



**Eulália Maria Mota
Santos**

**O Problema da Árvore de Suporte de Custo Mínimo
com Restrições de Peso**



**Eulália Maria Mota
Santos**

O Problema da Árvore de Suporte de Custo Mínimo com Restrições de Peso

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Doutor em Matemática, realizada sob a orientação científica da Doutora Maria Cristina Saraiva Requejo Agra, Professora Auxiliar do Departamento de Matemática da Universidade de Aveiro.

Apoio financeiro da Fundação para a Ciência e Tecnologia e do Fundo Social Europeu no âmbito do III Quadro Comunitário de Apoio e por fundos nacionais do Ministério da Ciência, Tecnologia e Ensino Superior - Bolsa de doutoramento com a referência SFRH/BD/46394/2008.

o júri
presidente

Doutor **Mário Guerreiro Silva Ferreira**
Professor Catedrático da Universidade de Aveiro

Doutora **Maria Teresa Nunes Chaves de Almeida**
Professora Catedrática do Instituto Superior de Economia e Gestão da Universidade de Lisboa

Doutor **Luís Eduardo Neves Gouveia**
Professor Catedrático da Faculdade de Ciências da Universidade de Lisboa

Doutor **Domingos Moreira Cardoso**
Professor Catedrático da Universidade de Aveiro

Doutor **José Manuel Vasconcelos Valério de Carvalho**
Professor Catedrático da Escola de Engenharia da Universidade do Minho

Doutora **Maria Cristina Saraiva Requejo Agra**
Professora Auxiliar da Universidade de Aveiro (Orientadora)

agradecimentos

Em primeiro lugar agradeço à minha orientadora, a Professora Doutora Maria Cristina Saraiva Requejo Agra pela oportunidade que me deu de realizar este trabalho sob sua orientação e pela disponibilidade e apoio que me prestou ao longo da sua elaboração.

Ao professor Agostinho Agra pela ajuda e sugestões.

Aos meus pais, ao Luís, à Érica e ao senhor Carlos Jorge agradeço o apoio que sempre me deram.

Aos meus amigos, Carlos Campos, Fátima Pina, Isabel Pego e Olga Oliveira agradeço a amizade e o apoio ao longo da realização deste trabalho.

Em termos institucionais, agradeço o apoio concedido pelas seguintes entidades durante o período de preparação deste trabalho: FCT - Fundação para a Ciência e Tecnologia e CIDMA - Centro de Investigação e desenvolvimento em Matemática e Aplicações da Universidade de Aveiro.

Para finalizar agradeço a todos os que têm contribuído direta ou indiretamente para a minha formação.

palavras-chave

Árvores com restrições de peso, Formulações, Relaxação Linear, Relaxação Lagrangeana, Desigualdades válidas, Separação, Heurísticas, *Feasibility Pump*, *Local Branching*.

resumo

Nesta tese abordam-se várias formulações e diferentes métodos para resolver o Problema da Árvore de Suporte de Custo Mínimo com Restrições de Peso (WMST – *Weight-constrained Minimum Spanning Tree Problem*). Este problema, com aplicações no desenho de redes de comunicações e telecomunicações, é um problema de Otimização Combinatória NP-difícil. O Problema WMST consiste em determinar, numa rede com custos e pesos associados às arestas, uma árvore de suporte de custo mínimo de tal forma que o seu peso total não exceda um dado limite especificado.

Apresentam-se e comparam-se várias formulações para o problema. Uma delas é usada para desenvolver um procedimento com introdução de cortes baseado em separação e que se tornou bastante útil na obtenção de soluções para o problema.

Tendo como propósito fortalecer as formulações apresentadas, introduzem-se novas classes de desigualdades válidas que foram adaptadas das conhecidas desigualdades de cobertura, desigualdades de cobertura estendida e desigualdades de cobertura levantada. As novas desigualdades incorporam a informação de dois conjuntos de soluções: o conjunto das árvores de suporte e o conjunto saco-mochila. Apresentam-se diversos algoritmos heurísticos de separação que nos permitem usar as desigualdades válidas propostas de forma eficiente.

Com base na decomposição Lagrangeana, apresentam-se e comparam-se algoritmos simples, mas eficientes, que podem ser usados para calcular limites inferiores e superiores para o valor ótimo do WMST. Entre eles encontram-se dois novos algoritmos: um baseado na convexidade da função Lagrangeana e outro que faz uso da inclusão de desigualdades válidas.

Com o objetivo de obter soluções aproximadas para o Problema WMST usam-se métodos heurísticos para encontrar uma solução inteira admissível. Os métodos heurísticos apresentados são baseados nas estratégias *Feasibility Pump* e *Local Branching*.

Apresentam-se resultados computacionais usando todos os métodos apresentados. Os resultados mostram que os diferentes métodos apresentados são bastante eficientes para encontrar soluções para o Problema WMST.

keywords

Trees with weight constraint, Formulations, Linear relaxation, Lagrangean relaxation, Valid inequalities, Separation, Heuristics, *Feasibility Pump*, *Local Branching*.

abstract

In this thesis we discuss several formulations and different methods to solve the Weight-constrained Minimum Spanning Tree Problem (WMST). This problem, with applications in the design of communication networks and telecommunications, is a NP-hard combinatorial optimization problem. The WMST problem aims at obtaining, in a network with costs and weights associated to its edges, a minimum cost spanning tree such that its total weight does not exceed a given specified parameter.

Various formulations to the problem are presented and compared. One is used to develop a procedure to introduce cuts based on separation and that became quite useful in obtaining solutions to the problem.

To strengthen the formulations, new classes of valid inequalities, adapted from the well-known cover inequalities, extended cover inequalities and lifted cover inequalities, are introduced.

These new inequalities incorporate information from two sets of solutions: the spanning trees set and the knapsack set. We present several separation heuristic algorithms that allow us to efficiently use the proposed valid inequalities.

Based on Lagrangean decomposition, simple and efficient algorithms are presented and compared. The algorithms can be used to obtain upper and lower bounds to the optimal value of the WMST problem. Among them are two new algorithms: one based on the convexity of the Lagrangean function and another making use of the inclusion of valid inequalities.

In order to obtain approximate solutions to the WMST problem, heuristic methods are used to find feasible solutions. The heuristic methods presented are based on the Feasibility Pump and Local Branching strategies.

We present computational results using all these methods. Results show that the different methods presented are very efficient for finding solutions to the WMST problem.

Conteúdo

1	Introdução	1
2	O Problema WMST	7
2.1	Descrição do Problema WMST	9
2.2	Propriedades do Problema WMST	11
2.3	Geração de Instâncias	15
3	Formulações	19
3.1	Formulações Naturais	21
3.2	Formulações Estendidas	24
3.2.1	Formulação de Fluxos	24
3.2.2	Formulação Miller-Tucker-Zemlin	26
3.2.3	Formulação de Peso Miller-Tucker-Zemlin	31
3.2.4	Formulação de Fluxo com especificação de Pesos	36
3.3	Procedimento com Introdução de Cortes	40
3.4	Experiências para Comparação das Formulações e Resultados Computacionais	46
3.4.1	Exemplo Comparativo das Formulações	46
3.4.2	Comparação das Formulações MTZ e WMTZ	48
3.4.3	Estudo Comparativo das Formulações	57
3.4.4	Resultados Computacionais	67
3.4.5	Síntese dos Resultados Computacionais	71
4	Algoritmos Lagrangeanos	73
4.1	Relaxação Lagrangeana para o Problema WMST	75
4.2	Algoritmo Lagrangeano Base	78

4.3	Comportamento Geométrico do Algoritmo Lagrangeano Base para o Problema WMST	81
4.4	Definição dos Valores Ponderados Utilizados no Algoritmo Lagrangeano Base para o Problema WMST	84
4.5	Experiências para Comparação dos Algoritmos Lagrangeanos e Resultados Computacionais	105
4.5.1	Exemplo Comparativo dos Algoritmos Lagrangeanos	105
4.5.2	Descrição dos Valores dos Parâmetros Usados nas Experiências Computacionais Realizadas	106
4.5.3	Estudo Comparativo dos Algoritmos Lagrangeanos e Resultados Computacionais	109
4.5.4	Síntese dos Resultados Computacionais	131
5	Desigualdades Válidas	133
5.1	Desigualdades de Cobertura (DC)	135
5.2	Desigualdades de Cobertura Implícita (DCI)	138
5.3	Desigualdades de Cobertura Implícita Estendida (DCIE)	141
5.4	Desigualdades de Cobertura Implícita Levantada (DCIL)	144
5.5	Desigualdades obtidas por Fixação de Variáveis	149
5.5.1	Desigualdades de Cobertura Implícita Levantada por <i>Down-Lifting</i> (DCILDL)	150
5.5.2	Desigualdades de Cobertura Implícita Levantada por <i>Up-Lifting</i> (DCILUL)	155
5.5.3	Desigualdades Generalizadas de Cobertura Implícita Levantada (DGCIL)	160
6	Algoritmos de Separação	167
6.1	Algoritmos de Separação Genéricos	168
6.1.1	Algoritmo de Separação para as DC	171
6.1.2	Algoritmo de Separação para as DCI	177
6.1.3	Algoritmo de Separação para as DCIE	182
6.1.4	Algoritmos de Separação para as DCIL	186
6.1.5	Algoritmo de Separação para as DCILDL	201
6.1.6	Algoritmo de Separação para as DCILUL	207
6.1.7	Algoritmo de Separação para as DGCIL	214

6.2	Experiências e Resultados Computacionais	220
6.2.1	Experiências realizadas com as DCs e as DCIs	221
6.2.2	Experiências Realizadas com as DCIEs	222
6.2.3	Experiências Realizadas com as DCILs	224
6.2.4	Experiências Realizadas com as DCILDLs	226
6.2.5	Experiências Realizadas com as DCILULs	228
6.2.6	Experiências Realizadas com as DGCILs	231
6.2.7	Comparação dos Algoritmos de Separação	233
6.2.8	Resultados Computacionais	237
6.2.9	Síntese dos Resultados Computacionais	239
7	Método <i>Feasibility Pump</i>	241
7.1	Heurística <i>Feasibility Pump</i> Básica Aplicada ao Problema WMST	242
7.2	Alteração da Função Objetivo na Heurística <i>Feasibility Pump</i> Aplicada ao Problema WMST	247
7.3	Extensões da Heurística <i>Feasibility Pump</i> para o Problema WMST . . .	249
7.3.1	Modo de Obtenção das Soluções x^*	249
7.3.2	Critérios de Arredondamento das Variáveis	250
7.3.3	Mecanismos de Perturbação	252
7.4	Comportamento da Heurística <i>Feasibility Pump</i>	253
7.5	Experiências e Resultados Computacionais	256
7.5.1	Descrição das Experiências Computacionais Realizadas	257
7.5.2	Exemplos Comparativos das Heurísticas <i>Feasibility Pump</i>	260
7.5.3	Resultados Computacionais	263
7.5.4	Síntese dos Resultados Computacionais	279
8	Método <i>Local Branching</i>	281
8.1	<i>Local Branching</i> Aplicado ao Problema WMST	282
8.2	Extensões do Método <i>Local Branching</i>	286
8.2.1	Solução de Referência Inicial	287
8.2.2	Mecanismos de Intensificação e de Diversificação	288
8.2.3	Critérios de Paragem	289
8.3	<i>Local Branching versus Feasibility Pump</i>	290
8.4	Descrição dos Algoritmos <i>Local Branching</i> para o WMST	292

8.5	Experiências para Comparação dos Algoritmos <i>Local Branching</i> e Resultados Computacionais	298
8.5.1	Exemplo Comparativo dos Algoritmos <i>Local Branching</i>	299
8.5.2	Estudo Comparativo dos Algoritmos <i>Local Branching</i>	300
8.5.3	Resultados Computacionais	308
8.5.4	Síntese dos Resultados Computacionais	317
9	Considerações Finais	319
	Anexos	332
	Notação Utilizada	335
	Siglas Utilizadas	343
	Lista de Figuras	349
	Lista de Tabelas	355
	Bibliografia	362

Capítulo 1

Introdução

O Problema da Árvore de Suporte de Custo Mínimo com Restrições de Peso, que designamos por WMST (*Weight-Constrained Minimum Spanning Tree Problem*), tem na literatura várias denominações e podemos encontrá-lo nas variantes de minimização e maximização. O problema foi mencionado pela primeira vez em 1982 por Aggarwal, Aneja e Nair [3], com a designação de *Minimum Spanning Tree Subject to a Side Constraint*. Outros autores utilizam a designação *Resource-Constrained Minimum Spanning Tree* [58] e ainda outros a de *Weight-Constraint Minimum Spanning Tree* [4, 37, 55, 65]. Para o caso da variante de maximização a designação mais comum é *Knapsack-Constrained Maximum Spanning Tree* [52, 66].

O objetivo do Problema WMST é a obtenção de uma árvore de suporte de custo mínimo, de tal modo que o peso total da árvore não exceda um determinado valor. A introdução da restrição de peso à árvore de suporte de custo mínimo transforma o WMST num problema NP-difícil [3].

O WMST é um problema de Otimização Combinatória que tem aplicação, por exemplo, no desenho de redes de telecomunicações e redes de comunicação. Em Henn [37] encontram-se dois exemplos de aplicações do Problema WMST: *design of physical systems subjected to limited budgets* e *minimum cost reliability constrained spanning tree*, os quais têm aplicação em problemas de redes de comunicação.

Na literatura, para este problema, podem ser encontradas fundamentalmente duas técnicas para obtenção de soluções utilizando métodos exatos. Uma das técnicas consiste no uso da relaxação Lagrangeana combinada com uma estratégia de *Branch and Bound* [3, 58] e outra consiste no uso de um algoritmo de tempo pseudo-polinomial

que usa a teoria espectral de grafos [38].

O artigo de Ravi e Goemans [54] descreve um esquema aproximado que usa a relaxação Lagrangeana e o artigo de Hong, Chung e Park [38] propõe um algoritmo de tempo pseudo-polinomial que usa a teoria espectral de grafos e também um esquema completo de aproximação bicritério. Os autores Hassin e Levin [35] melhoraram os resultados de [54] e consideram o problema como a interseção de matróides.

Em [66] os autores Yamada, Watanabe e Kataoka consideram o problema da árvore de suporte de custo máximo sujeita a restrições de peso. Provaram que o problema era NP-difícil, obtiveram limites superiores usando uma heurística de pesquisa local, utilizaram a relaxação Lagrangeana para a obtenção de limites inferiores, usaram o algoritmo *Branch and Bound* e por último um método para acelerar o tempo computacional. Referem ainda que se pode facilmente aplicar ao caso da minimização.

No artigo de Xue [65] é apresentado um simples, mas eficaz algoritmo primal-dual para obter soluções aproximadas para o problema de encontrar o caminho mais curto com restrições de peso (*Weight-Constrained Shortest Paths*) e para o Problema WMST. O autor apenas apresenta resultados computacionais para o algoritmo aplicado ao primeiro problema, mas afirma que tem intenção de estudar o segundo algoritmo para grafos aleatórios e comparar com soluções ótimas obtidas usando o Algoritmo *k Smallest Spanning Tree* [16].

Em Henn [37] é apresentado um estudo alargado de propriedades e relaxação Lagrangeana para o Problema WMST e um novo algoritmo *Branch and Bound*.

Em vez da introdução da restrição de peso pode-se incluir o peso da árvore como segundo objetivo. O problema resultante é de uma árvore de suporte biobjetivo/bicritério ([8, 31, 38, 53, 60, 61] entre muitos outros). Em Aggarwal, Aneja e Nair [3] são estabelecidas certas propriedades da solução ótima considerando uma árvore de suporte bicritério.

Em 2007, Raidl, Pirkwieser e Puchinger [52] apresentaram um algoritmo evolutivo híbrido para o Problema da Árvore de Suporte de Custo Máximo com Restrições Saco-mochila, baseado no algoritmo evolutivo proposto para o Problema da Árvore de Suporte de Custo Mínimo com Restrições de Grau [39], só adaptaram a inicialização e a variação dos operadores.

Recentemente, Requejo et al. [55] descreveram várias formulações para o problema, Requejo e Santos [57] apresentaram algoritmos baseados em relaxação Lagrangeana e Agra et al. [4] descreveram desigualdades válidas para o Problema WMST.

Nesta tese apresentamos várias formulações e diferentes métodos para resolver o Problema WMST, tais como, alguns procedimentos de introdução de cortes usando separação, o método de relaxação: linear e Lagrangeana, alguns métodos de geração e introdução de desigualdades válidas nas formulações usando separação, o método *Feasibility Pump* para obter uma solução inteira admissível, e por fim o método *Local Branching* para melhorar as soluções admissíveis.

Este trabalho está organizado da seguinte forma. No Capítulo 2 fazemos uma pequena abordagem a problemas de árvores com restrições adicionais, nos quais se enquadra o Problema WMST. De seguida descrevemos o Problema WMST, apresentamos algumas propriedades importantes e descrevemos o processo de geração de instâncias que servirão para realizar experiências e obter resultados computacionais com os diferentes métodos utilizados nesta tese para o Problema WMST.

No Capítulo 3 começamos por apresentar uma formulação genérica para o Problema WMST. De seguida, apresentamos duas formulações naturais: uma baseada nas desigualdades de eliminação de subcircuitos e a outra baseada em desigualdades de corte. Depois, apresentamos quatro formulações compactas estendidas: uma formulação de fluxos, duas formulações baseadas nas desigualdades Miller-Tucker-Zemlin e uma formulação de fluxos que especifica nos índices o peso do caminho desde a raiz. Devido ao uso destas formulações se tornar limitado na prática, vamos usar um procedimento, baseado em separação, no qual as restrições de corte vão sendo introduzidas no modelo. Este procedimento permite obter a solução ótima, ou uma solução admissível próxima da ótima. No final do capítulo apresentamos resultados computacionais de várias experiências realizadas com as formulações e também efetuamos um estudo computacional comparativo entre as várias formulações apresentadas.

No Capítulo 4 começamos por fazer uma revisão da literatura de alguns algoritmos Lagrangeanos existentes para o Problema WMST e para alguns problemas de Otimização Combinatória. Com o objetivo de obter soluções aproximadas para o Problema WMST, propomos um algoritmo baseado na relaxação Lagrangeana que denominamos de Algoritmo Lagrangeano Base para o Problema WMST e descrevemos o seu comportamento geométrico. Para o algoritmo base apresentado propomos diferentes variantes baseadas em algoritmos existentes para problemas de Otimização Combinatória. Neste Capítulo discutimos ainda os resultados computacionais efetuando um estudo comparativo do ponto de vista computacional entre os vários algoritmos derivados do Algoritmo

Lagrangeano Base.

No Capítulo 5 com o propósito de fortalecer as formulações apresentadas no Capítulo 3, discutimos classes de desigualdades válidas para o Problema WMST. Começamos por adaptar as conhecidas desigualdades de Cobertura para o Problema Saco-mochila Binário para o caso do Problema WMST e apresentamos novas desigualdades válidas baseadas numa estrutura em árvore, que designamos por Desigualdades de Cobertura Implícita. Propomos também Desigualdades de Cobertura Implícita Estendida baseadas nas conhecidas Desigualdades de Cobertura Estendida. Para fortalecer as desigualdades de Cobertura Implícita usamos a técnica de levantamento sequencial de variáveis e obtemos as Desigualdades de Cobertura Implícita Levantada. Por fim, propomos uma generalização das Desigualdades de Cobertura Implícita Levantada que podem fortalecer ainda mais as formulações, sendo estas baseadas na fixação de conjuntos de variáveis.

No Capítulo 6 descrevemos dois algoritmos heurísticos de separação genéricos para as classes de desigualdades válidas apresentadas no Capítulo 5. Num dos algoritmos é introduzida uma desigualdade válida, em cada iteração, enquanto que no outro podem ser introduzidas várias desigualdades válidas na mesma iteração. De seguida especificam-se as alterações a efetuar nos algoritmos heurísticos de separação genéricos para introduzir as desigualdades válidas descritas no Capítulo 5. Como para algumas classes de desigualdades válidas é necessário efetuar o levantamento de variáveis, apresentamos também um algoritmo heurístico baseado em relaxação Lagrangeana para determinação dos coeficientes das variáveis a efetuar levantamento. No final deste capítulo, apresentamos um estudo comparativo entre as várias experiências computacionais realizadas com os algoritmos heurísticos de separação propostos para as várias classes de desigualdades válidas.

No Capítulo 7 para obter uma solução inteira admissível descrevemos uma heurística denominada de Heurística *Feasibility Pump* Básica Aplicada ao Problema WMST. Com o intuito de melhorar a qualidade das soluções obtidas, apresentamos uma versão onde se altera a função objetivo na Heurística *Feasibility Pump* Básica aplicada ao Problema WMST. Para melhorar o desempenho das heurísticas *Feasibility Pump* propostas referimos algumas extensões do Método *Feasibility Pump* para o Problema WMST e apresentamos do ponto de vista geométrico o comportamento da Heurística *Feasibility Pump*, no caso geral e no caso particular do problema em estudo. Para finalizar o capítulo, discutimos algumas experiências computacionais realizadas, descrevemos quatro estratégias Heurísticas *Feasibility Pump* e comparamos essas estratégias,

com a Heurística do *Xpress* aplicada ao nodo raiz utilizando a Formulação WMTZ e também com a primeira solução inteira admissível obtida através da Formulação WMTZ quando usa o procedimento *Branch and Bound*.

No Capítulo 8 para obter uma solução inteira admissível aplicamos o Método *Local Branching* ao Problema WMST através da implementação de um algoritmo que denominamos de Algoritmo *Local Branching* Clássico Aplicado ao Problema WMST. Com o propósito de melhorar o desempenho do algoritmo quando aplicado ao Problema WMST são apresentadas algumas extensões do método. Para além de se compararem os dois métodos *Feasibility Pump* e *Local Branching*, também fazemos uma abordagem geométrica do comportamento da Heurística *Local Branching*. Neste capítulo descrevemos ainda vários algoritmos derivados do algoritmo clássico e apresentamos um estudo computacional comparativo entre eles. Comparamos também o melhor destes algoritmos com o Algoritmo *Branch and Bound* do Procedimento de Introdução de Cortes P-WMTZ+C apresentado no Capítulo 3.

Por fim, no Capítulo 9 apresentamos algumas considerações finais.

No final dos Capítulos 3, 4, 6, 7 e 8 são apresentadas sínteses dos resultados computacionais obtidos para esse capítulo.

Capítulo 2

O Problema WMST

Diversos problemas do mundo real podem ser modelados por grafos. Em muitos problemas, os grafos devem ter estruturas mais próximas do problema real, particularmente, os problemas com estrutura em árvore.

Os problemas de Otimização com topologia em árvore surgem num número surpreendentemente grande de aplicações, tais como, redes informáticas, redes de transporte, distribuição de energia, localização de instalações, produção industrial, telecomunicações e muitas outras.

O Problema da Árvore de Suporte de Custo Mínimo (MCST - *Minimum Cost Spanning Tree* ou, apenas, MST - *Minimum Spanning Tree*) é um problema de Otimização Combinatória [49] para o qual se conhecem algoritmos eficientes, que em tempo polinomial, permitem obter a solução ótima.

Se incluirmos uma restrição adicional ao Problema MST, o problema resultante pertence à chamada classe dos problemas NP-difíceis [21].

Na literatura podemos encontrar alguns problemas de árvores de suporte de custo mínimo com restrições adicionais que têm sido alvo de estudo. Para cada problema podemos encontrar muitas referências, mas apenas vamos referir algumas. São eles, o Problema da Árvore de Suporte de Custo Mínimo com:

- **Restrições de Capacidade** - *Capacited Minimum Spanning Tree* [23, 27]

Consiste em determinar uma árvore de suporte de custo mínimo centralizada num vértice do grafo 0 (raiz), com a restrição adicional de a soma dos pesos dos nós de qualquer sub-árvore conetada à raiz não poder ser maior que um dado valor Q (número natural).

- **Restrições de Grau** - *Degree-constrained Minimum Spanning Tree* [9, 48]

Pretende-se determinar uma árvore de suporte de custo mínimo tal que o grau dos vértices não exceda um valor máximo.

- **Restrições de Salto** - *Hop-constrained Minimum Spanning Tree* [25, 28]

Consiste em determinar a árvore de suporte de custo mínimo tal que o número de saltos no único caminho desde o nodo raiz até qualquer outro nodo da árvore não é superior a H (número natural).

- **Restrições de Diâmetro** - *Diameter-constrained Minimum Spanning Tree* [26, 56]

Neste problema pretende-se determinar uma árvore de suporte de custo mínimo sujeita a um limite D (número natural) para o seu diâmetro.

- **Restrições de Peso** - *Weight-constrained Minimum Spanning Tree* [3, 4]

Consiste em encontrar uma árvore de suporte de custo mínimo, de tal modo que o peso total da árvore não exceda um determinado valor W (número natural). Este é o problema em estudo ao longo da tese e vai ser descrito de forma detalhada na Secção 2.1.

A inclusão destas restrições tem a ver com propriedades adicionais que é necessário incluir nas redes/árvores. Por exemplo, as restrições de diâmetro e de salto estão relacionadas com restrições de atraso máximo e restrições de fiabilidade da rede, de modo a evitar a degradação da qualidade do sinal. As restrições de capacidade e de grau encontram-se relacionadas com restrições de capacidade de certos dispositivos instalados em alguns nós e as restrições de peso ou saco-mochila como também são conhecidas encontram-se relacionadas com restrições nos custos de instalação da rede.

Depois de enquadrar o problema em estudo como um problema de Árvore de Suporte de Custo Mínimo com Restrições Adicionais, na Secção 2.1 efetuamos uma descrição mais detalhada do Problema WMST. Na Secção 2.2 recordamos algumas propriedades importantes e, finalmente, na Secção 2.3 descrevemos o modo como foram geradas as instâncias que serão usadas ao longo de toda a tese para obtenção dos resultados computacionais dos vários métodos aplicados ao Problema WMST.

2.1 Descrição do Problema WMST

O Problema da Árvore de Suporte de Custo Mínimo com Restrições de Peso, WMST (*Weight-constrained Minimum Spanning Tree Problem*), é definido do seguinte modo.

Consideremos um grafo completo não orientado $G = (V, E)$, com o conjunto de nodos $V = \{0, 1, \dots, n-1\}$ e o conjunto de arestas $E = \{\{i, j\}, i, j \in V, i \neq j\}$. O conjunto V contém n nodos e o conjunto E contém $\frac{n(n-1)}{2}$ arestas. Cada aresta $e = \{i, j\} \in E$ tem associado um custo c_e e um peso w_e , ambos inteiros e positivos.

O Problema WMST consiste em encontrar uma árvore de suporte $T = (V, E_T)$ em G , $E_T \subset E$, de custo mínimo $C(T) = \sum_{e \in E_T} c_e$, onde o peso total da árvore $W(T) = \sum_{e \in E_T} w_e$ não exceda um dado limite W (número natural). No Problema WMST a restrição adicional que é incluída ao Problema MST é chamada de restrição de peso ou restrição saco-mochila (*Knapsack Constraint* ou *Weight Constraint*) e deve ser de tal forma que

$$\sum_{e \in E_T} w_e \leq W.$$

Assim, com a introdução desta restrição adicional, o Problema MST transforma-se no Problema WMST que é um problema de Otimização Combinatória NP-difícil [3]. Denota-se por X_{WMST} o conjunto de todas as soluções admissíveis do Problema WMST e por $\vartheta(WMST)$ o valor ótimo.

Exemplo 2.1.

Consideremos uma instância do problema, um exemplo de um grafo com 5 nodos, para o qual os custos e pesos associados a cada aresta se encontram indicados nas seguintes matrizes:

$$H = \begin{pmatrix} - & 6 & 3 & 8 & 7 \\ & - & 6 & 2 & 2 \\ & & - & 4 & 5 \\ & & & - & 5 \\ & & & & - \end{pmatrix} \quad \text{e} \quad C = \begin{pmatrix} - & 1 & 9 & 8 & 7 \\ & - & 1 & 15 & 16 \\ & & - & 30 & 16 \\ & & & - & 10 \\ & & & & - \end{pmatrix},$$

onde H e C são as matrizes de pesos e custos, respetivamente.

Pretendemos obter a árvore de suporte de custo mínimo com a restrição adicional de o peso da árvore não poder exceder o valor $W = 20$ ($W(T) \leq 20$). Usando, por exemplo, o Algoritmo de Kruskal ou Prim [6] obtemos a árvore de suporte de custo mínimo ilustrada na Figura 2.1.

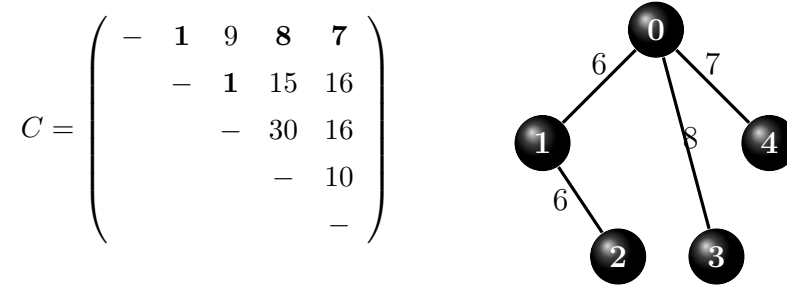


Figura 2.1: Árvore de suporte de custo mínimo.

Esta árvore tem custo $C(T) = 17$ e peso $W(T) = 27$ pelo que é não admissível para o Problema WMST, pois não verifica a restrição de peso, ou seja, $W(T) = 27 > W = 20$. Uma solução admissível seria, por exemplo, a árvore de suporte da Figura 2.2.

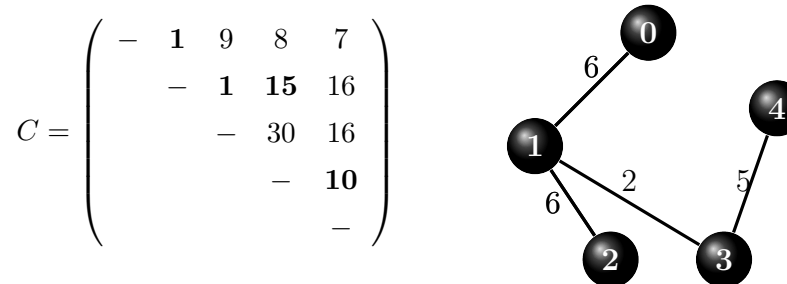


Figura 2.2: Árvore de suporte com custo 27 e peso 19.

A árvore representada na Figura 2.2 tem custo $C(T) = 27$ e peso $W(T) = 19$. Além de ser admissível dado que verifica a restrição de peso ($W(T) = 19 < 20$) ela corresponde à solução ótima do Problema WMST para a instância de 5 nodos considerada.

Um problema definido num grafo não orientado $G = (V, E)$ pode-se transformar num problema equivalente definido num grafo orientado, onde cada aresta do grafo não orientado $e = \{0, j\} \in E$ é substituída por um arco $(0, j)$ e cada aresta $e = \{i, j\} \in E, i \neq 0$, é substituída por dois arcos, o arco (i, j) e o arco (j, i) . Assim, no grafo orientado $G = (V, A)$, apenas se altera o conjunto de arcos, o qual é dado por $A = \{(i, j), i \in V, j \in V \setminus \{0\}, i \neq j\}$. Cada arco $(i, j) \in A$ herda os custos e os pesos da aresta correspondente $\{i, j\}$ do grafo não orientado. Deste modo, cada arco (i, j) tem associado um custo c_{ij} e um peso w_{ij} , ambos inteiros e positivos.

2.2 Propriedades do Problema WMST

Nesta secção vamos apresentar algumas propriedades do Problema WMST. Para o fazer vamos usar três árvores: a Árvore de Suporte de Custo Mínimo, a Árvore de Suporte de Peso Mínimo e a Árvore de Suporte Ponderada Mínima.

Árvore de Suporte de Custo Mínimo (MST)

O Problema MST consiste em encontrar uma árvore de suporte $T_c = (V, A_{T_c})$, $A_{T_c} \subset A$ em $G = (V, A)$ de custo mínimo $C(T_c) = \sum_{(i,j) \in A_{T_c}} c_{ij}$.

Para obtermos uma MST, árvore de suporte de custo mínimo, existem vários algoritmos polinomiais tais como os algoritmos de Sollin, Kruskal e Prim (ver [6]).

Árvore de Suporte de Peso Mínimo (MST_w)

O Problema MST_w consiste em encontrar uma árvore de suporte $T_w = (V, A_{T_w})$, $A_{T_w} \subset A$ em $G = (V, A)$ de peso mínimo $W(T_w) = \sum_{(i,j) \in A_{T_w}} w_{ij}$.

Neste caso também podemos recorrer aos algoritmos polinomiais referidos anteriormente para obtenção da árvore T_w (ver [6]).

As árvores T_c e T_w são duas árvores de G , sendo T_c a árvore de suporte de custo mínimo e T_w a árvore de suporte de peso mínimo. A árvore T_c tem associado um peso $W(T_c) = \sum_{(i,j) \in A_{T_c}} w_{ij}$ que, não sendo inferior ou igual a W , não é uma árvore admissível para o Problema WMST. A árvore T_w tem associado um custo $C(T_w) = \sum_{(i,j) \in A_{T_w}} c_{ij}$ sendo esta admissível, no caso do seu peso ser inferior ou igual a W . Verifica-se que os custos da árvore T_c e da árvore T_w correspondem a um limite inferior e superior, respetivamente, para o valor ótimo $\vartheta(WMST)$ do Problema WMST, o que nos permite escrever

$$C(T_c) \leq \vartheta(WMST) \leq C(T_w).$$

Podemos assumir igualmente a seguinte proposição.

Proposição 2.1.

O Problema WMST tem solução ótima se e só se

$$W(T_w) \leq W \leq W(T_c).$$

Facilmente se conclui que se $W(T_w) > W$, então o Problema WMST não tem solução. E temos também as seguintes proposições.

Proposição 2.2.

Se $W(T_c) \leq W$, então T_c é uma árvore correspondente à solução ótima para o Problema WMST.

Proposição 2.3.

Se $W(T_w) \leq W$ e $C(T_w) = C(T_c)$, então T_w é a árvore correspondente à solução ótima para o Problema WMST.

Apesar de T_w ser uma árvore admissível, no caso de $W(T_w) \leq W$, o objetivo é encontrar melhores soluções admissíveis do que a correspondente solução da árvore de suporte T_w .

Exemplo 2.2.

A árvore de suporte de custo mínimo T_c encontra-se representada na Figura 2.1 e temos que $C(T_c) = 17$ e $W(T_c) = 27$ pelo que T_c é uma árvore não admissível para o Problema WMST ($W(T_c) = 27 > 20 = W$).

A árvore de suporte de peso mínimo T_w encontra-se representada na figura seguinte.

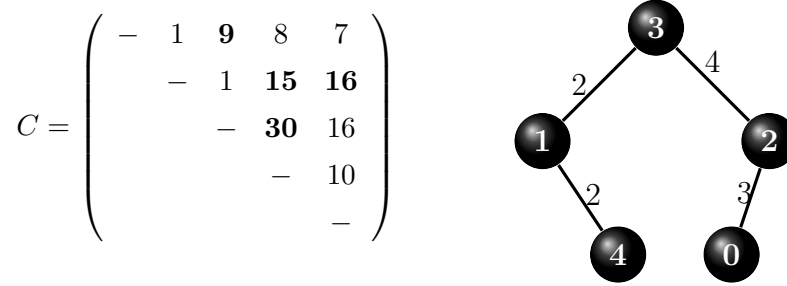


Figura 2.3: Árvore de suporte de peso mínimo.

A árvore T_w tem custo $C(T_w) = 70$ e peso $W(T_w) = 11$, pelo que T_w é uma árvore admissível para o Problema WMST ($W(T_w) = 11 < 20 = W$).

Assim, $17 \leq \vartheta(WMST) \leq 70$ e $11 \leq W = 20 \leq 27$.

Árvore de Suporte Ponderada Mínima (MST_P)

O Problema MST_P consiste em associar valores positivos a cada arco $(i, j) \in A$, os quais são combinação linear do custo e do peso, sendo estes da forma, $p_{ij} = aw_{ij} + bc_{ij}$, com a e b escalares reais não negativos.

O Problema MST_P consiste em encontrar a árvore de suporte $T_p = (V, A_{T_p})$, $A_{T_p} \subset A$, em $G = (V, A)$ de valor ponderado mínimo $P(T_p) = \sum_{(i,j) \in A_{T_p}} p_{ij}$ com um peso $W(T_p) = \sum_{(i,j) \in A_{T_p}} w_{ij}$ e um custo $C(T_p) = \sum_{(i,j) \in A_{T_p}} c_{ij}$. Para a obtenção desta árvore T_p também podemos recorrer aos algoritmos polinomiais descritos em [6].

Em particular:

- Se $a = 0$ e $b = 1$, então obtemos a árvore de suporte de custo mínimo T_c ($T_p \equiv T_c$).

- Se $a = 1$ e $b = 0$, então obtemos a árvore de suporte de peso mínimo T_w ($T_p \equiv T_w$).

Note-se que a árvore T_p pode ser uma árvore admissível, caso $W(T_p) \leq W$ ou uma árvore não admissível no caso de $W(T_p) > W$.

Exemplo 2.3.

Dependendo da escolha dos valores dos parâmetros a e b no cálculo dos valores ponderados de cada arco $(i, j) \in A$, podemos obter árvores admissíveis ou não admissíveis. Consideremos, por exemplo, os valores ponderados da forma, $p_{ij} = 0,5w_{ij} + 0,5c_{ij}$ para cada arco $(i, j) \in A$. Na Figura 2.4 encontram-se a matriz de valores ponderados L e a correspondente árvore de suporte ponderada mínima T_p .

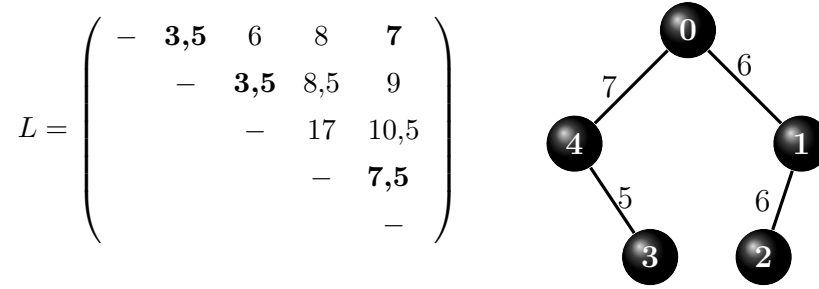


Figura 2.4: Árvore de suporte ponderada mínima ($p_{ij} = 0,5w_{ij} + 0,5c_{ij}$ para todos os $(i, j) \in A$).

A árvore T_p obtida na Figura 2.4 tem valor ponderado $P(T_p) = 21,5$, custo $C(T_p) = 19$ e peso $W(T_p) = 24$ pelo que é uma árvore não admissível para o Problema WMST, pois $W(T_p) = 24 > 20 = W$.

Caso se utilize, por exemplo, os valores ponderados da forma, $p_{ij} = w_{ij} + 0,2c_{ij}$ para cada arco $(i, j) \in A$ obtemos a matriz de valores ponderados L e a correspondente árvore de suporte ponderada mínima T_p representadas na Figura 2.5.

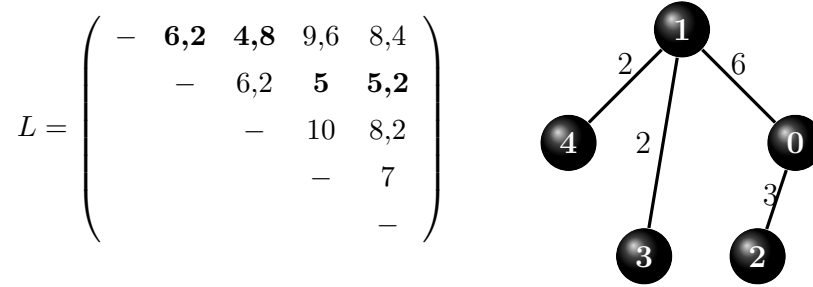


Figura 2.5: Árvore de suporte ponderada mínima ($p_{ij} = w_{ij} + 0,2c_{ij}$ para todos os $(i, j) \in A$).

A árvore T_p da Figura 2.5 tem valor ponderado $P(T_p) = 21,2$, custo $C(T_p) = 41$ e peso $W(T_p) = 13$, sendo uma árvore admissível para o Problema WMST.

2.3 Geração de Instâncias

Nesta secção descrevemos o modo como foram geradas as instâncias para o Problema WMST, as quais são usadas ao longo de toda a tese para obtenção de resultados computacionais dos vários métodos propostos. Para gerar as instâncias temos de ter em consideração que cada aresta $e = \{i, j\} \in E$ tem associado um custo c_e e um peso w_e e temos ainda de definir a constante W que limita o peso. Assim, para obter as matrizes de custos e de pesos e o valor da constante W geramos os seguintes três tipos de instâncias.

- INSTÂNCIAS EUCLIDEANAS (E)

Os custos c_e e os pesos w_e são obtidos usando distâncias Euclidianas. Para obter estas instâncias Euclidianas geramos aleatoriamente as coordenadas de n pontos/nodos numa grelha de dimensão 100×100 . O custo de cada aresta $e = \{i, j\} \in E$ é a parte inteira da distância Euclidiana entre os pontos/nodos i e j gerados na rede. Procedemos de modo semelhante para obter os pesos, gerando uma nova grelha.

O valor da constante W é obtido depois de geradas as matrizes de custos e de pesos. Para obter um valor (admissível) para o peso limite W temos de ter em consideração a Proposição 2.1, onde $W(T_w) \leq W \leq W(T_c)$. Assim, começamos

por obter o peso da árvore de suporte de custo mínimo $W(T_c) = W_c$ e o peso da árvore de suporte de peso mínimo $W(T_w) = W_w$ e depois testamos os seguintes valores para W_i ,

$$W_i = \frac{W_c + W_w}{2^i},$$

com $i \in \{1, \dots, 10\}$. No caso do valor de W_i não ser inteiro, efetua-se o arredondamento simétrico. Dos vários valores W_i testados escolheu-se para o valor de W aquele que tornava a instância mais difícil de resolver.

- INSTÂNCIAS ALEATÓRIAS (R)

Os custos c_e e os pesos w_e são gerados aleatoriamente no intervalo $[1; 1000]$.

O valor da constante W foi obtido como no grupo de instâncias Euclidianas e também após a geração das matrizes de custos e pesos.

- INSTÂNCIAS QUASE CAMINHOS (QC)

Os custos c_e são gerados de acordo com o conjunto de instâncias *Spanner* descrito em Pisinger [50]. Para obter os custos c_e usando o conjunto *spanner* neste grupo de instâncias são necessários, os parâmetros s e m inteiros e positivos, onde s é o tamanho do conjunto *spanner* e m é o limite superior do intervalo para o multiplicador, isto é, gera-se aleatoriamente um multiplicador m_a no intervalo $[1; m]$. Tendo fixado valores para os parâmetros s e m , os custos c_e são obtidos da seguinte forma:

1. gera-se aleatoriamente c_j no intervalo $[1; 100]$, $j \in \{1, \dots, s\}$;
2. obtêm-se os itens do conjunto *spanner* $c_k = \left\lceil \frac{2c_j}{m} \right\rceil$, $j, k \in \{1, \dots, s\}$;
3. gera-se aleatoriamente um multiplicador m_a (inteiro) no intervalo $[1; m]$ e escolhe-se de forma aleatória um custo c_k , $k \in \{1, \dots, s\}$;
4. obtém-se $c_e = m_a c_k$.

Após a realização de experiências computacionais com vários valores de s e m , verificamos que as instâncias são mais difíceis de resolver quando usamos $m = 10$ e valores inteiros entre 2 e 6 para o valor do parâmetro s .

Depois de gerados os custos, geram-se os pesos w_e de tal modo que a estrutura da correspondente solução seja um quase caminho, ou seja, um grafo com diâmetro grande, mas cujo diâmetro seja inferior a $n - 1$. Para gerar os pesos começa-se

por definir um valor para a constante W e inicializa-se o peso de cada aresta a 1. De seguida, obtém-se a árvore de suporte de custo mínimo e atribuem-se pesos elevados às arestas que pertencem a esta árvore. Para as restantes arestas, aquelas que não pertencem à MST, os pesos são gerados de forma a obter-se uma solução próxima de um caminho. Para o conseguir, os pesos das arestas que não pertencem à MST obtêm-se do seguinte modo

$$w_e = \frac{W}{n-1}(1-p) + r,$$

onde $p \in [0,5;1]$ e r é gerado aleatoriamente no intervalo $\left[1; \frac{W}{n-1}p\right]$. No caso das arestas continuarem a ter peso 1 usamos a seguinte fórmula para determinar os pesos

$$w_e = \frac{W}{n-1}(1 + a_1p) + a_2r,$$

onde a_1 e $a_2 \in [0;10]$. No caso dos valores dos pesos w_e não serem inteiros efetua-se o arredondamento simétrico. A constante W toma valores no intervalo $[1000;3500]$ consoante a dimensão n da instância.

Em todos os grupos de instâncias foram gerados exemplos de 10, 20, 40, 60, 80, 100, 150, 200, 300, 400, 500 e 1000 nodos. Nas instâncias Aleatórias e Euclidianas foram gerados 5 exemplos de cada dimensão, perfazendo um total de 60 instâncias em cada um dos grupos. Nas instâncias Quase Caminhos foram gerados 10 exemplos de cada dimensão com menos de 200 nodos e 5 exemplos para instâncias com 200 ou mais nodos. Assim no grupo de instâncias Quase Caminhos existem um total de 95 instâncias teste. Em suma, foram geradas 215 instâncias no total.

Capítulo 3

Formulações

O WMST é um problema de Otimização Combinatória que pode ser formulado em Programação Linear Inteira (PLI). Para um mesmo problema diferentes formulações podem diferir em termos dos correspondentes limites inferiores obtidos através da relaxação em programação linear. A qualidade do limite inferior obtido depende da formulação em PLI que se utiliza.

Para o Problema MST são conhecidas algumas formulações (ver Magnanti e Wolsey [45]). Para obter formulações para o Problema WMST pode-se facilmente adaptar uma formulação para o MST e adicionar a restrição que limita o peso das arestas na árvore de suporte. Esta restrição é uma restrição tipo saco-mochila. A introdução desta restrição torna o problema NP-difícil [3].

As formulações em Programação Linear Inteira para Problemas de Otimização Combinatória são, de um modo geral, consideradas formulações naturais ou formulações estendidas. Informalmente, e para problemas em árvores, podemos dizer que uma formulação natural usa apenas as variáveis de desenho topológico da árvore (uma única variável associada a cada arco do grafo). Uma formulação diz-se estendida se usa, adicionalmente, outras variáveis (associadas aos arcos ou não). Estas variáveis adicionais, apesar de não serem necessárias para a obtenção de uma formulação válida para o problema, contêm informação adicional que pode reduzir, consideravelmente, o número de restrições envolvidas no modelo. Além disso, o seu uso permite derivar formulações compactas (formulações que envolvem um número polinomial de restrições e variáveis).

Nesta secção apenas consideramos formulações orientadas para o Problema WMST definidas num grafo orientado. Sem perda de generalidade, consideramos o nodo 0 como o nodo raiz.

Considerem-se as variáveis binárias x_{ij} para todos os arcos $(i, j) \in A$, que indicam se o arco (i, j) está ou não na solução, ou seja,

$$x_{ij} = \begin{cases} 1, & \text{se o arco } (i, j) \text{ pertence à solução;} \\ 0, & \text{caso contrário.} \end{cases}$$

e para simplificar a notação assumimos que as variáveis x_{i0} , $i \in V \setminus \{0\}$ têm sempre valor nulo.

Vamos considerar que X_T representa o conjunto de todas as soluções admissíveis definidas por uma formulação para o Problema MST, no qual o conjunto de desigualdades definido para x , com $x = (x_{ij}) \in \mathbb{R}^{|A|}$, descreve o invólucro convexo das soluções inteiras do Problema MST. Assim, este conjunto de restrições garante que a solução tem uma estrutura de árvore (mais precisamente arborescência) de suporte. Uma formulação genérica para o Problema WMST pode ser dada pelo seguinte modelo:

$$(WMST) : \quad \min \quad \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (3.1)$$

$$s.a. \quad x \in X_T \quad (3.2)$$

$$\sum_{(i,j) \in A} w_{ij} x_{ij} \leq W. \quad (3.3)$$

A função objetivo, dada pela expressão (3.1), indica que se pretende minimizar o somatório dos custos dos arcos na solução. A desigualdade (3.3) é a restrição que limita o peso total da árvore de suporte e garante que a soma dos pesos dos arcos na solução não pode exceder um limite W , sendo esta denominada de restrição de peso.

Notamos que a validade de uma qualquer formulação para o Problema WMST obtida juntando a restrição (3.3) a uma conhecida formulação para o Problema MST (conjunto de restrições dadas por (3.2)) torna-se imediata. As quatro primeiras formulações que apresentamos (ES, CS, MF e MTZ) são obtidas desta forma. O processo de obtenção das duas últimas formulações (WMTZ e WE) já foi diferente.

Se substituirmos as restrições de integralidade $x_{ij} \in \{0, 1\}$, $(i, j) \in A$ por $x_{ij} \geq 0$, $(i, j) \in A$ obtemos a relaxação linear da Formulação WMST. De um modo geral, dado o conjunto X_{WMST} de soluções admissíveis, definido pela Formulação WMST, denotamos por X_L o conjunto de todas as soluções admissíveis da relaxação linear do Problema WMST.

Neste capítulo, apresentamos formulações naturais e estendidas para o Problema WMST e apresentamos também um procedimento com introdução de cortes baseado em separação.

Na Secção 3.1 fazemos uma breve apresentação de duas formulações naturais para o Problema WMST, uma onde a expressão designada por (3.2) representa desigualdades de eliminação de subcircuitos e a outra desigualdades de corte. Na Secção 3.2 são apresentadas quatro formulações compactas estendidas: uma formulação de fluxos, duas formulações baseadas nas desigualdades Miller-Tucker-Zemlin e uma formulação de fluxos que especifica nos índices o peso do caminho desde a raiz. Uma vez que o uso destas formulações na prática torna-se limitado, na Secção 3.3 vamos usar um procedimento, baseado em separação, no qual as restrições de corte vão sendo introduzidas e que nos permite obter a solução ótima, ou uma solução admissível próxima da ótima. Por fim, na Secção 3.4 apresentamos resultados computacionais de várias experiências realizadas com as formulações e efetuamos um estudo comparativo entre as várias formulações apresentadas em termos de qualidade do limite inferior obtido e tempo de execução na obtenção da solução ótima ou de uma solução admissível. No final da Secção 3.4, em forma de resumo do capítulo apresentamos uma síntese dos resultados computacionais obtidos.

3.1 Formulações Naturais

Nesta secção apresentamos duas formulações naturais para o Problema WMST, uma onde a expressão designada por (3.2) representa desigualdades de eliminação de subcircuitos e a outra desigualdades de corte. Nas duas formulações naturais que se seguem considere-se, apenas o conjunto de variáveis binárias orientadas x_{ij} .

Uma das formulações usa as desigualdades de eliminação de subcircuitos e a outra desigualdades de corte para assegurar a conexidade/prevenção de circuitos na solução.

Dados dois subconjuntos $S_1, S_2 \subseteq V$. O conjunto

$$A(S_1, S_2) = \{(i, j) \in A : i \in S_1, j \in S_2\}$$

designa o conjunto de arcos com um extremo em S_1 e outro extremo em S_2 e orientados

no sentido de S_1 para S_2 . Quando, $S_1 = S_2 = S$, $A(S)$ designa o conjunto de arcos com ambos os extremos em S . Seja $S^c = V \setminus S$, o conjunto complementar de S .

Formulação de Eliminação de Subcircuitos

A Formulação de Eliminação de Subcircuitos (ES) também conhecida por *Packing/Subsets Formulation* [45] no contexto do Problema WMST é dada por

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij}$$

s.a.

$$\sum_{(i,j) \in A} x_{ij} = |V| - 1 \quad (3.4)$$

$$\sum_{(i,j) \in A(S)} x_{ij} \leq |S| - 1, \quad S \subset V, S \neq \emptyset, 2 \leq |S| \leq |V| - 1 \quad (3.5)$$

$$\sum_{(i,j) \in A} w_{ij} x_{ij} \leq W \quad (3.6)$$

$$x_{ij} \in \{0, 1\}, \quad (i, j) \in A. \quad (3.7)$$

As restrições de cardinalidade (3.4) asseguram que são escolhidos exatamente $|V| - 1$ arcos para estarem na solução. Estas restrições podem ser substituídas pelo seguinte conjunto de restrições

$$\sum_{i \in V} x_{ij} = 1, \quad j \in V \setminus \{0\}, \quad (3.8)$$

que garantem que um e apenas um arco chega a cada nodo, exceto para o nodo raiz.

O conjunto de restrições (3.5) impede que o conjunto de arcos escolhidos para a solução contenha ciclos. Estas restrições são chamadas de restrições de eliminação de subcircuitos ou *packing constraints* e restringem o número de arcos que podem ser adicionados (*packed*) em qualquer conjunto de nodos S . Observe-se que este segundo conjunto de restrições com $|S| = 2$ implica $x_{ij} + x_{ji} \leq 1$, para qualquer arco $(i, j) \in A$, o que indica que o arco (i, j) e o seu simétrico (j, i) não podem estar simultaneamente na solução. Estas restrições (3.5) quando conjugadas com as restrições (3.8), garantem a obtenção de uma solução com estrutura de árvore de suporte.

A restrição de peso (3.6) assegura que o peso total dos arcos na solução não pode exceder W . Por fim, as restrições (3.7) são as restrições de integralidade das variáveis.

Formulação de Cortes

A Formulação de Cortes (CS) é conhecida por *CutSet Formulation* [45] e, no contexto do Problema WMST, é dada através do seguinte modelo:

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij}$$

s.a.

$$\sum_{i \in V} x_{ij} = 1, \quad j \in V \setminus \{0\} \quad (3.9)$$

$$\sum_{(i,j) \in A(S, S^c)} x_{ij} \geq 1, \quad S \subset V, S \neq \emptyset, 0 \in S \quad (3.10)$$

$$\sum_{(i,j) \in A} w_{ij} x_{ij} \leq W \quad (3.11)$$

$$x_{ij} \in \{0, 1\}, \quad (i, j) \in A. \quad (3.12)$$

Relativamente à formulação anterior, nesta formulação, apenas houve alteração no conjunto de restrições dado por (3.10). As restrições (3.10) estabelecem que, qualquer corte direcionado $A(S, S^c)$, separando o nó $0 \in S$ de qualquer outro conjunto de nós, S^c , deve conter pelo menos um arco. Estas desigualdades são chamadas de restrições de corte. As restrições (3.10) juntamente com as restrições (3.9) garantem a obtenção de uma solução com estrutura de árvore de suporte.

Substituindo as restrições de integralidade nas duas formulações por,

$$x_{ij} \geq 0, \quad (i, j) \in A, \quad (3.13)$$

obtemos, em ambos os modelos, a respetiva relaxação linear que designamos por ES_L e CS_L , respetivamente. A relaxação linear de ambos os modelos fornece o mesmo limite inferior, $\vartheta(ES_L) = \vartheta(CS_L)$ [45]. Como o número de desigualdades em ambos os conjuntos aumenta exponencialmente com o tamanho do modelo, vamos usar na Secção 3.3 um procedimento que efetua a inserção das restrições de corte no modelo apenas quando estas não estão a ser satisfeitas pela solução obtida usando a atual relaxação.

3.2 Formulações Estendidas

Com o objetivo de garantir a conexidade/prevenção de circuitos na solução, em vez de usar a família de desigualdades (3.5) ou (3.10) com um número exponencial de desigualdades, podemos usar formulações compactas estendidas. Nesta secção propomos quatro formulações estendidas: uma formulação de fluxos, duas formulações baseadas nas desigualdades Miller-Tucker-Zemlin e uma formulação de fluxos que especifica nos índices o peso do caminho desde a raiz.

3.2.1 Formulação de Fluxos

Começamos por apresentar uma formulação orientada com fluxos desagregados para o Problema WMST. Esta formulação obtém-se adicionando as restrições de peso a uma conhecida formulação de fluxos para o Problema MST (veja-se, por exemplo, Magnanti e Wolsey [45]).

Para além das variáveis binárias x_{ij} usamos também as variáveis de fluxo orientadas, f_{ij}^k , para todos os arcos $(i, j) \in A$ e $k \in V \setminus \{0, i\}$, que indicam se o arco (i, j) é ou não utilizado no caminho da raiz para o nodo k , ou seja,

$$f_{ij}^k = \begin{cases} 1, & \text{se o arco } (i, j) \text{ é utilizado no caminho da raiz até ao nodo } k \\ 0, & \text{caso contrário} \end{cases}$$

e para simplificar a notação assumimos também que as variáveis f_{i0}^k , $i \in V \setminus \{0\}$ e $k \in V \setminus \{0, i\}$ têm sempre valor nulo.

A Formulação de Fluxo (MF - *Multicommodity Flow Formulation*) para o Problema WMST é a seguinte:

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij}$$

s.a.

$$\sum_{i \in V} x_{ij} = 1, \quad j \in V \setminus \{0\} \quad (3.14)$$

$$\sum_{i \in V \setminus \{k\}} f_{ij}^k - \sum_{i \in V \setminus \{0\}} f_{ji}^k = \begin{cases} -1 & j = 0 \\ 0 & j \neq 0, k \\ 1 & j = k \end{cases}, \quad j \in V, k \in V \setminus \{0\} \quad (3.15)$$

$$f_{ij}^k \leq x_{ij}, \quad (i, j) \in A, k \in V \setminus \{0, i\} \quad (3.16)$$

$$\sum_{(i,j) \in A} w_{ij} x_{ij} \leq W \quad (3.17)$$

$$f_{ij}^k \geq 0, \quad (i, j) \in A, k \in V \setminus \{0, i\} \quad (3.18)$$

$$x_{ij} \in \{0, 1\}, \quad (i, j) \in A. \quad (3.19)$$

As restrições de cardinalidade (3.14) garantem que um e apenas um arco chega a cada nodo exceto para o nodo raiz.

As restrições (3.15) são as restrições de conservação de fluxo e estabelecem que a solução deve ter um caminho entre o nodo 0 e cada nodo k , para todo $k \in V \setminus \{0\}$. O primeiro conjunto de restrições de conservação de fluxo garantem que a raiz envia uma unidade de fluxo para o nodo k , o segundo conjunto de restrições garantem que a quantidade de fluxo que entra num nó é igual à quantidade de fluxo que sai desse nó e o terceiro conjunto de restrições implicam que o nodo k recebe uma unidade de fluxo.

As restrições de ligação (3.16) garantem que é possível enviar fluxo para cada nodo k através do arco (i, j) , apenas se o arco se encontra na solução. Juntamente com as restrições de conservação de fluxo estas restrições asseguram a ligação dos arcos da solução. A restrição de peso (3.17) assegura que o peso total dos arcos na solução não pode exceder W . As restrições (3.18) garantem que a quantidade de fluxo enviado entre dois quaisquer nodos é não negativa. Por fim, as restrições (3.19) são as restrições de integralidade das variáveis.

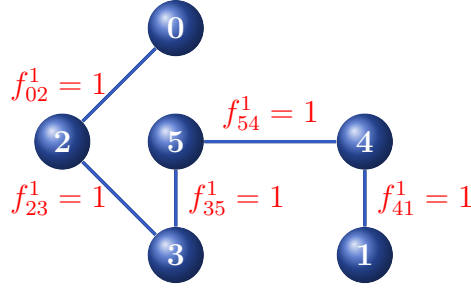


Figura 3.1: Exemplo com o valor das variáveis de fluxo na Formulação MF.

Na Figura 3.1 mostramos um exemplo onde podemos observar uma solução para o problema mostrando apenas o valor das variáveis de fluxo no caminho do nodo 0 para o nodo 1. Podemos observar que o nodo 0 envia uma unidade de fluxo para o nodo 1, pelo que o nodo 1 vai receber essa unidade de fluxo, para cada um dos restantes nodos do caminho a quantidade de fluxo que entra em cada nodo é igual à quantidade de fluxo que sai desse nó.

Substituindo as restrições de integralidade (3.19), por:

$$x_{ij} \geq 0, \quad (i, j) \in A, \quad (3.20)$$

obtemos a relaxação linear da Formulação MF que designamos por MF_L . Note que não é necessário incluir as restrições $x_{ij} \leq 1$ $((i, j) \in A)$ nem $f_{ij}^k \leq 1$ $((i, j) \in A, k \in V \setminus \{0, i\})$, pois estas restrições estão implícitas por (3.14) e (3.16).

A relaxação linear deste modelo fornece o mesmo limite inferior que os modelos da secção anterior, isto é, $\vartheta(MF_L) = \vartheta(ES_L) = \vartheta(CS_L)$ [45]. Resultado que não se altera por ter sido incluída a restrição de saco-mochila.

3.2.2 Formulação Miller-Tucker-Zemlin

Podemos obter uma formulação para o Problema WMST baseada nas restrições Miller-Tucker-Zemlin se substituirmos as restrições (3.15), (3.16) e (3.18) pelas restrições de eliminação de subcircuitos de Miller-Tucker-Zemlin (MTZ *Subtour Elimination Constraints*). Nesta formulação usamos as variáveis binárias usuais x_{ij}

$((i, j) \in A)$ que indicam se o arco (i, j) está ou não na solução e que são usadas para definir a topologia das soluções. Consideramos também as variáveis u_i ($i = 0, 1, \dots, n-1$), as quais definem a posição do nodo i na árvore, isto é, o número de arcos no caminho entre o nodo raiz e o nodo i .

A Formulação Miller-Tucker-Zemlin (MTZ) para o Problema WMST é a seguinte:

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij}$$

s.a.

$$\sum_{i \in V} x_{ij} = 1, \quad j \in V \setminus \{0\} \quad (3.21)$$

$$nx_{ij} + u_i \leq u_j + (n-1), \quad (i, j) \in A \quad (3.22)$$

$$\sum_{(i,j) \in A} w_{ij} x_{ij} \leq W \quad (3.23)$$

$$0 \leq u_i \leq n-1, \quad i \in V \quad (3.24)$$

$$x_{ij} \in \{0, 1\}, \quad (i, j) \in A. \quad (3.25)$$

Tal como vimos nas formulações anteriores as restrições de cardinalidade (3.21) garantem que um e apenas um arco chega a cada nodo exceto para o nodo raiz. A restrição de peso (3.23) assegura que o peso total dos arcos na solução não excede W . As restrições (3.25) são as restrições de integralidade das variáveis.

As restrições (3.22) são as conhecidas restrições de eliminação de subcircuitos dadas em Miller et al. [46] para o Problema do Caixeiro Viajante (TSP - *Travelling Salesman Problem*) e que foram adaptadas para o Problema MST com restrições de salto [24]. As restrições (3.24) asseguram que o número de arcos no caminho entre o nodo raiz e o nodo i é não negativo e que nunca excede o número de arestas da árvore.

Para qualquer árvore de suporte de peso admissível podemos sempre encontrar valores para as variáveis $u_j, \forall j \in V$, tais que (3.22) e (3.24) são satisfeitas. Se u_j é o número de arcos no caminho entre o nodo raiz e qualquer nodo j , $u_j = u_i + 1$ para todos os arcos (i, j) tal que $x_{ij} = 1$ e $u_0 = 0$. As restrições (3.22) e (3.24), para todos os arcos (i, j) tal que $x_{ij} = 0$, estão implícitas pela restrição de cardinalidade (3.21). Assim, as restrições (3.22) e (3.24) são válidas para o Problema WMST.

Se $x_{ij} = 1$, então $u_i \leq u_j - 1$.

Se $x_{ij} = 0$, então $u_i \leq u_j + (n - 1)$.

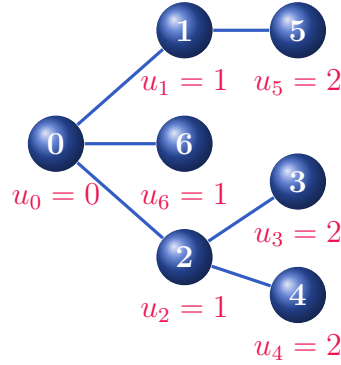


Figura 3.2: Exemplo do cálculo das posições dos nodos na árvore.

Na Figura 3.2 podemos observar um exemplo onde indicamos as várias posições dos nodos na árvore. Por exemplo, a posição do nodo 5 na árvore é dada pelo número de arcos no caminho entre o nodo raiz (0) e o nodo 5 que é igual a 2 arcos, $u_5 = 2$.

No exemplo da Figura 3.2, se a solução for um caminho, com nodo origem 0 e nodo destino 6, a posição do último nodo do caminho é $u_6 = 6$. As posições dos nodos nunca excedem o número de arestas da árvore ($u_j \leq n - 1$, $j \in V$).

Note-se que para a formulação do Problema WMST não é necessário conhecer as posições exatas dos nodos na árvore, o que é importante é que se o arco (i, j) está na solução, a diferença máxima entre as posições dos nodos j e i seja superior ou igual a um, isto é,

$$u_j - u_i \geq 1, \quad (i, j) \in A.$$

Substituindo as restrições de integralidade (3.25) por:

$$x_{ij} \geq 0, \quad (i, j) \in A, \quad (3.26)$$

obtemos a relaxação linear da Formulação MTZ que designamos por MTZ_L . Note que não é necessário incluir as restrições $x_{ij} \leq 1$ ($(i, j) \in A$), pois estas restrições estão implícitas por (3.21).

As restrições (3.22) podem ser fortalecidas (ver Desrochers e Laporte [15] e Gouveia [24]) da seguinte forma:

Desigualdades Levantadas 1 - MTZl1

$$(n - 2)x_{ji} + nx_{ij} + u_i \leq u_j + (n - 1), \quad (i, j) \in A \quad (3.27)$$

A validade destas restrições decorre do facto das variáveis x_{ij} e x_{ji} não poderem tomar simultaneamente o valor 1, $x_{ij} + x_{ji} \leq 1$. Se $x_{ji} = 0$, as restrições (3.27) estão implícitas por (3.22) e no caso de $x_{ji} = 1$, temos que $x_{ij} = 0$, o que significa que existe um arco direcionado que liga os dois nodos j e i , sendo que a diferença máxima entre a posição dos nodos i e j é inferior ou igual a 1, o que se verifica nas restrições (3.27), pois para este caso $u_i - u_j \leq 1$.

Desigualdades Levantadas 2 - MTZl2

$$\sum_{k \in V \setminus \{i, j\}} x_{kj} + nx_{ij} + u_i \leq u_j + (n - 1), \quad (i, j) \in A \quad (3.28)$$

Das restrições (3.21) temos que $\sum_{k \in V \setminus \{i, j\}} x_{kj} \leq 1$, pelo que $\sum_{k \in V \setminus \{i, j\}} x_{kj}$ e a variável x_{ij} não podem tomar simultaneamente o valor 1. Se $\sum_{k \in V \setminus \{i, j\}} x_{kj} = 0$, as restrições (3.28) estão implícitas por (3.22). Caso $\sum_{k \in V \setminus \{i, j\}} x_{kj} = 1$ temos que para algum $k \in V \setminus \{i, j\}$, $x_{kj} = 1$ e $x_{ij} = 0$, o que significa que existe um arco direcionado que liga os nodos k e j , e desta forma a diferença máxima entre a posição dos nodos i e j não pode ser superior a $n - 2$, o que está de acordo com as restrições (3.28), onde para este caso se obtém $u_i - u_j \leq n - 2$. Portanto, estas restrições são válidas para o Problema WMST.

Desigualdades Levantadas 3 - MTZl3

$$\sum_{k \in V \setminus \{i, j\}} (x_{ik} + x_{kj}) + nx_{ij} + u_i \leq u_j + (n - 1), \quad (i, j) \in A \quad (3.29)$$

Das restrições (3.21) temos que $\sum_{k \in V \setminus \{i,j\}} x_{kj} \leq 1$ e também $\sum_{(i,j) \in A} x_{ij} = n - 1$, então $\sum_{k \in V \setminus \{i,j\}} x_{ik} \leq n - 2$, pois o nodo i pode ligar a todos os nodos exceto ao nodo j . Deste modo, temos que o somatório $\sum_{k \in V \setminus \{i,j\}} (x_{ik} + x_{kj})$ pode tomar valores inferiores ou iguais a $n - 1$.

Se $\sum_{k \in V \setminus \{i,j\}} (x_{ik} + x_{kj}) = 0$, as restrições (3.29) estão implícitas por (3.22).

Se $\sum_{k \in V \setminus \{i,j\}} x_{ik} = 0$, as restrições (3.29) estão implícitas por (3.28).

Se $\sum_{k \in V \setminus \{i,j\}} (x_{ik} + x_{kj}) = p \leq n - 1$ com $p \in \mathbb{Z}^+$, ao substituírmos nas restrições (3.29) o somatório por p , obtemos $p + nx_{ij} + u_i \leq u_j + n - 1$. No caso de $\sum_{k \in V \setminus \{i,j\}} x_{kj} = 1$ temos que para algum $k \in V \setminus \{i,j\}$, $x_{kj} = 1$ e $x_{ij} = 0$, temos que $u_i - u_j \leq n - 1 - p < n - 1$. No caso de $\sum_{k \in V \setminus \{i,j\}} x_{kj} = 0$ e $x_{ij} = 1$ temos que $u_j - u_i \geq p + 1$. Ambas estas restrições são válidas para o Problema WMST.

Desigualdades Levantadas 4 - MTZl4

$$\sum_{k \in V \setminus \{i,j\}} x_{kj} + (n - 3)x_{ji} + nx_{ij} + u_i \leq u_j + (n - 1), \quad (i, j) \in A \quad (3.30)$$

Das restrições (3.21) temos que $\sum_{k \in V \setminus \{i,j\}} x_{kj} \leq 1$. Se $\sum_{k \in V \setminus \{i,j\}} x_{kj} = 0$, as restrições (3.30) transformam-se nas restrições

$$(n - 3)x_{ji} + nx_{ij} + u_i \leq u_j + (n - 1), \quad (i, j) \in A \quad (3.31)$$

A validade das restrições (3.31) decorre do facto das variáveis x_{ij} e x_{ji} não poderem tomar simultaneamente o valor 1, $x_{ij} + x_{ji} \leq 1$. Se $x_{ji} = 0$, as restrições (3.31) estão implícitas por (3.22) e no caso de $x_{ji} = 1$, temos que $x_{ij} = 0$, o que significa que existe um arco direcionado que liga os dois nodos j e i , sendo que a diferença máxima entre a posição dos nodos i e j é inferior ou igual a 2, o que se verifica nas restrições (3.31), pois para este caso $u_i - u_j \leq 2$. No caso do somatório $\sum_{k \in V \setminus \{i,j\}} x_{kj} = 1$ temos que para algum $k \in V \setminus \{i,j\}$, $x_{kj} = 1$ e $x_{ij} = 0$, pelo que as restrições (3.30) ficam $(n - 3)x_{ji} + u_i \leq u_j + n - 2$. Se $x_{ji} = 1$ temos que existe um arco direcionado que liga os nodos j e i , e desta forma a diferença máxima entre a posição dos nodos i e j não pode ser superior a 1, o que está de acordo com as restrições (3.30), onde para este caso se obtém $u_i - u_j \leq 1$. No caso de $x_{ji} = 0$, a diferença máxima entre a posição dos nodos i e j não pode ser superior a $n - 2$, o que está de acordo com as restrições (3.30), onde para este caso se obtém $u_i - u_j \leq n - 2$. Portanto, as restrições (3.30) são

válidas para o Problema WMST.

Às relaxações lineares das Formulações MTZl1, MTZl2, MTZl3 e MTZl4 designamos por MTZl1_L, MTZl2_L, MTZl3_L e MTZl4_L, respetivamente.

3.2.3 Formulação de Peso Miller-Tucker-Zemlin

Nesta formulação além das variáveis binárias x_{ij} ($(i, j) \in A$), que definem a topologia da solução, consideramos as variáveis p_i , ($i = 0, \dots, n-1$), as quais especificam o estado de peso do nodo i na árvore, isto é, indicam o valor do peso do caminho entre a origem e o nodo i , dado pela soma dos pesos dos arcos no caminho. A Formulação de Peso Miller-Tucker-Zemlin (WMTZ) é a que se segue:

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij}$$

s.a.

$$\sum_{i \in V} x_{ij} = 1, \quad j \in V \setminus \{0\} \quad (3.32)$$

$$w_{ij} x_{ij} + p_i \leq p_j + W(1 - x_{ij}), \quad (i, j) \in A \quad (3.33)$$

$$\sum_{(i,j) \in A} w_{ij} x_{ij} \leq W \quad (3.34)$$

$$0 \leq p_i \leq W, \quad i \in V \quad (3.35)$$

$$x_{ij} \in \{0, 1\}, \quad (i, j) \in A. \quad (3.36)$$

As restrições (3.32), (3.34) e (3.36) são as mesmas da formulação anterior. As restrições (3.33) são baseadas nas conhecidas restrições de eliminação de subcircuitos dadas em Miller et al. [46] para o Problema TSP. As restrições (3.35) impõem limites nas variáveis p_i , isto é, asseguram que a soma dos pesos no caminho entre o nodo origem e o nodo i é não negativo e nunca excede o limite W .

Proposição 3.1.

As restrições (3.33) evitam a existência de circuitos.

Demonstração.

Somando as restrições (3.33) para os arcos de um circuito \mathcal{C} ($x_{ij} = 1, (i, j) \in \mathcal{C}$) obtemos $\sum_{(i,j) \in \mathcal{C}} w_{ij} \leq 0$, o que contradiz o facto dos pesos w_{ij} serem positivos. \square

Para qualquer árvore de suporte de peso admissível podemos sempre encontrar valores para as variáveis $p_j, \forall j \in V$, tais que (3.33) e (3.35) são satisfeitas. Se p_j é o peso do caminho da origem a qualquer nodo j , $p_j = p_i + w_{ij}$ para todos os arcos (i, j) tal que $x_{ij} = 1$ e $p_0 = 0$. As restrições (3.33) e (3.35), para todos os arcos (i, j) tal que $x_{ij} = 0$, estão implícitas pela restrição de peso (3.34). Assim, as restrições (3.33) e (3.35) são válidas para o Problema WMST.

Se $x_{ij} = 1$, então $w_{ij} + p_i \leq p_j \Rightarrow p_i < p_j$.

Se $x_{ij} = 0$, então $p_i \leq p_j + W$.

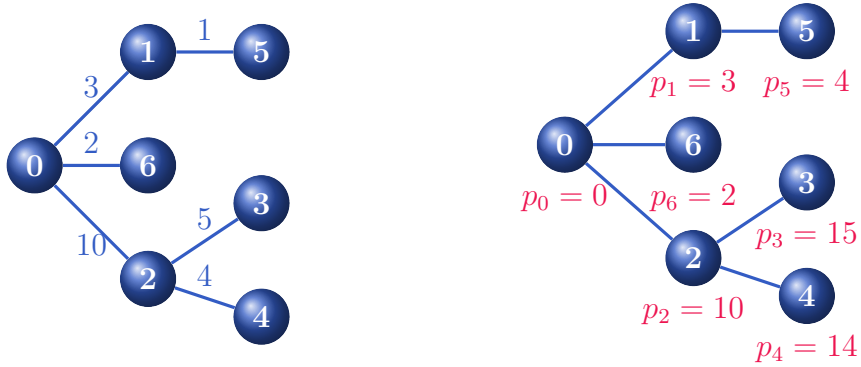


Figura 3.3: Exemplo do cálculo dos estados dos pesos dos nodos na árvore.

Na Figura 3.3 podemos observar como são obtidos os estados dos pesos de cada nodo na árvore. Por exemplo, o estado de peso do nodo 3 corresponde à soma dos pesos dos arcos $(0, 2)$ e $(2, 3)$, $p_3 = p_2 + w_{23} = 10 + 5 = 15$.

Se a solução do exemplo dado na Figura 3.3 for um caminho, com nodo origem 0 e nodo destino 6, o estado de peso do último nodo do caminho é igual ao peso da árvore de suporte obtida, $p_6 = W(T) \leq W$.

Note-se que para a formulação do Problema WMST não é necessário conhecer os estados de peso dos nodos na árvore, o que é importante é que se o arco (i, j) está na solução, a diferença máxima entre os estados de peso p_j e p_i seja superior ou igual a w_{ij} , isto é,

$$p_j - p_i \geq w_{ij}, \quad (i, j) \in A.$$

Substituindo as restrições de integralidade (3.36) por:

$$x_{ij} \geq 0, \quad (i, j) \in A, \quad (3.37)$$

obtemos a relaxação linear da Formulação WMTZ que designamos por WMTZ_L . Note que não é necessário incluir as restrições $x_{ij} \leq 1$ $((i, j) \in A)$, pois estas restrições estão implícitas por (3.32).

Notamos que as Formulações MTZ_L e WMTZ_L são incomparáveis e para comprovar este facto apresentamos o seguinte exemplo.

Exemplo 3.1.

Na Tabela 3.1 podemos observar os valores obtidos pela relaxação linear das Formulações MTZ e WMTZ em algumas instâncias de 10 e 20 nodos dos três grupos de instâncias, Quase Caminhos, Aleatórias e Euclidianas.

Instância	MTZ_L	WMTZ_L
QC10-1	70,850	68,087
R10-2	11856,200	11861,700
E10-1	29161,700	29076,100
QC20-1	1004,530	1017,800
R20-1	25226,500	25308,900
E20-1	40383,700	40368,600

Tabela 3.1: Comparação dos valores obtidos usando as Formulações MTZ_L e WMTZ_L .

Nas instâncias QC10-1, E10-1 e E20-1 o valor obtido pela Formulação MTZ_L é superior ao valor obtido pela Formulação WMTZ_L , mas nas instâncias R10-2, QC20-1

e R20-1 verifica-se o contrário, ou seja, o valor obtido pela Formulação MTZ_L é inferior ao valor obtido pela Formulação $WMTZ_L$. Deste modo verificamos que as Formulações MTZ_L e $WMTZ_L$ não são comparáveis em termos de valor da relaxação linear.

Tal como foi feito anteriormente para as restrições (3.22) da Formulação MTZ , as restrições (3.33) também podem ser fortalecidas (ver Desrochers e Laporte [15] e Gouveia [24]) da seguinte forma:

Desigualdades Levantadas 1 - $WMTZl1$

$$w_{ji}x_{ji} + w_{ij}x_{ij} + p_i \leq p_j + W(1 - x_{ij}), \quad (i, j) \in A \quad (3.38)$$

A validade destas restrições decorre do facto das variáveis x_{ij} e x_{ji} não poderem tomar simultaneamente o valor 1. Se $x_{ji} = 0$, as restrições (3.38) estão implícitas por (3.33) e no caso de $x_{ji} = 1$, temos que $x_{ij} = 0$, o que significa que existe um arco direccionado que liga os dois nodos j e i , sendo que a diferença máxima entre os estados de peso dos nodos i e j é inferior ou igual a $W - w_{ji}$, o que se verifica nas restrições (3.38), pois para este caso $p_i - p_j \leq W - w_{ji}$.

Desigualdades Levantadas 2 - $WMTZl2$

$$(W - w_{ji})x_{ji} + w_{ij}x_{ij} + p_i \leq p_j + W(1 - x_{ij}), \quad (i, j) \in A \quad (3.39)$$

Temos que as variáveis x_{ij} e x_{ji} não podem tomar simultaneamente o valor 1. Se $x_{ji} = 0$, as restrições (3.39) estão implícitas por (3.33) e no caso de $x_{ji} = 1$, temos que $x_{ij} = 0$, o que significa que existe um arco direccionado que liga os dois nodos j e i , sendo que a diferença máxima entre os estados de peso dos nodos i e j não excede o peso do arco (j, i) , o que se verifica nas restrições (3.39), pois para este caso $p_i - p_j \leq w_{ji}$. Portanto, estas restrições são válidas para o Problema $WMST$.

Desigualdades Levantadas 3 - WMTZl3

$$\sum_{k \in V \setminus \{i,j\}} w_{kj}x_{kj} + w_{ij}x_{ij} + p_i \leq p_j + W(1 - x_{ij}), \quad (i, j) \in A \quad (3.40)$$

Das restrições (3.32) temos que $\sum_{k \in V \setminus \{i,j\}} x_{kj} \leq 1$. No caso de $\sum_{k \in V \setminus \{i,j\}} x_{kj} = 0$, então $\sum_{k \in V \setminus \{i,j\}} w_{kj}x_{kj} = 0$ e as restrições (3.40) estão implícitas por (3.33). Caso para algum $k \in V \setminus \{i, j\}$, $x_{kj} = 1$, $\sum_{k \in V \setminus \{i,j\}} w_{kj}x_{kj} = w_{kj}$ e $x_{ij} = 0$, o que significa que existe um arco direcionado que liga os nodos k e j , e desta forma a diferença máxima entre os estados de peso dos nodos i e j não pode ser superior a $W - w_{kj}$, o que está de acordo com as restrições (3.40), onde para este caso se obtém $p_i - p_j \leq W - w_{kj} < W$. Portanto, estas restrições são válidas para o Problema WMST.

Desigualdades Levantadas 4 - WMTZl4

$$\sum_{k \in V \setminus \{i,j\}} (w_{kj}x_{kj} + w_{ik}x_{ik}) + w_{ji}x_{ji} + w_{ij}x_{ij} + p_i \leq p_j + W(1 - x_{ij}), \quad (i, j) \in A \quad (3.41)$$

Das restrições (3.32) temos que $\sum_{k \in V \setminus \{i,j\}} x_{kj} \leq 1$ e da restrição (3.34) temos que $\sum_{k \in V \setminus \{i,j\}} w_{ik}x_{ik} < W$, pois o nodo i pode ligar a todos os nodos exceto ao nodo j . Deste modo temos que o somatório $\sum_{k \in V \setminus \{i,j\}} (w_{kj}x_{kj} + w_{ik}x_{ik})$ toma valores inferiores ou iguais a $W + w_{kj}$ para algum $k \in V \setminus \{i, j\}$.

Se $\sum_{k \in V \setminus \{i,j\}} (w_{ik}x_{ik} + w_{kj}x_{kj}) = 0$, as restrições (3.41) estão implícitas por (3.38).

Se $\sum_{k \in V \setminus \{i,j\}} (w_{ik}x_{ik} + w_{kj}x_{kj}) = q \leq W + w_{kj}$ com $q \in \mathbb{Z}^+$, ao substituir nas restrições (3.41) o somatório por q , obtemos $q + w_{ji}x_{ji} + w_{ij}x_{ij} + p_i \leq p_j + W(1 - x_{ij})$. Caso $\sum_{k \in V \setminus \{i,j\}} x_{kj} = 1$ temos que para algum $k \in V \setminus \{i, j\}$, $x_{kj} = 1$, $\sum_{k \in V \setminus \{i,j\}} w_{kj}x_{kj} = w_{kj}$ e $x_{ij} = 0$, temos que $q + w_{ji}x_{ji} + p_i \leq p_j + W$. Se $x_{ji} = 0$, então $p_i - p_j \leq W - q < W$ e se $x_{ji} = 1$ temos que $p_i - p_j \leq W - q - w_{ji} < W$. No caso de $\sum_{k \in V \setminus \{i,j\}} x_{kj} = 0$, então $x_{ij} = 1$ e $x_{ji} = 0$, pelo que temos $p_j - p_i \geq q + w_{ij}$. Assim, estas restrições são válidas para o Problema WMST.

As relaxações lineares das Formulações WMTZl1, WMTZl2, WMTZl3 e WMTZl4 designamos por WMTZl1_L, WMTZl2_L, WMTZl3_L e WMTZl4_L, respetivamente.

3.2.4 Formulação de Fluxo com especificação de Pesos

Esta nova formulação estendida será denominada de WE. Para além das variáveis binárias usuais x_{ij} ($(i, j) \in A$) usamos variáveis de fluxo de peso direcionado $z_{ijk}^{h_1, h_2}$ (para todo $(i, j) \in A$, $k \in V \setminus \{0, i\}$ e $h_1 \leq h_2$, $h_1, h_2 = 0, \dots, W$), as quais especificam se o arco (i, j) é ou não utilizado no caminho da raiz até ao nodo k , sendo h_1 o peso do caminho da raiz ao nodo i e h_2 o peso do caminho do nodo raiz ao nodo j . Para simplificar a notação assumimos que as variáveis $z_{i0k}^{h_1, h_2}$ para todo $i \in V \setminus \{0\}$, $k \in V \setminus \{0, i\}$, $h_1 \leq h_2$, $h_1, h_2 = 0, \dots, W$ e as variáveis $z_{ijk}^{h_1, h_2}$ para todo $h_2 < h_1$ e $h_2 > W$ têm sempre valor nulo. A Formulação WE é dada pelo seguinte conjunto de restrições:

$$\begin{aligned} \min \quad & \sum_{(i,j) \in A} c_{ij} x_{ij} \\ \text{s.a.} \quad & \sum_{i \in V} x_{ij} = 1, \quad j \in V \setminus \{0\} \end{aligned} \quad (3.42)$$

$$\sum_{i \in V \setminus \{k\}} \sum_{h=0}^{\bar{h}} z_{ijk}^{h-w_{ij}, h} - \sum_{i \in V \setminus \{0\}} \sum_{h=0}^{\bar{h}} z_{jik}^{h, h+w_{ji}} = \begin{cases} -1 & j = 0, \bar{h} = 0 \\ 0 & j \neq 0, k, \bar{h} = W \\ 1 & j = k, \bar{h} = W \end{cases}, \quad j \in V, k \in V \setminus \{0\} \quad (3.43)$$

$$\sum_{h=0}^W z_{ijk}^{h, h+w_{ij}} \leq x_{ij}, \quad (i, j) \in A, k \in V \setminus \{0, i\} \quad (3.44)$$

$$z_{ijk}^{h_1, h_2} \geq 0, \quad (i, j) \in A, k \in V \setminus \{0, i\}, \quad (3.45)$$

$$x_{ij} \in \{0, 1\}, \quad (i, j) \in A. \quad (3.46)$$

As restrições de cardinalidade (3.42) garantem que um e apenas um arco chega a cada nodo exceto para o nodo raiz.

As restrições de conservação de fluxo (3.43) estabelecem que a solução deve conter um caminho entre o nodo 0 e o nodo k (para todo $k \in V \setminus \{0\}$). O primeiro conjunto de restrições de conservação de fluxo garantem que a raiz envia uma unidade de fluxo para o nodo k , o segundo conjunto de restrições garantem que a quantidade de fluxo que entra num nó é igual à quantidade de fluxo que sai desse nó e o terceiro conjunto de restrições implicam que o nodo k recebe uma unidade de fluxo.

As restrições de ligação (3.44) garantem que é possível enviar fluxo para cada nodo

k através do arco (i, j) apenas se o arco está na solução.

Juntamente com as restrições de conservação de fluxo estas restrições asseguram a ligação dos arcos da solução. As restrições (3.45) garantem que a quantidade de fluxo enviado entre dois quaisquer nodos é não negativa. Por fim, as restrições (3.46) são as restrições de integralidade das variáveis.

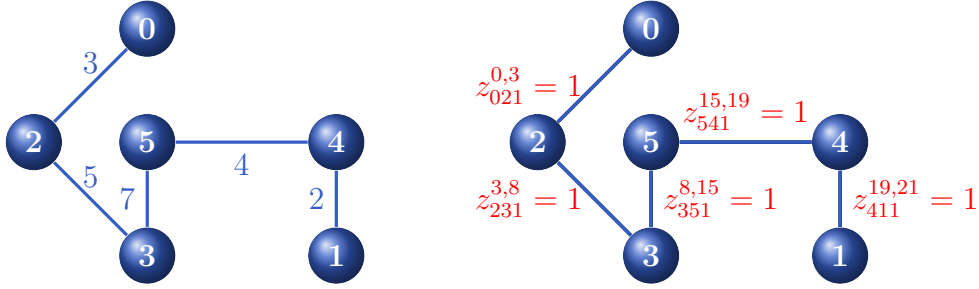


Figura 3.4: Exemplo com o valor das variáveis de fluxo na Formulação WE.

Na Figura 3.4 mostramos um exemplo onde podemos observar uma solução e indicamos apenas o valor das variáveis de fluxo no caminho do nodo raiz para o nodo 1. Podemos observar que o nodo 0 envia uma unidade de fluxo para o nodo 1, pelo que o nodo 1 vai receber essa unidade de fluxo, temos também que para os nodos que estão no caminho do nodo origem para o nodo 1 a quantidade de fluxo que entra em cada nodo é igual à quantidade de fluxo que sai desse nó. Assim as variáveis $z_{ij}^{h_1, h_2}$ para os arcos (i, j) no caminho do nodo origem para o nodo 1 terão valor um para valores apropriados de h_1 e h_2 . Por exemplo, para o nodo 3 o valor de h_1 corresponde ao peso do caminho do nodo raiz ao nodo 3, isto é, $h_1 = w_{02} + w_{23} = 3 + 5 = 8$ e o valor de h_2 corresponde ao peso do caminho do nodo raiz ao nodo 5, ou seja, $h_2 = w_{02} + w_{23} + w_{35} = 3 + 5 + 7 = 15$.

Proposição 3.2.

A Formulação WE é válida para o Problema WMST.

Demonstração.

Dada uma solução para o Problema WMST, facilmente, podemos construir uma solução (x, h) para a Formulação WE, onde as variáveis binárias usuais x_{ij} ($(i, j) \in A$) indicam se o arco pertence ou não à solução e as variáveis de fluxo de peso direcionado $z_{ijk}^{h_1, h_2}$ (para todo $(i, j) \in A$, $k \in V \setminus \{0, i\}$ e $h_1 \leq h_2$, $h_1, h_2 = 0, \dots, W$), indicam se o

arco (i, j) é ou não utilizado no caminho da raiz até ao nodo k , sendo h_1 o peso do caminho da raiz ao nodo i e h_2 o peso do caminho da raiz ao nodo j .

Tomemos uma solução (x, h) que satisfaz as restrições da Formulação WE acima e consideremos as variáveis de fluxo orientado f_{ij}^k , para todos os arcos $(i, j) \in A$ e $k \in V \setminus \{0, i\}$ definidas do seguinte modo

$$f_{ij}^k = \sum_{h=0}^W z_{ijk}^{h, h+w_{ij}}. \quad (3.47)$$

Quando substituimos (3.47) nas restrições (3.43) obtemos as restrições de conservação de fluxo usuais. Desta forma, conjuntamente, com as restrições de cardinalidade (3.42) e integralidade (3.46) obtemos uma árvore de suporte. Como $f_{ij}^k \geq 0$ para qualquer arco $(i, j) \in A$ e $k \in V \setminus \{0, i\}$, então temos que $z_{ijk}^{h_1, h_2} \geq 0$.

Além disso, se multiplicarmos cada restrição (3.44) por w_{ij} e depois adicioná-las para todos os arcos $(i, j) \in A$ obtemos a restrição de peso. O que nos garante que a árvore obtida satisfaz a restrição de peso. Desta forma verificamos que a solução (x, h) é capaz de determinar uma solução para o Problema WMST. \square

Substituindo as restrições de integralidade (3.46) por:

$$x_{ij} \geq 0, (i, j) \in A, \quad (3.48)$$

obtemos a relaxação linear da Formulação WE que designamos por WE_L . Note que não é necessário incluir as restrições $x_{ij} \leq 1$ $((i, j) \in A)$ nem $z_{ijk}^{h_1, h_2} \leq 1$ $((i, j) \in A, k \in V \setminus \{0, i\})$, pois estas restrições estão implícitas por (3.42) e (3.44).

Exemplo 3.2.

Neste exemplo apresentamos os diferentes valores obtidos pela relaxação linear das diferentes formulações para o Exemplo 2.1. Usando formulações diferentes podemos encontrar soluções ótimas da relaxação linear também diferentes. Neste exemplo de 5 nodos, para as diferentes formulações apresentadas ao longo deste capítulo, é possível encontrar quatro diferentes limites inferiores para o valor do custo, os quais se encontram na seguinte tabela.

Formulações	Relaxação Linear	
	Custo	Peso
MTZ, MTZl2, MTZl3, WMTZ e WMTZl1,	25,333	20
WMTZl3 e WMTZl4	25,3425	20
MTZl4	25,3913	20
ES, MF, WE, MTZl1, WMTZl2	25,4	20

Tabela 3.2: Comparação das formulações usando um exemplo de 5 nodos.

Os limites inferiores mais baixos são obtidos utilizando as Formulações MTZ, MTZl2, MTZl3, WMTZ e WMTZl1, com custo 25,333 e peso 20 e os limites inferiores mais elevados são obtidos através das Formulações ES, MF, WE, MTZl1 e WMTZl2. A melhor solução ótima da relaxação linear encontra-se representada na Figura 3.5.

$$\begin{aligned}
x_{01} &= 1 \\
x_{04} &= 0,2 \\
x_{12} &= 1 \\
x_{13} &= 0,8 \\
x_{34} &= 0,8 \\
x_{43} &= 0,2 \\
\text{restantes } x_{ij} &= 0, (i, j) \in A
\end{aligned}$$

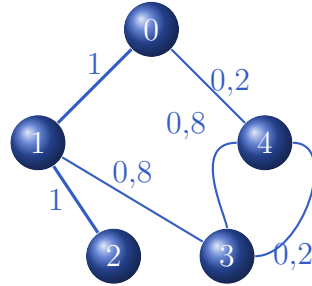


Figura 3.5: Solução ótima da relaxação linear obtida pelas Formulações ES, MF, WE, MTZl1 e WMTZl2.

O custo da solução ótima da relaxação linear obtida na Figura 3.5 é de 25,4 e o peso é de 20.

3.3 Procedimento com Introdução de Cortes

Neste procedimento as restrições de corte são introduzidas no modelo como cortes usando separação. Apenas quando a atual solução da relaxação considerada não satisfaz alguma restrição de corte é que essa restrição será adicionada ao modelo. Desta forma apenas são incluídas no modelo as restrições de corte necessárias para obter a solução ótima do Problema WMST.

Para adicionarmos as restrições de corte teremos de saber como identificar a restrição de corte a incluir no modelo. Para o fazer é necessário identificar um corte mínimo na solução atual obtida pela solução da relaxação. Um corte mínimo $A(S, S^c)$ pode ser determinado se aplicarmos o Algoritmo de *Ford-Fulkerson* [20] para encontrar o fluxo máximo entre a raiz, o nó 0, e cada um dos outros nós $i \in V \setminus \{0\}$. Denotamos por $f_{max}(0, i)$, $i \in V \setminus \{0\}$, o fluxo máximo entre o nodo zero/raiz e cada um dos restantes nodos do grafo.

Para descrever os principais passos para efetuar um procedimento genérico com introdução de cortes, vamos começar por denotar por FORM o modelo que será usado para inclusão dos cortes. Os diferentes passos necessários para este procedimento são os seguintes.

Procedimento com Introdução de Cortes

Passo 1: *Inicializar o conjunto de corte*

Considerar os conjuntos $S = \emptyset$ e $S^c = V$.

Passo 2: *Relaxação*

1) Introduzir o corte

$$\sum_{(i,j) \in A(S, S^c)} x_{ij} \geq 1 \quad S \subset V, S \neq \emptyset, 0 \in S.$$

2) Resolver uma relaxação da Formulação FORM.

Passo 3: *Obter o corte*

Para cada $i \in V \setminus \{0\}$ obter $f_{max}(0, i)$ considerando como capacidade dos arcos o valor da solução obtida pela relaxação no Passo 2.

Se $f_{max}(0, i) < 1$, então

Se $f_{max}(0, i) = 0$, então

o conjunto S é constituído pelos nodos $i \in V \setminus \{0\}$ para os quais foi possível enviar fluxo do nodo 0.

Caso contrário,

identificar o conjunto S para obter o conjunto de corte $A(S, S^c)$.

Voltar para o **Passo 2**.

Este processo termina quando o grafo estiver conectado e o fluxo máximo entre 0 e cada nodo $i \in V \setminus \{0\}$ for 1.

Passo 4: *Introduzir as restrições de subcircuito*

- 1) Introduzir as restrições $x_{ij} + x_{ji} \leq 1$, $i, j \in V$, $i \neq j$.
- 2) Resolver uma relaxação da Formulação FORM.

No Passo 1 começamos por inicializar o conjunto de corte, onde $S = \emptyset$ e $S^c = V$. No Passo 2 introduzimos o corte e resolvemos uma relaxação de uma formulação que designamos por FORM. Notamos que inicialmente (na primeira iteração) a relaxação não contém qualquer corte, é apenas constituída pelas restrições do modelo FORM sem cortes. Enquanto o fluxo máximo entre 0 e cada nodo $i \in V \setminus \{0\}$ não for um temos de usar o Algoritmo *Ford-Fulkerson* [20] para obter o fluxo máximo entre o nodo zero/raiz e cada um dos restantes nodos do grafo considerando como capacidade dos arcos o valor da solução obtida pela relaxação no Passo 2. Se o valor do fluxo máximo for inferior a um, temos de considerar dois casos. Um dos casos ocorre quando não é possível obter um fluxo entre o nodo 0 e o nodo $i \in V \setminus \{0\}$, então vamos considerar o conjunto S constituído pelos nodos $i \in V \setminus \{0\}$, para os quais foi possível enviar fluxo do nodo 0. O outro caso ocorre quando o fluxo máximo é positivo, onde temos de identificar o conjunto de corte. Em qualquer um destes casos voltamos ao Passo 2 para introduzir um novo corte e resolver uma nova relaxação. No Passo 4 são introduzidas as restrições de subcircuito e resolve-se uma nova relaxação da Formulação FORM. Deste modo foi obtido um valor para o limite inferior para o custo, o qual é o valor obtido pela relaxação linear deste procedimento com introdução de cortes.

Observe-se que o procedimento com introdução de cortes pode ser aplicado a um modelo que no Passo 2 denotámos por FORM. Nesta tese aplicamos este procedimento às Formulações CS, MTZ e WMTZ. Deste modo, podemos definir os seguintes procedimentos:

- **Procedimento P-CS**

Neste procedimento substituímos o modelo FORM no Passo 2 pela Formulação CS. A relaxação linear deste procedimento será designada por P-CS_L.

- **Procedimento P-MTZ+C**

Neste procedimento substituímos o modelo FORM no Passo 2 pela Formulação MTZ. A relaxação linear deste procedimento será designada por P-MTZ+C_L.

- **Procedimento P-WMTZ+C**

Neste procedimento substituímos o modelo FORM no Passo 2 pela Formulação WMTZ. A relaxação linear deste procedimento será designada por P-WMTZ+C_L.

Exemplo 3.3.

Para exemplificar o funcionamento do procedimento com introdução de cortes vamos considerar a instância E10-1 do grupo de instâncias Euclidianas.

Passo 1:

Considerar os conjuntos $S = \emptyset$ e $S^c = V$.

Passo 2:

Os valores de cada variável na solução da relaxação linear da Formulação WMTZ são os seguintes: $x_{03} = 1$; $x_{06} = 1$; $x_{17} = 0,5$; $x_{19} = 0,5$; $x_{91} = 0,5$; $x_{25} \approx 0,59$; $x_{52} \approx 0,41$; $x_{28} = 1$; $x_{32} \approx 0,59$; $x_{71} = 0,5$; $x_{79} = 0,5$; $x_{97} = 0,5$; $x_{85} \approx 0,41$ e $x_{94} = 1$ (restantes variáveis com valor nulo). O valor do limite inferior obtido para o custo é de 33619,7 com peso 24251 e na Figura 3.6 podemos encontrar a representação da solução através de um grafo, onde os arcos que não tomam o valor um estão representados a vermelho.



Figura 3.6: Representação da solução da relaxação linear da Formulação WMTZ.

Passo 3:

Como podemos observar na Figura 3.6, não é possível enviar fluxo do nodo 0 para o nodo 9, isto é, do nodo origem para o nodo 9, $f_{max}(0, 9) = 0$. Deste modo, a solução, neste caso, não é conexa pelo que formamos o conjunto $S = \{0, 2, 3, 5, 6, 8\}$ e o conjunto $S^c = \{1, 4, 7, 9\}$ e voltamos ao Passo 2.

Passo 2:

1) Introduzir o corte 1:

$$x_{01} + x_{04} + x_{07} + x_{09} + x_{21} + x_{24} + x_{27} + x_{29} + x_{31} + x_{34} + x_{37} + x_{39} + x_{51} + x_{54} + x_{57} + x_{59} + x_{61} + x_{64} + x_{67} + x_{69} + x_{81} + x_{84} + x_{87} + x_{89} \geq 1.$$

2) Os valores de cada variável na solução da relaxação linear da Formulação WMTZ com a introdução do corte 1 são os seguintes: $x_{03} = 1$; $x_{06} = 1$; $x_{17} = 1$; $x_{25} \approx 0,94$; $x_{52} \approx 0,06$; $x_{28} = 1$; $x_{32} \approx 0,94$; $x_{69} = 1$; $x_{85} = 0,06$; $x_{91} = 1$; e $x_{94} = 1$ (restantes variáveis com valor nulo). O valor do limite inferior obtido para o custo é de 36416,2 com peso 24251 e na Figura 3.7 podemos encontrar a representação da solução através de um grafo.

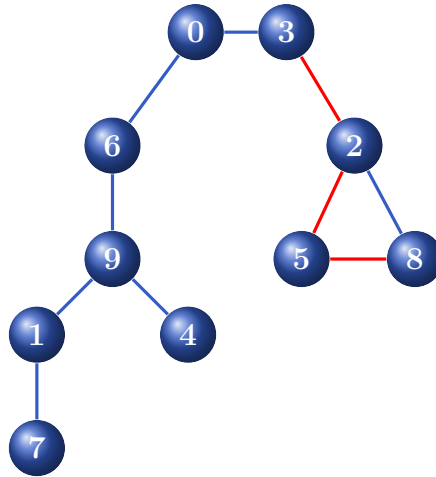


Figura 3.7: Representação da solução da relaxação linear da Formulação WMTZ após a introdução do corte 1.

Passo 3:

O fluxo máximo entre o nodo 0 e o nodo 8 é aproximadamente igual a 0,94, isto é, $f_{max}(0, 8) \approx 0,94$. Deste modo, podemos dizer que há um constrangimento na aresta $\{2, 3\}$ que nos permite formar os conjuntos $S = \{0, 1, 3, 4, 6, 7, 9\}$ e $S^c = \{2, 5, 8\}$ e voltamos ao Passo 2.

Passo 2:

1) Introduzir o corte 2:

$$x_{02} + x_{05} + x_{08} + x_{12} + x_{15} + x_{18} + x_{32} + x_{35} + x_{38} + x_{42} + x_{45} + x_{48} + x_{62} + x_{65} + x_{68} + x_{72} + x_{75} + x_{78} + x_{92} + x_{95} + x_{98} \geq 1.$$

2) Os valores de cada variável na solução da relaxação linear da Formulação WMTZ com a introdução do corte 2 são os seguintes: $x_{03} = 1$; $x_{05} \approx 0,08$; $x_{06} = 1$; $x_{17} = 1$; $x_{25} \approx 0,92$; $x_{52} \approx 0,08$; $x_{28} = 1$; $x_{32} \approx 0,92$; $x_{69} = 1$; $x_{91} = 1$ e $x_{94} = 1$ (restantes variáveis com valor nulo). O valor do limite inferior obtido para o custo é de 36530,6 com peso 24251 e na Figura 3.8 podemos encontrar a representação da solução através de um grafo.

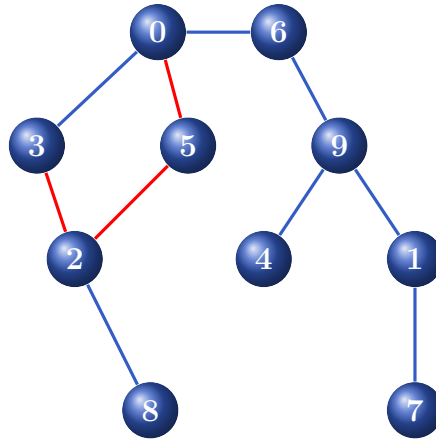


Figura 3.8: Representação da solução da relaxação linear da Formulação WMTZ após a introdução do corte 2.

Passo 3:

Como o fluxo máximo entre o nodo 0 e cada nodo $i \in V \setminus \{0\}$ é igual a 1, não é possível inserir mais cortes e passamos ao Passo 4.

Passo 4:

Neste passo são inseridas 72 restrições do tipo $x_{ij} + x_{ji} \leq 1$, $i, j \in V$, $i \neq j$ e resolve-se novamente a relaxação linear da Formulação WMTZ. A solução obtida encontra-se representada na Figura 3.8 e o valor da relaxação linear é $\vartheta(\text{P-WMTZ} + C_L) = 36530,6$.

Como a solução ótima desta instância tem custo 39983 e peso 23608, o *gap* do limite inferior obtido com a aplicação deste procedimento é de $\frac{39983 - 36530,6}{39983} \times 100 = 8,63\%$.

3.4 Experiências para Comparação das Formulações e Resultados Computacionais

Atendendo a que existem diferenças entre as várias formulações apresentadas para o Problema WMST é necessário avaliar qual é a mais eficiente. Neste sentido pretendemos avaliar a qualidade do valor do limite inferior obtido pela relaxação linear de cada formulação e os tempos de execução. Para avaliar a qualidade do limite inferior obtido em cada um dos modelos calculamos o $gap = \frac{OPT - \vartheta(WMST_L)}{OPT} \times 100$, onde $\vartheta(WMST_L)$ é o valor do limite inferior obtido pela relaxação linear do modelo em causa e OPT é o valor ótimo ($\vartheta(WMST)$) ou o melhor valor de uma solução admissível obtida, e, portanto, um limite superior para o valor do custo.

Para resolver os modelos de Programação Linear Inteira apresentados ao longo deste capítulo usamos o *software Xpress 7.3* (*Xpress-Optimizer 23.01.03* e *Xpress-Mosel 3.4.0*) [1] e todos os testes foram efetuados num Intel(R) Core(TM)2 Duo CPU (T7100) 2.00 GHz processador e 4Gb de RAM.

3.4.1 Exemplo Comparativo das Formulações

Para comparar as formulações vamos começar por usar um exemplo de 10 nodos com $W = 79$, para o qual os valores obtidos nos vários modelos se encontram na Tabela 3.3. Na primeira coluna da referida tabela, encontram-se as designações atribuídas a cada uma das formulações apresentadas ao longo deste capítulo, as três colunas seguintes contêm os valores obtidos pela relaxação linear, isto é, o valor do limite inferior para o custo, o valor do correspondente peso e o tempo de execução (em segundos). As últimas duas colunas contêm o tempo de execução (em segundos) e o número de nodos na árvore de pesquisa do Algoritmo *Branch and Bound*.

Para este exemplo, a solução ótima corresponde a uma árvore de suporte com custo 335 e peso 61.

Formulação	Relaxação Linear			Algoritmo <i>Branch and Bound</i>	
	Custo	Peso	Tempo	Tempo	#Nodos
ES	279,435	79	0,063	0,093	1
P-CS	279,435	79	0,101	0,143	1
MF	279,435	79	0,047	1,903	3
MTZ	178,750	79	0,032	0,078	1
MTZl1	279,435	79	0,032	0,078	9
MTZl2	180,594	79	0,031	0,078	1
MTZl3	184,728	79	0,015	0,203	3
MTZl4	275,330	79	0,015	0,063	3
WMTZ	169,632	79	0,031	0,078	1
WMTZl1	174,397	79	0,032	0,109	13
WMTZl2	279,435	79	0,031	0,078	5
WMTZl3	174,653	79	0,015	0,109	1
WMTZl4	175,490	79	0,032	0,234	9
WE	279,435	79	0,655	1,201	9
P-MTZ+C	279,435	79	0,138	0,206	7
P-MTZl1+C	279,435	79	0,031	0,097	6
P-MTZl2+C	279,435	79	0,141	0,2	3
P-MTZl3+C	279,435	79	0,142	0,212	5
P-MTZl4+C	279,435	79	0,048	0,132	9
P-WMTZ+C	279,435	79	0,102	0,175	17
P-WMTZl1+C	279,435	79	0,133	0,183	1
P-WMTZl2+C	279,435	79	0,029	0,157	67
P-WMTZl3+C	279,435	79	0,13	0,229	5
P-WMTZl4+C	279,435	79	0,136	0,26	22

Tabela 3.3: Comparação das formulações usando um exemplo de 10 nodos.

O melhor valor obtido pela relaxação linear das diferentes formulações foi 279,435, este valor está assinalado a negrito na Tabela 3.3. Para o exemplo de 10 nodos considerado obtemos os mesmos valores da relaxação linear nas Formulações ES, MF, MTZl1, WMTZl2 e WE. Em todas as formulações onde se aplica o procedimento com introdução de cortes também se obtém o mesmo valor da relaxação linear 279,435. Notamos que comparando a qualidade do valor do limite inferior obtido pelas desigualdades levantadas podemos verificar que tanto no exemplo de 5 nodos (Exemplo 3.2) como neste exemplo de 10 nodos, as Desigualdades Levantadas 1 na Formulação MTZ (MTZl1) e as Desigualdades Levantadas 2 na Formulação WMTZ (WMTZl2) são as que apresentam melhores resultados de entre as várias desigualdades levantadas

derivadas das Formulações MTZ e WMTZ.

3.4.2 Comparação das Formulações MTZ e WMTZ

Para cada uma das Formulações MTZ e WMTZ apresentámos nas Secções 3.2.2 e 3.2.3 algumas desigualdades denominadas de Desigualdades Levantadas. Nesta secção começamos por comparar, computacionalmente, os valores dos limites inferiores obtidos e dos tempos de execução usando cada uma das desigualdades levantadas propostas na Formulação MTZ. De seguida, efetuamos o mesmo estudo para as desigualdades levantadas propostas para a Formulação WMTZ. Esta comparação é efetuada com o objetivo de averiguar qual das desigualdades levantadas, em cada uma das Formulações MTZ e WMTZ, produz melhor valor do limite inferior da relaxação linear. Depois de escolhidas as desigualdades que quando introduzidas no modelo produzem melhor valor do limite inferior da relaxação linear de cada uma das Formulações MTZ e WMTZ, pretendemos comparar, computacionalmente, as duas formulações.

Formulação MTZ

Para comparar, computacionalmente, os valores dos limites inferiores obtidos usando cada uma das desigualdades levantadas propostas para a Formulação MTZ elaborámos as Tabelas 3.4 e 3.5. A primeira coluna destas tabelas refere-se à designação da instância e as restantes colunas contêm os valores dos limites inferiores obtidos pelas Formulações MTZ, MTZl1, MTZl2, MTZl3 e MTZl4. A Tabela 3.4 contém instâncias de 10 nodos e a Tabela 3.5 contém instâncias de 20 nodos dos três grupos de instâncias em estudo. Notamos que no interior de ambas as tabelas se encontram duas linhas horizontais que separam os três grupos de instâncias, Quase Caminhos, Aleatórias e Euclidianas.

Instância	MTZ _L	MTZl1 _L	MTZl2 _L	MTZl3 _L	MTZl4 _L
QC10-1	70,850	102,846	71,354	73,437	101,750
QC10-2	180,558	277,893	181,160	183,068	276,205
QC10-3	101,189	282,320	102,561	104,173	269,248
QC10-4	148,118	282,940	149,592	154,284	276,805
QC10-5	295,290	480,818	298,061	323,849	480,818
QC10-6	88,687	128,771	88,998	89,246	128,771
QC10-7	269,666	364,790	270,000	277,005	364,065
QC10-8	267,364	474,800	268,739	276,392	473,219
QC10-9	122,943	217,663	124,044	127,471	211,672
QC10-10	413,217	533,522	413,654	418,521	531,433
R10-1	12589,3	17066,5	12661	12746,6	17066,5
R10-2	11856,2	16824	11932,1	12808,6	16824
R10-3	6402,33	8796,34	6429,56	6558,05	8796,34
R10-4	10659,3	16627,4	10755,4	11256,4	16627,4
R10-5	23911,9	26048,3	23976,6	24559	26012,6
E10-1	29161,7	33619,7	29202,2	29427,5	33619,7
E10-2	25480,6	32734,9	25621,2	26213,4	32290,9
E10-3	25501,9	29698,6	25532,3	25648,3	29698,6
E10-4	14870,3	22379,5	15046,1	15612,9	22223,1
E10-5	35837,3	38040,1	35898,7	35977,8	37998,6

Tabela 3.4: Comparação dos valores da relaxação linear da Formulação MTZ com as várias desigualdades levantadas em instâncias de 10 nodos.

Instância	MTZ _L	MTZl1 _L	MTZl2 _L	MTZl3 _L	MTZl4 _L
QC20-1	1017,800	1477,310	1019,100	1030,800	1470,370
QC20-2	1028,600	1345,330	1029,520	1034,930	1343,920
QC20-3	844,136	1293,780	845,667	853,500	1290,690
QC20-4	400,352	565,526	400,581	404,039	565,269
QC20-5	168,315	440,339	168,484	169,235	438,258
QC20-6	607,602	847,329	608,394	613,906	847,329
QC20-7	519,714	1022,540	521,196	529,368	1022,540
QC20-8	211,335	296,717	211,638	213,445	296,543
QC20-9	176,805	559,421	177,886	181,276	554,623
QC20-10	774,781	1300,880	775,725	785,813	1293,250
R20-1	25308,900	30518,100	25330,500	25516,300	30412,500
R20-2	26211,500	31057,400	26227,100	26754,800	31012,300
R20-3	25357,100	30826,400	25377,800	25748,700	30826,400
R20-4	23844,000	30596,300	23879,200	24126,400	30555,100
R20-5	24327,700	28647,700	24333,400	24686,500	28576,800
E20-1	40368,600	50041,500	40404,300	40560,200	50041,500
E20-2	62689,600	77740,400	62727,500	63072,200	77740,400
E20-3	45953,400	57072,100	46004,100	46289,000	57063,300
E20-4	44808,000	51277,900	44812,600	44845,700	51274,500
E20-5	44445,400	54287,900	44481,700	44922,600	54287,900

Tabela 3.5: Comparação dos valores da relaxação linear da Formulação MTZ com as várias desigualdades levantadas em instâncias de 20 nodos.

Os valores a negrito nas Tabelas 3.4 e 3.5 correspondem ao melhor valor do limite inferior obtido. Podemos observar que se verifica a seguinte relação entre os valores obtidos pela relaxação linear nas diversas formulações para todas as instâncias consideradas.

$$\vartheta(MTZ_L) \leq \vartheta(MTZl2_L) \leq \vartheta(MTZl3_L) \leq \vartheta(MTZl4_L) \leq \vartheta(MTZl1_L),$$

qualquer uma das desigualdades quando introduzidas no modelo melhora o valor do limite inferior obtido pela Formulação MTZ, sendo a Formulação WMTZl1 a que apresenta melhores limites inferiores e a Formulação MTZ a que apresenta piores resultados.

Para comparar, computacionalmente, os tempos de execução obtidos usando cada uma das desigualdades levantadas propostas na Formulação MTZ elaborámos o gráfico da Figura 3.9, onde se apresentam os tempos médios, em segundos, necessários para a obtenção do valor ótimo em cada uma das Formulações MTZ, MTZl1, MTZl2, MTZl3 e MTZl4 em instâncias de 10, 20 e 40 nodos.

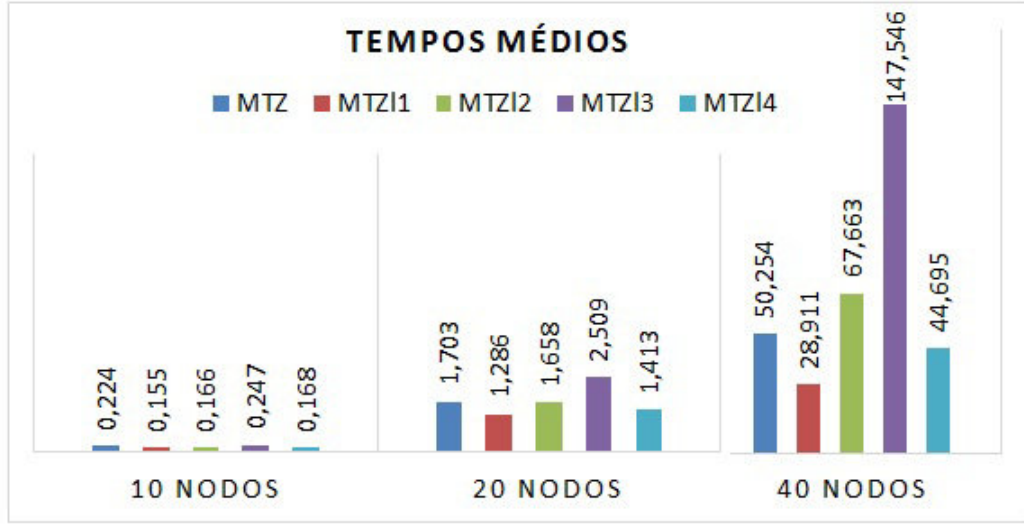


Figura 3.9: Tempos médios, em segundos, necessários para a obtenção do valor ótimo da Formulação MTZ com as várias desigualdades levantadas em instâncias de 10, 20 e 40 nodos.

Através do gráfico da Figura 3.9 é possível observar que os tempos médios mais baixos são obtidos usando a Formulação MTZl1 e os tempos médios mais elevados são obtidos usando a Formulação MTZl3.

Assim, verificamos que os melhores resultados são obtidos quando as Desigualdades Levantadas 1 (Formulação MTZl1) são incorporadas no modelo quer em termos de melhor valor do limite inferior quer em termos de melhores tempos médios de execução necessários para a obtenção do valor ótimo. Deste modo e daqui em diante consideramos a Formulação MTZ com as desigualdades (3.33) substituídas por

$$(n - 2)x_{ji} + nx_{ij} + u_i \leq u_j + (n - 1), \quad (i, j) \in A.$$

Formulação WMTZ

Para comparar, computacionalmente, os valores dos limites inferiores obtidos usando cada uma das desigualdades levantadas propostas na Formulação WMTZ elaborámos as Tabelas 3.6 e 3.7. A primeira coluna destas tabelas refere-se à designação da instância e as restantes colunas contêm os valores dos limites inferiores obtidos pelas Formulações WMTZ, WMTZl1, WMTZl2, WMTZl3 e WMTZl4. A Tabela 3.6 contém instâncias de 10 nodos e a Tabela 3.7 contém instâncias de 20 nodos dos três grupos de instâncias em estudo. Notamos que no interior de ambas as tabelas se encontram duas linhas horizontais que separam os três grupos de instâncias, Quase Caminhos, Aleatórias e Euclideanas.

Instância	WMTZ _L	WMTZl1 _L	WMTZl2 _L	WMTZl3 _L	WMTZl4 _L
QC10-1	68,087	69,272	102,846	69,457	71,898
QC10-2	178,512	180,943	277,893	181,397	183,665
QC10-3	93,654	96,854	282,320	97,740	99,806
QC10-4	142,483	148,007	282,940	149,518	158,827
QC10-5	283,673	292,998	480,818	295,012	327,451
QC10-6	87,994	89,096	128,771	89,886	90,336
QC10-7	268,808	270,369	364,790	270,711	281,230
QC10-8	262,573	268,037	474,800	271,337	283,628
QC10-9	115,192	118,295	217,663	119,823	124,517
QC10-10	411,754	413,537	533,522	413,868	423,625
R10-1	12422,800	12750,700	17066,500	12867,300	12917,000
R10-2	11861,700	12297,900	16824,000	12413,300	13101,400
R10-3	6388,140	6553,230	8796,340	6572,760	6607,750
R10-4	10245,500	10584,700	16627,400	10913,600	12016,400
R10-5	23718,600	24000,800	26048,300	24130,900	25094,600
E10-1	29076,100	29261,400	33619,700	29380,900	29572,000
E10-2	25284,200	26005,700	32734,900	26286,900	27053,600
E10-3	25426,300	25570,000	29698,600	25589,100	25656,400
E10-4	14822,700	15902,900	22379,500	16139,300	16773,000
E10-5	35540,900	35835,700	38040,100	35887,100	35954,700

Tabela 3.6: Comparação dos valores da relaxação linear da Formulação WMTZ com as várias desigualdades levantadas em instâncias de 10 nodos.

Instância	WMTZ _L	WMTZl1 _L	WMTZl2 _L	WMTZl3 _L	WMTZl4 _L
QC20-1	1004,530	1014,120	1477,310	1014,680	1036,750
QC20-2	1024,370	1035,710	1345,330	1036,980	1042,310
QC20-3	828,600	839,914	1293,780	841,609	846,907
QC20-4	399,060	401,685	565,526	401,775	406,760
QC20-5	166,906	169,220	440,339	169,879	170,985
QC20-6	600,583	607,423	847,329	607,927	616,592
QC20-7	503,581	519,391	1022,540	521,910	532,799
QC20-8	208,947	211,816	296,717	212,757	216,625
QC20-9	166,153	180,256	559,421	181,285	185,302
QC20-10	767,331	779,376	1300,880	779,893	791,667
R20-1	25226,500	25506,700	30518,100	25586,600	25812,600
R20-2	25957,000	26133,800	31057,400	26134,200	26645,600
R20-3	25199,800	25400,000	30826,400	25424,000	25933,200
R20-4	23735,000	24212,000	30596,300	24275,800	24599,600
R20-5	24156,300	24254,800	28647,700	24261,500	24534,200
E20-1	40383,700	40966,700	50041,500	41065,700	41170,000
E20-2	62536,800	63020,600	77740,400	63083,100	63547,700
E20-3	45546,500	46001,400	57072,100	46068,100	46691,700
E20-4	44812,100	45239,800	51277,900	45243,500	45256,000
E20-5	44326,000	44801,000	54287,900	44881,200	45349,100

Tabela 3.7: Comparação dos valores da relaxação linear da Formulação WMTZ com as várias desigualdades levantadas em instâncias de 20 nodos.

Os valores a negrito nas Tabelas 3.6 e 3.7 correspondem ao melhor valor do limite inferior obtido. Podemos observar que se verifica a seguinte relação entre os valores obtidos pela relaxação linear nas diversas formulações para todas as instâncias consideradas.

$$\vartheta(WMTZ_L) \leq \vartheta(WMTZl1_L) \leq \vartheta(WMTZl3_L) \leq \vartheta(WMTZl4_L) \leq \vartheta(WMTZl2_L),$$

qualquer uma das desigualdades quando introduzidas no modelo melhora o valor do limite inferior obtido pela Formulação WMTZ, sendo a Formulação WMTZl2 a que apresenta melhores limites inferiores e a Formulação WMTZ a que apresenta piores resultados.

Para comparar, computacionalmente, os tempos de execução obtidos usando cada uma das desigualdades levantadas propostas na Formulação WMTZ elaborámos o gráfico da Figura 3.10, onde se apresentam os tempos médios, em segundos, necessários para a obtenção do valor ótimo em cada uma das Formulações WMTZ, WMTZl1, WMTZL2, WMTZl3 e WMTZl4 em instâncias de 10, 20 e 40 nodos.

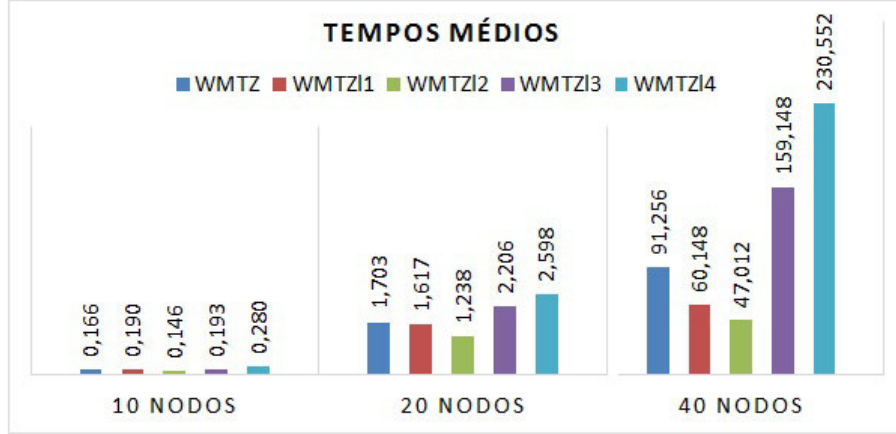


Figura 3.10: Tempos médios, em segundos, necessários para a obtenção do valor ótimo da Formulação WMTZ com as várias desigualdades levantadas em instâncias de 10, 20 e 40 nodos.

Através do gráfico da Figura 3.10 é possível observar que os tempos médios mais baixos são obtidos usando a Formulação WMTZl2 e os tempos médios mais elevados são obtidos usando a Formulação WMTZl4.

Assim, verificamos que os melhores resultados são obtidos quando as Desigualdades Levantadas 2 (Formulação WMTZl2) são incorporadas no modelo. Deste modo e daqui em diante consideramos a Formulação WMTZ com as desigualdades (3.33) substituídas por

$$(W - w_{ji})x_{ji} + w_{ij}x_{ij} + p_i \leq p_j + W(1 - x_{ij}), \quad (i, j) \in A.$$

Comparação entre as Formulações MTZ e WMTZ

A partir deste momento vamos usar as Formulações MTZ e WMTZ para representar as Formulações MTZl1 e WMTZl2, respetivamente. Além disso, podemos verificar que nas instâncias de 10 e 20 nodos nos três grupos de instâncias (ver Tabelas 3.4, 3.5, 3.6 e 3.7) e de estudos computacionais realizados para as restantes instâncias verificamos a seguinte relação

$$\vartheta(MTZ_L) = \vartheta(WMTZ_L).$$

Pretendemos agora verificar qual, dentre estas duas formulações, a que apresenta melhores tempos computacionais na obtenção do valor do limite inferior para o custo. Nas tabelas que se seguem apresentamos, para cada grupo de instâncias em teste, os tempos médios, em segundos, para obtenção do valor da relaxação linear acompanhados dos respetivos desvios padrões em cada uma das Formulações MTZ e WMTZ para instâncias até 1000 nodos.

	Tempo Médio		Desvio Padrão	
N.º Nodos	MTZ _L	WMTZ _L	MTZ _L	WMTZ _L
10	0,003	0,003	0,007	0,001
20	0,005	0,008	0,008	0,001
40	0,022	0,024	0,008	0,002
60	0,050	0,056	0,007	0,002
80	0,106	0,113	0,014	0,005
100	0,223	0,200	0,070	0,014
150	0,721	0,788	0,119	0,223
200	1,807	1,679	0,254	0,125
300	6,101	4,807	2,842	0,264
400	18,561	11,538	11,473	0,407
500	26,068	23,001	1,132	0,653
1000	195,298	76,388	36,262	14,615

Tabela 3.8: Comparação dos tempos médios (em segundos) e dos desvios padrões necessários para a obtenção do valor da relaxação linear nas Formulações MTZ e WMTZ em instâncias QC.

	Tempo Médio		Desvio Padrão	
N. ^o Nodos	MTZ _L	WMTZ _L	MTZ _L	WMTZ _L
10	0,000	0,000	0,000	0,000
20	0,003	0,009	0,007	0,008
40	0,025	0,022	0,008	0,009
60	0,144	0,069	0,058	0,014
80	0,219	0,196	0,037	0,101
100	0,387	0,331	0,168	0,152
150	1,105	1,098	0,318	0,245
200	2,683	2,271	0,229	0,550
300	8,250	6,683	0,446	2,826
400	20,336	10,446	1,089	1,786
500	60,651	31,782	7,682	13,284
1000	154,099	115,664	48,808	8,702

Tabela 3.9: Comparação dos tempos médios (em segundos) e dos desvios padrões necessários para a obtenção do valor da relaxação linear nas Formulações MTZ e WMTZ em instâncias R.

	Tempo Médio		Desvio Padrão	
N. ^o Nodos	MTZ _L	WMTZ _L	MTZ _L	WMTZ _L
10	0,003	0,000	0,007	0,000
20	0,016	0,006	0,000	0,008
40	0,031	0,025	0,016	0,014
60	0,081	0,050	0,007	0,007
80	0,147	0,103	0,023	0,008
100	0,275	0,212	0,034	0,060
150	0,839	0,418	0,138	0,051
200	2,043	0,933	0,205	0,160
300	7,506	2,480	1,751	0,168
400	16,049	6,318	1,064	0,478
500	39,225	11,132	6,477	0,093
1000	172,568	73,114	62,173	4,347

Tabela 3.10: Comparação dos tempos médios (em segundos) e dos desvios padrões necessários para a obtenção do valor da relaxação linear nas Formulações MTZ e WMTZ em instâncias E.

Nas instâncias Quase Caminhos, os tempos médios gastos na obtenção do valor da relaxação linear de ambas as Formulações MTZ e WMTZ são semelhantes. De notar apenas que para instâncias de 1000 nodos, os tempos médios usando a Formulação MTZ_L aumentam para mais do dobro dos tempos médios usando a Formulação $WMTZ_L$. Em relação às instâncias Aleatórias para mais de 200 nodos usando a Formulação $WMTZ_L$ obtêm-se menores tempos médios, sendo os tempos médios semelhantes para as restantes instâncias Aleatórias. Nas instâncias Euclidianas até 100 nodos os tempos médios são semelhantes, mas para instâncias com mais de 100 nodos usando a Formulação $WMTZ_L$ obtêm-se menores tempos médios. Nos vários grupos de instâncias de entre as Formulações MTZ_L e $WMTZ_L$, em geral, usando a Formulação $WMTZ_L$ obtêm-se tempos médios de execução inferiores.

3.4.3 Estudo Comparativo das Formulações

Nesta secção apresentamos e comparamos o tempo de execução e a qualidade das soluções encontradas usando as formulações e os procedimentos apresentados ao longo do capítulo. Para cada grupo de instâncias apresentamos algumas tabelas, onde se reúnem os *gap* (em percentagem) encontrados na obtenção de um limite inferior para o valor do custo e os tempos (em segundos) encontrados na obtenção do valor ótimo ou de um limite superior para o valor do custo. Notamos que o $gap = \frac{OPT - \vartheta(WMST_L)}{OPT} \times 100$, onde $\vartheta(WMST_L)$ é o valor do limite inferior obtido para o valor do custo e OPT é o valor ótimo ($\vartheta(WMST)$) ou o melhor valor de uma solução admissível obtida, e, portanto, um limite superior para o valor do custo. Nas várias tabelas que vamos apresentar a designação *Mem* representa falta de memória. No Algoritmo *Branch and Bound* usando as Formulações ES, MF, WE, MTZ e WMTZ foi imposto um tempo de execução de 5000 segundos e usando os procedimentos de introdução de cortes um limite no tempo de execução de 10800 segundos. Em relação aos Procedimentos P-MTZ+C e P-WMTZ+C, apenas apresentamos os resultados computacionais relativos ao Procedimento P-WMTZ+C dado que na subsecção anterior, a relaxação linear da Formulação WMTZ foi a que apresentou tempos médios de execução inferiores.

Instâncias Quase Caminhos

Neste grupo de instâncias não foi possível correr as instâncias com 40 ou mais nodos usando as Formulações ES e WE por falta de memória, sendo assim apenas é possível

comparar as várias formulações e procedimentos com introdução de cortes para as instâncias de 10 e 20 nodos. Quanto à Formulação MF apresenta falta de memória para instâncias com mais de 150 nodos. Usando as Formulações MTZ e WMTZ e os Procedimentos P-CS e P-WMTZ+C é possível obter um valor para o limite inferior para instâncias que tenham até 1000 nodos.

Em termos de qualidade do valor do limite inferior, as formulações apresentadas podem ser divididas em dois grupos como se pode observar na Tabela 3.11 através da existência de um risco vertical. Na Tabela 3.11 a primeira coluna refere-se à designação da instância, a segunda coluna contém o *gap* obtido usando as Formulações ES_L , MF_L e WE_L e os Procedimentos P-CS_L e P-WMTZ+C_L. A coluna oito contém o *gap* obtido usando as Formulações MTZ_L e WMTZ_L. Nas restantes colunas podemos observar os tempos de execução (em segundos) necessários para a obtenção do valor ótimo usando as Formulações ES, MF, WE, MTZ e WMTZ e os Procedimentos P-CS e P-WMTZ+C.

Instância	<i>Gap</i>		Tempos				<i>Gap</i>	Tempos	
		ES	MF	WE	P-CS	P-WMTZ+C		MTZ	WMTZ
QC10-1	38,78	0,16	0,28	3,08	0,06	0,08	38,78	0,08	0,11
QC10-2	23,66	0,19	0,84	3,42	0,12	0,22	23,66	0,11	0,05
QC10-3	24,71	0,23	0,28	2,62	0,05	0,11	24,71	0,16	0,10
QC10-4	22,27	0,41	1,51	3,18	0,11	0,11	22,27	0,14	0,17
QC10-5	17,00	0,23	0,56	3,23	0,03	0,09	17,24	0,19	0,14
QC10-6	28,06	0,19	0,98	2,98	0,23	0,11	28,06	0,08	0,12
QC10-7	16,90	0,16	0,56	3,48	0,14	0,09	16,90	0,11	0,14
QC10-8	17,51	0,53	1,50	3,46	0,09	0,13	17,71	0,13	0,15
QC10-9	17,24	0,38	1,28	3,54	0,08	0,06	17,24	0,08	0,06
QC10-10	16,79	0,42	0,70	2,93	0,16	0,09	18,79	0,17	0,28
QC20-1	6,56	1179,24	19,45	43,16	0,90	0,95	6,56	0,89	0,72
QC20-2	8,04	4946,99	14,47	37,78	0,64	1,09	8,04	0,84	1,12
QC20-3	5,15	1192,27	42,79	58,82	0,95	1,12	5,15	1,25	1,13
QC20-4	16,07	1351,41	7,78	33,43	0,67	0,73	16,09	1,01	0,95
QC20-5	19,12	1379,94	7,53	34,74	0,63	0,62	19,50	0,94	0,55
QC20-6	8,34	1812,30	33,77	66,62	1,23	1,45	8,89	1,09	1,60
QC20-7	8,33	3073,37	71,39	61,98	2,25	3,61	8,54	2,17	2,12
QC20-8	7,18	1874,57	12,06	34,94	0,66	1,28	7,28	1,08	0,90
QC20-9	9,91	1010,29	30,70	57,78	0,95	1,81	9,92	2,09	0,91
QC20-10	10,04	4218,94	11,68	44,51	1,00	0,83	10,04	1,05	1,35

Tabela 3.11: Comparação das formulações em instâncias QC de 10 e 20 nodos.

Os *gaps* obtidos usando as Formulações ES_L , MF_L e WE_L e os Procedimentos $P-CS_L$ e $P-WMTZ+C_L$ são iguais. Estes *gaps* são inferiores aos obtidos através do uso das Formulações MTZ_L e $WMTZ_L$ em 40% das instâncias apresentadas na Tabela 3.11 (8 em 20 instâncias).

Os tempos de execução usando a Formulação ES começam a ser bastante elevados para instâncias de 20 nodos. O uso da Formulação MF torna-se mais rápido do que o uso da Formulação WE. Em geral, para instâncias de 10 e 20 nodos os tempos de execução são inferiores usando o Procedimento P-CS.

Na Tabela 3.12 também podemos observar a existência de um risco vertical, o qual indica que as formulações apresentadas também podem ser divididas, em termos de qualidade do valor do limite inferior, em dois grupos. Para instâncias com 40, 60 e 80 nodos só é possível comparar as Formulações MF, MTZ e WMTZ e os Procedimentos P-CS e P-WMTZ+C dado que as Formulações ES e WE apresentam falta de memória. Na Tabela 3.12 a primeira coluna refere-se à designação da instância, a segunda coluna contém o *gap* obtido usando a Formulação MF_L e os Procedimentos $P-CS_L$ e $P-WMTZ+C_L$. A coluna seis contém o *gap* obtido usando as Formulações MTZ_L e $WMTZ_L$. Nas restantes colunas podemos observar os tempos de execução (em segundos) necessários para a obtenção do valor ótimo ou de um limite superior para o valor do custo usando as Formulações MF, MTZ e WMTZ e os Procedimentos P-CS e P-WMTZ+C.

Note-se que nas instâncias com 40, 60 e 80 nodos usando a Formulação MF não foi possível no tempo limite imposto (5000 segundos) provar a otimalidade em nenhuma das instâncias.

	<i>Gap</i>		Tempos		<i>Gap</i>	Tempos	
Instância	MF		P-CS	P-WMTZ+C		MTZ	WMTZ
QC40-1	5,83	5012,23 ^a	61,43	29,89	5,92	32,61	67,98
QC40-2	7,18	5010,42 ^a	55,87	49,55	7,18	42,20	28,68
QC40-3	5,60	5012,01 ^a	27,69	34,77	5,77	38,19	5,51
QC40-4	5,38	5007,71 ^a	164,14	88,26	5,70	186,48	148,97
QC40-5	6,40	5009,16 ^a	37,21	25,88	6,42	36,78	24,81
QC40-6	5,95	5010,74 ^a	45,20	28,58	5,95	15,27	24,16
QC40-7	8,15	5010,32 ^a	45,44	101,58	8,15	36,03	54,28
QC40-8	6,73	5030,13 ^a	28,14	17,33	6,77	5,71	21,07
QC40-9	6,60	5011,85 ^a	25,08	14,64	6,66	11,53	28,88
QC40-10	5,81	5009,64 ^a	62,62	27,06	5,81	24,07	30,00
QC60-1	4,54	5328,16 ^a	345,54	473,61	4,58	487,59	857,77
QC60-2	4,56	5145,07 ^a	418,85	610,92	4,56	621,32	456,46
QC60-3	2,87	5137,45 ^a	304,87	554,68	2,87	297,38	584,50
QC60-4	4,36	5151,68 ^a	1121,22	1250,37	4,36	872,04	2100,82
QC60-5	5,48	5124,27 ^a	400,90	214,55	5,48	234,70	189,30
QC60-6	5,26	5153,44 ^a	185,12	201,32	5,26	227,57	248,73
QC60-7	5,46	5176,42 ^a	972,59	783,62	5,46	2361,43	1032,19
QC60-8	5,11	5128,38 ^a	310,45	414,94	5,13	394,41	332,79
QC60-9	6,00	5108,40 ^a	563,06	603,84	6,00	752,00	710,41
QC60-10	4,66	8223,73 ^a	169,76	66,36	4,66	1110,18	107,88
QC80-1	3,29	6159,59 ^a	700,68	4605,85	3,33	2524,85	1129,14
QC80-2	4,30	5424,06 ^a	1726,52	3618,82	4,33	1336,79	1013,28
QC80-3	3,07	5425,27 ^a	1592,13	2102,90	3,13	4283,73	992,12
QC80-4	3,28	5660,65 ^a	3104,76	732,30	3,31	2968,99	4999,24 ^a
QC80-5	4,92	5777,09 ^a	1164,13	1424,12	4,96	1860,88	1288,65
QC80-6	2,88	6238,23 ^a	9679,97	7280,69	3,02	3888,80	4999,48 ^a
QC80-7	3,39	5976,62 ^a	1731,47	1916,47	3,44	853,57	2951,94
QC80-8	3,04	5944,65 ^a	480,45	393,98	3,04	4897,01	325,45
QC80-9	4,17	5747,04 ^a	2363,90	1255,64	4,37	2736,49	4999,84 ^a
QC80-10	5,45	5508,36 ^a	1793,40	2025,78	5,59	4883,93	2541,65

Tabela 3.12: Comparação das formulações em instâncias QC de 40, 60 e 80 nodos.

^a Não foi possível provar a otimalidade no tempo imposto.

Na Tabela 3.12 continuamos a observar que os *gaps* usando a Formulação MF_L e os Procedimentos $P-CS_L$ e $P-WMTZ+C_L$ são um pouco mais baixos em 56,67% das instâncias apresentadas (17 em 30 instâncias). Pela observação da Tabela 3.12 verificamos que em 15 instâncias usando o Procedimento $P-CS$ obtêm-se tempos de execução inferiores e nas outras 15 instâncias os tempos de execução são menores usando o Procedimento $P-WMTZ+C$.

Já vimos que os dois procedimentos com introdução de cortes apresentam os mesmos *gaps*. Para averiguar qual dos procedimentos é mais rápido na obtenção de um limite inferior elaborámos a Tabela 3.13, onde se encontram os tempos médios (em segundos) e os respetivos desvios padrões obtidos para instâncias de 10 a 80 nodos.

	Tempos PL		Desvio Padrão	
N.º Nodos	$P-CS_L$	$P-WMTZ+C_L$	$P-CS_L$	$P-WMTZ+C_L$
10	0,011	0,003	0,007	0,006
20	0,233	0,025	0,061	0,008
40	7,541	0,086	1,201	0,015
60	49,439	0,201	9,570	0,042
80	214,574	0,587	36,718	0,188

Tabela 3.13: Comparação dos Procedimentos $P-CS$ e $P-WMTZ+C$ em termos de tempo médio de execução (em segundos) da relaxação linear em instâncias QC de 10, 20, 40, 60 e 80 nodos.

Na Tabela 3.13 podemos observar que para a obtenção de um limite inferior para o valor do custo, os tempos médios e correspondentes desvios padrões são menores usando o Procedimento $P-WMTZ+C_L$.

Instâncias Aleatórias

Nas instâncias Aleatórias usando a Formulação WE só é possível executar a instância R10-1, não sendo possível correr as restantes instâncias por falta de memória, sendo assim apenas é possível comparar as várias formulações e procedimentos com introdução de cortes para a instância R10-1. A Formulação ES apresenta falta de memória nas instâncias R20-1, R20-2, R20-3 e para instâncias com 40 ou mais nodos. Quanto à Formulação MF apresenta falta de memória para instâncias com mais de 200 nodos.

Usando as Formulações MTZ e WMTZ e os Procedimentos P-CS e P-WMTZ+C é possível obter um valor para o limite inferior para instâncias que tenham até 1000 nodos.

Em termos de qualidade do valor do limite inferior, as formulações apresentadas podem ser divididas em dois grupos como se pode observar na Tabela 3.14 através da existência de um risco vertical. Na Tabela 3.14 a primeira coluna refere-se à designação da instância, a segunda coluna contém o *gap* obtido usando as Formulações ES_L , MF_L e WE_L e os Procedimentos P-CS_L e P-WMTZ+C_L. A coluna oito contém o *gap* obtido usando as Formulações MTZ_L e WMTZ_L. Nas restantes colunas podemos observar os tempos de execução (em segundos) necessários para a obtenção do valor ótimo usando as Formulações ES, MF, WE, MTZ e WMTZ e os Procedimentos P-CS e P-WMTZ+C.

Instância	<i>Gap</i>	Tempos					<i>Gap</i>	Tempos	
		ES	MF	WE	P-CS	P-WMTZ+C		MTZ	WMTZ
R10-1	10,70	0,14	1,12	109,75	0,06	0,06	11,59	0,13	0,08
R10-2	18,96	0,16	0,55	<i>Mem</i>	0,12	0,23	18,96	0,05	0,06
R10-3	19,25	0,16	0,66	<i>Mem</i>	0,03	0,05	42,21	0,16	0,13
R10-4	14,96	0,45	1,31	<i>Mem</i>	0,25	0,22	14,96	0,06	0,09
R10-5	11,42	0,14	0,28	<i>Mem</i>	0,06	0,37	11,42	0,03	0,03
R20-1	3,96	<i>Mem</i>	3,84	<i>Mem</i>	0,27	1,01	4,79	0,17	0,27
R20-2	3,97	<i>Mem</i>	2,36	<i>Mem</i>	0,41	1,92	3,97	0,12	0,09
R20-3	3,94	<i>Mem</i>	8,14	<i>Mem</i>	0,45	0,45	7,61	0,77	0,33
R20-4	3,89	2458,65	1,42	<i>Mem</i>	0,44	0,66	3,89	0,64	0,09
R20-5	4,63	3183,96	4,59	<i>Mem</i>	0,50	1,62	4,63	0,28	0,16

Tabela 3.14: Comparação das formulações em instâncias R de 10 nodos.

Os *gaps* obtidos usando as Formulações ES_L , MF_L e WE_L e os Procedimentos P-CS_L e P-WMTZ+C_L são iguais. Estes *gaps* são inferiores aos obtidos através do uso das Formulações MTZ_L e WMTZ_L em 40% das instâncias apresentadas na Tabela 3.14 (4 em 10 instâncias).

Na instância R10-1 obtêm-se os valores ótimos mais rapidamente usando os Procedimentos P-CS e P-WMTZ+C. Para a única instância que é possível resolver usando a Formulação WE obtém-se o tempo mais elevado. Nas restantes instâncias de 10 nodos e nas instâncias R20-1, R20-2 e R20-3, em geral, usando a Formulação MF obtêm-se tempos mais elevados.

Os tempos de execução usando a Formulação ES são bastante elevados para as instâncias R20-4 e R20-5, as únicas de 20 nodos que resolve. O Procedimento P-CS para instâncias de 10 e 20 nodos parece ser o que, em geral, apresenta tempos de execução inferiores.

Na Tabela 3.15 também podemos observar a existência de um risco vertical, o qual indica que as formulações apresentadas também podem ser divididas, em termos de qualidade do valor do limite inferior, em dois grupos. Para instâncias com 40, 60 e 80 nodos só é possível comparar as Formulações MF, MTZ e WMTZ e os Procedimentos P-CS e P-WMTZ+C dado que as Formulações ES e WE apresentam falta de memória. Na Tabela 3.15 a primeira coluna refere-se à designação da instância, a segunda coluna contém o *gap* obtido usando a Formulação MF_L e os Procedimentos $P-CS_L$ e $P-WMTZ+C_L$. A coluna seis contém o *gap* obtido usando as Formulações MTZ_L e $WMTZ_L$. Nas restantes colunas podemos observar os tempos de execução (em segundos) necessários para a obtenção do valor ótimo ou de um limite superior para o valor do custo usando as Formulações MF, MTZ e WMTZ e os Procedimentos P-CS e P-WMTZ+C.

Instância	<i>Gap</i>		Tempos		<i>Gap</i>	Tempos	
	MF	P-CS	P-WMTZ+C			MTZ	WMTZ
R40-1	1,84	26,35	3,68	4,51	1,84	0,44	0,28
R40-2	1,72	34,93	6,26	3,68	1,72	0,67	0,33
R40-3	1,74	27,86	9,50	5,73	5,32	4,99	11,22
R40-4	2,03	35,32	10,28	2,57	2,03	0,41	0,28
R40-5	1,53	62,85	16,44	4,18	1,53	0,58	0,50
R60-1	0,63	91,82	58,66	4,51	0,63	1,36	0,70
R60-2	0,56	124,89	54,24	6,07	0,56	1,33	0,98
R60-3	0,52	369,99	34,65	6,93	0,52	1,30	0,81
R60-4	0,70	418,42	44,15	7,27	1,59	9,50	6,30
R60-5	0,53	318,72	52,43	5,03	0,59	2,17	2,75
R80-1	0,34	595,91	153,36	15,19	0,34	2,37	1,33
R80-2	0,40	973,80	143,55	16,18	0,97	24,26	22,11
R80-3	0,44	240,03	164,03	12,42	0,78	4,34	2,59
R80-4	0,37	346,58	158,05	11,17	0,37	1,86	1,14
R80-5	0,38	1127,72	176,53	16,29	0,93	5,46	6,18
R100-1	0,10	1172,18	524,12	12,90	1,75	1515,59	1328,88
R100-2	0,13	2934,95	369,84	28,80	1,67	2635,51	4999,92 ^a
R100-3	0,02	651,38	670,05	23,10	0,20	3,92	3,99
R100-4	0,06	1273,22	619,69	16,80	0,82	18,43	20,70
R100-5	0,04	1788,42	560,92	21,81	1,04	19,86	40,28

Tabela 3.15: Comparação das formulações em instâncias R de 40, 60, 80 e 100 nodos.

^a Não foi possível provar a otimalidade no tempo imposto.

Os *gaps* obtidos usando a Formulação MF_L e os Procedimentos $P-CS_L$ e $P-WMTZ+C_L$ são inferiores aos *gaps* obtidos usando as Formulações MTZ_L e $WMTZ_L$ em 55% das instâncias apresentadas na Tabela 3.15 (11 em 20 instâncias). À medida que se aumenta a dimensão das instâncias, verificamos que o tempo de execução é inferior usando o Procedimento $P-WMTZ+C$. Note-se que usando a Formulação MF obtêm-se tempos de execução mais elevados.

Instâncias Euclidianas

Usando a Formulação WE todas as instâncias Euclidianas apresentam falta de memória. A Formulação ES não corre por falta de memória para instâncias de 20 ou mais nodos. Quanto à Formulação MF apresenta falta de memória para instâncias

com mais de 200 nodos. Usando as Formulações MTZ e WMTZ e os Procedimentos P-CS e P-WMTZ+C é possível obter um valor para o limite inferior para instâncias que tenham até 1000 nodos.

Em termos de qualidade do valor do limite inferior, as formulações apresentadas podem ser divididas em dois grupos como se pode observar na Tabela 3.16 através da existência de um risco vertical. Na Tabela 3.16 a primeira coluna refere-se à designação da instância, a segunda coluna contém o *gap* obtido usando as Formulações ES_L e MF_L e os Procedimentos P-CS_L e P-WMTZ+C_L. A coluna sete contém o *gap* das Formulações MTZ_L e WMTZ_L. Nas restantes colunas podemos observar os tempos de execução (em segundos) necessários para a obtenção do valor ótimo usando as Formulações ES, MF, MTZ e WMTZ e os Procedimentos P-CS e P-WMTZ+C.

Instância	Gap		Tempos			Gap		Tempos	
		ES	MF	P-CS	P-WMTZ+C		MTZ	WMTZ	
E10-1	8,63	0,27	0,90	0,08	0,09	15,92	0,17	0,28	
E10-2	10,42	0,17	0,22	0,05	0,03	11,33	0,36	0,14	
E10-3	8,85	0,22	1,16	0,06	0,09	15,65	0,33	0,38	
E10-4	9,51	0,17	0,05	0,02	0,17	14,86	0,30	0,09	
E10-5	7,76	0,17	0,19	0,05	0,16	7,76	0,05	0,06	
E20-1	3,92	Mem	5,07	0,36	0,61	6,39	1,14	1,64	
E20-2	2,42	Mem	0,53	0,16	0,17	13,13	1,50	1,17	
E20-3	2,61	Mem	5,07	0,78	1,59	4,91	1,05	0,67	
E20-4	3,08	Mem	2,92	0,41	0,81	9,31	1,20	1,28	
E20-5	2,40	Mem	4,10	0,53	1,28	10,16	4,65	3,76	
E40-1	0,88	Mem	56,24	13,57	6,41	8,31	100,09	389,14	
E40-2	1,05	Mem	47,92	11,79	7,08	7,08	17,69	41,36	
E40-3	0,79	Mem	37,88	9,17	4,74	3,15	6,65	11,06	
E40-4	0,95	Mem	166,67	6,01	5,66	7,09	13,00	34,45	
E40-5	0,81	Mem	56,24	10,95	4,09	4,28	6,55	10,11	

Tabela 3.16: Comparação das formulações em instâncias E de 10 nodos.

Os *gaps* obtidos usando as Formulações ES_L e MF_L e os Procedimentos P-CS_L e P-WMTZ+C_L são inferiores aos *gaps* obtidos usando as Formulações MTZ_L e WMTZ_L em 93,33% das instâncias apresentadas na Tabela 3.16 (14 em 15 instâncias).

Os tempos de execução mais elevados obtêm-se usando a Formulação MF. Nas instâncias de 10 e 20 nodos observamos tempos inferiores quando se utiliza o Proce-

dimento P-CS, mas para instâncias de 40 nodos os tempos de execução são inferiores usando o procedimento P-WMTZ+C.

Na Tabela 3.17 também podemos observar a existência de um risco vertical, o qual indica que as formulações apresentadas também podem ser divididas, em termos de qualidade do valor do limite inferior, em dois grupos. Para instâncias com 60, 80 e 100 nodos só é possível comparar as Formulações MF, MTZ e WMTZ e os Procedimentos P-CS e P-WMTZ+C dado que as Formulações ES e WE apresentam falta de memória. Na Tabela 3.17 a primeira coluna refere-se à designação da instância, a segunda coluna contém o *gap* obtido usando a Formulação MF_L e os Procedimentos $P-CS_L$ e $P-WMTZ+C_L$. A coluna seis contém o *gap* obtido usando as Formulações MTZ_L e $WMTZ_L$. Nas restantes colunas podemos observar os tempos de execução (em segundos) necessários para a obtenção do valor ótimo ou de um limite superior para o valor do custo usando as Formulações MF, MTZ e WMTZ e os Procedimentos P-CS e P-WMTZ+C.

Instância	<i>Gap</i>		Tempos		<i>Gap</i>	Tempos	
		MF	P-CS	P-WMTZ+C		MTZ	WMTZ
E60-1	0,39	464,96	43,67	11,75	3,10	76,66	47,35
E60-2	0,49	2545,94	69,44	11,36	5,92	579,28	755,92
E60-3	0,34	254,42	35,68	7,85	4,35	793,01	532,07
E60-4	0,39	322,67	39,73	10,62	7,29	710,44	4768,68
E60-5	0,34	902,43	33,24	20,22	9,20	5000,41 ^a	4999,99 ^a
E80-1	0,22	397,72	175,42	29,89	7,18	5000,32 ^a	5000,33 ^a
E80-2	0,24	408,71	283,39	29,27	3,47	161,85	142,68
E80-3	0,23	5068,13	151,38	61,28	7,88	4999,64 ^a	5000,23 ^a
E80-4	0,25	1659,03	160,81	42,79	7,19	5000,31 ^a	5000,48 ^a
E80-5	0,21	5967,62 ^a	224,94	42,35	8,28	5000,47 ^a	5000,51 ^a
E100-1	0,11	6056,17 ^a	686,10	78,61	3,35	2032,71	2462,64
E100-2	0,06	4376,29	518,48	98,70	7,62	4999,99 ^a	5000,17 ^a
E100-3	0,09	3827,08	476,77	166,30	6,00	4999,63 ^a	5000,31 ^a
E100-4	0,10	1384,07	478,98	102,98	5,35	5001,39 ^a	5000,79 ^a
E100-5	0,10	2665,98	527,56	135,39	6,79	5000,34 ^a	5000,96 ^a

Tabela 3.17: Comparação das formulações em instâncias E de 20, 40, 60 e 80 nodos.

^a Não foi possível provar a otimalidade no tempo imposto.

Os *gaps* obtidos usando a Formulação MF_L e os Procedimentos $P-CS_L$ e

P-WMTZ+C_L são inferiores aos *gaps* obtidos usando as Formulações MTZ_L e WMTZ_L em todas as instâncias apresentadas na Tabela 3.17. Usando a Formulação MF obtêm-se tempos de execução mais elevados. Em instâncias de 60, 80 e 100 nodos, usando o Procedimento P-WMTZ+C obtêm-se tempos de execução inferiores.

3.4.4 Resultados Computacionais

Na subsecção anterior verificámos que, em geral, o Procedimento P-WMTZ+C é o mais eficiente nos três grupos de instâncias consideradas. Para este procedimento nas 215 instâncias testadas, não foi possível obter solução inteira admissível em 14 instâncias, sendo 4 delas instâncias de 1000 nodos do grupo Quase Caminhos, 1 instância de 1000 nodos do grupo Aleatórias e 9 instâncias de 500 e 1000 nodos do grupo Euclidianas.

Em instâncias Quase Caminhos a partir de 100 nodos (exceto QC100-10) não é possível resolver as instâncias até provar a otimalidade, o que corresponde a 46,32% das instâncias (44 instâncias em 95). Em relação às instâncias Aleatórias apenas não é possível provar a otimalidade em 3 instâncias de 1000 nodos (5% das instâncias) e por fim no grupo de instâncias Euclidianas não foi possível provar a otimalidade em 25% das instâncias (15 instâncias em 60), sendo elas instâncias com mais de 300 nodos.

Na Tabela 3.18 apresentamos para os três grupos de instâncias os tempos médios, em segundos, e os respetivos desvios padrões obtidos com a aplicação do Procedimento P-WMTZ+C na obtenção do valor ótimo ou de um limite superior para o valor ótimo. A primeira coluna contém o número de nodos das instâncias, as três colunas seguintes apresentam os tempos médios, em segundos, necessários para a obtenção do valor ótimo ou de um limite superior para o valor ótimo e as três últimas colunas contêm os correspondentes desvios padrões.

	Tempo Médio			Desvio Padrão		
N.º Nodos	QC	R	E	QC	R	E
10	0,109	0,187	0,109	0,042	0,136	0,211
20	1,350	1,133	0,893	0,867	0,625	0,558
40	41,756	4,134	5,597	29,749	1,188	1,213
60	517,422	5,960	12,358	338,821	1,188	4,650
80	2535,656	14,249	41,115	2095,439	2,325	13,012
100	10597,763	20,683	116,395	646,021	6,096	34,540
150	10809,301	95,072	585,157	3,413	37,464	181,108
200	10826,760	310,588	1905,269	9,115	222,722	737,561
300	10931,065	1566,215	12368,562	68,410	866,040	3070,597
400	11054,431	2647,825	23590,768	159,831	854,527	5483,945
500	11307,302	9225,342	27611,014	267,266	2740,351	7483,639
1000	13082,191	17168,960	31071,300	823,942	2724,852	86,227

Tabela 3.18: Tempos médios, em segundos, e respetivos desvios padrões obtidos com a aplicação do Procedimento P-WMTZ+C na obtenção do valor ótimo ou de um limite superior para o valor ótimo em cada grupo de instâncias.

Nas instâncias de 10 e 20 nodos, os tempos médios de execução obtidos com a aplicação do Procedimento P-WMTZ+C nos vários grupos de instâncias são semelhantes. Para instâncias com mais de 20 e menos de 300 nodos, verificamos que as instâncias Quase Caminhos são as que apresentam maior tempo médio na obtenção do valor ótimo ou de um limite superior para o valor ótimo, depois temos as instâncias Euclidianas e por fim as instâncias Aleatórias. Nas instâncias de 300 ou mais nodos o comportamento dos tempos altera-se, sendo as instâncias Euclidianas as que apresentam maior tempo médio de execução.

Dado que através da observação da Tabela 3.18 se verificou uma grande diferença nos tempos médios de execução do Procedimento P-WMTZ+C (tempo de execução da relaxação linear + tempo de execução do Algoritmo *Branch and Bound*) nos três grupos de instâncias em teste, decidimos analisar os tempos médios de execução da relaxação linear e os tempos médios de execução do Algoritmo *Branch and Bound* nos vários grupos de instâncias. Com esta análise pretendemos averiguar se existem diferentes comportamentos a nível destes dois tempos médios. Para efetuar a análise elaborámos os gráficos das Figuras 3.11, 3.12 e 3.13 para os grupos de instâncias Aleatórias, Euclidianas e Quase Caminhos para instâncias com mais de 40 nodos. Nestes gráficos encontram-se as percentagens de tempo médio obtidas com a execução da relaxação

linear e as percentagens de tempo médio obtidas com a execução do Algoritmo *Branch and Bound*. Estas duas percentagens distinguem-se nos gráficos por duas cores diferentes. Cada gráfico contém ainda uma tabela onde se pode observar o tempo médio, em segundos, de execução da relaxação linear (Tempo PL) e o tempo médio, em segundos, de execução do Algoritmo *Branch and Bound* (Tempo B&B).

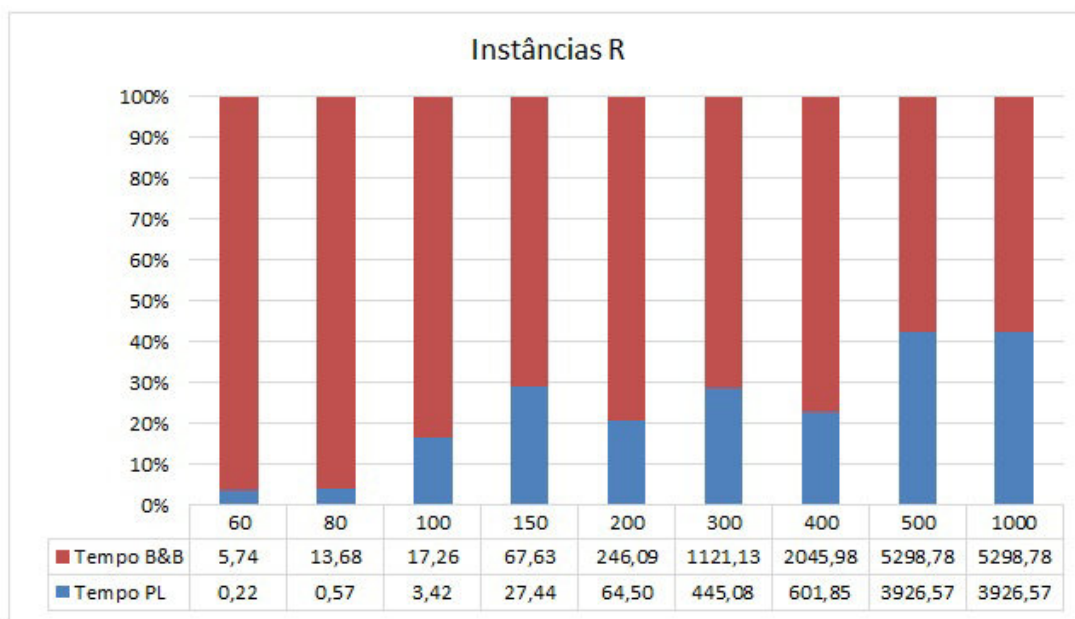


Figura 3.11: Comparação das percentagens de tempo médio de execução (em segundos) da relaxação linear e do Algoritmo *Branch and Bound* do Procedimento P-WMTZ+C em instâncias R.

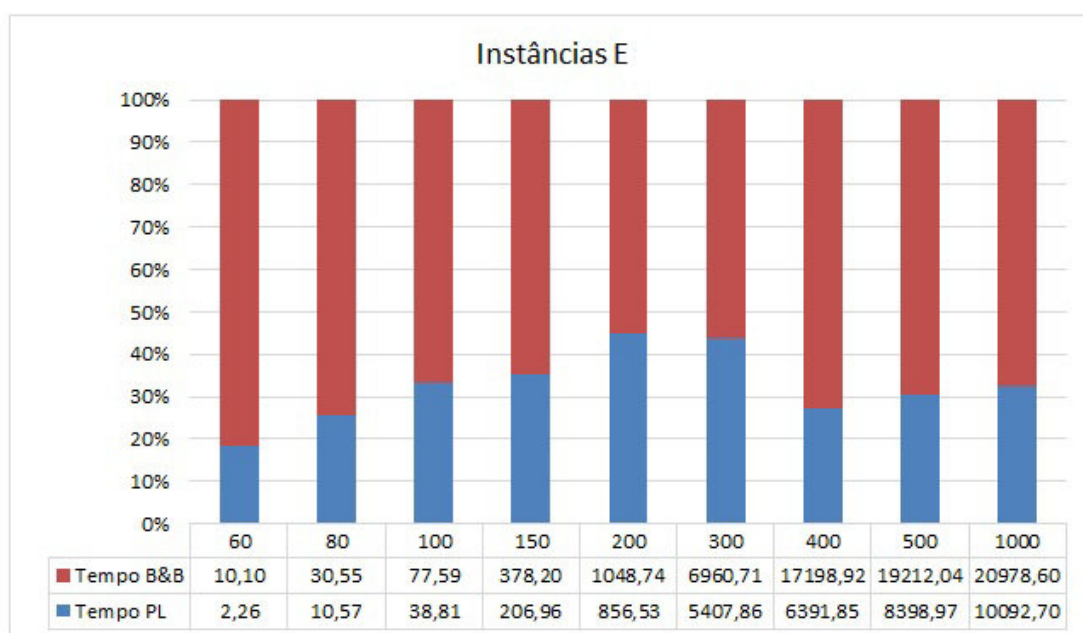


Figura 3.12: Comparação das percentagens de tempo médio de execução (em segundos) da relaxação linear e do Algoritmo *Branch and Bound* do Procedimento P-WMTZ+C em instâncias E.

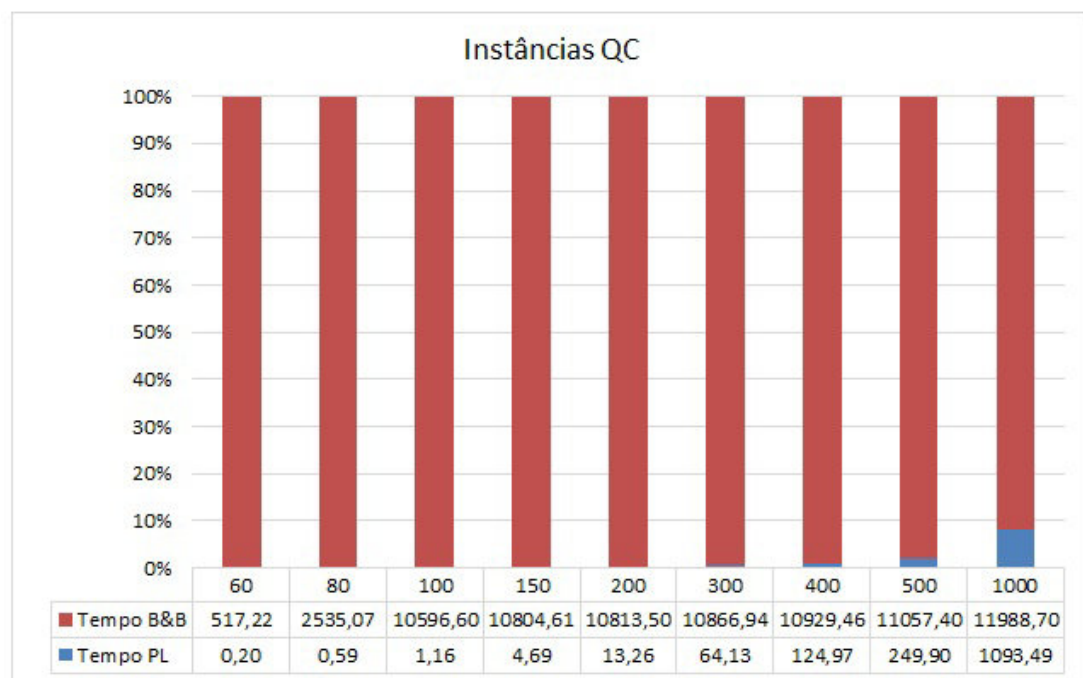


Figura 3.13: Comparação das percentagens de tempo médio de execução (em segundos) da relaxação linear e do Algoritmo *Branch and Bound* do Procedimento P-WMTZ+C em instâncias QC.

Podemos observar que nos vários grupos de instâncias existem dois comportamentos diferentes. Nas instâncias Aleatórias e Euclideanas os comportamentos dos tempos médios são semelhantes. Em instâncias de dimensão superior a 100 nodos, em média, mais de 20% do tempo é ocupado a resolver a relaxação linear e o restante do tempo na execução do Algoritmo *Branch and Bound* para encontrar o valor ótimo ou um limite superior para o valor ótimo. Este comportamento não se verifica no grupo de instâncias Quase Caminhos, pois neste grupo de instâncias o valor da relaxação linear é obtido rapidamente em comparação com o tempo de execução do Algoritmo *Branch and Bound* para encontrar o valor ótimo ou um limite superior para o valor ótimo. Nas instâncias Quase Caminhos mais de 90% do tempo médio de execução é gasto no Algoritmo *Branch and Bound*. Notamos que a percentagem representada a azul nos gráficos das Figuras 3.11 e 3.12 poderia ser maior, no caso de não ser introduzido um procedimento de interrupção da relaxação linear. Este procedimento consistiu em interromper a relaxação linear quando o *gap*, em percentagem, entre duas soluções consecutivas for inferior a 0,000001, mais do que 20 vezes ou quando o limite de tempo na introdução de cortes exceder 10000 segundos (limite imposto).

3.4.5 Síntese dos Resultados Computacionais

Os resultados computacionais mostram a seguinte relação entre os valores obtidos na relaxação linear

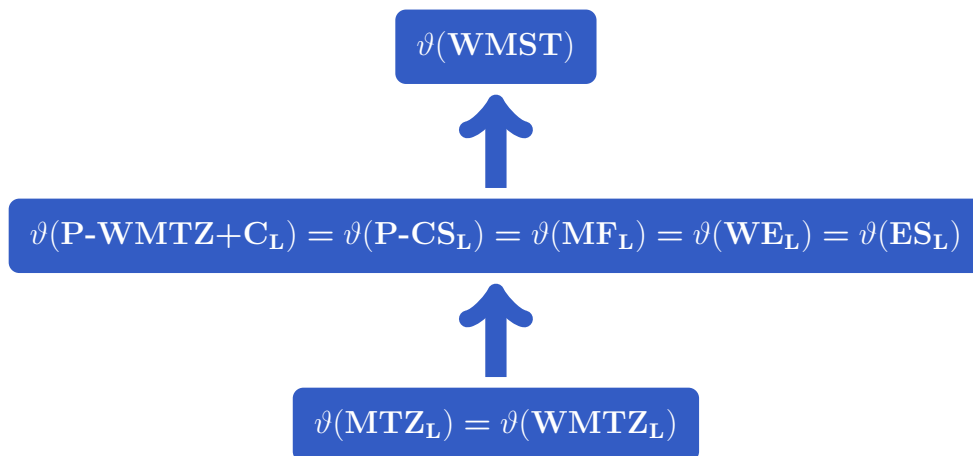


Figura 3.14: Relação entre os valores obtidos pela relaxação linear das diferentes formulações e procedimentos de introdução de cortes.

Os valores dos limites inferiores obtidos usando as Formulações MTZ e WMTZ encontram-se mais afastados do valor ótimo e os obtidos usando as Formulações ES, MF e WE e os Procedimentos P-CS e P-WMTZ+C encontram-se mais próximos do valor ótimo.

De entre as duas Formulações MTZ e WMTZ a que, em geral, apresenta tempos médios de execução da relaxação linear inferiores é a Formulação WMTZ.

De entre todas as formulações e procedimentos de introdução de cortes usando separação, o Procedimento P-WMTZ+C é o que apresenta os melhores resultados em termos de tempo de execução da relaxação linear e do Algoritmo *Branch and Bound* para todos os grupos de instâncias em teste.

Em instâncias Aleatórias e Euclidianas de maiores dimensões, a resolução da relaxação linear torna-se lenta, enquanto que nas instâncias Quase Caminhos é bastante rápida.

Capítulo 4

Algoritmos Lagrangeanos

Neste capítulo vamos apresentar algoritmos baseados numa relaxação Lagrangeana que determinam uma solução aproximada para o Problema WMST. Este tipo de algoritmos são muito comuns para o Problema do Caminho Mais Curto com Restrições de Peso (CSP - *Constrained Shortest Path*) [13, 33, 40, 41, 64, 65]. Notamos que a forma de resolução através desta técnica é muito semelhante para ambos os problemas CSP e WMST, de tal modo que é frequente nos artigos referentes ao CSP aparecer a observação de que o mesmo pode ser aplicado ao Problema WMST. Neste capítulo, para além de apresentarmos resultados computacionais para algoritmos adaptados do CSP para o WMST, apresentamos três novos algoritmos para o WMST e os respetivos resultados computacionais.

O trabalho de Xue [65] descreve dois algoritmos: um para encontrar soluções aproximadas para o Problema CSP e outro para o Problema WMST. Os resultados computacionais para o primeiro algoritmo mostraram ser bastante bons, encontrando soluções ótimas em mais de 80% dos casos e soluções próximas das ótimas nos restantes casos. Não são apresentados quaisquer resultados computacionais para o algoritmo para o WMST. Os autores apenas manifestam o seu interesse em fazê-lo futuramente para grafos aleatórios e comparar com soluções ótimas obtidas usando o Algoritmo *k Smallest Spanning Tree* [16].

No artigo de Block e Gutin [13] podemos consultar um algoritmo aproximado bastante eficiente para problemas de Otimização Combinatória com dois parâmetros associados a elementos combinatórios em geral. Embora o algoritmo não seja polinomial, o autor fornece algumas evidências teóricas e práticas que mostram que o algoritmo pode ser bastante rápido em muitos casos.

Jüttner et al. [41] desenvolveram o Algoritmo LARAC (*Lagrangean Relaxation*

Based Agregated Cost) para resolver a relaxação Lagrangeana proposta para o Problema CSP. Os autores aplicam uma aproximação algébrica e estabelecem vários resultados relacionados com a estrutura de soluções ótimas da relaxação Lagrangeana. Jüttner [40] provou a complexidade do Algoritmo LARAC. No artigo de Xiao et al. [64] foi apresentada a equivalência de certos algoritmos, simplesmente designados LARAC, os quais foram apresentados independentemente em alguns trabalhos anteriores [13, 33, 41]. Este artigo também apresenta um estudo algébrico que estabelece diversas novas propriedades de solução ótima e um novo algoritmo chamado LARAC-BIN baseado numa pesquisa binária.

Yamada et al. [66] apresentam um algoritmo baseado numa relaxação Lagrangeana para o Problema da Árvore de Suporte de Custo Máximo com Restrições de Peso, o qual é semelhante ao Algoritmo LARAC-BIN descrito em [64] para o Problema CSP.

Amado e Barcia [7] apresentam uma relaxação para o Problema da Mochila Matroidal (*Matroidal Knapsack*), sendo casos especiais deste problema, o Problema Saco-mochila de Múltipla Escolha e o Problema WMST, sendo este denominado pelos autores de Problema da Árvore de Suporte de um Grafo com uma Restrição de Capacidade.

Com o objetivo de obter soluções aproximadas para o Problema WMST, na Secção 4.1 fazemos uma breve descrição da relaxação Lagrangeana para o problema. Nas duas secções seguintes (Secção 4.2 e 4.3) descrevemos um algoritmo genérico baseado na relaxação Lagrangeana, que denominamos de Algoritmo Lagrangeano Base para o Problema WMST, fazemos uma análise da complexidade e apresentamos a ideia geométrica do algoritmo. Este algoritmo descreve e uniformiza os passos comuns a todos os algoritmos e na Secção 4.4 apresentamos vários algoritmos tendo todos como princípio o mesmo algoritmo base apresentado. Na Secção 4.5 descrevemos algumas experiências computacionais realizadas e apresentamos os resultados computacionais, onde se efetua uma comparação entre os vários algoritmos, a nível de tempo e da qualidade das soluções aproximadas obtidas. Para finalizar o capítulo apresentamos uma síntese dos resultados computacionais obtidos.

4.1 Relaxação Lagrangeana para o Problema WMST

Nesta secção, considera-se a aplicação da relaxação Lagrangeana ao Problema WMST, sendo esta uma técnica clássica usada como alternativa à relaxação de Programação Linear para calcular limites inferiores para o valor ótimo e encontrar boas soluções para Problemas de Otimização com Restrições.

Para obtermos a relaxação Lagrangeana da formulação genérica para o problema WMST que se encontra na parte introdutória do Capítulo 3, associamos um multiplicador de Lagrange λ ($\lambda \geq 0$) à restrição de peso (3.3) e incluímos essa restrição, à moda Lagrangeana, na função objetivo. Obtemos, deste modo o seguinte problema relaxado.

$$(WMST_\lambda) : \quad -\lambda W \quad + \quad \min \quad \sum_{(i,j) \in A} (c_{ij} + \lambda w_{ij}) x_{ij} \\ \text{s.a.} \quad x \in X_T,$$

onde $x = (x_{ij}) \in \mathbb{R}^{|A|}$ e X_T descrevem o invólucro convexo das soluções inteiras do Problema MST. Para todos os multiplicadores não negativos λ , as soluções com estrutura em árvore para este problema relaxado dão-nos limites inferiores para o valor ótimo, isto é,

$$\vartheta(WMST_\lambda) \leq \vartheta(WMST).$$

O Problema relaxado $WMST_\lambda$ pode ser resolvido usando qualquer algoritmo polinomial conhecido para o Problema MST [6]. Se, para cada multiplicador $\lambda \geq 0$, definirmos os valores ponderados $p_{ij}^\lambda = c_{ij} + \lambda w_{ij}$ associados a cada arco $(i, j) \in A$ e T_{p^λ} for a correspondente árvore de suporte ponderada mínima com custo $C(T_{p^\lambda})$ e peso $W(T_{p^\lambda})$, então a função Lagrangeana pode-se escrever do seguinte modo

$$\vartheta(WMST_\lambda) = -\lambda W + P(T_{p^\lambda}),$$

onde $P(T_{p^\lambda}) = C(T_{p^\lambda}) + \lambda W(T_{p^\lambda})$. Seja T_{λ^*} a árvore correspondente à solução ótima do Problema WMST, com custo $C(T_{\lambda^*})$ e peso $W(T_{\lambda^*})$, então consegue-se facilmente mostrar que o valor da relaxação Lagrangeana é um limite inferior para o valor ótimo, isto é,

$$\begin{aligned} \vartheta(WMST_\lambda) &= -\lambda W + P(T_{p^\lambda}) \\ &\leq -\lambda W + P(T_{\lambda^*}) \end{aligned}$$

$$\begin{aligned}
&= -\lambda W + C(T_{\lambda^*}) + \lambda W(T_{\lambda^*}) \\
&= C(T_{\lambda^*}) + \lambda(W(T_{\lambda^*}) - W) \\
&\leq C(T_{\lambda^*}) \\
&= \vartheta(WMST).
\end{aligned}$$

Para obter o melhor limite inferior há que resolver o Problema Dual Lagrangeano, isto é, precisamos de maximizar a função $\vartheta(WMST_\lambda)$ para todo $\lambda \geq 0$, ou seja,

$$\begin{aligned}
\vartheta^* &:= \max_{\lambda \geq 0} \vartheta(WMST_\lambda) \\
&\text{s.a. } \lambda \geq 0.
\end{aligned}$$

Seja λ^* o valor de λ que maximiza $\vartheta(WMST_\lambda)$.

De seguida apresentam-se duas propriedades da função $\vartheta(WMST_\lambda)$.

Lema 4.1.

Para qualquer $\lambda \geq 0$, $\frac{d\vartheta(WMST_\lambda)}{d\lambda} = W(T_\lambda) - W$ é um subgradiente de $\vartheta(WMST_\lambda)$, onde a árvore de suporte T_λ corresponde à solução ótima do problema relaxado $WMST_\lambda$.

Lema 4.2.

Para qualquer $\lambda \geq 0$, $\vartheta(WMST_\lambda)$ é uma função linear por partes e côncava.

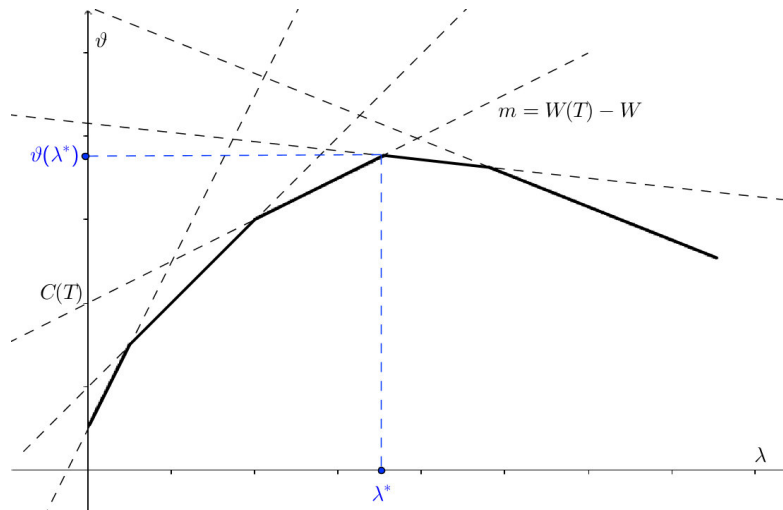


Figura 4.1: Representação gráfica de $\vartheta(WMST_\lambda)$ em função do valor de λ .

A cada árvore de suporte T com custo $C(T)$ e peso $W(T)$, podemos fazer corresponder a função linear $\vartheta(\lambda) = C(T) + \lambda(W(T) - W)$ com $\lambda \geq 0$, a qual tem ordenada na origem $C(T)$ e declive $W(T) - W$. Quando obtemos uma árvore não admissível, temos que $W(T) > W$, o que corresponde a um declive positivo. No caso da árvore ser admissível, então $W(T) \leq W$, ou seja, $W(T) - W \leq 0$, o que corresponde a um declive negativo ou nulo. O gráfico da Figura 4.1, representa o invólucro superior de todas as retas correspondentes a todas as árvores de suporte do grafo G . Podemos observar que a função $\vartheta(WMST_\lambda)$ é linear por partes e côncava.

A relaxação Lagrangeana pode ser resolvida, na grande maioria das vezes, usando o Método do Subgradiente [59]. Este método começa por inicializar o multiplicador de Lagrange, λ_0 . Depois, iterativamente, resolve o Problema relaxado $WMST_{\lambda_k}$, atualiza, em cada iteração o multiplicador de Lagrange para $\lambda_{k+1} = \max\{0, \lambda_k + s_k d_k\}$ usando a direção d_k e o tamanho do passo s_k e finalmente verifica se o critério de paragem é satisfeito.

Uma escolha apropriada para a direção d_k e para o tamanho do passo s_k produz um método convergente.

Para a direção d_k podemos usar as ideias de Held, Wolfe e Crowder [36]

$$d_k = \sum_{(i,j) \in A} w_{ij} x_{ij}^k - W.$$

Como à solução $x^k = (x_{ij}^k)$ do problema relaxado $WMST_{\lambda_k}$, corresponde a árvore de suporte $T_{p^{\lambda_k}}$, temos que $\sum_{(i,j) \in A} w_{ij} x_{ij}^k$ é o peso dessa árvore. Assim, temos que

$$d_k = W(T_{p^{\lambda_k}}) - W.$$

Podemos usar o tamanho do passo s_k de acordo com [59]

$$s_k = \rho \frac{C(T_w) - \vartheta(WMST_{\lambda_k})}{(\sum_{(i,j) \in A} w_{ij} x_{ij}^k - W) d_k},$$

onde $0 < \rho < 2$, $C(T_w)$ é um limite superior para aproximar o valor ótimo do Problema WMST e $\vartheta(WMST_{\lambda_k}) = -\lambda_k W + P(T_{p^{\lambda_k}})$. Portanto, o tamanho do passo s_k é dado por

$$s_k = \rho \frac{C(T_w) - P(T_{p^{\lambda_k}}) + \lambda_k W}{(W(T_{p^{\lambda_k}}) - W) d_k}.$$

Assim, em cada iteração atualiza-se o multiplicador λ_k e definem-se os valores ponderados $p_{ij}^{\lambda_k} = c_{ij} + \lambda_k w_{ij}$ associados a cada arco $(i, j) \in A$ e obtém-se uma árvore de suporte ponderada mínima $T_{p^{\lambda_k}}$ com valor ponderado $P(T_{p^{\lambda_k}}) = C(T_{p^{\lambda_k}}) + \lambda_k W(T_{p^{\lambda_k}})$.

4.2 Algoritmo Lagrangeano Base

Esta abordagem algorítmica determina sucessivamente árvores de suporte até encontrar o melhor valor da variável dual Lagrangeana λ . O objetivo do algoritmo é procurar melhores soluções admissíveis do que a correspondente solução da árvore T_w usando combinações lineares do custo e do peso das soluções das árvores T_c e T_w . As sucessivas árvores de suporte obtidas podem ser admissíveis ou não admissíveis. No caso da árvore obtida ser admissível atualiza-se o valor do limite superior (LS) para o custo, caso contrário, atualiza-se o valor do limite inferior (LI) para o custo.

Apresentamos de seguida, em forma esquematizada, o Algoritmo Lagrangeano Base (ALagB) para o Problema WMST.

Algoritmo Lagrangeano Base (ALagB) para o Problema WMST

Passo 1: *Inicializações*

Passo 1.1: *Obter um limite inferior*

Obter a árvore de suporte de custo mínimo $T_c = (V, A_{T_c})$.

Se $W(T_c) \leq W$, então

T_c é a árvore correspondente à solução ótima. STOP.

Caso contrário,

$$T_\beta = T_c.$$

Calcular $C(T_\beta)$.

Passo 1.2: *Inicializar o intervalo (quando aplicável)*

Inicializar o intervalo $[\ell_0, u_0]$ e a árvore $T_{\ell_0} = T_c$ (obtida no Passo 1.1).

Calcular os valores ponderados $p_{ij} = u_0 w_{ij} + c_{ij}$ para cada arco $(i, j) \in A$.

Obter a árvore de suporte ponderada mínima $T_{u_0} = (V, A_{T_{u_0}})$.

Se $W(T_{u_0}) > W$, então

não existe solução, STOP.

Caso contrário,

$$T_\alpha = T_{u_0}.$$

Calcular $C(T_\alpha)$ e $P(T_\alpha)$ e ir para o **Passo 2**.

Passo 1.3: *Obter um limite superior*

Obter a árvore de suporte de peso mínimo $T_w = (V, A_{T_w})$.

Se $W(T_w) > W$, então

não existe solução, STOP.

Caso contrário,

$$T_\alpha = T_w.$$

Calcular $C(T_\alpha)$ e $P(T_\alpha)$.

Passo 2: *Obter uma nova árvore*

Calcular os valores ponderados $p_{ij} = aw_{ij} + bc_{ij}$ para cada arco $(i, j) \in A$.

Obter a árvore de suporte ponderada mínima $T_p = (V, A_{T_p})$ e calcular $C(T_p)$, $W(T_p)$ e $P(T_p)$.

Passo 3: *Atualização de limites*

Se $W(T_p) \leq W$, então

atualizar o LS, isto é, se $C(T_p) \leq C(T_\alpha)$ substituir T_α por T_p .

Caso contrário,

atualizar o LI, isto é, se $C(T_p) \geq C(T_\beta)$ substituir T_β por T_p .

Passo 4: *Critério de paragem*

Se $|P(T_\alpha) - P(T_p)| \leq tol$, então

T_α é a árvore correspondente à solução aproximada, STOP.

Caso contrário,

ir para o **Passo 2**.

No Passo 1.1 é obtida a árvore de suporte de custo mínimo T_c . Se esta não verificar a restrição de peso, então foi encontrado um limite inferior para o valor do custo e $T_\beta = T_c$. No caso de verificação da restrição de peso significa que foi encontrada a solução ótima e o algoritmo termina. No Passo 1.3 é obtida a árvore de suporte de peso mínimo, T_w . Se esta não verificar a restrição de peso, então não existe solução e o algoritmo termina. Caso a restrição de peso seja verificada, então foi encontrado um limite superior para o valor do custo e $T_\alpha = T_w$.

O Passo 1.2 só é aplicável quando os limites inferior e superior forem inicializados tendo em conta o intervalo $[\ell_0, u_0]$, onde a árvore correspondente a ℓ_0 é a árvore de suporte de custo mínimo ($T_{\ell_0} = T_c$) obtida no Passo 1.1 e para obter a árvore de suporte correspondente a u_0 é necessário calcular os valores ponderados $p_{ij} = u_0 w_{ij} + c_{ij}$, para cada arco $(i, j) \in A$, o parâmetro u_0 depende do algoritmo utilizado e será definido na próxima secção. Se a árvore de suporte ponderada mínima T_{u_0} verificar a restrição de peso é obtido um limite superior para o valor do custo e $T_\alpha = T_{u_0}$ e neste caso vamos para o Passo 2. Caso contrário, não existe solução para o Problema WMST e o algoritmo termina.

No Passo 2 são determinados os valores ponderados $p_{ij} = aw_{ij} + bc_{ij}$ para cada arco $(i, j) \in A$ e é obtida a árvore de suporte ponderada mínima T_p . Os valores dos parâmetros a e b que utilizamos para definir os valores ponderados p_{ij} vão depender do algoritmo utilizado e serão definidos na próxima secção. No Passo 3 atualizam-se os limites, isto é, no caso da restrição de peso ser verificada e o custo da árvore T_p não ser superior ao custo da árvore T_α , atualizamos o valor do limite superior e substituímos T_α por T_p . Caso a restrição de peso seja violada e o custo da árvore T_p não seja inferior ao custo da árvore T_β , atualizamos o valor do limite inferior e substituímos T_β por T_p . Os Passos 2 e 3 são executados até que se verifique $|P(T_\alpha) - P(T_p)| \leq tol$, e quando tal acontece o algoritmo termina, sendo T_α a árvore correspondente à solução aproximada.

A complexidade do Algoritmo ALagB depende do algoritmo utilizado para a obtenção das sucessivas árvores de suporte e do número de árvores que é possível determinar.

Quanto ao algoritmo que determina as árvores de suporte, a sua complexidade é $O(\varphi(m, n))$ [5], onde a função $\varphi(m, n) = m \log(n)$ depende de duas variáveis, sendo elas o número de arestas $m = |E|$ e o número de nodos $n = |V|$ do grafo $G = (V, E)$. No Passo 1 do Algoritmo ALagB, são obtidas duas árvores e para a obtenção de cada

uma delas usamos um algoritmo que efetua $O(\varphi(m, n))$ operações.

Se K for o número total de árvores de suporte que se podem formar no grafo G , então o algoritmo vai terminar ao fim de $O(\log(K))$ iterações, pois o número de árvores de suporte a determinar no pior dos casos é proporcional a K . Portanto, os Passos 2, 3 e 4 do algoritmo obtêm-se fazendo $O(\log(K)) \times O(\varphi(m, n))$ operações. Logo, no Algoritmo ALagB para obter os valores dos limites inferior e superior são necessárias $O(\log(K) \varphi(m, n))$ operações.

4.3 Comportamento Geométrico do Algoritmo Lagrangeano Base para o Problema WMST

Para estudar o comportamento geométrico do algoritmo descrito na secção anterior consideramos um sistema de coordenadas cartesianas, onde o eixo horizontal representa os pesos e o eixo vertical os custos. Os pesos e os custos das árvores T_c e T_w podem ser representados como pontos com coordenadas $A = (W(T_c), C(T_c))$ e $B = (W(T_w), C(T_w))$ (ver Figura 4.2). A equação da reta AB é da forma $aW + bC = d$, onde $a = C(T_w) - C(T_c)$, $b = W(T_c) - W(T_w)$ e $d = W(T_c)C(T_w) - W(T_w)C(T_c)$.

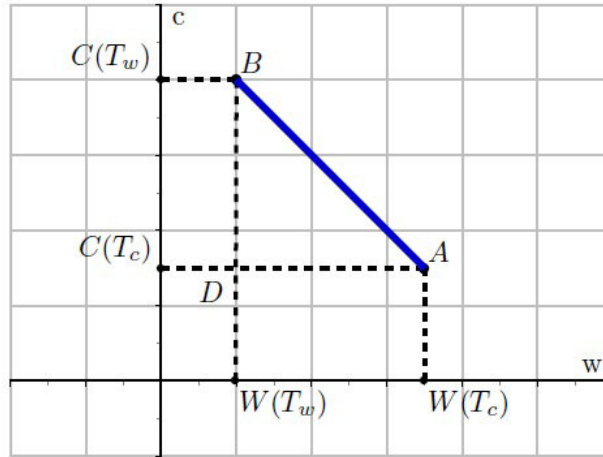


Figura 4.2: Representação geométrica dos pesos e dos custos das árvores T_c e T_w .

Como existe a restrição de peso, temos que introduzir na figura anterior a reta de equação $w = W$ (apresentada a vermelho nas Figuras 4.3, 4.4 e 4.5). Dessa introdução

podem ocorrer três situações:

- A primeira situação ocorre quando a árvore T_w não satisfaz a restrição de peso e então não há solução para o Problema WMST (Figura 4.3).

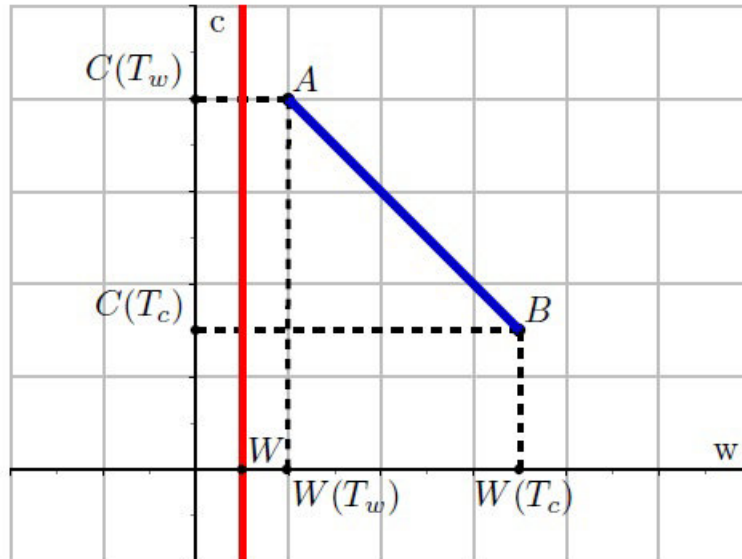


Figura 4.3: Representação geométrica dos pesos e dos custos das árvores T_c e T_w e da reta $w = W$ com $W(T_w) > W$.

Note-se que no caso de não existir nenhuma árvore com custo inferior a $C(T_w)$ e $W(T_w) = W$, então T_w é a árvore correspondente à solução ótima.

- A segunda situação ocorre quando a árvore T_c satisfaz a restrição de peso e então concluímos que T_c é a árvore correspondente à solução ótima (Figura 4.4).

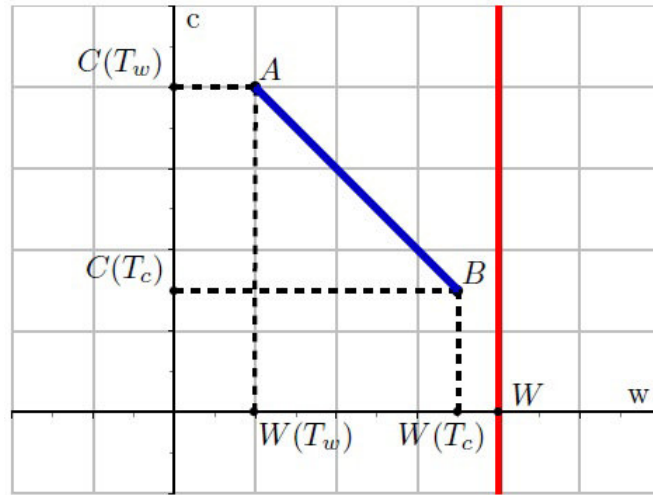


Figura 4.4: Representação geométrica dos pesos e dos custos das árvores T_c e T_w e da reta $w = W$ com $W(T_c) \leq W$.

- A terceira e última situação ocorre quando não se verificam as condições anteriores (ver Figura 4.5). Nesta situação pretendemos obter uma árvore de suporte cujo valor de custo se aproxime do valor ótimo. Para a obter associamos novos valores p_{ij} a cada arco $(i, j) \in A$ e obtemos uma árvore de suporte ponderada mínima $T_p = (V, A_{T_p})$, onde $W(T_w) < W(T_p) < W(T_c)$ e $C(T_c) < C(T_p) \leq C(T_w)$.

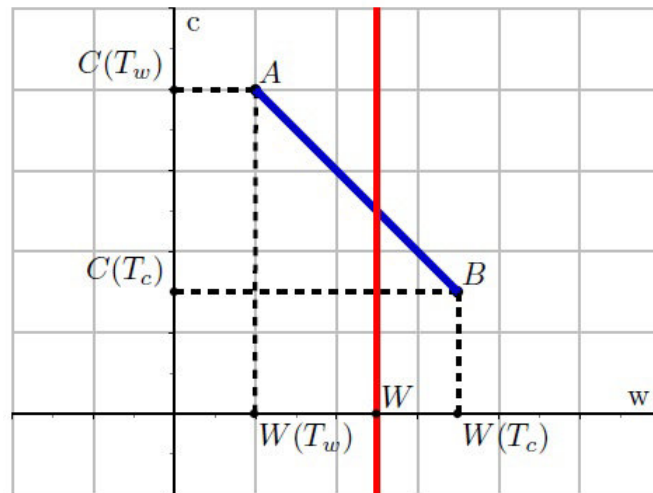


Figura 4.5: Representação geométrica dos pesos e dos custos das árvores T_c e T_w e da reta $w = W$, onde $W(T_w) < W < W(T_c)$.

Se o ponto $(W(T_p), C(T_p))$ correspondente à árvore T_p pertence à reta AB , então isso significa que encontramos a solução aproximada, que corresponde à árvore T_p se esta for uma árvore admissível, isto é, $W(T_p) \leq W$.

Se o ponto $(W(T_p), C(T_p))$ correspondente à árvore T_p não pertence à reta AB e a árvore T_p for admissível, então substituímos T_w por T_p e obtemos um novo valor para o limite superior, caso contrário substituímos T_c por T_p e atualizamos o valor do limite inferior. Após termos atualizado um dos limites, repetimos o Passo 2 do Algoritmo ALagB, ou seja, voltamos a obter uma nova árvore de suporte ponderada mínima T_p .

Saliente-se que as sucessivas árvores T_w que designamos no Algoritmo ALagB por T_α são sempre árvores admissíveis, enquanto que as árvores T_c , que designamos no Algoritmo ALagB por T_β , são sempre árvores não admissíveis.

Recordamos que o objetivo é encontrar soluções admissíveis com custo inferior ao da árvore T_w .

4.4 Definição dos Valores Ponderados Utilizados no Algoritmo Lagrangeano Base para o Problema WMST

No Algoritmo ALagB os valores ponderados definidos no Passo 2 são da seguinte forma:

$$p_{ij} = aw_{ij} + bc_{ij}$$

para cada arco $(i, j) \in A$. Consoante os valores atribuídos aos parâmetros a e b assim vamos obter diferentes algoritmos. De seguida, discutimos diferentes formas de obtenção dos valores ponderados.

Algoritmo Lagrangeano 1 (Alg1)

Neste algoritmo consideramos que os valores ponderados p_{ij} ($(i, j) \in A$) são caracterizados por associar o parâmetro $a = 1 - \gamma_k$ para os pesos e o parâmetro $b = \gamma_k$ para os custos segundo o artigo [65], ou seja,

$$p_{ij} = (1 - \gamma_k)w_{ij} + \gamma_k c_{ij}$$

onde, o parâmetro $\gamma_k \in [0, 1]$ é obtido da seguinte forma:

$$\gamma_k = \begin{cases} \frac{W(T_\beta) - W(T_\alpha)}{C(T_\alpha) - C(T_\beta) + W(T_\beta) - W(T_\alpha)}, & \text{se a iteração } k \text{ for ímpar} \\ \frac{\gamma_a + \gamma_b}{2}, & \text{se a iteração } k \text{ for par.} \end{cases}$$

Neste algoritmo o Passo 1.2 não é necessário. Os valores de γ_a e γ_b são inicializados a 0 e 1 respetivamente. Estes valores são atualizados no Passo 3. O valor de γ_a é atualizado para γ_k quando a árvore T_p obtida verifica a restrição de peso. O valor de γ_b é atualizado para γ_k quando a árvore T_p obtida não verifica a restrição de peso.

De seguida, apresenta-se uma descrição do Passo 3 para este algoritmo cuja alteração é apenas a atualização dos valores γ_a e γ_b .

Passo 3: *Atualização de limites*

Se $W(T_p) \leq W$, então

atualizar $\gamma_a := \gamma_k$;

atualizar o LS, isto é, se $C(T_p) \leq C(T_\alpha)$ substituir T_α por T_p .

Caso contrário,

atualizar $\gamma_b := \gamma_k$;

atualizar o LI, isto é, se $C(T_p) \geq C(T_\beta)$ substituir T_β por T_p .

Em particular, quando $\gamma_k = 1$, T_p corresponde a T_c da MST, e quando $\gamma_k = 0$, T_p corresponde a T_w da MST_W . A ideia deste algoritmo é encontrar um intervalo, o mais pequeno possível $[\gamma_a, \gamma_b] \subset [0, 1]$, tal que o peso da árvore T_α é menor ou igual a W e o custo da árvore correspondente à solução ótima esteja entre o custo da árvore T_β e

o da árvore T_α . Os valores γ_a e γ_b serão obtidos usando o Método da Bissecção nas iterações pares.

Tendo em conta que os valores de $P(T_\alpha)$ e $P(T_\beta)$ se aproximam, então temos,

$$\gamma_k C(T_\alpha) + (1 - \gamma_k)W(T_\alpha) = \gamma_k C(T_\beta) + (1 - \gamma_k)W(T_\beta)$$

o que é equivalente a ter,

$$\frac{1 - \gamma_k}{\gamma_k} = \frac{C(T_\alpha) - C(T_\beta)}{W(T_\beta) - W(T_\alpha)}.$$

Assim, o valor de λ_k obtido nas sucessivas relaxações Lagrangeanas é dado por:

$$\lambda_k = \frac{1 - \gamma_k}{\gamma_k}.$$

Para exemplificarmos os vários algoritmos propostos ao longo desta secção vamos considerar o Exemplo 2.1.

Exemplo 4.1.

Na seguinte tabela sumariamos os resultados da aplicação do Algoritmo Alg1 para o Exemplo 2.1.

$C(T_\beta)$	$W(T_\beta)$	$C(T_\alpha)$	$W(T_\alpha)$	γ_a	γ_b	γ_k	$C(T_p)$	$W(T_p)$	$P(T_\alpha)$	$P(T_p)$	$\vartheta(\lambda_k)$
17	27	70	11	0	1	0,5	19	24	40,5	21,5	23
19	24	70	11	0	0,5	0,203	41	13	22,984	18,688	13,538
19	24	41	13	0,203	0,5	0,352	27	19	22,844	21,813	25,156
19	24	27	19	0,352	0,5	0,385	27	19	22,077	22,077	25,4

Tabela 4.1: Resultados da aplicação do Algoritmo Alg1 para o exemplo de 5 nodos.

A primeira coluna dá-nos os sucessivos valores dos limites inferiores obtidos para o valor do custo e a terceira coluna dá-nos os sucessivos valores dos limites superiores obtidos para o valor do custo. Os custos e os pesos das árvores de suporte que são, sucessivamente, atualizadas encontram-se a negrito na tabela. O algoritmo usa 4

iterações do Passo 2 para obter a solução ótima com custo 27 e peso 19. A árvore T_α é atualizada duas vezes, enquanto que a árvore T_β é atualizada uma só vez. O valor obtido através da relaxação Lagrangeana é de 25,4.

Algoritmo Lagrangeano 2 (Alg2)

Nos algoritmos descritos em [13, 64] aplicados ao problema do caminho mais curto com restrições de peso, os valores ponderados de cada arco $(i, j) \in A$ são obtidos das seguintes formas:

$$p_{ij} = aw_{ij} + bc_{ij} \quad \text{ou} \quad p_{ij} = \lambda w_{ij} + c_{ij}$$

onde $a = C(T_\alpha) - C(T_\beta)$, $b = W(T_\beta) - W(T_\alpha)$ e $\lambda = \frac{a}{b}$. Podemos ver em [64] que estes valores ponderados são equivalentes. Nesta tese, para descrever um algoritmo aproximado para o Problema WMST vamos associar o parâmetro $a = \lambda_k$ aos pesos e $b = 1$ aos custos,

$$p_{ij} = \lambda_k w_{ij} + c_{ij},$$

onde $\lambda_k = \frac{C(T_\alpha) - c(T_\beta)}{W(T_\beta) - W(T_\alpha)}$ e k designa o número da iteração.

Exemplo 4.2.

Na seguinte tabela resumamos os resultados da aplicação do Algoritmo Alg2 para o Exemplo 2.1.

$C(T_\beta)$	$W(T_\beta)$	$C(T_\alpha)$	$W(T_\alpha)$	λ_k	$C(T_p)$	$W(T_p)$	$P(T_\alpha)$	$P(T_p)$	$\vartheta(\lambda_k)$
17	27	70	11	3,313	41	13	106,438	84,063	17,813
17	27	41	13	1,714	27	19	63,286	59,571	25,286
17	27	27	19	1,25	19	24	50,75	49	24
19	24	27	19	1,6	27	19	57,4	57,4	25,4

Tabela 4.2: Resultados da aplicação do Algoritmo Alg2 para o exemplo de 5 nodos.

A primeira coluna dá-nos os sucessivos valores dos limites inferiores obtidos para o valor do custo e a terceira coluna dá-nos os sucessivos valores dos limites superiores obtidos para o valor do custo. Os custos e os pesos das árvores de suporte que são, sucessivamente, atualizadas encontram-se a negrito na tabela. O algoritmo usa 4 iterações do Passo 2 para obter a solução ótima com custo 27 e peso 19. A árvore T_α é atualizada duas vezes, enquanto que a árvore T_β é atualizada uma só vez. O valor obtido através da relaxação Lagrangeana é de 25,4.

Algoritmo Lagrangeano 3 (Alg3)

Neste algoritmo os valores ponderados de cada arco $(i, j) \in A$ são obtidos como descritos na Secção 4.1, ou seja, os parâmetros a e b são obtidos como no algoritmo anterior, sendo

$$p_{ij} = \lambda_k w_{ij} + c_{ij},$$

onde o multiplicador λ_k é obtido tal como descrito no Método do Subgradiente.

O Método do Subgradiente é uma generalização do Método dos Gradientes, no qual o gradiente da função é substituído por um subgradiente para obter uma nova direção de busca.

Método do Subgradiente

Passo 1: Inicializar o multiplicador de Lagrange

$$\text{Seja } \lambda_0 = \frac{C(T_\alpha) - C(T_\beta)}{W(T_\beta) - W}.$$

Passo 2: Resolver iterativamente o problema relaxado $WMST_{\lambda_k}$

Obter a árvore de suporte $T_{p^{\lambda_k}}$ tal que $\vartheta(WMST_{\lambda_k}) = -\lambda_k W + P(T_{p^{\lambda_k}})$, onde

$$\lambda_k = \max \left\{ 0, \lambda_{k-1} + \rho \frac{C(T_\alpha) - P(T_{p^{\lambda_{k-1}}}) + \lambda_{k-1} W}{W(T_{p^{\lambda_{k-1}}}) - W} \right\}.$$

- Na direção usar as ideias de Held, Wolfe e Crowder [36]

$$d_k = W(T_{p^{\lambda_k}}) - W.$$

- No tamanho do passo usar as ideias de Shor [59]

$$s_k = \rho \frac{C(T_\alpha) - P(T_{p^{\lambda_k}}) + \lambda_k W}{(W(T_{p^{\lambda_k}}) - W)d_k} \quad \text{com } 0 < \rho < 2.$$

Passo 4: *Cr terio de paragem*

Definido no Passo 4 do Algoritmo ALagB para o Problema WMST apresentado na Sec  o 4.2.

No Cap tulo 3 us mos v rias formula  es para encontrar o valor da relaxa  o linear e para resolver o Problema WMST. A principal vantagem de utilizar o M todo do Subgradiente para obter o valor da relaxa  o Lagrangeana comparando com a relaxa  o linear usando as formula  es   a rapidez em termos de tempo computacional. A grande desvantagem reside na escolha do valor de ρ , a qual pode tornar o m todo mais lento e tamb m piorar a qualidade do limite inferior obtido.

Exemplo 4.3.

Na seguinte tabela resumamos os resultados da aplica  o do Algoritmo Alg3 para o Exemplo 2.1 para diferentes valores de ρ .

ρ	Itera��es	N.� Atualiza��es T_c	N.� Atualiza��es T_w	λ	$\vartheta(\lambda)$
0,9	4	1	2	3,086	19,4
0,8	4	1	2	2,743	21,8
0,7	9	4	4	1,700	25,3
0,6	7	3	3	2,649	22,406
0,5	7	3	3	1,679	25,321
0,49	7	3	3	1,662	25,338
0,48	7	3	3	1,645	25,355
0,47	6	2	3	2,258	23,969

Tabela 4.3: Resultados da relaxa  o Lagrangeana aplicando o Algoritmo Alg3 para diferentes valores de ρ .

No Algoritmo Alg3 foram utilizados v rios valores de ρ (Tabela 4.3). Neste exemplo, para valores de ρ pr ximos de 0,48 o valor da relaxa  o Lagrangeana aproxima-se do

valor da relaxação linear ($25,355 < 25,4$). Foram necessárias seis iterações do Passo 2 para obter a solução ótima com custo 27 e peso 19. As árvores T_α e T_β são atualizadas três vezes cada uma.

Algoritmo Lagrangeano 4 (Alg4)

Neste algoritmo para encontrar o valor do λ ótimo usamos a ideia da técnica de pesquisa binária, apresentada em [64] para reduzir o intervalo $[\ell_0, u_0]$. Os valores ponderados são obtidos da mesma forma que nos Algoritmos Alg2 e Alg3, isto é,

$$p_{ij} = \lambda_k w_{ij} + c_{ij},$$

onde $\lambda_k = \frac{\ell_k + u_k}{2}$, ou seja, λ_k é o ponto médio entre ℓ_k e u_k , sendo k o número da iteração.

No Passo 1.2 o intervalo $[\ell_0, u_0]$ é inicializado fazendo $\ell_0 = 0$ e $u_0 = \frac{C(T_\alpha) - C(T_\beta)}{W - W(T_\alpha)}$. O valor u_k é atualizado no Passo 3 quando a restrição de peso for verificada e ℓ_k é atualizado caso a restrição de peso não seja verificada.

De seguida apresentamos a descrição do Passo 3 para este algoritmo cuja única alteração é na atualização dos limites inferior e superior do intervalo $[\ell_k, u_k]$.

Passo 3: *Atualização de limites*

Se $W(T_p) \leq W$, então

atualizar o limite superior do intervalo $u_k := \lambda_k$;

atualizar o LS, isto é, se $C(T_p) \leq C(T_\alpha)$ substituir T_α por T_p .

Caso contrário,

atualizar o limite inferior do intervalo $\ell_k := \lambda_k$;

atualizar o LI, isto é, se $C(T_p) \geq C(T_\beta)$ substituir T_β por T_p .

No artigo de Yamada et al. [66] é apresentado um algoritmo semelhante a este, mas para o Problema da Árvore de Suporte de Custo Máximo com Restrições de Peso,

onde é alterada a inicialização do limite superior do intervalo u_0 e também o critério de paragem. Ao adaptar esse algoritmo para o Problema WMST ficamos com

- a inicialização do limite superior do intervalo $u_0 = -\frac{d\vartheta(WMST_\lambda)}{d\lambda}$;
- o critério de paragem utilizado é $|u_k - \ell_k| < tol$ em vez de $|P(T_\alpha) - P(T_p)| \leq tol$.

Exemplo 4.4.

Na seguinte tabela resumamos os resultados da aplicação do Algoritmo Alg4 para o Exemplo 2.1.

$C(T_\beta)$	$W(T_\beta)$	$C(T_\alpha)$	$W(T_\alpha)$	ℓ_k	u_k	λ_k	$C(T_p)$	$W(T_p)$	$P(T_\alpha)$	$P(T_p)$	$\vartheta(\lambda_k)$
17	27	70	11	0	5,889	2,944	41	13	102,389	79,278	20,389
17	27	41	13	0	2,944	1,472	19	24	60,139	54,333	24,889
19	24	41	13	1,472	2,944	2,208	33	16	69,708	68,333	24,167
19	24	33	16	1,472	2,208	1,84	27	19	62,444	61,965	25,160
19	24	27	19	1,472	1,84	1,655	27	19	58,469	58,469	25,344

Tabela 4.4: Resultados da aplicação do Algoritmo Alg4 para o exemplo de 5 nodos.

A primeira coluna dá-nos os sucessivos valores dos limites inferiores obtidos para o valor do custo e a terceira coluna dá-nos os sucessivos valores dos limites superiores obtidos para o valor do custo. Os custos e os pesos das árvores de suporte que são, sucessivamente, atualizadas encontram-se a negrito na tabela. Para este exemplo, o Algoritmo Alg4 usando a inicialização $\ell_0 = 0$ e $u_0 = \frac{C(T_w) - C(T_c)}{W - W(T_w)} \approx 5,8889$ usa cinco iterações do Passo 2 para obter a solução ótima com custo 27 e peso 19. A árvore T_α é atualizada três vezes, enquanto que, a árvore T_β é atualizada uma só vez. O valor obtido pela relaxação Lagrangeana está próximo do valor da relaxação linear ($25,344 < 25,4$).

Se usarmos a inicialização proposta em [66] e adaptada para o Problema WMST, isto é, $u_0 = -\frac{d\vartheta(WMST_\lambda)}{d\lambda} = 9$ e o critério de paragem $|u_k - \ell_k| < 0,001$, são necessárias catorze iterações do Passo 2 para obter a solução ótima com custo 27 e peso 19. A árvore T_α é atualizada nove vezes, enquanto que, a árvore T_β é atualizada cinco vezes. O valor obtido pela relaxação Lagrangeana está muito próximo do valor da relaxação linear ($25,3998 < 25,4$).

Algoritmo Lagrangeano 5 (Alg5)

O algoritmo que apresentamos de seguida é baseado no artigo de Amado e Barcia [7]. Este algoritmo inicializa o intervalo $[\ell_0, u_0]$ fazendo $\ell_0 = 0$ e $u_0 = U$, onde $U := \max_{(i,j) \in A} \{c_{ij}, w_{ij}\}$, ou seja, o máximo valor entre os custos e os pesos associados a todos os arcos $(i, j) \in A$ do grafo G .

Iterativamente o intervalo $[\ell_k, u_k]$ é reduzido até que $u_k - \ell_k \leq \frac{1}{U^2}$ e, consequentemente, é identificada a solução ótima no intervalo reduzido. Fazendo $a = \lambda_k = \frac{\ell_k + u_k}{2}$ e $b = 1$ o intervalo reduzido é obtido por comparação dos valores de $\vartheta(WMST_{\ell_k})$, $\vartheta(WMST_{\lambda_k})$ e $\vartheta(WMST_{u_k})$. Para simplificar a notação iremos usar $\vartheta(\ell_k)$, $\vartheta(\lambda_k)$ e $\vartheta(u_k)$ em vez de $\vartheta(WMST_{\ell_k})$, $\vartheta(WMST_{\lambda_k})$ e $\vartheta(WMST_{u_k})$, respetivamente.

Inicialmente $\vartheta(\ell_0) = C(T_{\ell_0}) = C(T_c)$, que corresponde ao custo da árvore de suporte de custo mínimo obtido no Passo 1.1 e o valor de $\vartheta(u_0)$ é obtido usando a árvore de suporte T_{u_0} de valor ponderado mínimo $P(T_{u_0})$ (Passo 1.2). Comparando com o Algoritmo ALagB, o Passo 1.3 pode ser omitido e os Passos 3 e 4 irão sofrer modificações que passamos a descrever de seguida.

Passo 3: *Atualização de limites*

Se $\vartheta(\lambda_k) \geq \max \{\vartheta(\ell_k), \vartheta(u_k)\}$, então

calcular $\lambda_k^a := \frac{\ell_k + \lambda_k}{2}$ e $\lambda_k^b := \frac{\lambda_k + u_k}{2}$ e obter $\vartheta(\lambda_k^a)$ e $\vartheta(\lambda_k^b)$.

Se $\vartheta(\lambda_k^a) \geq \vartheta(\lambda_k)$, então

atualizar o limite superior do intervalo $u_k := \lambda_k$ e $\lambda_k := \lambda_k^a$;

atualizar o LS, isto é, se $C(T_{\lambda_k}) \leq C(T_{u_k})$ substituir T_{u_k} por T_{λ_k} .

Caso contrário,

se $\vartheta(\lambda_k^b) \geq \vartheta(\lambda_k)$, então

atualizar o limite inferior do intervalo $\ell_k := \lambda_k$ e $\lambda_k := \lambda_k^b$;

atualizar o LI, isto é, se $C(T_{\ell_k}) \leq C(T_{\lambda_k})$ substituir T_{ℓ_k} por T_{λ_k} .

Caso contrário,

atualizar ambos os limites do intervalo $\ell_k := \lambda_k^a$ e $u_k := \lambda_k^b$;

atualizar o LS, isto é, se $C(T_{\lambda_k}) \leq C(T_{u_k})$ substituir T_{u_k} por T_{λ_k} ;

atualizar o LI, isto é, se $C(T_{\ell_k}) \leq C(T_{\lambda_k})$ substituir T_{ℓ_k} por T_{λ_k} .

Caso contrário,

se $\vartheta(\ell_k) \leq \vartheta(\lambda_k) \leq \vartheta(u_k)$, então

atualizar o limite inferior do intervalo $\ell_k := \lambda_k$ e $\lambda_k := \frac{\lambda_k + u_k}{2}$;

atualizar o LI, isto é, se $C(T_{\ell_k}) \leq C(T_{\lambda_k})$ substituir T_{ℓ_k} por T_{λ_k} .

Caso contrário,

atualizar o limite superior do intervalo $u_k := \lambda_k$ e $\lambda_k := \frac{\ell_k + \lambda_k}{2}$;

atualizar o LS, isto é, se $C(T_{\lambda_k}) \leq C(T_{u_k})$ substituir T_{u_k} por T_{λ_k} .

Passo 4: Critério de paragem

Se $(u_k - \ell_k) \leq \frac{1}{U^2}$, então

Se $(W(T_{u_k}) - W)(W(T_{\ell_k}) - W) > 0$, então

seja T a árvore T_{u_k} ou T_{ℓ_k} para a qual o $\max\{\vartheta(\ell_k), \vartheta(u_k)\}$ ocorre.

Caso contrário,

seja T_{λ_k} , com $\lambda_k = \frac{C(T_{u_k}) - C(T_{\ell_k})}{W(T_{\ell_k}) - W(T_{u_k})}$ de valor mínimo $P(T_{\lambda_k})$.

Se $W(T_{\lambda_k}) \leq W$, então

$$T = T_{\lambda_k}.$$

Caso contrário,

$$T = T_{u_k}.$$

A solução aproximada corresponde à árvore T , STOP.

Caso contrário,

ir para o **Passo 2**.

No Passo 4, quando o critério de paragem for satisfeito podemos identificar duas retas, uma correspondente à árvore T_{ℓ_k} e outra à árvore T_{u_k} , com declives $W(T_{\ell_k}) - W$ e $W(T_{u_k}) - W$, respetivamente. Note-se que os custos das árvores T_{ℓ_k} e T_{u_k} correspondem ao limite inferior e superior para o valor ótimo do problema,

$$C(T_{\ell_k}) \leq \vartheta(WMST) \leq C(T_{u_k}).$$

Se os declives de ambas as retas têm o mesmo sinal, então o limite inferior para a solução ótima corresponde à árvore cujo valor da relaxação Lagrangeana é maior,

$$\vartheta(\lambda^*) = \max\{\vartheta(\ell_k), \vartheta(u_k)\}.$$

Caso os declives tenham sinais contrários, $(\lambda^*, \vartheta(\lambda^*))$ corresponde ao ponto de interseção destas duas retas. Para o obtermos calculamos um novo λ_k , dado por:

$$\lambda_k = \frac{C(T_{u_k}) - C(T_{\ell_k})}{W(T_{\ell_k}) - W(T_{u_k})}$$

que corresponde à abcissa do ponto de interseção das duas retas. De seguida com este ponto de interseção das duas retas obtém-se uma nova árvore de suporte T_{λ_k} . Se esta verificar a restrição de peso, então a solução aproximada corresponde à árvore T_{λ_k} , caso contrário, a solução aproximada corresponde à árvore T_{u_k} , uma vez que, neste caso a árvore T_{u_k} verifica a restrição de peso.

Proposição 4.1.

O valor do multiplicador de Lagrange λ_k é limitado

$$0 \leq \lambda_k \leq \frac{C(T_w) - C(T_c)}{W - W(T_w)}.$$

Demonstração.

Qualquer árvore de suporte admissível T verifica

$$C(T_c) \leq C(T) \leq C(T_w) \text{ e } W(T_w) \leq W(T) \leq W \leq W(T_c).$$

A reta definida pelos pontos de coordenadas $(W(T_w), C(T_w))$ e $(W(T_c), C(T_c))$ tem declive $m = \frac{C(T_c) - C(T_w)}{W(T_c) - W(T_w)} \leq 0$ e equação $y = C(T_w) + m(x - W(T_w))$.

Se o declive é negativo, temos que $\lambda_k \geq 0$, dado que $\lambda_k = \frac{C(T_w) - C(T_c)}{W(T_c) - W(T_w)}$.

Para a árvore ser admissível o seu peso tem de ser inferior ou igual a W , isto é, $x \leq W$, assim $y = C(T_w) + m(x - W(T_w)) \geq C(T_w) + m(W - W(T_w))$. Além disso, qualquer árvore admissível T deve ser tal que $C(T) \geq C(T_w) + m(W - W(T_w))$. Como a árvore T_c é não admissível, temos $C(T_c) \leq C(T_w) + m(W - W(T_w))$. Portanto, $-m \leq \frac{C(T_w) - C(T_c)}{W - W(T_w)}$, o que significa que $\lambda_k \leq \frac{C(T_w) - C(T_c)}{W - W(T_w)}$.

Logo, $0 \leq \lambda_k \leq \frac{C(T_w) - C(T_c)}{W - W(T_w)}$.

□

Exemplo 4.5.

Na seguinte tabela resumamos os resultados da aplicação do Algoritmo Alg5 para o Exemplo 2.1.

ℓ_k	u_k	λ_k	λ_k^a	λ_k^b	$\vartheta(\ell_k)$	$\vartheta(u_k)$	$\vartheta(\lambda_k)$	$\vartheta(\lambda_k^a)$	$\vartheta(\lambda_k^b)$
0	5,889	2,944	1,472	—	17	−0,222	20,389	24,889	—
0	2,944	1,472	0,736	2,208	17	20,389	24,889	21,944	24,167
0,736	2,208	1,472	1,104	1,840	21,944	24,167	24,889	23,417	25,160
1,472	2,208	1,840	1,656	1,840	24,889	24,167	25,160	25,344	25,160
1,472	1,840	1,656	1,564	1,748	24,889	25,160	25,344	25,257	25,252
1,564	1,748	1,656	1,610	1,748	25,257	25,252	25,344	25,390	25,252
1,564	1,656	1,610	1,587	1,633	25,257	25,344	25,390	25,349	25,367
1,587	1,633	1,610	1,599	1,633	25,349	25,367	25,390	25,395	25,367
1,587	1,610	1,599	1,593	1,605	25,349	25,390	25,395	25,372	25,396
1,599	1,611	1,605	1,602	1,605	25,395	25,390	25,396	25,398	25,396
1,599	1,605	1,602	1,600	1,605	25,395	25,396	25,398	25,400	25,400
1,599	1,602	1,600	1,600	1,601	25,395	25,398	25,400	25,400	25,400
1,600	1,601	1,600	1,600	1,601	25,398	25,399	25,400	25,400	25,400
1,600	1,601								

Tabela 4.5: Resultados da aplicação do Algoritmo Alg5 para obtenção do intervalo $[\ell_k, u_k]$.

Para este exemplo de 5 nodos, o Algoritmo Alg5 usando a inicialização $\ell_0 = 0$ e $u_0 = U = \max_{(i,j) \in A} \{c_{ij}, w_{ij}\} = 30$ precisa de quinze iterações e é necessário determinar 38 árvores de suporte para obter a solução ótima no intervalo $[1,6; 1,601]$.

Tendo em conta a Proposição 4.1 verificamos que podemos utilizar para este exemplo uma nova inicialização do intervalo $[\ell_0, u_0]$, fazendo $\ell_0 = 0$ e $u_0 = \frac{C(T_w) - C(T_c)}{W - W(T_w)} \approx 5,8889$. Usando estes valores, são precisas treze iterações e é necessário determinar 36 árvores de suporte para a obtenção do intervalo $[1,5998; 1,6005]$ o qual, contém $\lambda^* = 1,6$.

Os limites inferior e superior para o valor ótimo são, respetivamente, $C(T_{\ell_k}) = 19$ e $C(T_{u_k}) = 27$.

O valor obtido através da relaxação Lagrangeana é igual ao valor da relaxação linear 25,4 usando qualquer uma das inicializações propostas.

Algoritmo Lagrangeano 6 (Alg6)

Devido às propriedades da convexidade da função Lagrangeana, os limites do intervalo $[\ell_k, u_k]$ podem ser atualizados tendo em conta os declives das retas. Assim, podemos simplificar o Algoritmo Alg5.

Se os declives das retas correspondentes a λ_k e ℓ_k têm o mesmo sinal, então é atualizado o limite inferior do intervalo ℓ_k . Caso contrário, se os declives das retas correspondentes a λ_k e u_k têm o mesmo sinal, então é atualizado o limite superior do intervalo u_k . Caso um dos declives seja nulo, então temos de atualizar ambos os limites inferior e superior do intervalo. De seguida apresenta-se o novo Passo 3 simplificado.

Passo 3: *Atualização de limites*

Se $(W(T_{\lambda_k}) - W)(W(T_{\ell_k}) - W) > 0$, então

atualizar o limite inferior do intervalo, $\ell_k := \lambda_k$ e $\lambda_k := \frac{\lambda_k + u_k}{2}$;

atualizar o LI, isto é, se $C(T_{\lambda_k}) \geq C(T_{\ell_k})$ substitui-se T_{ℓ_k} por T_{λ_k} .

Caso contrário,

se $(W(T_{\lambda_k}) - W)(W(T_{u_k}) - W) > 0$, então

atualizar o limite superior do intervalo, $u_k := \lambda_k$ e $\lambda_k := \frac{\ell_k + \lambda_k}{2}$;

atualizar o LS, isto é, se $C(T_{\lambda_k}) \leq C(T_{u_k})$ substitui-se T_{u_k} por T_{λ_k} .

Caso contrário,

atualizar ambos os limites do intervalo $\ell_k := \frac{\ell_k + \lambda_k}{2}$ e $u_k := \frac{\lambda_k + u_k}{2}$;

atualizar o LS, isto é, se $C(T_{\lambda_k}) \leq C(T_{u_k})$ substituir T_{u_k} por T_{λ_k} ;

atualizar o LI, isto é, se $C(T_{\ell_k}) \leq C(T_{\lambda_k})$ substituir T_{ℓ_k} por T_{λ_k} .

O critério de paragem deste algoritmo será efetuado de acordo com o Passo 4 do Algoritmo Alg5.

Exemplo 4.6.

Na seguinte tabela resumamos os resultados da aplicação do Algoritmo Alg6 para o Exemplo 2.1.

ℓ_k	u_k	λ_k	$W(T_{\ell_k}) - W$	$W(T_{u_k}) - W$	$W(T_{\lambda_k}) - W$	$\vartheta(\ell_k)$	$\vartheta(u_k)$	$\vartheta(\lambda_k)$
0	5,8889	2,9444	7	-7	-7	17	-0,2222	20,3889
0	2,9444	1,4722	7	-7	4	17	20,3889	24,8889
1,4722	2,9444	2,2083	4	-7	-4	24,8889	20,3889	24,1667
1,4722	2,2083	1,8403	4	-4	-1	24,8889	24,1667	25,1597
1,4722	1,8403	1,6563	4	-1	-1	24,8889	25,1597	25,3438
1,4722	1,6563	1,5642	4	-1	4	24,8889	25,3438	25,2569
1,5642	1,6563	1,6102	4	-1	-1	25,2569	25,3438	25,3898
1,5642	1,6102	1,5872	4	-1	4	25,2569	25,3898	25,3490
1,5872	1,6102	1,5987	4	-1	4	25,3490	25,3898	25,3950
1,5987	1,6102	1,6045	4	-1	-1	25,3950	25,3898	25,3955
1,5987	1,6045	1,6016	4	-1	-1	25,3950	25,3955	25,3984
1,5987	1,6016	1,6002	4	-1	-1	25,3950	25,3984	25,3998
1,5987	1,6002	1,5995	4	-1	4	25,3950	25,3998	25,3978
1,5995	1,6002							

Tabela 4.6: Resultados da aplicação do Algoritmo Alg6 para obtenção do intervalo $[\ell_k, u_k]$.

Na tabela encontram-se a negrito os declives com o mesmo sinal. Para este exemplo, no Algoritmo Alg6 usando a inicialização $\ell_0 = 0$ e $u_0 = U = \max_{(i,j) \in A} \{c_{ij}, w_{ij}\} = 30$, são necessárias quinze iterações e é necessário encontrar 18 árvores de suporte para obtenção da solução ótima.

Tendo em conta a Proposição 4.1 verificamos que podemos utilizar para este exemplo uma nova inicialização do intervalo $[\ell_0, u_0]$, fazendo $\ell_0 = 0$ e $u_0 = \frac{C(T_w) - C(T_c)}{W - W(T_w)} \approx 5,8889$. Usando estes valores, são necessárias treze iterações e obtém-se 16 árvores de suporte para obtenção do intervalo $[1,5995; 1,6002]$ (Tabela 4.6) que contém o valor $\lambda^* = 1,6$. Os limites inferiores e superiores para o valor ótimo são, respetivamente, $C(T_{\ell_k}) = 19$ e $C(T_{u_k}) = 27$. O valor obtido pela relaxação Lagrangeana é de 25,4 usando qualquer uma das inicializações propostas.

Algoritmo Lagrangeano 7 (Alg7)

Este algoritmo tem por objetivo obter melhores limites inferiores. Portanto, em vez de considerar apenas a restrição de peso,

$$\sum_{(i,j) \in A} w_{ij}x_{ij} \leq W,$$

vamos introduzir a seguinte desigualdade válida,

$$\sum_{(i,j) \in S} x_{ij} \leq |S| - 1, \quad (4.1)$$

designada de Desigualdade de Cobertura [4]. Observe-se que poderá ser utilizada qualquer outra desigualdade válida. Adicionando a desigualdade (4.1) ao modelo genérico, obtemos a seguinte formulação:

$$\begin{aligned} (\text{WMST-C}) : \quad & \min \sum_{(i,j) \in A} c_{ij}x_{ij} \\ & s.a. \quad x \in X_T \\ & \sum_{(i,j) \in A} w_{ij}x_{ij} \leq W \\ & \sum_{(i,j) \in S} x_{ij} \leq |S| - 1. \end{aligned} \quad (4.2)$$

$$(4.3)$$

Onde $x = (x_{ij}) \in \mathbb{R}^{|A|}$ e X_T descrevem o invólucro convexo das soluções inteiras do Problema MST. Associando à restrição de peso (4.2) um multiplicador de Lagrange λ e associando à desigualdade válida (4.3) um outro multiplicador de Lagrange ν , e incluindo na função objetivo uma penalização destas duas restrições, podemos definir o seguinte problema relaxado:

$$\begin{aligned} (\text{WMST-C}_{\lambda\nu}) : \quad & -\lambda W - \nu(|S| - 1) + \min \sum_{(i,j) \in A} (c_{ij} + \lambda w_{ij})x_{ij} + \min \sum_{(i,j) \in S} \nu x_{ij} \\ & s.a. \quad x \in X_T. \end{aligned}$$

Para todos os multiplicadores não negativos λ e ν as soluções para este problema relaxado dão-nos limites inferiores para o valor ótimo. A proposição que se segue permite concluir que os limites inferiores obtidos através desta relaxação Lagrangeana são melhores do que os obtidos com a relaxação Lagrangeana descrita anteriormente.

Proposição 4.2.

Seja $\lambda \geq 0$ e $\nu \geq 0$, então

$$\vartheta(WMST_\lambda) \leq \vartheta(\text{WMST-C}_{\lambda\nu}) \leq \vartheta(WMST).$$

Demonstração.

Para demonstrar a primeira desigualdade consideramos a função dual Lagrangeana,

$$\begin{aligned} \vartheta(\text{WMST-C}_{\lambda\nu}) &= -\lambda W - \nu(|S| - 1) + \min_{x \in X_T} \left(\sum_{(i,j) \in A} (c_{ij} + \lambda w_{ij}) x_{ij} + \sum_{(i,j) \in S} \nu x_{ij} \right) \\ &= -\lambda W - \nu|S| + \nu + P(T_\lambda) + \nu \min_{x \in X_T} \sum_{(i,j) \in S} x_{ij} \end{aligned}$$

Como $\min_{x \in X_T} \sum_{(i,j) \in S} x_{ij} \geq |S| - 1$, então obtemos,

$$\begin{aligned} \vartheta(\text{WMST-C}_{\lambda\nu}) &\geq -\lambda W - \nu|S| + \nu + P(T_\lambda) + \nu(|S| - 1) \\ &= -\lambda W + P(T_\lambda) \\ &= \vartheta(WMST_\lambda). \end{aligned}$$

Logo, $\vartheta(WMST_\lambda) \leq \vartheta(\text{WMST-C}_{\lambda\nu})$.

Para provar a segunda desigualdade começamos por notar que a desigualdade considerada é uma desigualdade válida para o Problema WMST e, portanto, a Formulação WMST-C é uma formulação para o problema. Observamos agora que, sendo $\text{WMST-C}_{\lambda\nu}$ uma relaxação linear, pela Teoria da Dualidade Lagrangeana, obtemos um limite inferior para o valor ótimo do problema.

□

O objetivo agora é o de maximizar a função dual Lagrangeana $\vartheta(\text{WMST-C}_{\lambda\nu})$, ou seja, encontrar os valores de λ e ν que a maximizam.

Para obter o valor do parâmetro λ , vamos começar por assumir que ν é uma constante e vamos reduzir o intervalo $[\ell_k, u_k]$ até que $u_k - \ell_k \leq \frac{1}{U^2}$. Para tal, inicialmente considera-se um valor pequeno para ν , que designamos por $\nu_i = \frac{1}{U}$, com $U := \max_{(i,j) \in A} \{c_{ij}, w_{ij}\}$, e considera-se o intervalo $[\ell_0, u_0]$ inicializado com $\ell_0 = 0$ e $u_0 = \frac{C(T_w) - C(T_c)}{W - W(T_w)}$. Depois desta inicialização, vamos aplicar o Passo 3 do Algoritmo

Alg6 para reduzir o intervalo $[\ell_k, u_k]$. Durante o processo iterativo de redução do intervalo $[\ell_k, u_k]$, obtém-se uma desigualdade do tipo (4.3) quando obtemos uma árvore de suporte não admissível T_{ℓ_k} . Para construir a desigualdade (4.3) é necessário optar pela escolha de uma ordem de inclusão do conjunto de arestas $E_{T_{\ell_k}}$ na desigualdade. Para escolher a ordenação utilizamos os seguintes procedimentos:

Procedimento 1

Ordenar as arestas do conjunto $E_{T_{\ell_k}}$ por ordem crescente de custo.

Procedimento 2

Ordenar as arestas do conjunto $E_{T_{\ell_k}}$ por ordem decrescente de peso.

Procedimento 3

Ordenar as arestas do conjunto $E_{T_{\ell_k}}$ por ordem decrescente de valor $\frac{c_{ij}}{w_{ij}}$.

Utilizando os procedimentos anteriores forma-se um conjunto $S \subset E_{T_{\ell_k}}$ de arestas tal que a soma dos pesos das arestas contidas em S é superior a W , ou seja, verifica-se a desigualdade $\sum_{e \in S} w_e > W$, onde $e = \{i, j\}$. Logo, S é uma cobertura e também pode ser uma cobertura minimal, no caso de ao remover qualquer aresta da cobertura S , a soma dos pesos das restantes arestas contidas na cobertura for inferior ou igual a W , isto é, para toda a aresta $f = \{k, t\} \in S$ temos $\sum_{e \in S - \{f\}} w_e \leq W$.

Note-se que quando introduzimos uma desigualdade válida ao Problema WMST, associamos à desigualdade (4.3) um multiplicador de Lagrange ν e desta forma as arestas contidas em S vão ser penalizadas na função objetivo.

No Capítulo 5 são apresentadas desigualdades válidas para o Problema WMST.

Exemplo 4.7.

Vamos agora exemplificar como obtemos uma desigualdade válida. Considere-se que foi obtida a árvore de suporte não admissível com custo 17 e peso 27, e cujas arestas são as seguintes $\{0, 1\}; \{1, 2\}; \{0, 4\}; \{0, 3\}$.

Para efetuar o Procedimento 1, as arestas são inseridas na desigualdade por ordem crescente de custo, isto é, $\{0, 1\} \rightsquigarrow \{1, 2\} \rightsquigarrow \{0, 4\} \rightsquigarrow \{0, 3\}$.

Como,

$$w_{01} = 6 < 20$$

$$w_{01} + w_{12} = 6 + 6 = 12 < 20$$

$$w_{01} + w_{12} + w_{04} = 12 + 7 = 19 < 20$$

$$w_{01} + w_{12} + w_{04} + w_{03} = 19 + 8 = 27 > 20,$$

obtemos a desigualdade válida $x_{01} + x_{03} + x_{04} + x_{12} \leq 3$.

No caso de se efetuar o Procedimento 2, então ordenam-se as arestas por ordem decrescente de peso, ou seja, $\{0, 3\} \rightsquigarrow \{0, 4\} \rightsquigarrow \{0, 1\} \rightsquigarrow \{1, 2\}$.

Como,

$$w_{03} = 8 < 20$$

$$w_{03} + w_{04} = 8 + 7 = 15 < 20$$

$$w_{03} + w_{04} + w_{01} = 15 + 6 = 21 > 20,$$

obtemos a desigualdade válida $x_{01} + x_{03} + x_{04} \leq 2$.

Esta desigualdade também é obtida com a aplicação do Procedimento 3.

Observe-se que a desigualdade obtida no Procedimento 2 e 3 é mais forte do que a obtida no Procedimento 1.

Já vimos como reduzir o intervalo do parâmetro λ e como obter uma desigualdade válida. Decorre ainda a necessidade de determinar o valor do multiplicador de Lagrange ν que faz com que $\vartheta(\text{WMST-C}_{\lambda\nu})$ seja máximo.

Proposição 4.3.

Dado um intervalo $[\ell_k, u_k]$, o multiplicador de Lagrange ν é dado por:

$$\nu = \frac{(C(T_{u_k}) - C(T_{\ell_k})) (W - W(T_{u_k}))}{S_{\ell_k}(W - W(T_{u_k})) - W(T_{u_k}) + W(T_{\ell_k})}. \quad (4.4)$$

Demonstração.

Ao longo dos algoritmos descritos anteriormente estamos sucessivamente a obter duas árvores de suporte: uma não admissível, T_{ℓ_k} , e a outra admissível, T_{u_k} . Pretendemos, que as relaxações destas duas árvores se aproximem, isto é, queremos saber quando é que os dois planos correspondentes se intersejam, mais precisamente quando

$$C(T_{\ell_k}) + \ell_k W(T_{\ell_k}) + \nu S_{\ell_k} = C(T_{u_k}) + u_k W(T_{u_k}) + \nu S_{u_k}, \quad (4.5)$$

onde S_{ℓ_k} e S_{u_k} designam o número de arestas contidas em S na árvore T_{ℓ_k} e T_{u_k} , respetivamente.

Quando a árvore obtida é admissível, não é possível obter-se uma desigualdade válida, neste caso $S_{u_k} = 0$. Substituindo $S_{u_k} = 0$ na equação (4.5) e tendo em conta que os valores de ℓ_k e u_k se aproximam de λ_k , obtemos

$$C(T_{\ell_k}) - C(T_{u_k}) + \nu S_{\ell_k} = \lambda_k (W(T_{u_k}) - W(T_{\ell_k})).$$

Resolvendo a equação anterior em ordem a ν obtém-se,

$$\nu = \frac{C(T_{u_k}) - C(T_{\ell_k})}{S_{\ell_k}} + \lambda_k \frac{W(T_{u_k}) - W(T_{\ell_k})}{S_{\ell_k}}. \quad (4.6)$$

Como $\frac{W(T_{u_k}) - W(T_{\ell_k})}{S_{\ell_k}} < 0$, concluímos que estas retas têm declive negativo.

Necessitamos agora de encontrar retas cujo declive seja positivo para podermos fazer a respetiva interseção com as retas definidas pela equação (4.6).

Sabemos que em cada iteração o valor do limite superior pode ser atualizado $LS = C(T_{u_k})$, portanto

$$\vartheta(\lambda_k, \nu) \leq C(T_{u_k}) \Leftrightarrow C(T_{u_k}) + \lambda_k (W(T_{u_k}) - W) + \nu \leq C(T_{u_k})$$

Resolvendo a inequação anterior em ordem a ν obtemos,

$$\nu \leq \lambda_k (W - W(T_{u_k})) \quad (4.7)$$

onde $W - W(T_{u_k}) > 0$, ou seja, o declive das retas $\nu = \lambda_k (W - W(T_{u_k}))$ é positivo.

Em cada iteração vamos obter os sucessivos valores das interseções das retas (4.6) e (4.7). Resolvendo as equações (4.6) e (4.7) em ordem a λ_k , vem,

$$\lambda_k = \frac{S_{\ell_k} \nu - C(T_{u_k}) + C(T_{\ell_k})}{W(T_{u_k}) - W(T_{\ell_k})} \quad (4.8)$$

$$\lambda_k = \frac{\nu}{W - W(T_{u_k})}. \quad (4.9)$$

Igualando (4.8) e (4.9) vem,

$$\frac{S_{\ell_k} \nu - C(T_{u_k}) + C(T_{\ell_k})}{W(T_{u_k}) - W(T_{\ell_k})} = \frac{\nu}{W - W(T_{u_k})}.$$

Resolvendo a equação anterior em ordem a ν obtemos,

$$\nu = \frac{(C(T_{u_k}) - C(T_{\ell_k}))(W - W(T_{u_k}))}{S_{\ell_k}(W - W(T_{u_k})) - W(T_{u_k}) + W(T_{\ell_k})}. \quad (4.10)$$

□

É no Passo 4 que a seguir se descreve que se obtém o valor ν do multiplicador de Lagrange e se calcula o maior valor das relaxações das árvores $T_{\ell_k, \nu}$ e $T_{u_k, \nu}$.

Passo 4: Critério de paragem

Se $u_k - \ell_k \leq \frac{1}{U^2}$, então

$$\text{Obter } \nu = \frac{(C(T_{u_k}) - C(T_{\ell_k}))(W - W(T_{u_k}))}{S_{\ell_k}(W - W(T_{u_k})) - W(T_{u_k}) + W(T_{\ell_k})}.$$

Seja T a árvore $T_{u_k, \nu}$ ou $T_{\ell_k, \nu}$ para o qual $\max\{\vartheta(\ell_k, \nu), \vartheta(u_k, \nu)\}$ ocorre.

A solução aproximada corresponde à árvore T , STOP.

Caso contrário,

voltar ao **Passo 2**.

Exemplo 4.8.

Para exemplificar o Algoritmo Alg7 utilizamos o Exemplo 2.1. Vamos aplicar o Procedimento 2 para obtenção da desigualdade de cobertura.

Inicialização de valores:

$$\ell_0 = 0, \quad u_0 = \frac{70 - 17}{20 - 11} = \frac{53}{9} \simeq 5,89, \quad U = \max\{30, 8\} = 30 \quad \text{e} \quad \nu_i = \frac{1}{30}.$$

Iteração 1: Inicializar T_{ℓ_k} e T_{u_k} .

$$T_{\ell_k} : C(T_{\ell_k}) = 17, W(T_{\ell_k}) = 27 \text{ e } S_{\ell_k} = 3.$$

$$T_{u_k} : C(T_{u_k}) = 41 \text{ e } W(T_{u_k}) = 13.$$

$$\text{Desigualdade válida: } x_{03} + x_{04} + x_{01} \leq 2.$$

$$\text{Intervalo obtido: } [0; 2,9(4)].$$

Iteração 2: Atualizar T_{ℓ_k} .

$$T_{\ell_k} : C(T_{\ell_k}) = 19, W(T_{\ell_k}) = 24 \text{ e } S_{\ell_k} = 4.$$

$$T_{u_k} : C(T_{u_k}) = 41 \text{ e } W(T_{u_k}) = 13.$$

$$\text{Desigualdade valida: } x_{04} + x_{01} + x_{12} + x_{43} \leq 3.$$

$$\text{Intervalo obtido: } [1,47(2); 2,9(4)].$$

Iterao 3: Atualizar T_{u_k} .

$$T_{\ell_k} : C(T_{\ell_k}) = 19, W(T_{\ell_k}) = 24 \text{ e } S_{\ell_k} = 4.$$

$$T_{u_k} : C(T_{u_k}) = 33 \text{ e } W(T_{u_k}) = 16.$$

$$\text{Intervalo obtido: } [1,47(2); 2,208(3)].$$

Iterao 4: Atualizar T_{u_k} .

$$T_{\ell_k} : C(T_{\ell_k}) = 19, W(T_{\ell_k}) = 24 \text{ e } S_{\ell_k} = 4.$$

$$T_{u_k} : C(T_{u_k}) = 27 \text{ e } W(T_{u_k}) = 19.$$

$$\text{Intervalo obtido: } [1,47(2); 1,840278].$$

Nas restantes treze iteraes obtem-se os mesmos resultados da Iterao 4 com exceo do intervalo reduzido que  sucessivamente atualizado. Quando se atualiza ℓ_k  adicionada a desigualdade valida $x_{01} + x_{04} + x_{12} + x_{43} \leq 3$. O intervalo reduzido obtido  $[1,594428; 1,595147]$.

$$\text{No Passo 4 obtem-se } \nu = \frac{(27 - 19)(20 - 19)}{4 \times (20 - 19) - 19 + 24} = \frac{8}{9} \simeq 0,89.$$

$$\text{Assim, } \vartheta(\text{WMST-C}_{\lambda\nu}) = \max\{26,266602; 26,293742\} = 26,293742.$$

O valor da relaxao linear neste exemplo de 5 nodos  de 25,4 e o valor timo  de 27, logo o *gap*  $\frac{27 - 25,4}{27} \times 100 = 5,93\%$. Aplicando o Algoritmo Alg7 com qualquer um dos trs procedimentos o valor da relaxao Lagrangeana  de 26,293742. O *gap* passa a ser $\frac{27 - 26,293742}{27} \times 100 = 2,62\%$ e obtemos assim uma reduo de 3,31%.

4.5 Experiências para Comparação dos Algoritmos Lagrangeanos e Resultados Computacionais

Nesta secção descrevemos algumas experiências computacionais realizadas com o objetivo de definir parâmetros e estratégias a serem adotadas para a utilização dos algoritmos Lagrangeanos expostos. Apresentamos igualmente os resultados computacionais obtidos quando utilizamos os algoritmos descritos ao longo deste capítulo para resolver o conjunto de instâncias geradas e descritas na Secção 2.3 e fazemos uma comparação entre os algoritmos em termos de tempo de execução e de qualidade das soluções aproximadas obtidas. Pretendemos também averiguar se o valor obtido pela relaxação Lagrangeana dos diversos algoritmos corresponde ao valor da relaxação linear do Procedimento P-WMTZ+C.

Os algoritmos Lagrangeanos foram implementados em C++ e todos os testes foram efetuados num Intel(R) Core(TM)2 Duo CPU (T7100) 2.00 GHz processador e 4Gb de RAM.

4.5.1 Exemplo Comparativo dos Algoritmos Lagrangeanos

Com o objetivo de comparar os diferentes algoritmos Lagrangeanos descritos ao longo deste capítulo elaborámos a Tabela 4.7, onde se resume a informação já dada para o exemplo de 5 nodos que foi utilizado ao longo deste capítulo para exemplificar o comportamento de cada um dos algoritmos Lagrangeanos. Para este exemplo com 5 nodos, a solução ótima tem custo 27 e peso 19 e o valor de $W = 20$.

	$C(T_\alpha)$	$W(T_\alpha)$	Iter.	λ_k	ℓ_k	u_k	$\vartheta(\lambda_k)$	ν	$\vartheta(\lambda_k, \nu)$	Gap
Alg1	27	19	4	1,5970	–	–	25,400	–	–	5,93
Alg2	27	19	4	1,6000	–	–	25,400	–	–	5,93
Alg3	27	19	7	1,6450	–	–	25,355	–	–	6,09
Alg4	27	19	5	1,6550	1,4720	1,8400	25,344	–	–	6,13
Alg5	27	19	13	1,6000	1,6000	1,6010	25,400	–	–	5,93
Alg6	27	19	13	1,6000	1,5995	1,6002	25,400	–	–	5,93
Alg7	27	19	13	1,5948	1,5944	1,5951	–	0,89	26,294	2,62

Tabela 4.7: Comparação dos algoritmos Lagrangeanos no Exemplo 2.1.

Para o exemplo de 5 nodos considerado podemos observar que todos os algoritmos Lagrangeanos obtiveram a solução aproximada correspondente à solução ótima, isto é, foi obtida a árvore de suporte com custo 27 e peso 19. Os Algoritmos Alg3 e Alg4 são os que apresentam um valor de *gap* mais elevado. Nos Algoritmos Alg1, Alg2, Alg5 e Alg6 o valor obtido pela relaxação Lagrangeana é igual ao valor obtido pela relaxação linear. Como já tinha sido observado, o Algoritmo Alg7 melhora o valor da relaxação Lagrangeana havendo uma redução no *gap* de 3,31%.

4.5.2 Descrição dos Valores dos Parâmetros Usados nas Experiências Computacionais Realizadas

Um parâmetro comum a todos os algoritmos Lagrangeanos apresentados é a tolerância *tol* usada no critério de paragem. Após efetuarmos algumas experiências computacionais com vários valores de *tol* no intervalo $[0,0001; 0,1]$ decidimos utilizar $tol = 0,001$.

Para a execução do Algoritmo Alg3 é necessário definir o parâmetro ρ e para tal, foram efetuadas diversas experiências com vários valores de ρ no intervalo $]0, 2[$, nos três grupos de instâncias em estudo.

Nas instâncias Quase Caminhos, ao diminuirmos o valor de ρ , a qualidade das soluções obtidas melhora, isto é, os valores obtidos para o limite superior e para o valor da relaxação Lagrangeana aproximam-se cada vez mais dos correspondentes valores obtidos através dos Algoritmos Alg2, Alg5 e Alg6.

No gráfico da Figura 4.6 podemos observar que nas instâncias Quase Caminhos quanto mais pequeno é o valor de ρ maior é o tempo de execução do algoritmo.

Com o objetivo de encontrar soluções de melhor qualidade e de forma rápida decidimos utilizar para as instâncias Quase Caminhos o valor $\rho = 0,001$.

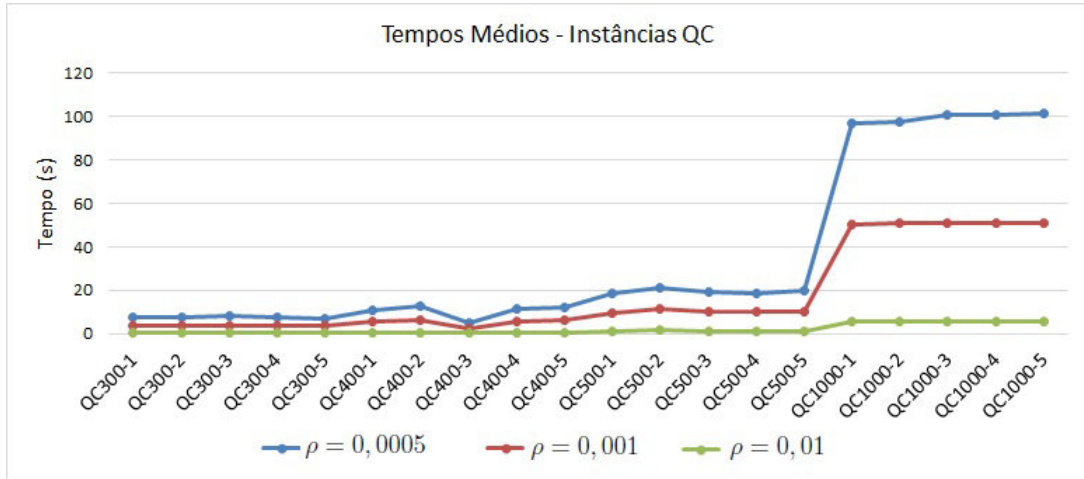


Figura 4.6: Comparação dos tempos de execução em instâncias QC para diferentes valores de ρ .

Nas instâncias Aleatórias e Euclidianas, verificamos que ao diminuirmos o valor de ρ para as instâncias de maiores dimensões, as soluções são de pior qualidade.

Para que o Algoritmo Alg3 seja convergente nos grupos de instâncias Aleatórias e Euclidianas é necessário adotar diferentes valores de ρ para as diferentes dimensões das instâncias. Após diversas experiências decidimos utilizar os valores de ρ que se encontram nas Tabelas 4.8 e 4.9.

N.º Nodos	10	20	40	60	80	100
R	0,001	0,0065	0,075	0,25	0,095	0,125
E	0,0001	0,0001	0,0001	0,0001	0,0001	0,04

Tabela 4.8: Valores de ρ utilizados no Algoritmo Alg3 nas instâncias Aleatórias e Euclidianas de 10 a 100 nodos.

N.º Nodos	150	200	300	400	500	1000
R	0,085	0,06	0,045	0,04	0,04	0,03
E	0,04	0,04	0,04	0,04	0,035	0,015

Tabela 4.9: Valores de ρ utilizados no Algoritmo Alg3 nas instâncias Aleatórias e Euclidianas de 150 a 1000 nodos.

No Algoritmo Alg5 foram referidas, na Subsecção 4.4, as seguintes duas inicializações para o extremo superior do intervalo $[\ell_0, u_0]$:

- inicialização proposta por [7] (Alg5IB): $u_0 = U = \max_{(i,j) \in A} \{c_{ij}, w_{ij}\}$;
- nova inicialização proposta na Subsecção 4.4 (Alg5IN): $u_0 = \frac{C(T_w) - C(T_c)}{W - W(T_w)}$.

Das experiências realizadas com ambas as inicializações e a título de exemplo, construímos a Tabela 4.10 indicando o número médio de iterações, o número médio de árvores de suporte obtidas e o tempo de execução (em segundos) usando cada uma das inicializações nas diferentes dimensões das instâncias Quase Caminhos (QC).

	Alg5IB			Alg5IN		
N.º Nodos	N.º. Iterações	N.º. Árvores	Tempo (s)	N.º. Iterações	N.º. Árvores	Tempo (s)
10	27,1	66,6	0,004	18,7	57,7	0,002
20	26	64,3	0,008	19	56,4	0,002
40	25	61,9	0,007	19,2	57,1	0,006
60	24,8	63,2	0,018	19,3	57,2	0,018
80	24	60,9	0,023	19,1	55,9	0,024
100	24	61,9	0,037	19,8	58	0,028
150	25	63,3	0,064	20,5	58,3	0,066
200	23	59,2	0,102	20,4	57	0,091
300	23	57,8	0,172	20	54,6	0,165
400	23	57,2	0,279	18,2	50,6	0,260
500	23	59,4	0,449	20	55,6	0,429
1000	22,2	61,4	1,745	21,6	59,6	1,708

Tabela 4.10: Comparação das duas estratégias de inicialização do Algoritmo Alg5 em instâncias QC.

Podemos observar na Tabela 4.10 que com a inicialização proposta na Subsecção 4.4 é possível reduzir o número médio de iterações e o número médio de árvores determinadas, e consequentemente, o tempo de execução do algoritmo. Para os outros dois grupos de instâncias Aleatórias e Euclidianas obtêm-se as mesmas conclusões obtidas nas instâncias Quase Caminhos.

Relativamente ao Algoritmo Alg7 foram propostos três procedimentos de ordenação de arestas na Subsecção 4.4. Para todos os grupos de instâncias e usando todos os procedimentos de ordenação de arestas, o valor obtido através da relaxação Lagrangeana aumenta comparativamente ao valor obtido pela relaxação Lagrangeana dos outros algoritmos, sendo de realçar que os valores mais próximos da solução ótima são obtidos usando o Procedimento 2 em 85,26% das instâncias Quase Caminhos (81 em 95 instâncias). No grupo de instâncias Aleatórias, o Procedimento 2 é melhor em 98,33% das instâncias (59 em 60 instâncias) e no grupo de instâncias Euclideanas, é melhor em 81,67% das instâncias (49 em 60 instâncias). Os tempos de execução dos três procedimentos são muito semelhantes.

4.5.3 Estudo Comparativo dos Algoritmos Lagrangeanos e Resultados Computacionais

Nesta secção apresentamos e comparamos os resultados computacionais obtidos através dos algoritmos Lagrangeanos descritos na Secção 4.2 para as instâncias geradas e descritas na Secção 2.3.

Para avaliar a qualidade dos valores obtidos pelos algoritmos Lagrangeanos apresentados comparou-se o valor do limite superior obtido em cada algoritmo Lagrangeano com o valor ótimo ou o melhor valor do limite superior obtido usando o Procedimento P-WMTZ+C descrito no Capítulo 3.

Sejam LS o valor do limite superior obtido para o valor do custo usando os algoritmos Lagrangeanos, $\vartheta(WMST_\lambda)$ o valor obtido pela relaxação Lagrangeana usando os algoritmos Lagrangeanos e OPT o valor ótimo ($\vartheta(WMST)$) ou o melhor valor obtido até ao momento através do Procedimento P-WMTZ+C. Como todos estes algoritmos obtêm um limite superior (LS) e um limite inferior (LI) (valor da relaxação Lagrangeana), podemos considerar dois tipos de *gap*, o *gap* correspondente ao valor do limite superior que é dado por

$$Gap_{LS} = \frac{LS - OPT}{OPT} \times 100,$$

a este valor passaremos a designar de *gap* do limite superior (Gap_{LS}) e o *gap* correspondente ao valor do limite inferior obtido através da relaxação Lagrangeana que é

dado por

$$Gap_R = \frac{OPT - \vartheta(WMST_\lambda)}{OPT} \times 100,$$

a este valor passaremos a designar de *gap* da relaxação Lagrangeana (Gap_R).

Os testes desta subsecção consistem no uso de todos os algoritmos apresentados neste capítulo para obter soluções para as 215 instâncias dos grupos Quase Caminhos, Aleatórias e Euclidianas e cuja descrição foi efetuada na Secção 2.3.

Vamos começar por efetuar uma análise geral aos algoritmos Lagrangeanos e em seguida passamos a estudar o comportamento em termos de qualidade e tempo de execução dos vários algoritmos em cada grupo de instâncias em separado. Em forma de síntese, no final do capítulo são apresentadas as principais conclusões obtidas.

Na Tabela 4.11 podemos observar em cada grupo de instâncias em estudo as percentagens de *gaps* nulos obtidas através do limite superior nos vários Algoritmos Lagrangeanos propostos. Na última linha da tabela encontram-se as percentagens de $Gap_{LS} = 0$ obtidas tendo em consideração todas as instâncias em estudo.

	Alg1	Alg2	Alg3	Alg4	Alg5	Alg6	Alg7
QC	6,32	35,79	16,84	6,32	35,79	35,79	37,89
R	5,00	6,67	1,67	5,00	6,67	6,67	5,00
E	25,00	30,00	11,67	23,33	30,00	30,00	28,33
% $Gap_{LS} = 0$	11,16	26,05	11,16	10,70	26,05	26,05	26,05

Tabela 4.11: Percentagens de *gaps* nulos obtidas através do LS nos vários algoritmos por cada grupo de instâncias.

Em geral, dos vários algoritmos em estudo, os Algoritmos Alg2, Alg5, Alg6 e Alg7 são os que apresentam a maior percentagem de *gaps* nulos 26,05% (56 instâncias em 215). Nos grupos de instâncias Quase Caminhos e Euclidianas é onde se obtêm a maior percentagem de *gaps* nulos.

Na Tabela 4.12 podemos observar em cada grupo de instâncias em estudo, as percentagens de gap do limite superior que são inferiores aos gap da relaxação Lagrangeana obtidos nos vários algoritmos Lagrangeanos propostos. Na última linha da tabela encontram-se as percentagens de gap do limite superior que são inferiores aos gap da relaxação Lagrangeana obtidas tendo em consideração todas as instâncias em estudo.

	Alg1	Alg2	Alg3	Alg4	Alg5	Alg6	Alg7
QC	54,74	75,79	67,37	97,89	75,79	75,79	76,84
R	16,67	20,00	11,67	13,33	20,00	20,00	18,33
E	38,33	40,00	25,00	31,67	40,00	40,00	35,00
% $Gap_{LS} < Gap_R$	39,53	50,23	40,00	55,81	50,23	50,23	48,84

Tabela 4.12: Percentagens de Gap_{LS} que são inferiores aos Gap_R nos vários algoritmos por cada grupo de instâncias.

Observamos uma maior percentagem de Gap_{LS} inferior ao Gap_R nas instâncias Quase Caminhos, isto é, na maior parte destas instâncias o valor do limite superior obtido encontra-se mais próximo do valor ótimo comparativamente ao valor do limite inferior. Já nos outros dois grupos de instâncias verifica-se um comportamento diferente, o que significa que a relaxação Lagrangeana nos grupos de instâncias Aleatórias e Euclidianas apresenta soluções mais próximas da solução ótima. Portanto, no grupo de instâncias Quase Caminhos é preferível encontrar um valor para o limite superior para o custo, pois este encontra-se mais próximo do valor ótimo, enquanto que nos grupos de instâncias Aleatórias e Euclidianas é mais conveniente a escolha do valor do limite inferior para o custo dado pelo valor da relaxação Lagrangeana.

Os Algoritmos Alg2, Alg5 e Alg6 mostraram-se apropriados nos grupos de instâncias considerados para encontrar um valor muito próximo do valor da relaxação linear, pois os valores obtidos através da aplicação da relaxação Lagrangeana aproximam-se dos valores da relaxação linear obtidos através do Procedimento P-WMTZ+C descrito na Secção 3.3.

Em 11,67% das instâncias do grupo Aleatórias (7 em 60 instâncias) e em 25% das instâncias Euclidianas (15 em 60 instâncias), mais precisamente as instâncias com mais de 300 nodos, o valor obtido através da relaxação Lagrangeana encontra-

se mais próximo do valor ótimo do que o valor da relaxação linear do Procedimento P-WMTZ+C. Este facto ocorre, pois para instâncias do grupo Aleatórias e Euclidianas com mais de 300 nodos, a relaxação linear do Procedimento P-WMTZ+C é interrompida quando o *gap*, em percentagem, entre duas soluções consecutivas é inferior a 0,000001, mais do que 20 vezes ou quando o limite de tempo na introdução de cortes exceder 10000 segundos.

De seguida e para facilitar a análise, vamos comparar os algoritmos, em cada grupo de instâncias em estudo e para tal separamos as instâncias em 12 subgrupos (um por cada valor para o número de nodos). Para analisar a qualidade das soluções obtidas são apresentados para cada algoritmo os *gap* médios, em percentagem, do valor do limite superior, Gap_{LS} , e do valor da relaxação Lagrangeana, Gap_R , e os respetivos desvios padrões (ver Tabelas 4.13, 4.14, 4.17, 4.18, 4.21 e 4.22). Para se poder avaliar a rapidez dos algoritmos Lagrangeanos são apresentados também para cada algoritmo, os valores dos tempos médios de execução (em segundos) nos vários subgrupos de nodos com os respetivos desvios padrões (ver Tabelas 4.15, 4.16, 4.19, 4.20, 4.23 e 4.24).

Instâncias Quase Caminhos

Através da observação das Tabelas 4.13 e 4.14 conclui-se que os Algoritmos Alg1 e Alg4 apresentam Gap_{LS} e Gap_R muito elevados quando comparados com os correspondentes $gaps$ dos restantes algoritmos. Os Algoritmos Alg2, Alg5 e Alg6 determinam as mesmas soluções. Em relação ao Algoritmo Alg3 podemos dizer que os Gap_R se aproximam dos Gap_R dos Algoritmos Alg2, Alg5 e Alg6 e os Gap_{LS} são mais elevados do que os Gap_{LS} dos Algoritmos Alg2, Alg5 e Alg6 mas, apesar disso e, em geral, são melhores do que os dos Algoritmos Alg1 e Alg4.

No gráfico da Figura 4.7 podemos comparar os Gap_R do Algoritmo Alg7 com os do Algoritmo Alg6 (os Algoritmos Alg2 e Alg5 têm Gap_R igual ao Algoritmo Alg6).

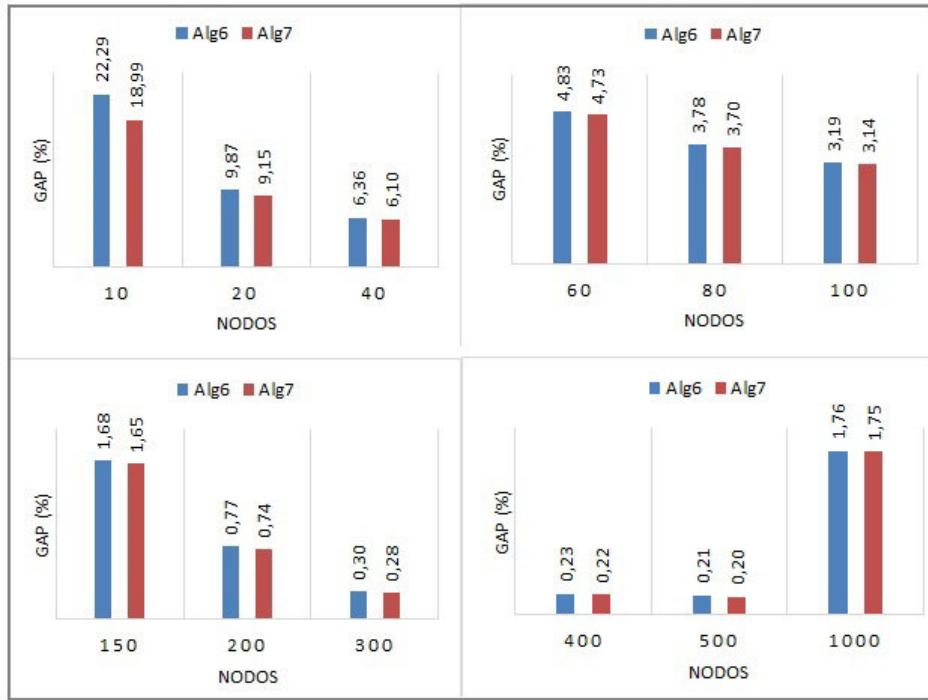


Figura 4.7: Comparação dos Gap_R médios, em percentagem, entre os Algoritmos Alg6 e Alg7 nas instâncias QC.

O Algoritmo Alg7 é o que apresenta $gaps$ da relaxação Lagrangeana mais baixos em todas as instâncias. Em relação ao Gap_{LS} , o Algoritmo Alg6 apresenta $gaps$ mais baixos do que o Algoritmo Alg7, em 7,37% dos casos (7 instâncias em 95) e o Algoritmo Alg7 apresenta menores $gaps$ em 4,21% dos casos (4 instâncias em 95). Nas restantes 84 instâncias os Gap_{LS} são iguais nos dois algoritmos.

	Alg1		Alg2		Alg3		Alg4		Alg5		Alg6		Alg7	
N.º Nodos	<i>Gap_R</i>	<i>Gap_{LS}</i>	<i>Gap_R</i>	<i>Gap_{LS}</i>	<i>Gap_R</i>	<i>Gap_{LS}</i>	<i>Gap_R</i>	<i>Gap_{LS}</i>	<i>Gap_R</i>	<i>Gap_{LS}</i>	<i>Gap_R</i>	<i>Gap_{LS}</i>	<i>Gap_R</i>	<i>Gap_{LS}</i>
10	46,516	45,433	22,292	7,138	23,551	37,666	28,855	28,688	22,292	7,138	22,292	7,138	18,987	7,138
20	26,899	17,027	9,874	0,574	10,732	6,031	31,769	17,533	9,874	0,574	9,874	0,574	9,145	0,574
40	15,654	18,451	6,362	0,579	7,004	9,332	35,398	23,925	6,362	0,579	6,362	0,579	6,101	1,545
60	11,061	11,658	4,831	0,022	5,277	7,870	40,532	12,949	4,831	0,022	4,831	0,022	4,729	0,022
80	10,767	9,798	3,778	0,058	4,277	7,888	42,424	9,798	3,778	0,058	3,778	0,058	3,701	0,058
100	7,406	6,083	3,192	1,048	3,417	6,349	43,984	7,147	3,192	1,048	3,192	1,048	3,141	0,346
150	11,774	8,559	1,680	0,773	1,867	3,893	42,587	9,439	1,680	0,773	1,680	0,773	1,651	0,445
200	4,618	3,716	0,771	0,719	0,831	1,837	42,050	3,716	0,771	0,719	0,771	0,719	0,739	1,837
300	3,647	3,050	0,299	1,628	0,350	2,399	42,251	3,050	0,299	1,628	0,299	1,628	0,282	1,628
400	10,074	9,291	0,232	1,433	0,675	9,291	40,651	9,291	0,232	1,433	0,232	1,433	0,219	2,030
500	3,455	3,314	0,207	1,265	0,224	2,351	41,860	3,894	0,207	1,265	0,207	1,265	0,198	1,771
1000	3,598	0,292	1,757	0,292	1,806	0,292	44,696	0,292	1,757	0,292	1,757	0,292	1,754	0,292

Tabela 4.13: *Gaps* médios, em percentagem, obtidos pelos algoritmos Lagrangeanos nas instâncias QC.

	Alg1		Alg2		Alg3		Alg4		Alg5		Alg6		Alg7	
N.º Nodos	<i>Gap_R</i>	<i>Gap_{LS}</i>	<i>Gap_R</i>	<i>Gap_{LS}</i>	<i>Gap_R</i>	<i>Gap_{LS}</i>	<i>Gap_R</i>	<i>Gap_{LS}</i>	<i>Gap_R</i>	<i>Gap_{LS}</i>	<i>Gap_R</i>	<i>Gap_{LS}</i>	<i>Gap_R</i>	<i>Gap_{LS}</i>
10	26,556	41,545	7,061	8,075	7,058	45,160	11,237	28,365	7,061	8,075	7,061	8,075	6,756	8,075
20	23,118	8,109	4,380	0,610	4,588	9,238	8,347	7,768	4,380	0,610	4,380	0,610	4,074	0,610
40	9,213	10,500	0,840	0,614	1,230	10,831	3,919	5,751	0,840	0,614	0,840	0,614	0,762	2,974
60	6,314	4,918	0,863	0,029	1,020	5,937	3,096	2,721	0,863	0,029	0,863	0,029	0,853	0,029
80	5,351	1,740	0,884	0,051	1,155	4,487	1,198	1,740	0,884	0,051	0,884	0,051	0,883	0,051
100	4,791	2,538	0,274	1,690	0,275	1,444	1,394	0,602	0,274	1,690	0,274	1,690	0,280	1,087
150	8,054	3,462	0,360	1,271	0,582	4,545	1,589	1,750	0,360	1,271	0,360	1,271	0,361	0,931
200	3,300	0,911	0,345	0,679	0,330	1,500	2,141	0,911	0,345	0,679	0,345	0,679	0,350	1,500
300	1,597	0,923	0,214	1,629	0,194	1,539	1,966	0,923	0,214	1,629	0,214	1,629	0,218	1,629
400	8,370	10,450	0,057	1,867	0,677	10,450	4,709	10,450	0,057	1,867	0,057	1,867	0,052	1,966
500	3,565	1,903	0,082	1,295	0,083	2,040	1,856	1,049	0,082	1,295	0,082	1,295	0,079	2,108
1000	1,704	0,652	1,115	0,652	1,157	0,652	3,638	0,652	1,115	0,652	1,115	0,652	1,116	0,652

Tabela 4.14: Desvios Padrões dos *gap* obtidos pelos algoritmos Lagrangeanos nas instâncias QC.

Na Tabela 4.15 encontram-se representados os tempos médios, em segundos, de execução dos vários algoritmos Lagrangeanos para as instâncias Quase Caminhos e na Tabela 4.16 encontram-se os respectivos desvios padrões.

Da observação das tabelas referidas e do gráfico da Figura 4.8 podemos constatar que de todos os algoritmos Lagrangeanos apresentados o mais lento é o Algoritmo Alg3.

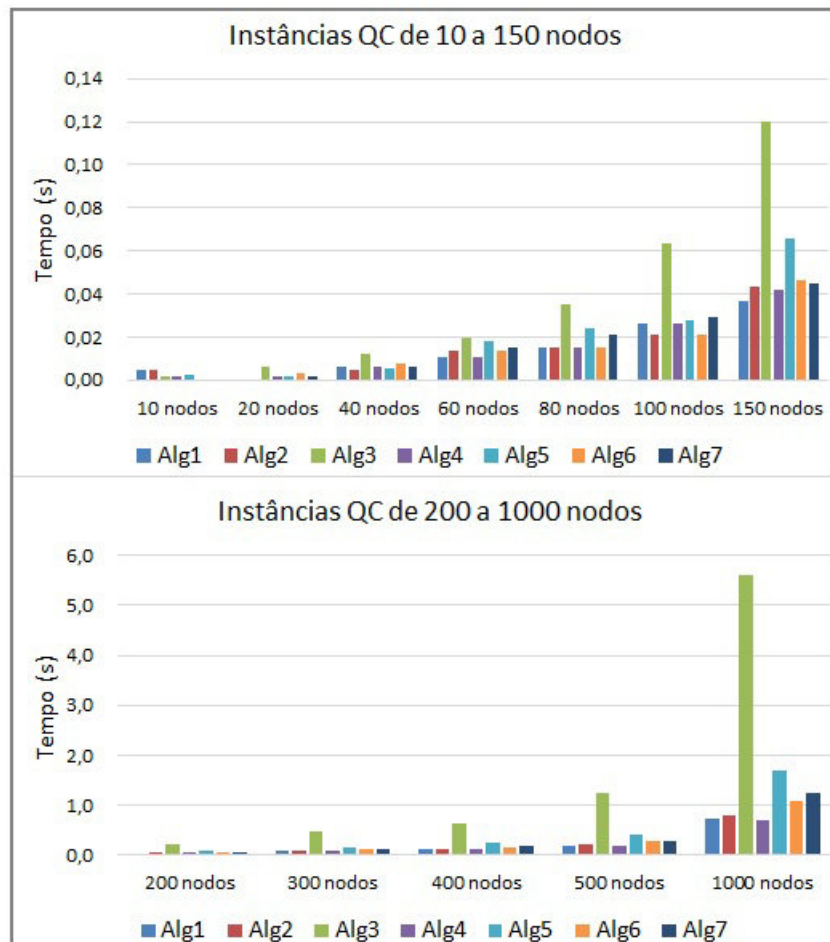


Figura 4.8: Tempos médios, em segundos, obtidos pelos algoritmos Lagrangeanos (instâncias QC).

Podemos também observar que de todos os algoritmos o Algoritmo Alg5 é o segundo a demorar mais tempo. É de realçar ainda que todos os algoritmos com exceção do Algoritmo Alg3 correm em menos de 2 segundos. Se considerarmos apenas instâncias com menos de 200 nodos, os algoritmos com exceção do Algoritmo Alg3 correm, em média, em menos de 0,08 segundos.

N.º Nodos	Alg1	Alg2	Alg3	Alg4	Alg5	Alg6	Alg7
10	0,005	0,005	0,002	0,002	0,002	0,000	0,000
20	0,000	0,000	0,006	0,002	0,002	0,003	0,002
40	0,006	0,005	0,012	0,006	0,006	0,008	0,006
60	0,011	0,014	0,020	0,011	0,018	0,014	0,015
80	0,015	0,015	0,036	0,015	0,024	0,015	0,021
100	0,026	0,021	0,064	0,026	0,028	0,021	0,029
150	0,037	0,043	0,120	0,042	0,066	0,046	0,045
200	0,043	0,056	0,231	0,056	0,091	0,065	0,062
300	0,087	0,087	0,465	0,081	0,165	0,112	0,118
400	0,127	0,130	0,649	0,115	0,260	0,171	0,174
500	0,190	0,212	1,254	0,184	0,429	0,271	0,283
1000	0,742	0,801	5,622	0,689	1,708	1,076	1,232

Tabela 4.15: Tempos médios, em segundos, obtidos pelos algoritmos Lagrangeanos (instâncias QC).

N.º Nodos	Alg1	Alg2	Alg3	Alg4	Alg5	Alg6	Alg7
10	0,010	0,015	0,005	0,005	0,004	0,000	0,000
20	0,000	0,000	0,008	0,005	0,001	0,006	0,005
40	0,008	0,007	0,006	0,008	0,002	0,008	0,008
60	0,007	0,005	0,008	0,007	0,002	0,009	0,000
80	0,007	0,007	0,007	0,007	0,004	0,010	0,008
100	0,008	0,013	0,016	0,008	0,010	0,011	0,011
150	0,018	0,022	0,013	0,023	0,021	0,024	0,014
200	0,013	0,024	0,026	0,009	0,010	0,013	0,011
300	0,008	0,014	0,039	0,007	0,015	0,013	0,021
400	0,007	0,009	0,148	0,014	0,015	0,011	0,007
500	0,006	0,008	0,193	0,012	0,016	0,007	0,006
1000	0,028	0,014	0,100	0,034	0,079	0,036	0,047

Tabela 4.16: Desvios padrões dos tempos obtidos pelos algoritmos Lagrangeanos (instâncias QC).

Os Algoritmos Alg3 e Alg5 são os que, em geral, demoram mais tempo. Quanto ao Algoritmo Alg3, vimos na Subsecção 4.5.2 que é necessário escolher um valor de ρ adequado, este facto pode aumentar o tempo de execução. Para explicar o motivo que leva o Algoritmo Alg5 a demorar mais tempo elaborámos o gráfico da Figura 4.9 que compara o número médio de árvores de suporte obtidas usando cada um dos Algoritmos Alg5 e Alg6.

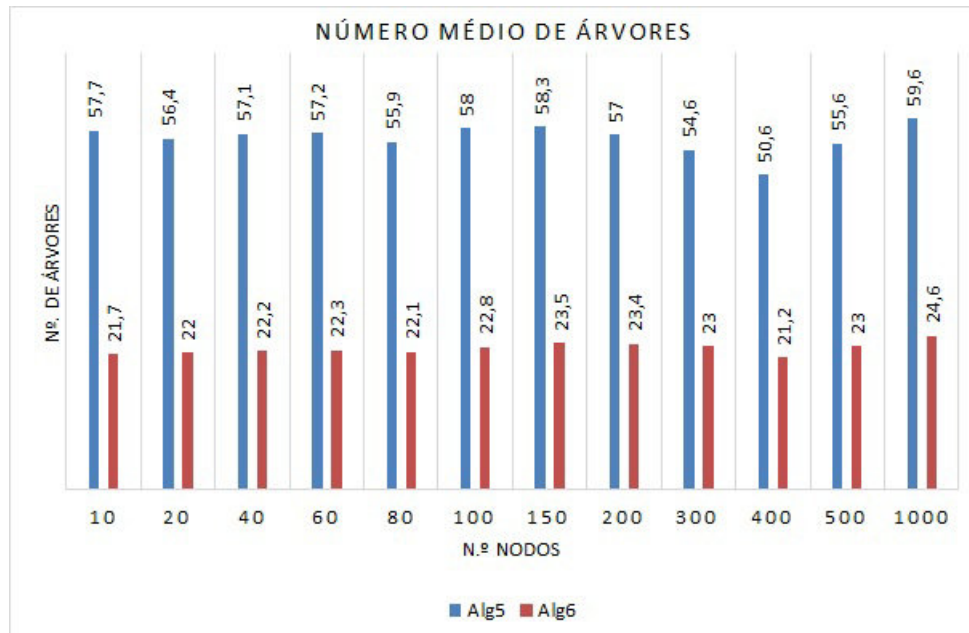


Figura 4.9: Número médio de árvores de suporte obtidas pelos Algoritmos Alg5 e Alg6 nas instâncias QC.

Em geral, o tempo de execução do Algoritmo Alg6 é inferior ao tempo de execução do Algoritmo Alg5, o que é facilmente explicado pelo facto do número médio de árvores formadas ao longo da execução do algoritmo diminuir para mais do que metade no Algoritmo Alg6 (Figura 4.9).

Notamos que nas 4 instâncias de 1000 nodos do grupo de instâncias Quase Caminhos em que não tinha sido possível encontrar um valor para o limite superior através do Procedimento P-WMTZ+C, com a aplicação dos vários algoritmos Lagrangeanos foi possível obter um valor para o limite superior para o custo.

Instâncias Aleatórias

Através da observação das Tabelas 4.17 e 4.18 conclui-se que, ao contrário do que acontecia para as instâncias Quase Caminhos, neste grupo de instâncias os Algoritmos Alg1 e Alg4, em geral, não apresentam *gaps* muito elevados quer do limite superior quer da relaxação Lagrangeana, quando comparados com os restantes algoritmos.

Os Algoritmos Alg2, Alg5 e Alg6 determinam as mesmas soluções. Em relação ao Algoritmo Alg3 podemos dizer que os Gap_R se aproximam dos Gap_R dos Algoritmos Alg2, Alg5 e Alg6 e os Gap_{LS} são os piores de todos os algoritmos.

No gráfico da Figura 4.10 podemos comparar os Gap_R do Algoritmo Alg7 com os do Algoritmo Alg6 (os Algoritmos Alg2 e Alg5 têm Gap_R igual ao Algoritmo Alg6).

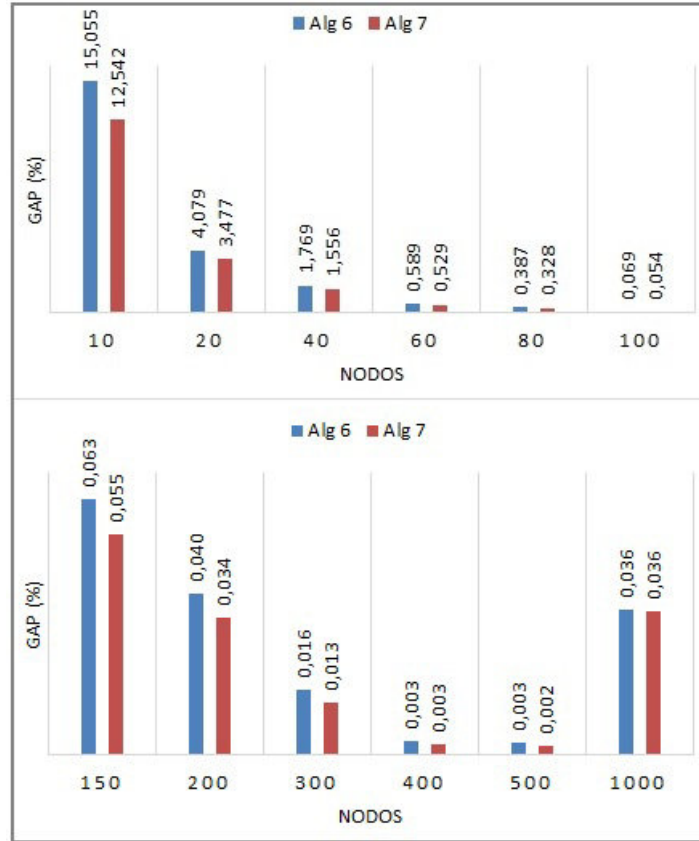


Figura 4.10: Comparação dos Gap_R médios, em percentagem, entre os Algoritmos Alg6 e Alg7 nas instâncias R.

O Algoritmo Alg7 é o que apresenta Gap_R mais baixo em todas as instâncias Aleatórias. Os Gap_{LS} são iguais em todas as instâncias.

	Alg1		Alg2		Alg3		Alg4		Alg5		Alg6		Alg7	
N.º Nodos	Gap_R	Gap_{LS}	Gap_R	Gap_{LS}	Gap_R	Gap_{LS}	Gap_R	Gap_{LS}	Gap_R	Gap_{LS}	Gap_R	Gap_{LS}	Gap_R	Gap_{LS}
10	15,055	3,840	15,055	3,840	15,612	7,713	16,780	3,840	15,055	3,840	15,055	3,840	12,542	3,840
20	4,196	6,208	4,079	5,285	7,304	22,292	6,350	14,955	4,079	5,285	4,079	5,285	3,477	5,285
40	1,788	2,743	1,769	2,743	3,549	8,828	1,889	2,743	1,769	2,743	1,769	2,743	1,556	2,743
60	0,622	1,312	0,589	1,312	0,818	2,775	0,622	1,312	0,589	1,312	0,589	1,312	0,529	1,312
80	0,401	1,505	0,387	1,505	1,255	5,343	0,447	2,242	0,387	1,505	0,387	1,505	0,328	1,505
100	0,072	0,547	0,069	0,547	0,250	3,581	0,070	0,547	0,069	0,547	0,069	0,547	0,054	0,547
150	0,064	0,787	0,063	0,787	0,132	2,068	0,065	0,787	0,063	0,787	0,063	0,787	0,055	0,787
200	0,044	0,533	0,040	0,533	0,299	4,111	0,046	0,704	0,040	0,533	0,040	0,533	0,034	0,533
300	0,016	0,567	0,016	0,567	0,057	1,811	0,020	0,713	0,016	0,567	0,016	0,567	0,013	0,567
400	0,003	0,145	0,003	0,145	0,030	0,930	0,004	0,145	0,003	0,145	0,003	0,145	0,003	0,145
500	0,003	0,220	0,003	0,220	0,060	1,445	0,004	0,275	0,003	0,220	0,003	0,220	0,002	0,220
1000	0,036	0,143	0,036	0,071	0,038	0,303	0,036	0,143	0,036	0,071	0,036	0,071	0,036	0,117

Tabela 4.17: Gap médios, em percentagem, obtidos pelos algoritmos Lagrangeanos nas instâncias R.

	Alg1		Alg2		Alg3		Alg4		Alg5		Alg6		Alg7	
N.º Nodos	<i>Gap_R</i>	<i>Gap_{LS}</i>	<i>Gap_R</i>	<i>Gap_{LS}</i>	<i>Gap_R</i>	<i>Gap_{LS}</i>	<i>Gap_R</i>	<i>Gap_{LS}</i>	<i>Gap_R</i>	<i>Gap_{LS}</i>	<i>Gap_R</i>	<i>Gap_{LS}</i>	<i>Gap_R</i>	<i>Gap_{LS}</i>
10	4,034	3,430	4,034	3,430	3,950	8,592	3,811	3,430	4,034	3,430	4,034	3,430	3,685	3,430
20	0,365	4,670	0,312	5,006	3,816	28,414	3,873	17,027	0,312	5,006	0,312	5,006	0,142	5,006
40	0,183	1,676	0,182	1,676	3,073	12,972	0,267	1,676	0,182	1,676	0,182	1,676	0,209	1,676
60	0,117	1,074	0,077	1,074	0,205	3,226	0,090	1,074	0,077	1,074	0,077	1,074	0,066	1,074
80	0,029	1,320	0,037	1,320	1,862	8,213	0,077	1,645	0,037	1,320	0,037	1,320	0,034	1,320
100	0,047	0,264	0,045	0,264	0,294	3,901	0,046	0,264	0,045	0,264	0,045	0,264	0,042	0,264
150	0,035	0,684	0,035	0,684	0,123	1,619	0,036	0,684	0,035	0,684	0,035	0,684	0,034	0,684
200	0,019	0,379	0,019	0,379	0,418	5,374	0,025	0,515	0,019	0,379	0,019	0,379	0,018	0,379
300	0,001	0,621	0,001	0,621	0,043	1,256	0,005	0,692	0,001	0,621	0,001	0,621	0,002	0,621
400	0,003	0,098	0,003	0,098	0,054	1,417	0,003	0,098	0,003	0,098	0,003	0,098	0,003	0,098
500	0,002	0,136	0,002	0,136	0,106	1,861	0,002	0,223	0,002	0,136	0,002	0,136	0,002	0,136
10000	0,046	0,120	0,046	0,081	0,046	0,183	0,046	0,120	0,046	0,081	0,046	0,081	0,046	0,085

Tabela 4.18: Desvios Padrões dos *gap* obtidos pelos algoritmos Lagrangeanos nas instâncias R.

No gráfico da Figura 4.11 encontram-se representados os tempos médios, em segundos, de execução dos vários algoritmos Lagrangeanos para as instâncias Aleatórias. Estes tempos médios encontram-se registados na Tabela 4.19 e os correspondentes desvios padrões na Tabela 4.20.

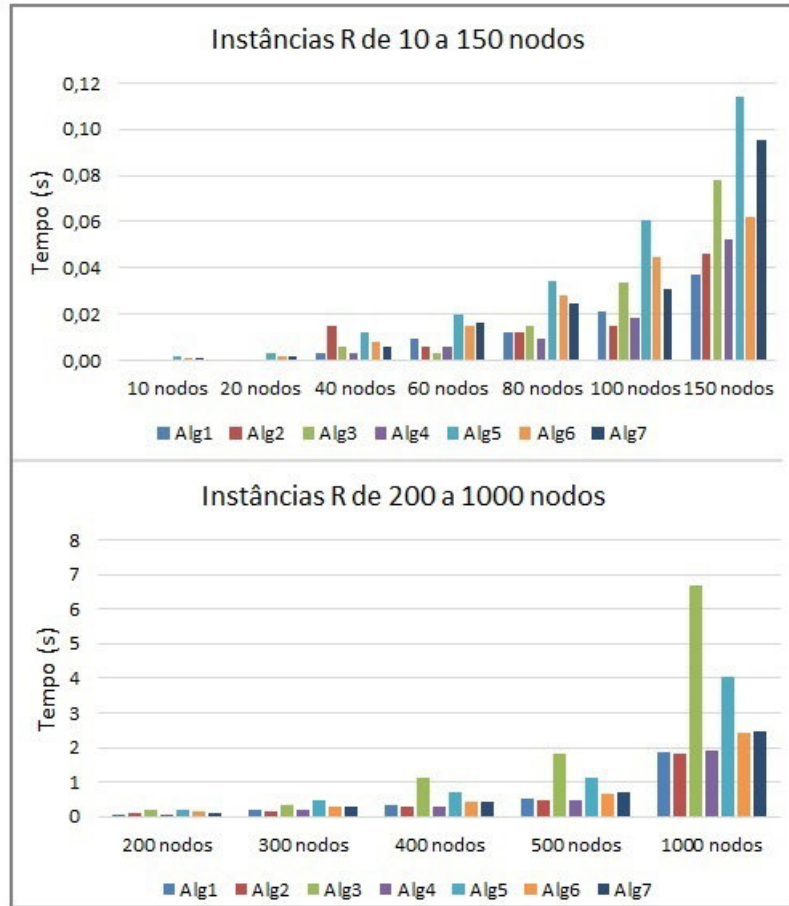


Figura 4.11: Tempos médios, em segundos, obtidos pelos algoritmos Lagrangeanos (instâncias R).

Em relação ao algoritmo que demora mais tempo na obtenção das soluções aproximadas podemos observar através do gráfico da Figura 4.11 dois comportamentos diferentes. Em instâncias com um número de nodos inferior ou igual a 300, o Algoritmo Alg5 é o que apresenta maiores tempos médios de execução, enquanto que para as restantes instâncias é o Algoritmo Alg3 que apresenta tempos mais elevados.

É de notar que todos os algoritmos com exceção do Algoritmo Alg3 correm em menos do que 5 segundos. Se considerarmos apenas instâncias com menos de 200 nodos, estas correm, em média, em menos de 0,12 segundos.

N.º Nodos	Alg1	Alg2	Alg3	Alg4	Alg5	Alg6	Alg7
10	0,000	0,000	0,000	0,000	0,001	0,001	0,001
20	0,000	0,000	0,000	0,000	0,003	0,002	0,002
40	0,003	0,015	0,006	0,003	0,012	0,008	0,006
60	0,009	0,006	0,003	0,006	0,020	0,015	0,016
80	0,012	0,012	0,015	0,009	0,034	0,028	0,024
100	0,021	0,015	0,034	0,018	0,060	0,045	0,031
150	0,037	0,046	0,078	0,052	0,114	0,062	0,095
200	0,078	0,081	0,190	0,075	0,205	0,149	0,121
300	0,208	0,162	0,315	0,215	0,463	0,275	0,303
400	0,318	0,302	1,120	0,309	0,699	0,446	0,436
500	0,502	0,480	1,812	0,477	1,101	0,661	0,686
1000	1,853	1,828	6,686	1,922	4,034	2,405	2,477

Tabela 4.19: Tempos médios, em segundos, obtidos pelos algoritmos Lagrangeanos (instâncias R).

N.º Nodos	Alg1	Alg2	Alg3	Alg4	Alg5	Alg6	Alg7
10	0,000	0,000	0,000	0,000	0,001	0,000	0,000
20	0,000	0,000	0,000	0,000	0,001	0,001	0,001
40	0,007	0,000	0,008	0,007	0,005	0,004	0,001
60	0,008	0,008	0,007	0,008	0,003	0,004	0,005
80	0,007	0,007	0,000	0,008	0,004	0,016	0,007
100	0,009	0,000	0,013	0,007	0,032	0,027	0,007
150	0,008	0,011	0,019	0,009	0,029	0,015	0,078
200	0,000	0,007	0,102	0,007	0,087	0,067	0,043
300	0,067	0,008	0,243	0,124	0,100	0,043	0,120
400	0,052	0,018	0,349	0,007	0,101	0,064	0,055
500	0,063	0,047	1,048	0,067	0,062	0,075	0,073
1000	0,035	0,063	1,931	0,147	0,423	0,074	0,142

Tabela 4.20: Desvios padrões dos tempos obtidos pelos algoritmos Lagrangeanos (instâncias R).

Para justificar o motivo que leva o Algoritmo Alg5 a demorar mais tempo elaborámos o gráfico da Figura 4.12 que compara o número médio de árvores de suporte obtidas usando cada um dos Algoritmos Alg5 e Alg6.

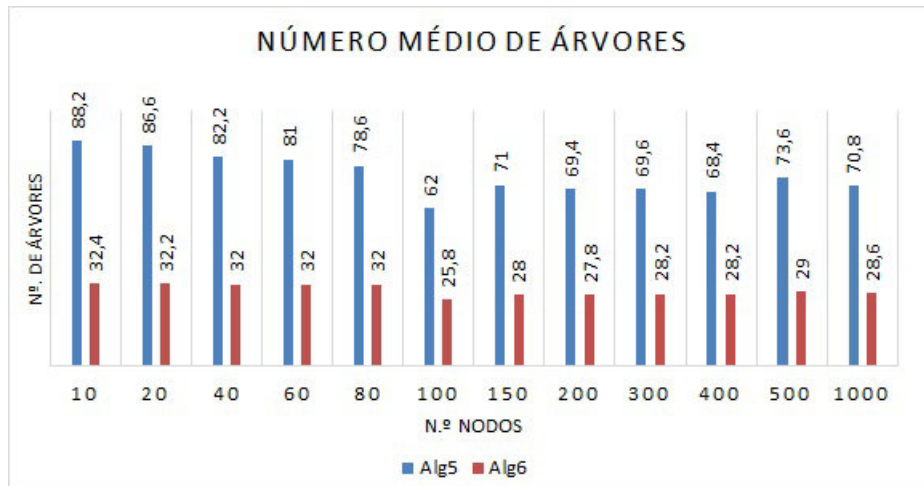


Figura 4.12: Número médio de árvores de suporte obtidas pelos Algoritmos Alg5 e Alg6 nas instâncias R.

Tal como vimos para o caso do grupo de instâncias Quase Caminhos, neste grupo de instâncias também, em geral, o tempo de execução do Algoritmo Alg6 é inferior ao tempo de execução do Algoritmo Alg5, o que é facilmente explicado pelo facto do número médio de árvores formadas ao longo da execução do algoritmo diminuir para metade no Algoritmo Alg6 (Figura 4.12).

Notamos que nas 2 instâncias de 1000 nodos do grupo de instâncias Aleatórias em que não tinha sido possível encontrar um valor para o limite superior através do Procedimento P-WMTZ+C, com a aplicação dos vários algoritmos Lagrangeanos foi possível obter um valor para o limite superior para o custo.

Instâncias Euclidianas

Através da observação das Tabelas 4.21 e 4.22 conclui-se que ao contrário do que acontecia para as instâncias Quase Caminhos, neste grupo de instâncias os Algoritmos Alg1 e Alg4, em geral, não apresentam *gaps* muito elevados quer do limite superior quer da relaxação Lagrangeana quando comparados com os restantes algoritmos tal como se verificou no grupo de instâncias Aleatórias.

Os Algoritmos Alg2, Alg5 e Alg6 determinam as mesmas soluções. Em relação ao Algoritmo Alg3 podemos dizer que os Gap_R e Gap_{LS} se aproximam dos Gap_R e Gap_{LS} dos Algoritmos Alg2, Alg5 e Alg6.

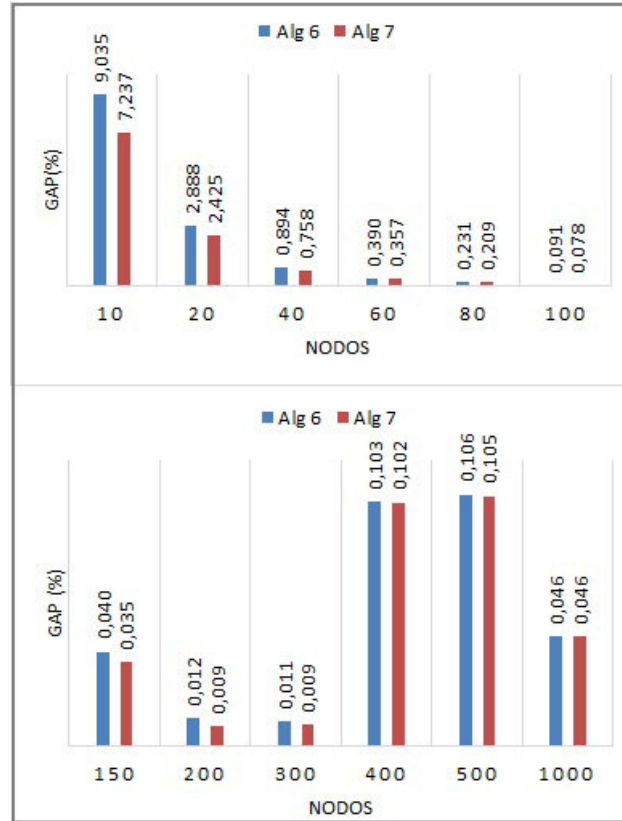


Figura 4.13: Comparação dos Gap_R médios, em percentagem, entre os Algoritmos Alg6 e Alg7 nas instâncias E.

O Algoritmo Alg7 é o que apresenta Gap_R mais baixo em todas as instâncias Euclidianas. Os Gap_{LS} são iguais exceto numa instância de 1000 nodos (E1000-4) onde o *gap* do Algoritmo Alg6 é mais baixo.

	Alg1		Alg2		Alg3		Alg4		Alg5		Alg6		Alg7	
N.º Nodos	Gap_R	Gap_{LS}	Gap_R	Gap_{LS}	Gap_R	Gap_{LS}	Gap_R	Gap_{LS}	Gap_R	Gap_{LS}	Gap_R	Gap_{LS}	Gap_R	Gap_{LS}
10	9,035	5,274	9,035	5,274	9,039	5,274	10,816	5,274	9,035	5,274	9,035	5,274	7,237	5,274
20	2,938	2,250	2,888	2,250	2,888	2,250	6,058	11,323	2,888	2,250	2,888	2,250	2,425	2,250
40	0,894	1,959	0,894	1,959	0,894	1,959	0,969	1,959	0,894	1,959	0,894	1,959	0,758	1,959
60	0,399	0,890	0,390	0,747	0,394	0,747	0,428	1,039	0,390	0,747	0,390	0,747	0,357	0,747
80	0,239	0,576	0,231	0,576	0,232	0,576	0,239	0,576	0,231	0,576	0,231	0,576	0,209	0,576
100	0,104	0,706	0,091	0,563	0,120	1,378	0,127	1,480	0,091	0,563	0,091	0,563	0,078	0,563
150	0,043	0,284	0,040	0,284	0,058	0,628	0,041	0,284	0,040	0,284	0,040	0,284	0,035	0,284
200	0,013	0,402	0,012	0,402	0,064	1,416	0,012	0,402	0,012	0,402	0,012	0,402	0,009	0,402
300	0,011	0,280	0,011	0,219	0,014	0,280	0,012	0,280	0,011	0,219	0,011	0,219	0,009	0,219
400	0,103	0,000	0,103	0,000	0,115	0,500	0,103	0,000	0,103	0,000	0,103	0,000	0,102	0,000
500	0,106	0,034	0,106	0,000	0,114	0,572	0,106	0,108	0,106	0,000	0,106	0,000	0,105	0,000
1000	0,046	0,007	0,046	0,000	0,052	0,496	0,046	0,030	0,046	0,000	0,046	0,000	0,046	0,007

Tabela 4.21: Gap médios, em percentagem, obtidos pelos algoritmos Lagrangeanos nas instâncias E.

	Alg1		Alg2		Alg3		Alg4		Alg5		Alg6		Alg7	
N.º Nodos	<i>Gap_R</i>	<i>Gap_{LS}</i>	<i>Gap_R</i>	<i>Gap_{LS}</i>	<i>Gap_R</i>	<i>Gap_{LS}</i>	<i>Gap_R</i>	<i>Gap_{LS}</i>	<i>Gap_R</i>	<i>Gap_{LS}</i>	<i>Gap_R</i>	<i>Gap_{LS}</i>	<i>Gap_R</i>	<i>Gap_{LS}</i>
10	0,993	3,493	0,993	3,493	0,994	3,493	2,833	3,493	0,993	3,493	0,993	3,493	0,915	3,493
20	0,610	2,685	0,640	2,685	0,640	2,685	2,726	11,121	0,640	2,685	0,640	2,685	0,459	2,685
40	0,106	0,390	0,106	0,390	0,106	0,390	0,126	0,390	0,106	0,390	0,106	0,390	0,099	0,390
60	0,056	0,793	0,059	0,708	0,060	0,708	0,094	1,336	0,059	0,708	0,059	0,708	0,066	0,708
80	0,026	0,428	0,016	0,428	0,015	0,428	0,013	0,428	0,016	0,428	0,016	0,428	0,016	0,428
100	0,049	0,675	0,029	0,665	0,040	0,890	0,076	1,526	0,029	0,665	0,029	0,665	0,027	0,665
150	0,018	0,194	0,016	0,194	0,025	0,578	0,017	0,194	0,016	0,194	0,016	0,194	0,015	0,194
200	0,008	0,195	0,007	0,195	0,069	1,585	0,007	0,195	0,007	0,195	0,007	0,195	0,006	0,195
300	0,040	0,156	0,040	0,187	0,042	0,156	0,040	0,156	0,040	0,187	0,040	0,187	0,040	0,187
400	0,115	0,000	0,115	0,000	0,120	0,687	0,115	0,000	0,115	0,000	0,115	0,000	0,115	0,000
500	0,057	0,076	0,056	0,000	0,064	0,497	0,056	0,242	0,056	0,000	0,056	0,000	0,056	0,000
1000	0,041	0,016	0,041	0,000	0,040	0,439	0,041	0,050	0,041	0,000	0,041	0,000	0,041	0,016

Tabela 4.22: Desvios Padrões dos *gap* obtidos pelos algoritmos Lagrangeanos nas instâncias E.

No gráfico da Figura 4.14 encontram-se representados os tempos médios, em segundos, de execução dos vários algoritmos Lagrangeanos para as instâncias Euclidianas. Estes tempos médios encontram-se registados na Tabela 4.23 e os correspondentes desvios padrões na Tabela 4.24.

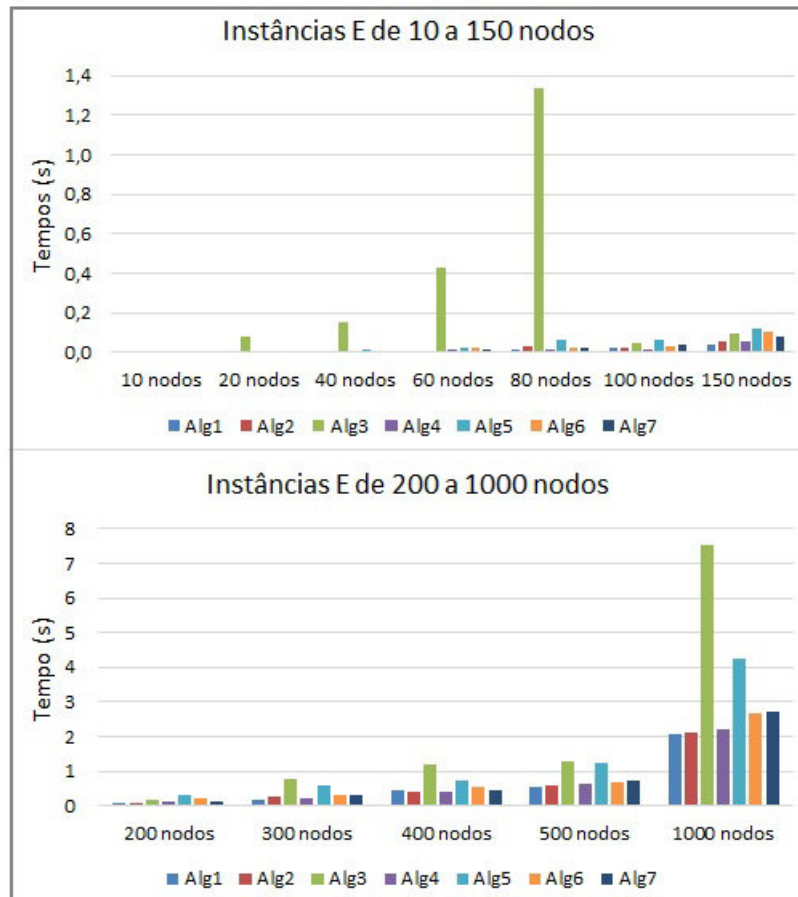


Figura 4.14: Tempos médios, em segundos, obtidos pelos algoritmos Lagrangeanos (instâncias E).

Na maior parte das instâncias o algoritmo que demora mais tempo na obtenção das soluções aproximadas é o Algoritmo Alg3. O segundo algoritmo a demorar mais tempo na execução é o Algoritmo Alg5. Todos os algoritmos com exceção do Algoritmo Alg3 correm em menos de 5 segundos. Se observarmos apenas o gráfico correspondente às instâncias de 10 a 150 nodos, todos os algoritmos com exceção do Algoritmo Alg3 correm, em média, em menos de 0,2 segundos.

N.º Nodos	Alg1	Alg2	Alg3	Alg4	Alg5	Alg6	Alg7
10	0,003	0,003	0,003	0,000	0,008	0,003	0,000
20	0,009	0,000	0,084	0,003	0,006	0,001	0,003
40	0,003	0,003	0,149	0,006	0,015	0,010	0,004
60	0,009	0,006	0,427	0,012	0,022	0,022	0,013
80	0,015	0,031	1,332	0,015	0,062	0,022	0,025
100	0,021	0,025	0,046	0,018	0,064	0,033	0,043
150	0,043	0,053	0,100	0,052	0,118	0,107	0,076
200	0,090	0,078	0,174	0,128	0,318	0,200	0,125
300	0,174	0,258	0,792	0,230	0,572	0,336	0,297
400	0,449	0,408	1,213	0,387	0,723	0,528	0,471
500	0,539	0,596	1,307	0,617	1,253	0,684	0,739
1000	2,084	2,130	7,531	2,187	4,234	2,669	2,722

Tabela 4.23: Tempos médios, em segundos, obtidos pelos algoritmos Lagrangeanos (instâncias E).

N.º Nodos	Alg1	Alg2	Alg3	Alg4	Alg5	Alg6	Alg7
10	0,007	0,007	0,007	0,000	0,015	0,004	0,000
20	0,008	0,000	0,171	0,007	0,004	0,001	0,004
40	0,007	0,007	0,032	0,008	0,007	0,006	0,004
60	0,008	0,008	0,107	0,007	0,005	0,013	0,004
80	0,011	0,035	0,271	0,000	0,038	0,002	0,004
100	0,009	0,009	0,025	0,007	0,024	0,006	0,019
150	0,007	0,014	0,045	0,014	0,024	0,060	0,016
200	0,030	0,000	0,058	0,094	0,094	0,089	0,031
300	0,007	0,099	0,500	0,074	0,118	0,091	0,076
400	0,072	0,086	0,790	0,071	0,052	0,133	0,051
500	0,101	0,093	0,800	0,106	0,117	0,056	0,101
1000	0,085	0,094	2,808	0,072	0,120	0,086	0,066

Tabela 4.24: Desvios padrões dos tempos obtidos pelos algoritmos Lagrangeanos (instâncias E).

Para justificar o motivo que leva o Algoritmo Alg5 a demorar mais tempo elaborámos o gráfico da Figura 4.15 que compara o número médio de árvores de suporte obtidas usando cada um dos Algoritmos Alg5 e Alg6.

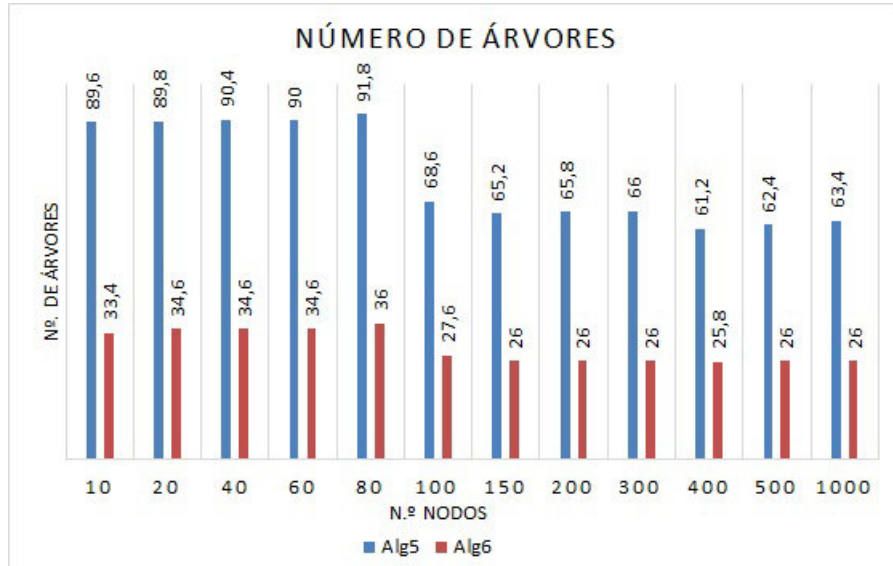


Figura 4.15: Número médio de árvores de suporte obtidas pelos Algoritmos Alg5 e Alg6 nas instâncias E.

O tempo de execução do Algoritmo Alg6 é inferior ao tempo de execução do Algoritmo Alg5 tal como ocorreu nos outros dois grupos de instâncias, o que se explica pelo número médio de árvores obtidas no Algoritmo Alg5 ser mais do que o dobro do Algoritmo Alg6 (Figura 4.15).

Notamos que nas 9 instâncias de 500 e 1000 nodos do grupo de instâncias Euclidianas onde não tinha sido possível encontrar um valor para o limite superior através do Procedimento P-WMTZ+C, com a aplicação dos vários algoritmos Lagrangeanos foi possível obter um valor para o limite superior para o custo e em 6 instâncias de 400 e 500 nodos foi possível encontrar um valor para o limite superior para o custo melhor do que o obtido através da aplicação do Procedimento P-WMTZ+C.

4.5.4 Síntese dos Resultados Computacionais

Os resultados computacionais mostram que todos os algoritmos Lagrangeanos apresentados demoram menos de 13 segundos. Pelo que se conclui serem bastante rápidos na obtenção do valor da relaxação Lagrangeana e de um limite superior para o Problema WMST.

Os Algoritmos Alg2, Alg5, Alg6 e Alg7 revelaram ser bastante eficientes em todos os grupos de instâncias, pois com a aplicação destes algoritmos foram obtidas algumas soluções ótimas e outras muito próximas das soluções ótimas. Destes o Algoritmo Alg5 tem a desvantagem de demorar mais tempo.

Os valores obtidos pela relaxação Lagrangeana dos Algoritmos Alg2, Alg5 e Alg6 são iguais aos da relaxação linear do Procedimento P-WMTZ+C apresentado no Capítulo 3. O Algoritmo Alg7 obtém melhores valores da relaxação Lagrangeana comparativamente à relaxação linear do Procedimento P-WMTZ+C.

A relaxação Lagrangeana nos grupos de instâncias Aleatórias e Euclidianas apresenta soluções mais próximas da solução ótima, enquanto que no grupo de instâncias Quase Caminhos é preferível encontrar um valor para o limite superior para o custo, pois este encontra-se mais próximo do valor ótimo.

Capítulo 5

Desigualdades Válidas

Com o propósito de fortalecer as formulações apresentadas no Capítulo 3, neste capítulo discutimos classes de desigualdades válidas para o Problema WMST. As desigualdades válidas podem ser incorporadas no modelo usando um algoritmo de planos de corte. Assim sendo é de interesse desenvolver classes de desigualdades válidas fortes de modo a tornarem o algoritmo de planos de corte eficiente.

Neste capítulo começamos por apresentar uma pequena introdução sobre desigualdades válidas, particularizando ao caso das desigualdades de saco-mochila. Na Secção 5.1 fazemos uma abordagem às conhecidas Desigualdades de Cobertura para o Problema Saco-mochila Binário adaptando-as para o caso do Problema WMST. Na Secção 5.2 apresentamos novas desigualdades válidas que consideram simultaneamente a estrutura de árvore de suporte e a estrutura de saco-mochila. Designámos estas novas desigualdades por Desigualdades de Cobertura Implícita. Tendo por base o processo usado para obter as conhecidas Desigualdades de Cobertura Estendidas, propomos na Secção 5.3 estender as Desigualdades de Cobertura Implícita, formando as Desigualdades de Cobertura Implícita Estendida. Na Secção 5.4 fortalecemos as desigualdades das secções anteriores usando a técnica de levantamento de variáveis e apresentamos as Desigualdades de Cobertura Implícita Levantada. Por fim, na Secção 5.5 propomos uma generalização das desigualdades que podem fortalecer ainda mais as formulações, sendo estas baseadas na fixação de conjuntos de variáveis.

Dada uma desigualdade

$$\pi x \leq \pi_0, \quad (5.1)$$

diz-se que é válida para um conjunto $X_{WMST} \subseteq \mathbb{R}^{|A|}$ se ela for verificada por todos os pontos $x \in X_{WMST}$.

Para o caso do Problema WMST, o conjunto das soluções admissíveis X_{WMST} , pode ser visto como a interseção de dois conjuntos conhecidos: $X_{WMST} = X_T \cap X_K$, onde X_T é o conjunto formado pelas árvores de suporte e X_K é o conjunto saco-mochila binário definido pelos pontos $x \in \mathbb{R}^{|A|}$ tais que

$$\sum_{(i,j) \in A} w_{ij} x_{ij} \leq W, \quad (5.2)$$

e,

$$x_{ij} \in \{0, 1\}, \quad (i, j) \in A. \quad (5.3)$$

As desigualdades válidas para X_T e para X_K são também válidas para X_{WMST} . Enquanto que a descrição poliédrica do invólucro convexo de X_T , $P_T = \text{conv}(X_T)$, é bem conhecida (ver, por exemplo [45]), para a caracterização poliédrica do invólucro convexo dos conjuntos saco-mochila $P_K = \text{conv}(X_K)$, só são conhecidas descrições parciais. Este poliedro, é provavelmente, um dos poliedros que ocorre na Otimização Combinatória mais estudado.

Teoricamente as desigualdades válidas mais fortes são aquelas que definem facetas de P_K . Seja,

$$F = \left\{ x \in P_K : \sum_{(i,j) \in A} \pi_{ij} x_{ij} = \pi_0 \right\}.$$

A desigualdade (5.1) diz-se que define face de P_K se satisfaz a igualdade do conjunto F ($F \neq \emptyset$) e diz-se que define uma faceta de P_K se define uma face de P_K e se $\dim(F) = \dim(P_K) - 1$, onde $\dim(P_K)$ é a dimensão de P_K .

Como, em geral, $P_{WMST} = \text{conv}(X_{WMST})$ está estritamente contido em $P_T \cap P_K$, existem soluções fracionárias que não podem ser cortadas por desigualdades válidas derivadas para P_T ou P_K . Por esse motivo, propomos também o uso de desigualdades válidas derivadas para P_{WMST} que tomam em consideração propriedades dos dois conjuntos simultaneamente.

5.1 Desigualdades de Cobertura (DC)

As Desigualdades de Cobertura foram introduzidas independentemente por Balas [10], Hammer et al. [32] e Wolsey [62] no contexto do Problema Saco-mochila Binário, mas podem ser aplicadas a qualquer Problema Linear Binário.

Para escrever o que se segue usamos as referências [10, 32, 62, 63]. Dado o conjunto E , constituído por todas as arestas do grafo $G = (V, E)$ define-se uma cobertura do seguinte modo.

Definição 5.1. Cobertura

O conjunto $S \subset E$ é uma Cobertura, se a soma dos pesos das arestas contidas em S for superior a W , ou seja, se se verificar a seguinte desigualdade

$$\sum_{e \in S} w_e > W.$$

Definição 5.2. Cobertura Minimal

O conjunto $S \subset E$ é uma Cobertura Minimal, se ao removermos qualquer aresta f da Cobertura S , a soma dos pesos das restantes arestas contidas na cobertura for inferior ou igual a W , ou seja,

$$\sum_{e \in S \setminus \{f\}} w_e \leq W.$$

Proposição 5.1.

Se o conjunto $S \subset E$ é uma Cobertura então, a seguinte desigualdade

$$\sum_{e \in S} x_e \leq |S| - 1, \tag{5.4}$$

é válida para o conjunto de soluções admissíveis $X_{W_{MST}}$ e é chamada de Desigualdade de Cobertura (DC).

A prova da proposição anterior pode ser consultada em [63].

Se considerarmos duas Coberturas S_1 e S_2 com $S_1 \subset S_2$, em que S_1 é uma Cobertura Minimal, as correspondentes desigualdades de cobertura são dadas por

$$\sum_{e \in S_1} x_e \leq |S_1| - 1 \quad \text{e} \quad \sum_{e \in S_2} x_e \leq |S_2| - 1.$$

Dado que $S_1 \subset S_2$ também se verifica que $|S_1| < |S_2|$ e assim a primeira desigualdade é mais forte do que a segunda. Note-se que a desigualdade $\sum_{e \in S_2} x_e \leq |S_2| - 1$ pode ser obtida como a soma de $\sum_{e \in S_1} x_e \leq |S_1| - 1$ com as desigualdades $x_e \leq 1$, para todo $e \in S_2 \setminus S_1$. Portanto, tal como observado em [10, 32, 62], as desigualdades de cobertura são mais fortes quando a cobertura considerada for a cobertura minimal.

Para qualquer subconjunto $S \subset E$, $|S| = s$, considera-se

$$P_S = \text{Conv} \left\{ x \in \{0, 1\}^s : \sum_{e \in S} w_e x_e \leq W \right\}.$$

Se o conjunto S é uma Cobertura Minimal, então a desigualdade (5.4) define uma faceta de P_S (ver, por exemplo, [10, 11, 32, 62]).

Exemplo 5.1.

Considere-se o Exemplo 2.1 de 5 nodos definido no Capítulo 2 onde o conjunto $E = \{\{0, 1\}, \{0, 2\}, \{0, 3\}, \{0, 4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}\}$.

Considerando, por exemplo, os seguintes conjuntos $S_1 = \{\{0, 1\}, \{0, 3\}, \{0, 4\}, \{1, 2\}\}$, $S_2 = \{\{0, 1\}, \{0, 4\}, \{1, 2\}, \{3, 4\}\}$ e $S_3 = \{\{0, 1\}, \{0, 4\}, \{1, 2\}, \{2, 3\}\}$.

O conjunto S_1 é uma Cobertura, pois a soma dos pesos das arestas do conjunto S_1 é maior do que 20 ($w_{01} + w_{03} + w_{04} + w_{12} = 6 + 8 + 7 + 6 = 27 > 20$), mas não é uma Cobertura Minimal dado que se removermos a aresta $\{1, 2\}$ da Cobertura S_1 a soma dos pesos das restantes arestas contidas na cobertura não é inferior ou igual a 20 ($w_{01} + w_{03} + w_{04} = 6 + 8 + 7 = 21 > 20$). Assim, $x_{01} + x_{03} + x_{04} + x_{12} \leq 3$ é uma Desigualdade de Cobertura, onde a Cobertura S_1 não é minimal.

Os conjuntos S_2 e S_3 são também Coberturas. A soma dos pesos das arestas do conjunto S_2 é $w_{01} + w_{04} + w_{12} + w_{34} = 6 + 7 + 6 + 5 = 24 > 20$ e para o conjunto S_3 é $w_{01} + w_{04} + w_{12} + w_{23} = 6 + 7 + 6 + 4 = 23 > 20$. Em ambas as Coberturas S_2 e S_3 se removermos qualquer aresta, a soma dos pesos das restantes arestas da cobertura será menor do que 20, logo ambos os conjuntos S_2 e S_3 são Coberturas Minimais.

Por conseguinte, dado que S_2 e S_3 são Coberturas Minimais, podemos construir as seguintes Desigualdades de Cobertura

$$x_{01} + x_{04} + x_{12} + x_{34} \leq 3 \quad \text{e} \quad x_{01} + x_{04} + x_{12} + x_{23} \leq 3, \quad (5.5)$$

que são válidas no conjunto das soluções admissíveis $X_{W_{MST}}$.

As coberturas podem assim incluir várias combinações de variáveis, contudo algumas coberturas não cortam a solução fracionária. Por exemplo, as desigualdades formadas pelas Coberturas S_1 e S_3 não cortam a solução da relaxação linear dada no Exemplo 3.2, isto é, se substituirmos nas desigualdades (5.5) o valor das variáveis da relaxação linear obtemos $3 \leq 3$ e $2,2 < 3$, respetivamente, sendo 3 o valor do segundo membro das desigualdades, pelo que estas coberturas não cortam a solução da relaxação linear.

Note-se que, com as arestas de cada uma das três coberturas apresentadas no Exemplo 5.1, é possível formar três árvores de suporte diferentes, mas como se pode ver no Exemplo 5.2 a seguir, nem sempre é possível formar uma árvore de suporte com as arestas de uma cobertura.

Exemplo 5.2.

Usando novamente o Exemplo 2.1 de 5 nodos podemos considerar a Cobertura $S_4 = \{\{0, 1\}, \{0, 3\}, \{0, 4\}, \{3, 4\}\}$ (para o qual $w_{01} + w_{03} + w_{04} + w_{34} = 6 + 8 + 7 + 5 = 26 > 20$). Se representarmos num grafo estas arestas, podemos observar na Figura 5.1 que as arestas da Cobertura S_4 não formam uma árvore de suporte.

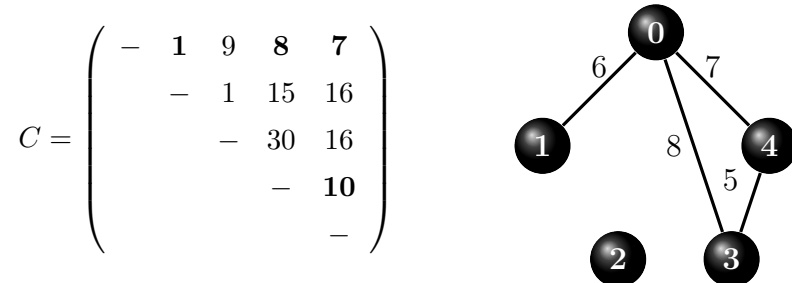


Figura 5.1: Representação das arestas da Cobertura S_4 no grafo.

5.2 Desigualdades de Cobertura Implícita (DCI)

Com o objetivo de tornar as desigualdades de cobertura mais fortes, vamos introduzir uma nova classe de desigualdades válidas que toma em consideração propriedades dos conjuntos P_T e P_K simultaneamente. Para obter a nova classe de desigualdades vamos associar às desigualdades de cobertura alguma informação proveniente da estrutura em árvore.

Definição 5.3. Cobertura Implícita

A um conjunto $S \subset E$, tal que as arestas de S não formam ciclos, chama-se Cobertura Implícita (CI) se para cada árvore de suporte $T = (V, E_T)$ tal que $S \subset E_T$ se verifica a seguinte desigualdade

$$W(T) = \sum_{e \in E_T} w_e > W.$$

Para verificar se o conjunto S define uma CI, consideramos as arestas do conjunto S e, sucessivamente, adicionamos arestas de $E \setminus S$ até formarmos uma árvore de suporte de peso mínimo, T . No caso da árvore de suporte T , que contém as arestas de S , não verificar a restrição de peso, então o conjunto S é uma CI. Note-se que usamos a árvore T apenas para validar a Cobertura Implícita S .

Definição 5.4. Cobertura Implícita Minimal

O conjunto $S \subset E$, tal que as arestas de S não formam ciclos, é uma Cobertura Implícita Minimal, se ao removermos qualquer aresta f da Cobertura Implícita S , o peso da árvore de menor peso que contém as arestas de $S \setminus \{f\}$ for inferior ou igual a W , ou seja,

$$\sum_{e \in E_T \setminus \{f\}} w_e \leq W.$$

As Desigualdades de Cobertura Implícita generalizam as conhecidas Desigualdades de Cobertura, nos casos em que as coberturas não formam ciclos.

Proposição 5.2.

Se o conjunto $S \subset E$ é uma CI, então a desigualdade

$$\sum_{e \in S} x_e \leq |S| - 1, \quad (5.6)$$

é válida para o conjunto de soluções admissíveis X_{WMST} e é chamada de Desigualdade de Cobertura Implícita (DCI).

Demonstração.

Suponhamos que $x^* \in X_{WMST}$ é uma solução que corresponde a uma árvore de suporte T_Q ($x_j^* = 1$ se $j \in Q$ e $x_j^* = 0$ se $j \notin Q$) e não satisfaz a DCI, então temos de mostrar que $x^* \notin X_{WMST}$.

Se x^* não satisfaz a DCI, então $\sum_{e \in S} x_e^* \geq |S|$, logo $|Q \cap S| = |S|$ e assim $S \subseteq Q$. Como S é uma CI, então para validar a Cobertura Implícita S foi utilizada uma árvore de suporte de peso mínimo, T , tal que $S \subset E_T$.

Como $\sum_{e \in Q} w_e x_e^* = \sum_{e \in Q} w_e \geq \sum_{e \in E_T} w_e > W$ e assim $x^* \notin X_{WMST}$.

□

Quando S é uma CI com um único elemento, $S = \{e\}$, a DCI correspondente é simplesmente $x_e = 0$. Estas desigualdades particulares podem ser usadas num procedimento inicial de preprocessamento para reduzir o tamanho do modelo.

Exemplo 5.3.

No Exemplo 2.1 podemos considerar o conjunto $S = \{\{0, 1\}, \{0, 4\}, \{1, 2\}\}$, cuja árvore correspondente está representada pelas arestas a vermelho na Figura 5.2, onde podemos observar que as arestas de S não formam ciclos. Podemos adicionar à árvore a aresta de peso mínimo $\{1, 3\}$. Desta forma completamos a árvore, obtendo a árvore de suporte T , onde $E_T = \{\{0, 1\}, \{0, 4\}, \{1, 2\}, \{1, 3\}\}$ que corresponde a uma solução não admissível com peso 21. Dado que o peso da árvore é $21 > 20$, o conjunto S é

uma CI. Se removermos qualquer aresta de S , o peso da árvore resultante é inferior a $W = 20$ e portanto o conjunto S define uma Cobertura Implícita Minimal.

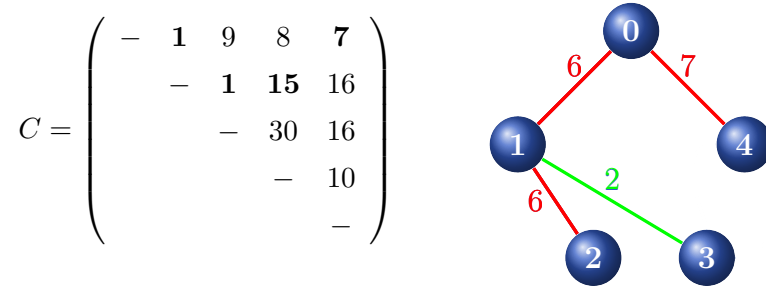


Figura 5.2: Árvore de suporte com custo 24 e peso 21.

Assim, usando as arestas do conjunto S podemos construir a seguinte DCI

$$x_{01} + x_{04} + x_{12} \leq 2 \quad (5.7)$$

válida para o conjunto X_{WMST} das soluções admissíveis. Se considerarmos a solução da relaxação linear dada no Exemplo 3.2 verificamos que esta desigualdade corta a solução fracionária $(2, 2 > 2)$.

Comparação entre DC e DCI

De seguida apresentamos um exemplo para mostrar as diferenças existentes entre a Desigualdade de Cobertura e a Desigualdade de Cobertura Implícita.

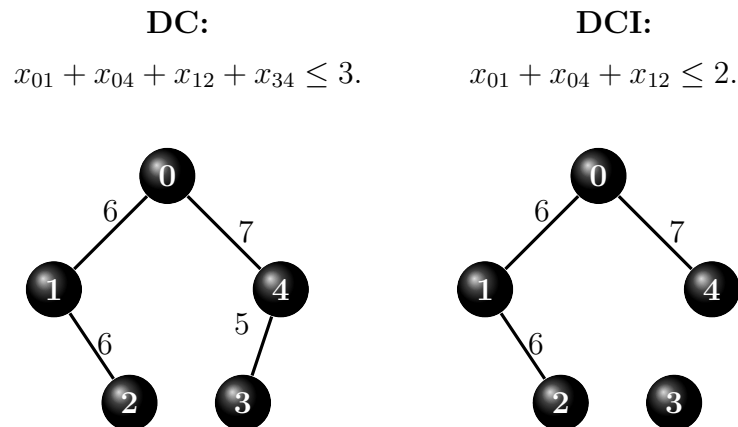


Figura 5.3: Desigualdade de Cobertura e Desigualdade de Cobertura Implícita.

Na Figura 5.3 podemos observar que nas DCIs apenas usamos a estrutura de árvore para validar a cobertura, não sendo o conjunto S constituído por todas as arestas da árvore de suporte, enquanto que nas DCs o conjunto S pode ou não ser formado pelo conjunto de arestas da árvore de suporte (no Exemplo 5.2 podemos ver que a cobertura aí considerada não era formada pelo conjunto de arestas da árvore de suporte). Ambas as coberturas representadas são minimais, a DCI é mais forte do que a DC, pois a desigualdade (5.7) e $x_{34} \leq 1$ implicam a desigualdade $x_{01} + x_{04} + x_{12} + x_{34} \leq 3$ obtida através da Cobertura S_2 no Exemplo 5.1. Com este exemplo, constatamos que a estrutura em árvore é importante na construção de desigualdades válidas, permitindo a obtenção de desigualdades mais fortes que as desigualdades de cobertura.

5.3 Desigualdades de Cobertura Implícita Estendida (DCIE)

Segundo Balas [10] podemos obter desigualdades mais fortes se estendermos as desigualdades de cobertura. De modo semelhante podemos também obter desigualdades mais fortes se estendermos as desigualdades de cobertura implícita.

O seguinte resultado, pode ser consultado em [10, 62, 63], sendo a prova análoga à da Proposição 5.1 (ver [63]).

Proposição 5.3.

Seja $S \subset E$ uma Cobertura e seja

$$S' = \left\{ e \in E \setminus S : w_e \geq \max\{w_f : f \in S\} \right\}.$$

A seguinte desigualdade

$$\sum_{e \in S} x_e + \sum_{e \in S'} x_e \leq |S| - 1$$

é válida para o conjunto de soluções admissíveis X_{WMST} e é chamada de Desigualdade de Cobertura Estendida (DCE).

O facto de estas desigualdades serem válidas para $X_{W_{MST}}$ e dominarem (se $S' \neq \emptyset$) as desigualdades de cobertura não garante que definam facetas.

Há que ter atenção à forma de realizar a extensão, pois uma extensão da DCI pode conduzir a desigualdades não válidas tal como podemos observar no seguinte exemplo.

Exemplo 5.4.

Neste exemplo as matrizes de pesos e custos são dadas por:

$$H = \begin{pmatrix} - & 6 & \mathbf{7} & 8 & 7 \\ & - & 6 & 2 & 2 \\ & & - & 4 & 5 \\ & & & - & 5 \\ & & & & - \end{pmatrix} \quad \text{e} \quad C = \begin{pmatrix} - & 1 & 9 & 8 & 7 \\ & - & 1 & 15 & 16 \\ & & - & 30 & 16 \\ & & & - & 10 \\ & & & & - \end{pmatrix}.$$

O exemplo apresentado foi adaptado do Exemplo 2.1, com uma única alteração na matriz de pesos, o peso da aresta $\{0, 2\}$ alterou para $w_{02} = 7$. A solução da relaxação linear e a solução ótima não sofrem alteração (ver Exemplos 3.2 e 2.1, respetivamente).

Considere-se a DCI, $x_{12} + x_{24} + x_{34} \leq 2$. Seja $\theta = \max\{w_{12}, w_{24}, w_{34}\} = \max\{6, 5, 5\} = 6$ e como $w_{01} = 6 \geq \theta, w_{04} = 7 \geq \theta, w_{02} = 7 \geq \theta$ e $w_{03} = 8 \geq \theta$, o conjunto $S' = \{\{0, 1\}, \{0, 2\}, \{0, 3\}, \{0, 4\}\}$. Logo, a desigualdade obtida é dada por

$$x_{12} + x_{24} + x_{34} + \mathbf{x_{01}} + \mathbf{x_{02}} + \mathbf{x_{03}} + \mathbf{x_{04}} \leq 2,$$

que é uma desigualdade não válida. Tal acontece pois, por exemplo, a solução ótima desta instância do problema corresponde à árvore de suporte da Figura 2.2. Esta árvore não verifica a desigualdade acima, pois para esta árvore teremos $x_{12} + x_{24} + x_{34} + x_{01} + x_{02} + x_{03} + x_{04} = 1 + 0 + 1 + 1 + 0 + 0 + 0 = 3 > 2$, sendo 2 o valor do segundo membro da desigualdade.

A extensão que é efetuada das desigualdades de cobertura é baseada apenas nos valores dos pesos das arestas. Para estender as desigualdades de cobertura implícita, vamos ter de considerar não só os pesos das arestas mas, também, a informação relativa à estrutura de árvore. Notamos que, mais uma vez, é necessário usar informação relativa à estrutura da árvore para desenvolver desigualdades válidas para este problema.

Proposição 5.4.

Seja $S \subset E$ uma CI e seja

$$S' = \left\{ e \in E \setminus S : w_e \geq \max\{w_f : f \in S\} \text{ e as arestas de } S \cup \{e\} \text{ formam ciclo} \right\}.$$

A Desigualdade de Cobertura Implícita Estendida (DCIE)

$$\sum_{e \in S} x_e + \sum_{e \in S'} x_e \leq |S| - 1$$

é válida para o conjunto de soluções admissíveis X_{WMST} .

Demonstração.

Suponhamos que existem conjuntos S e S' nas condições dadas na Proposição 5.4. Seja $x^* \in X_{WMST}$ uma solução que corresponde a uma árvore de suporte, T , e que não satisfaz a DCIE, isto é,

$$\sum_{e \in S} x_e + \sum_{e \in S'} x_e \geq |S|.$$

Seja $\bar{S} = \{j \in S : x_j^* = 1\}$ e $\bar{S}' = \{j \in S' : x_j^* = 1\}$. Então $\sum_{j \in (S \cup S')} x_j^* = |\bar{S}| + |\bar{S}'|$. Por construção do conjunto S' , na solução x^* a soma de variáveis fixas a um (nos conjuntos S e S') não pode exceder $|S|$ e, portanto, $|\bar{S}| + |\bar{S}'| = |S|$. Seja ainda $E_T = \{j \in E : x_j^* = 1\}$ o conjunto de arestas da árvore T que não satisfaz a DCIE e $R = E_T \setminus (\bar{S} \cup \bar{S}')$. Considere-se o grafo conexo $S \cup T$. Como $\bar{S} \subseteq S$ e os elementos de \bar{S}' formam ciclos com os elementos de S , a árvore de suporte deste grafo que contém o conjunto S é T' tal que $E_{T'} = S \cup R$ e

$$\sum_{j \in E} w_j x_j^* = \sum_{j \in \bar{S} \cup \bar{S}' \cup R} w_j \geq \sum_{j \in S \cup R} w_j > W,$$

onde a primeira desigualdade resulta de $|\bar{S}| + |\bar{S}'| = |S|$ e da definição dos pesos de S' e a segunda desigualdade resulta da suposição de que S é uma CI. Logo $x^* \notin X_{WMST}$, o que contradiz a suposição inicial. □

Vamos tentar estender a DCI $x_{01} + x_{04} + x_{12} \leq 2$ obtida no Exemplo 5.3. Se considerarmos as arestas que formam ciclo com as arestas da CI, ou seja, as arestas $\{1, 4\}$, $\{0, 2\}$ e $\{2, 4\}$ (ver Figura 5.4) nenhuma tem peso superior a 7 (sendo $7 = \max\{w_{01}, w_{04}, w_{12}\}$) pelo que não é possível obter uma DCIE para este exemplo.

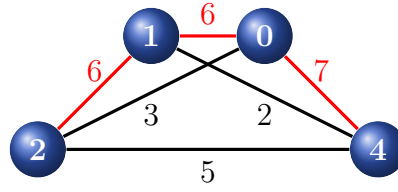


Figura 5.4: Ciclos formados com as arestas da Cobertura Implícita S do Exemplo 5.3 .

Assim, quando o conjunto $S' = \emptyset$, temos apenas uma DCI. De seguida apresentamos um exemplo em que $S' \neq \emptyset$.

Exemplo 5.5.

Considere-se o Exemplo 5.4 e a DCI, $x_{01} + x_{04} + x_{12} \leq 2$. A aresta $\{0, 2\}$ forma ciclo com as arestas da CI e $w_{02} = \max\{w_{01}, w_{04}, w_{12}\} = 7$, logo $S' = \{\{0, 2\}\}$. Incluindo esta variável correspondente a esta aresta na DCI obtemos a DCIE

$$x_{01} + x_{04} + x_{12} + \mathbf{x}_{02} \leq 2,$$

que é válida para o conjunto X_{WMSST} das soluções admissíveis.

5.4 Desigualdades de Cobertura Implícita Levantada (DCIL)

As desigualdades discutidas nas três secções anteriores podem ser fortalecidas através da aplicação de uma técnica designada por levantamento (*lifting*). Este processo é efetuado por introdução de novas variáveis na desigualdade ou pela modificação de coeficientes existentes na desigualdade que permitam fortalecer a desigualdade. Gomory [22] foi o primeiro a utilizar a técnica de levantamento.

Ao adicionarmos variáveis ao lado esquerdo da desigualdade de cobertura implícita obtém-se a chamada Desigualdade de Cobertura Implícita Levantada.

Definição 5.5. *Desigualdade de Cobertura Implícita Levantada*

Dada uma DCI $\sum_{e \in S} x_e \leq |S| - 1$, a seguinte desigualdade válida

$$\sum_{e \in S} x_e + \sum_{e \in E \setminus S} \beta_e x_e \leq |S| - 1,$$

com $\beta_e \geq 0$, para todas as arestas $e \in E \setminus S$, é chamada de Desigualdade de Cobertura Implícita Levantada (DCIL).

As Desigualdades de Cobertura Implícita Estendida, abordadas na secção anterior, são casos particulares das Desigualdades de Cobertura Implícita Levantada em que apenas efetuamos o levantamento de algumas variáveis, tendo essas variáveis coeficiente um.

Para incluir outras variáveis na desigualdade válida com o objetivo de a fortalecer, podem ser encontradas na literatura duas técnicas: a técnica de levantamento sequencial de variáveis e a técnica de levantamento simultâneo de variáveis. Nesta tese apenas se aborda a técnica de levantamento sequencial onde os coeficientes β_e são introduzidos um a um de forma sequencial e sendo a sequência de levantamento das variáveis pré-estabelecida.

Vamos de seguida ver como determinar os coeficientes das variáveis a efetuar levantamento.

Dada a DCI $\sum_{e \in S} x_e \leq |S| - 1$ e a DCIL $\sum_{e \in S} x_e + \sum_{e \in R} \beta_e x_e \leq |S| - 1$, onde $R \subseteq E \setminus (S \cup \{f\})$ é o conjunto de variáveis que já foram sujeitas a levantamento, podemos fazer o levantamento da variável x_f , com $f \in E \setminus (S \cup R)$, calculando o coeficiente β_f tal que $0 \leq \beta_f \leq g(S, R, \beta)$, onde

$$g(S, R, \beta) = \min \left\{ |S| - 1 - \sum_{e \in S} x_e - \sum_{e \in R} \beta_e x_e : x \in X_{WMST}, x_f = 1 \right\}. \quad (5.8)$$

Caso o problema de minimização seja inadmissível atribuiremos a $g(S, R, \beta)$ o valor zero.

Proposição 5.5.

Dada a DCI $\sum_{e \in S} x_e \leq |S| - 1$ e a DCIL $\sum_{e \in S} x_e + \sum_{e \in R} \beta_e x_e \leq |S| - 1$, onde $R \subseteq E \setminus (S \cup \{f\})$ é o conjunto de variáveis que já foram sujeitas a levantamento, então

$$\sum_{e \in S} x_e + \sum_{e \in R} \beta_e x_e + \beta_f x_f \leq |S| - 1, \quad (5.9)$$

é válida para o conjunto das soluções admissíveis X_{WMST} se $\beta_f \leq g(S, R, \beta)$.

Demonstração.

Se $x_f = 0$, temos que a desigualdade (5.9) é a DCIL que é válida para o conjunto das soluções admissíveis X_{WMST} . No caso em que $x_f = 1$ na desigualdade (5.9) ficamos com

$$\beta_f \leq |S| - 1 - \sum_{e \in S} x_e - \sum_{e \in R} \beta_e x_e,$$

pelo que a sua validade para o conjunto das soluções admissíveis X_{WMST} resulta da própria definição da função $g(S, R, \beta)$.

□

Exemplo 5.6.

Consideremos a DCI $x_{01} + x_{04} + x_{12} \leq 2$ obtida no Exemplo 5.3. Para fazer o levantamento da variável x_{03} calcula-se o seu coeficiente β_{03} da seguinte forma:

$$\beta_{03} = \min\{2 - x_{01} - x_{04} - x_{12} : x \in X_{WMST}, x_{03} = 1\} = 1.$$

Usando o levantamento sequencial exato para o cálculo dos restantes coeficientes verificamos que estes são todos nulos. A Desigualdade de Cobertura Implícita Levantada é dada por:

$$x_{01} + x_{04} + x_{12} + \mathbf{x}_{03} \leq 2.$$

Proposição 5.6.

Se β_e é inteiro para todo $e \in R$, então $g(S, R, \beta)$ em 5.8 é inteiro.

Demonstração.

Admitamos que o conjunto das soluções admissíveis do problema de minimização considerado em 5.8 não é vazio. Se as variáveis desse problema são binárias (inteiras) e os coeficientes da função objetivo, os valores dos $\beta_e, e \in R$, são inteiros, então o seu valor ótimo tem de ser inteiro.

□

Note-se que diferentes sequências de levantamento de variáveis podem produzir diferentes DCILs. Tal também se verifica com as Desigualdades de Cobertura Levantadas (ver Balas [10] e Wolsey [62]).

Exemplo 5.7.

Neste exemplo as matrizes de pesos e custos são dadas por:

$$H = \begin{pmatrix} - & 7 & 7 & 5 & 5 \\ & - & 5 & 8 & 5 \\ & & - & 6 & 7 \\ & & & - & 5 \\ & & & & - \end{pmatrix} \quad \text{e} \quad C = \begin{pmatrix} - & 1 & 9 & 8 & 7 \\ & - & 1 & 15 & 16 \\ & & - & 30 & 16 \\ & & & - & 10 \\ & & & & - \end{pmatrix}.$$

Consideremos $W = 20$ e a DCI, $x_{01} + x_{02} + x_{03} \leq 2$. Se pretendermos efetuar o levantamento das variáveis x_{04} e x_{12} , o cálculo dos seus coeficientes pode ser efetuado segundo duas sequências diferentes. Podemos calcular primeiro o coeficiente da variável x_{04} e depois o da variável x_{12} e fazemos,

$$\beta_{04} = \min\{2 - x_{01} - x_{02} - x_{03} : x \in X_{WMST}, x_{04} = 1\} = 1.$$

$$\beta_{12} = \min\{2 - x_{01} - x_{02} - x_{03} - x_{04} : x \in X_{WMST}, x_{12} = 1\} = 0.$$

Deste modo obtém-se a seguinte DCIL

$$x_{01} + x_{02} + x_{03} + \mathbf{x}_{04} \leq 2.$$

Se calcularmos primeiro o coeficiente da variável x_{12} e depois o da variável x_{04} fazemos,

$$\beta_{12} = \min\{2 - x_{01} - x_{02} - x_{03} : x \in X_{WMST}, x_{12} = 1\} = 1.$$

$$\beta_{04} = \min\{2 - x_{01} - x_{02} - x_{03} - x_{12} : x \in X_{WMST}, x_{04} = 1\} = 0.$$

Usando esta segunda sequência obtemos a DCIL

$$x_{01} + x_{02} + x_{03} + \mathbf{x}_{12} \leq 2.$$

Verificamos assim que utilizando diferentes sequências de levantamento de variáveis é possível obter diferentes Desigualdades de Cobertura Implícita Levantada.

Gerar DCILs usando o levantamento sequencial exato envolve a resolução de muitos subproblemas o que pode ser computacionalmente dispendioso. Então, uma questão que surge é como calcular os coeficientes β_e de tal forma que não se torne computacionalmente dispendioso, ou seja, procuram-se métodos mais rápidos e simples pois, o cálculo de forma eficiente de coeficientes é um elemento crucial para o uso com sucesso de DCILs.

Observemos que em vez de obter o valor ótimo de $g(S, R, \beta)$ podemos usar uma relaxação de $g(S, R, \beta)$. Testamos duas relaxações, a relaxação linear e a relaxação Lagrangeana onde o conjunto X_{WMST} é substituído pelo conjunto de soluções admissíveis da sua relaxação linear, X_L :

$$g_L(S, R, \beta) = \min \left\{ |S| - 1 - \sum_{e \in S} x_e - \sum_{e \in R} \beta_e x_e : x \in X_L, x_f = 1 \right\}.$$

Caso o problema de minimização seja inadmissível atribuiremos a $g(S, R, \beta)$ o valor zero.

De testes computacionais realizados com os grupos de instâncias Quase Caminhos, Aleatórias e Euclidianas verificámos que o uso da relaxação linear para determinar os coeficientes de forma sequencial torna-se bastante lento. Deste modo, no Capítulo 6 vamos apresentar um algoritmo heurístico para o cálculo dos coeficientes de levantamento usual, *down-lifting* e *up-lifting* baseado em relaxação Lagrangeana.

Exemplo 5.8.

Dada a DCI $x_{01} + x_{04} + x_{12} \leq 2$ obtida no Exemplo 5.3. Podemos efetuar o levantamento das variáveis usando o levantamento sequencial aproximado baseado na relaxação linear ou na relaxação Lagrangeana. O coeficiente β_{03} é obtido da seguinte forma:

$$\beta_{03} = \min\{2 - x_{04} - x_{01} - x_{12} : x \in X_L, x_{03} = 1\} = 0,5$$

e os restantes coeficientes são todos nulos. Assim, a DCIL é dada por:

$$x_{01} + x_{04} + x_{12} + 0,5x_{03} \leq 2.$$

Se arredondarmos o valor do coeficiente da variável x_{03} para 1, usando a Proposição 5.6, obtemos a mesma desigualdade que obtivemos usando o levantamento sequencial exato (ver Exemplo 5.6).

5.5 Desigualdades obtidas por Fixação de Variáveis

Quando se introduzem desigualdades levantadas não existem conjuntos de variáveis fixas à priori. Contudo, podem ser criadas novas classes de desigualdades de levantamento generalizadas derivadas do procedimento usual de levantamento, onde o levantamento das variáveis é efetuado fixando à priori o valor de conjuntos de variáveis. Nesta tese utilizamos três estratégias diferentes de fixação de variáveis que originam três tipos de desigualdades. Na primeira começamos por fixar apenas um conjunto de variáveis a um, na segunda apenas um conjunto de variáveis a zero e por último fixamos dois conjuntos de variáveis em simultâneo, um conjunto de variáveis fixas a zero e um conjunto de variáveis fixas a um. Uma questão importante é a escolha do conjunto ou conjuntos de variáveis a fixar à priori.

5.5.1 Desigualdades de Cobertura Implícita Levantada por *Down-Lifting* (DCILDL)

Para obter Desigualdades de Cobertura Implícita Levantada através do designado *Down-Lifting* é necessário começar por fixar um conjunto de variáveis a um. Designamos por E_1 o conjunto de variáveis a fixar a um e por X^{E_1} o conjunto restrito definido da seguinte forma

$$X^{E_1} = X_{WMST} \cap \{x : x_f = 1, f \in E_1\},$$

onde as variáveis $f \in E_1$ encontram-se fixas a um, $x_f = 1$. Notamos que o conjunto X^{E_1} é uma restrição do conjunto X_{WMST} .

De seguida vamos definir uma cobertura implícita para o conjunto restrito X^{E_1} . Vamos assumir $X^{E_1} \neq \emptyset$.

Definição 5.6. *Cobertura Implícita para o conjunto restrito X^{E_1}*

O conjunto $S \subset E \setminus E_1$, cujas arestas de $S \cup E_1$ não formam ciclos, é uma Cobertura Implícita para o conjunto restrito X^{E_1} , se para cada árvore de suporte $T = (V, E_T)$ tal que $S \subset E_T$ e $E_1 \subset E_T$ se verifica a seguinte desigualdade

$$W(T) = \sum_{e \in E_T} w_e > W.$$

Para verificar que $S \subset E \setminus E_1$ é uma CI para o conjunto restrito X^{E_1} , construímos uma sub-árvore com as arestas do conjunto S e com as arestas do conjunto E_1 , e sucessivamente, adicionamos arestas de $E_f = E \setminus (S \cup E_1)$ à sub-árvore até formarmos uma árvore de suporte de peso mínimo, T , contendo as arestas dos conjuntos S e E_1 . No caso da árvore de suporte T , não verificar a restrição de peso, então S é uma CI para o conjunto restrito X^{E_1} .

Proposição 5.7.

Se $S \subset E \setminus E_1$ é uma CI para o conjunto restrito X^{E_1} , então a Desigualdade de Cobertura Implícita

$$\sum_{e \in S} x_e \leq |S| - 1$$

é válida para o conjunto restrito X^{E_1} .

Demonstração.

Seja E_1 o conjunto de variáveis fixas a 1 e $x^* \in X^{E_1} = X_{WMST} \cap \{x : x_f = 1, f \in E_1\}$ uma solução que corresponde a uma árvore de suporte, T_Q ($x_e^* = 1$ se $e \in Q$ e $x_e^* = 0$ se $e \notin Q$), notamos que as arestas de E_1 estão na árvore T_Q .

Suponhamos que a solução x^* correspondente à árvore T_Q não satisfaz a DCI, então $\sum_{e \in S} x_e^* \geq |S|$, logo $|Q \cap S| = |S|$ e assim $S \subseteq Q$. Contudo S é uma CI para o conjunto restrito X^{E_1} , para validar a Cobertura Implícita S foi utilizada uma árvore de suporte de peso mínimo T , contendo arestas de S e de E_1 , ou seja, $S \cup E_1 \subset E_T$. Como

$$\sum_{e \in Q} w_e x_e^* = \sum_{e \in Q} w_e \geq \sum_{e \in E_T} w_e > W.$$

Assim $x^* \notin X_{WMST}$, logo $x^* \notin X^{E_1}$.

□

Observação 1.

A DCI para o conjunto restrito X^{E_1} não tem de ser necessariamente válida para X_{WMST} . A desigualdade válida é obtida apenas quando se faz o levantamento das variáveis do conjunto E_1 (x_e , $e \in E_1$).

Exemplo 5.9.

Considere-se o Exemplo 2.1 de 5 nodos, onde os valores de cada variável na solução da relaxação linear (ver Figura 3.5) são os seguintes: $x_{01} = 1$; $x_{04} = 0,2$; $x_{12} = 1$; $x_{13} = 0,8$; $x_{34} = 0,8$ e $x_{43} = 0,2$ (restantes variáveis com valor nulo).

Seja $E_1 = \{\{0, 1\}, \{1, 2\}\}$ o conjunto de variáveis fixas a um. As arestas de E_1 tomam o valor um na solução da relaxação linear e não pertencem à árvore de suporte de peso mínimo (ver Figura 2.3).

De seguida, ordenam-se as restantes arestas não fixas a um, ou seja, as arestas de $E \setminus E_1$. Considere-se, por exemplo, a seguinte ordenação:

$$\{0, 4\} \rightarrow \{3, 4\} \rightarrow \{1, 3\} \rightarrow \{0, 3\} \rightarrow \{2, 4\} \rightarrow \{2, 3\} \rightarrow \{0, 2\} \rightarrow \{1, 4\}.$$

Para formar a DCI para o conjunto restrito $X^{E_1} = X_{WMST} \cap \{x_e : x_{01} = x_{12} = 1\}$ considere-se a aresta $\{0, 4\}$ e com esta aresta forma-se uma árvore de suporte de peso mínimo que contenha as arestas de E_1 , a qual se encontra representada na Figura 5.5.

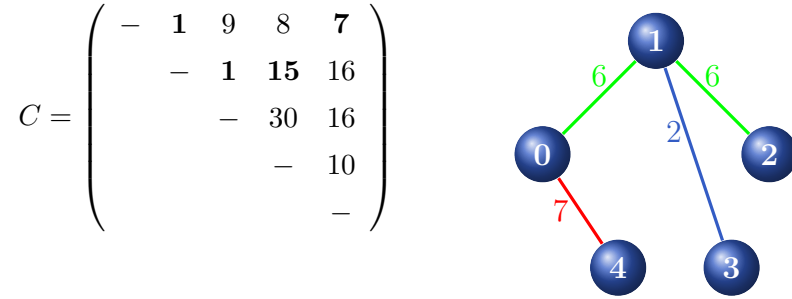


Figura 5.5: Árvore de suporte com custo 24 e peso 21.

Como $W(T) = 21 > 20$, obteve-se uma árvore de suporte não admissível e $S = \{\{0, 4\}\}$ é uma CI para o conjunto restrito X^{E_1} , então a DCI obtida é $x_{04} \leq 0$, a qual é válida para o conjunto restrito X^{E_1} . Notamos que as arestas $\{0, 2\}$, $\{0, 4\}$, $\{1, 3\}$ e $\{1, 4\}$, formam uma árvore de suporte com peso 14 e custo 47, portanto uma solução admissível de X_{WMST} , e na qual $x_{04} = 1$ o que evidencia o facto da desigualdade $x_{04} \leq 0$ não ser válida para X_{WMST} .

De uma forma genérica, se considerarmos que $E_f = E \setminus (S \cup E_1)$ e $R \subseteq E_1 \cup E_f$ é o conjunto de variáveis que já foram sujeitas a levantamento, para calcular os coeficientes das variáveis de $(E_1 \cup E_f) \setminus R$ resolve-se o seguinte subproblema:

$$\beta_t = \min \left\{ |S| - 1 - \sum_{e \in S} x_e - \sum_{e \in R \cap E_f} \beta_e x_e - \sum_{e \in R \cap E_1} \beta_e (1 - x_e) : x \in X(t) \right\}, \quad (5.10)$$

onde $X(t)$ é definido de uma das seguintes formas:

$$X(t) = \left\{ x \in X_{WMST} : x_e = 1, e \in E_1 \setminus (R \cup \{t\}), x_t = 0 \right\}, t \in E_1 \setminus R. \quad (5.11)$$

$$X(t) = \left\{ x \in X_{WMST} : x_e = 1, e \in E_1 \setminus R, x_t = 1 \right\}, t \in E_f \setminus R. \quad (5.12)$$

Nestes casos, se $X(t) = \emptyset$ atribuímos ao coeficiente β_t o valor zero ($\beta_t = 0$). Para efetuar o chamado *down-lifting*, ou seja, o levantamento das variáveis fixas a um, usamos o conjunto definido em (5.11). Neste caso, estamos a atribuir o valor zero à variável $x_t, t \in E_1 \setminus R$ da qual se está a fazer o *down-lifting* e as restantes variáveis de E_1 ($E_1 \setminus (R \cup \{t\})$) das quais ainda não efetuámos o levantamento tomam o valor um.

Para efetuar o levantamento das variáveis de E_f , ou seja, o chamado levantamento usual usamos o conjunto definido em (5.12). Neste caso, estamos a atribuir o valor um à variável $x_t, t \in E_f \setminus R$, da qual se está a fazer o levantamento e atribuímos também o valor um às variáveis fixas a um das quais ainda não efetuámos o levantamento ($E_1 \setminus R$).

Após o levantamento sequencial de todas as variáveis do conjunto R obtemos uma Desigualdade de Cobertura Implícita Levantada por *Down-Lifting*.

Proposição 5.8.

Dada a CI $\sum_{e \in S} x_e \leq |S| - 1$ para o conjunto restrito X^{E_1} , $E_f = E \setminus (S \cup E_1)$ e $R \subseteq E_1 \cup E_f$, a seguinte desigualdade válida

$$\sum_{e \in S} x_e + \sum_{e \in R \cap E_f} \beta_e x_e + \sum_{e \in R \cap E_1} \beta_e (1 - x_e) \leq |S| - 1.$$

é chamada de Desigualdade de Cobertura Implícita Levantada por *Down-Lifting* (DCILDL) e é válida no conjunto das soluções admissíveis X_{WMST} , onde os coeficientes β_e são obtidos sequencialmente resolvendo o subproblema (5.10).

A prova desta proposição é análoga à prova da Proposição 5.5.

No caso do conjunto $E_1 = \emptyset$, ou se todos os coeficientes correspondentes às variáveis fixas a um forem nulos, então a desigualdade anterior é uma simples DCIL.

Exemplo 5.10.

Considere-se a DCI $x_{04} \leq 0$ válida para o conjunto restrito

$$X^{E_1} = X_{WMST} \cap \{x_e : x_{01} = x_{12} = 1\},$$

obtida no Exemplo 5.9. Para efetuar o levantamento das variáveis de E_1 podemos começar por calcular sequencialmente os coeficientes associados a cada uma das variáveis fixas a um, isto é,

$$\beta_{01} = \min\{-x_{04} : x \in X_{WMST}, x_{01} = 0, x_{12} = 1\} = -1;$$

$$\beta_{12} = \min\{1 - x_{01} - x_{04} : x \in X_{WMST}, x_{12} = 0\} = -1.$$

Após o cálculo dos coeficientes das variáveis fixas a um obtém-se a seguinte desigualdade

$$x_{04} + \mathbf{x}_{01} + \mathbf{x}_{12} \leq 2.$$

De seguida para tornar mais forte a desigualdade anterior efetua-se o levantamento das variáveis de $E_f = E \setminus (S \cup E_1) = \{\{0, 2\}, \{0, 3\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}\}$, usando o levantamento sequencial usual, ou seja, calculam-se da seguinte forma os coeficientes das variáveis a efetuar o levantamento

$$\beta_{14} = \min\{2 - x_{01} - x_{04} - x_{12} : x \in X_{WMST}, x_{14} = 1\} = 0;$$

$$\beta_{02} = \min\{2 - x_{01} - x_{04} - x_{12} : x \in X_{WMST}, x_{02} = 1\} = 0;$$

$$\beta_{23} = \min\{2 - x_{01} - x_{04} - x_{12} : x \in X_{WMST}, x_{23} = 1\} = 0;$$

$$\beta_{24} = \min\{2 - x_{01} - x_{04} - x_{12} : x \in X_{WMST}, x_{24} = 1\} = 0;$$

$$\beta_{03} = \min\{2 - x_{01} - x_{04} - x_{12} : x \in X_{WMST}, x_{03} = 1\} = 1;$$

$$\beta_{13} = \min\{2 - x_{01} - x_{04} - x_{12} - x_{03} : x \in X_{WMST}, x_{13} = 1\} = 0;$$

$$\beta_{34} = \min\{2 - x_{01} - x_{04} - x_{12} - x_{03} : x \in X_{WMST}, x_{34} = 1\} = 0.$$

A DCILDL é dada por

$$x_{01} + x_{04} + x_{12} + \mathbf{x}_{03} \leq 2,$$

que é válida para o conjunto das soluções admissíveis X_{WMST} , corta a solução fracionária $(2, 2 > 2)$ e define uma faceta de P_{WMST} (verificado através do *software* PORTA [14]).

Observação 2.

Como se fixaram a um variáveis cujas correspondentes arestas não pertencem à árvore de suporte de peso mínimo, os seus coeficientes obtidos usando *down-lifting* são inferiores ou iguais a zero ($\beta_e \leq 0$).

5.5.2 Desigualdades de Cobertura Implícita Levantada por *Up-Lifting* (DCILUL)

Para obter Desigualdades de Cobertura Implícita Levantada através do designado *Up-Lifting* é necessário começar por fixar a zero um conjunto de variáveis. Designamos por E_0 o conjunto de variáveis fixas a zero e por X^{E_0} o conjunto restrito definido da seguinte forma

$$X^{E_0} = X_{WMST} \cap \{x : x_f = 0, f \in E_0\},$$

onde as variáveis $f \in E_0$ se encontram fixas a zero, $x_f = 0$. Notamos que o conjunto X^{E_0} é uma restrição do conjunto X_{WMST} .

De seguida vamos definir uma cobertura implícita para o conjunto restrito X^{E_0} . Vamos assumir $X^{E_0} \neq \emptyset$.

Definição 5.7. *Cobertura Implícita para o conjunto restrito X^{E_0}*

O conjunto $S \subset E \setminus E_0$, cujas arestas de S não formam ciclos, é uma Cobertura Implícita para o conjunto restrito X^{E_0} , se para cada árvore de suporte $T = (V, E_T)$ tal que $S \subset E_T$ e $E_0 \not\subset E_T$ se verifica a seguinte desigualdade

$$W(T) = \sum_{e \in E_T} w_e > W.$$

Para verificar que $S \subset E \setminus E_0$ é uma CI para o conjunto restrito X^{E_0} , construímos uma sub-árvore com as arestas do conjunto S e, sucessivamente, adicionamos arestas de $E_f = E \setminus (S \cup E_0)$ à sub-árvore até formarmos uma árvore de suporte de peso mínimo, T , contendo as arestas de S e não contendo as arestas de E_0 . No caso da árvore de suporte T , não verificar a restrição de peso, então S é uma CI para o conjunto restrito X^{E_0} .

Proposição 5.9.

Se $S \subset E \setminus E_0$ é uma CI para o conjunto restrito X^{E_0} , então a Desigualdade de Cobertura Implícita

$$\sum_{e \in S} x_e \leq |S| - 1$$

é válida para o conjunto restrito X^{E_0} .

A prova desta proposição é análoga à prova da Proposição 5.7.

Observação 3.

A DCI para o conjunto restrito X^{E_0} não tem de ser necessariamente válida para X_{WMST} . A desigualdade válida é obtida apenas quando se faz o levantamento das variáveis do conjunto E_0 (x_e , $e \in E_0$).

Exemplo 5.11.

Considere-se o Exemplo 2.1 de 5 nodos, onde os valores de cada variável na solução da relaxação linear (ver Figura 3.5) são os seguintes: $x_{01} = 1$; $x_{04} = 0,2$; $x_{12} = 1$; $x_{13} = 0,8$; $x_{34} = 0,8$ e $x_{43} = 0,2$ (restantes variáveis com valor nulo).

Seja $E_0 = \{\{0, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}\}$ o conjunto de variáveis fixas a zero. As arestas de E_0 pertencem à árvore de suporte de peso mínimo (ver Figura 2.3) e não tomam o valor um na solução da relaxação linear.

De seguida ordenam-se as restantes arestas não fixas a zero, ou seja, as arestas de $E \setminus E_0$. Considere-se, por exemplo, a seguinte ordenação:

$$\{0, 1\} \rightarrow \{1, 2\} \rightarrow \{3, 4\} \rightarrow \{0, 4\} \rightarrow \{0, 3\} \rightarrow \{2, 4\}.$$

Para formar a DCI para o conjunto restrito $X^{E_0} = X_{WMST} \cap \{x_e : x_{02} = x_{13} = x_{14} = x_{23} = 0\}$ considere-se a aresta $\{0, 1\}$ e com esta aresta forma-se uma árvore de suporte de peso mínimo que não contenha as arestas de E_0 , a qual se encontra representada na Figura 5.6.

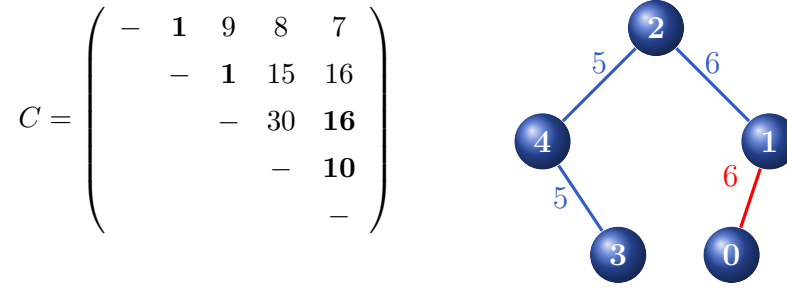


Figura 5.6: Árvore de suporte com custo 28 e peso 22.

Como $W(T) = 22 > 20$, obteve-se uma árvore de suporte não admissível e $S = \{\{0, 1\}\}$ é uma CI para o conjunto restrito X^{E_0} , então a DCI obtida é $x_{01} \leq 0$, a qual é válida para o conjunto restrito X^{E_0} . Notamos que as arestas $\{0, 1\}$, $\{0, 2\}$, $\{1, 3\}$ e $\{1, 4\}$, formam uma árvore com peso 13 e custo 41, portanto uma solução admissível de X_{WMST} , e na qual $x_{01} = 1$ o que evidencia o facto da desigualdade $x_{01} \leq 0$ não ser válida para X_{WMST} .

De uma forma genérica, se considerarmos que $E_f = E \setminus (S \cup E_0)$ e $R \subseteq E_0 \cup E_f$ é o conjunto de variáveis que já foram sujeitas a levantamento, para calcular os coeficientes das variáveis de $(E_0 \cup E_f) \setminus R$ resolve-se o seguinte subproblema:

$$\beta_t = \min \left\{ |S| - 1 - \sum_{e \in S} x_e - \sum_{e \in R \cap (E_0 \cup E_f)} \beta_e x_e : x \in X(t) \right\}, \quad (5.13)$$

onde $X(t)$ é definido da seguinte forma:

$$X(t) = \left\{ x \in X_{WMST} : x_e = 0, e \in E_0 \setminus (R \cup \{t\}), x_t = 1 \right\}, t \in E_0 \setminus R. \quad (5.14)$$

$$X(t) = \left\{ x \in X_{WMST} : x_e = 0, e \in E_0 \setminus R, x_t = 1 \right\}, t \in E_f \setminus R. \quad (5.15)$$

Nestes casos, se $X(t) = \emptyset$ atribuímos ao coeficiente β_t o valor zero ($\beta_t = 0$). Para efetuar o chamado *up-lifting*, ou seja, o levantamento das variáveis fixas a zero, usamos o conjunto definido em (5.14). Neste caso, estamos a atribuir o valor um à variável x_t , $t \in E_0 \setminus R$ da qual se está a fazer o *up-lifting* e as restantes variáveis de E_0 ($E_0 \setminus (R \cup \{t\})$) das quais ainda não efetuámos o levantamento tomam o valor zero.

Para efetuar o levantamento das variáveis de E_f , ou seja, o chamado levantamento usual usamos o conjunto definido em (5.15). Neste caso, estamos a atribuir o valor um à variável $x_t, t \in E_f \setminus R$, da qual se está a fazer o levantamento e às variáveis de $E_0 \setminus R$ atribuímos o valor zero.

Após o levantamento de todas as variáveis do conjunto R obtemos uma Desigualdade de Cobertura Implícita Levantada por *Up-Lifting*.

Proposição 5.10.

Dada a DCI $\sum_{e \in C} x_e \leq |C| - 1$ para o conjunto restrito X^{E_0} , $E_f = E \setminus (S \cup E_0)$ e $R \subseteq E_0 \cup E_f$ a seguinte desigualdade válida

$$\sum_{e \in S} x_e + \sum_{e \in R \cap (E_0 \cup E_f)} \beta_e x_e \leq |S| - 1,$$

é chamada de Desigualdade de Cobertura Implícita Levantada por *Up-Lifting* (DCILUL) e é válida no conjunto das soluções admissíveis X_{WMST} , onde os coeficientes β_e são obtidos sequencialmente resolvendo o subproblema (5.13).

A prova desta proposição é análoga à prova da Proposição 5.5.

No caso do conjunto $E_0 = \emptyset$, ou se todos os coeficientes β_e correspondentes às variáveis fixas a zero forem nulos, então a desigualdade anterior é uma simples DCIL.

Exemplo 5.12.

Considere-se a DCI $x_{01} \leq 0$ válida para o conjunto restrito

$$X^{E_0} = X_{WMST} \cap \{x_e : x_{02} = x_{13} = x_{14} = x_{23} = 0\},$$

obtida no Exemplo 5.11. Para efetuar o levantamento das variáveis de E_0 podemos começar por calcular os coeficientes associados a cada uma das variáveis fixas a zero e obtemos

$$\begin{aligned}
\beta_{13} &= \min\{-x_{01} : x \in X_{WMST}, x_{13} = 1, x_{02} = x_{14} = x_{23} = 0\} = -1; \\
\beta_{23} &= \min\{-x_{01} + x_{13} : x \in X_{WMST}, x_{23} = 1, x_{14} = x_{02} = 0\} = 0; \\
\beta_{02} &= \min\{-x_{01} + x_{13} : x \in X_{WMST}, x_{02} = 1, x_{14} = 0\} = -1; \\
\beta_{14} &= \min\{-x_{01} + x_{13} + x_{02} : x \in X_{WMST}, x_{14} = 1\} = -1.
\end{aligned}$$

Após o cálculo dos coeficientes das variáveis fixas a zero obtém-se a seguinte desigualdade

$$x_{01} - x_{02} - x_{13} - x_{14} \leq 0.$$

De seguida para tornar mais forte a desigualdade anterior efetua-se o levantamento das variáveis de $E_f = E \setminus (S \cup E_0) = \{\{0, 3\}, \{0, 4\}, \{1, 2\}, \{2, 4\}, \{3, 4\}\}$, usando o levantamento sequencial usual, ou seja, calculam-se da seguinte forma os coeficientes das variáveis a efetuar levantamento.

$$\begin{aligned}
\beta_{24} &= \min\{-x_{01} + x_{13} + x_{02} + x_{14} : x \in X_{WMST}, x_{24} = 1\} = 0; \\
\beta_{03} &= \min\{-x_{01} + x_{13} + x_{02} + x_{14} : x \in X_{WMST}, x_{03} = 1\} = 0; \\
\beta_{04} &= \min\{-x_{01} + x_{13} + x_{02} + x_{14} : x \in X_{WMST}, x_{04} = 1\} = 0; \\
\beta_{34} &= \min\{-x_{01} + x_{13} + x_{02} + x_{14} : x \in X_{WMST}, x_{34} = 1\} = 0; \\
\beta_{12} &= \min\{-x_{01} + x_{13} + x_{02} + x_{14} : x \in X_{WMST}, x_{12} = 1\} = 0.
\end{aligned}$$

Como todos os coeficientes obtidos através do levantamento usual são nulos então, a DCILUL é dada por

$$x_{01} - x_{02} - x_{13} - x_{14} \leq 0,$$

que é válida para o conjunto das soluções admissíveis X_{WMST} e corta a solução fracionária $(0, 2 > 0)$.

Observação 4.

Note-se que no caso do Problema WMST podem ser obtidas desigualdades válidas diferentes quando se deixam as variáveis livres ou quando se fixam variáveis a zero.

Exemplo 5.13.

Considere-se a DCI $x_{01} + x_{04} + x_{12} \leq 2$ obtida no Exemplo 5.3. Ao efetuar o levantamento das variáveis x_{34} e x_{13} , obtemos os seguintes coeficientes

$$\beta_{34} = \min\{2 - x_{01} - x_{04} - x_{12} : x \in X_{WMST}, x_{34} = 1\} = 0;$$

$$\beta_{13} = \min\{2 - x_{01} - x_{04} - x_{12} : x \in X_{WMST}, x_{13} = 1\} = 0.$$

Após o cálculo dos coeficientes das variáveis x_{34} e x_{13} obtém-se a seguinte desigualdade

$$x_{01} + x_{04} + x_{12} \leq 2.$$

No caso de fixarmos a variável $x_{13} = 0$ obtemos a DCI $x_{01} + x_{04} + x_{12} \leq 2$ válida para o conjunto restrito $X^{E_0} = X_{WMST} \cap \{x_e : x_{13} = 0\}$. Ao efetuarmos o levantamento das variáveis x_{34} e x_{13} , obtemos os seguintes coeficientes

$$\beta_{34} = \min\{2 - x_{01} - x_{04} - x_{12} : x \in X_{WMST}, x_{13} = 0, x_{34} = 1\} = 1;$$

$$\beta_{13} = \min\{2 - x_{01} - x_{04} - x_{12} - x_{34} : x \in X_{WMST}, x_{13} = 1\} = -1;$$

Após o cálculo dos coeficientes das variáveis a efetuar levantamento obtemos a desigualdade

$$x_{01} + x_{04} + x_{12} + x_{34} - x_{13} \leq 2.$$

Assim, verificamos que ao deixar a variável x_{13} livre ou ao fixá-la a zero se obtêm diferentes desigualdades válidas para X_{WMST} .

5.5.3 Desigualdades Generalizadas de Cobertura Implícita Levantada (DGCIL)

Para obter Desigualdades Generalizadas de Cobertura Implícita Levantada fixam-se à priori dois conjuntos de variáveis: um conjunto de variáveis fixas a zero, que se designa por E_0 e um conjunto de variáveis fixas a um, que se designa por E_1 . Seja X^U o conjunto restrito definido da seguinte forma

$$X^U = X_{WMST} \cap \{x : x_e = 0, e \in E_0, x_e = 1, e \in E_1\},$$

onde $U = E_0 \cup E_1$ e as variáveis $e \in E_0$ encontram-se fixas a zero e as variáveis $e \in E_1$ encontram-se fixas a um. Notamos que o conjunto X^U é uma restrição do conjunto X_{WMST} .

De seguida vamos definir uma cobertura implícita para o conjunto restrito X^U . Vamos assumir $X^U \neq \emptyset$.

Definição 5.8. *Cobertura Implícita para o conjunto restrito X^U*

O conjunto $S \subset E \setminus U$ em que $U = E_0 \cup E_1$, cujas arestas de $S \cup E_1$ não formam ciclos, é uma Cobertura Implícita para o conjunto restrito X^U , se para cada árvore de suporte $T = (V, E_T)$ tal que $S \subset E_T$, $E_0 \not\subset E_T$ e $E_1 \subset E_T$ se verifica a seguinte desigualdade

$$W(T) = \sum_{e \in E_T} w_e > W.$$

Para verificar que $S \subset E \setminus U$ é uma CI para o conjunto restrito X^U , construímos uma sub-árvore com as arestas do conjunto S e do conjunto E_1 e, sucessivamente, adicionamos arestas de $E_f = E \setminus (S \cup U)$ à sub-árvore até formarmos uma árvore de suporte de peso mínimo, T , contendo as arestas dos conjuntos S e E_1 . No caso da árvore de suporte T não verificar a restrição de peso, então S é uma CI para o conjunto restrito X^U .

Proposição 5.11.

Seja $S \subset E \setminus U$ uma CI para o conjunto restrito X^U , então a Desigualdade de Cobertura Implícita

$$\sum_{e \in S} x_e \leq |S| - 1$$

é válida para o conjunto restrito X^U .

A prova desta proposição é análoga à prova da Proposição 5.7.

Observação 5.

Observe-se que a DCI para o conjunto restrito X^U não tem de ser necessariamente válida para X_{WMST} . A desigualdade válida é obtida apenas quando se faz o levantamento das variáveis do conjunto $U = E_0 \cup E_1$ (x_e , $e \in E_0 \cup E_1$).

Exemplo 5.14.

Considere-se o Exemplo 2.1 de 5 nodos, onde os valores de cada variável na solução da relaxação linear (ver Figura 3.5) são os seguintes: $x_{01} = 1$; $x_{04} = 0,2$; $x_{12} = 1$; $x_{13} = 0,8$; $x_{34} = 0,8$ e $x_{43} = 0,2$ (restantes variáveis com valor nulo).

Seja $E_0 = \{\{0, 2\}, \{2, 3\}, \{1, 4\}\}$ o conjunto de variáveis fixas a zero, as arestas de E_0 pertencem à árvore de suporte de peso mínimo (ver Figura 2.3) e tomam o valor zero na solução da relaxação linear. Seja $E_1 = \{\{0, 1\}, \{1, 2\}\}$ o conjunto de variáveis fixas a um. As arestas de E_1 tomam o valor um na solução da relaxação linear e não pertencem à árvore de suporte de peso mínimo (ver Figura 2.3).

De seguida, ordenam-se as restantes arestas não fixas, ou seja, as arestas de $E \setminus (E_0 \cup E_1)$. Considere-se, por exemplo, a seguinte ordenação:

$$\{0, 4\} \rightarrow \{3, 4\} \rightarrow \{1, 3\} \rightarrow \{0, 3\} \rightarrow \{2, 4\}.$$

Para formar a DCI para o conjunto restrito $X^U = X_{WMST} \cap \{x_e : x_{01} = x_{12} = 1, x_{02} = x_{23} = x_{14} = 0\}$ considere-se a aresta $\{0, 4\}$ e com esta forma-se a árvore de suporte de peso mínimo da Figura 5.7, esta contém as arestas de E_1 e não contém as arestas de E_0 .

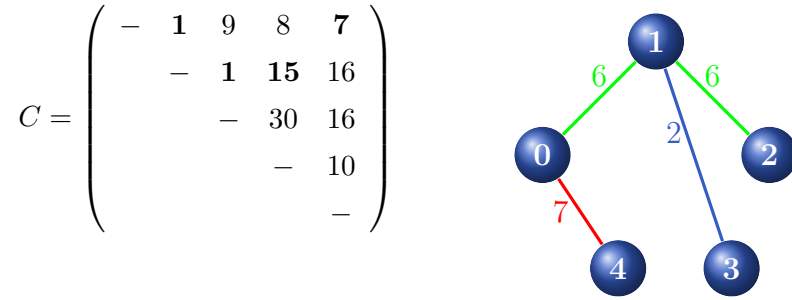


Figura 5.7: Árvore de suporte com custo 24 e peso 21.

Como $W(T) = 21 > 20$, obteve-se uma árvore de suporte não admissível e $S = \{\{0, 4\}\}$ é uma CI para o conjunto restrito X^U , então a DCI obtida é $x_{04} \leq 0$, a qual é válida para o conjunto restrito X^U . Notamos que as arestas $\{0, 2\}$, $\{0, 4\}$, $\{1, 3\}$ e $\{1, 4\}$, formam uma árvore de suporte com peso 14 e custo 47, portanto uma solução admissível de X_{WMST} , e na qual $x_{04} = 1$ o que evidencia o facto da desigualdade $x_{04} \leq 0$ não ser válida para X_{WMST} .

De uma forma genérica para calcular, sequencialmente, os coeficientes das variáveis de $(E_0 \cup E_1 \cup E_f) \setminus R$, onde $E_f = E \setminus (S \cup E_0 \cup E_1)$ e $R \subseteq E_0 \cup E_1 \cup E_f$ resolve-se o seguinte subproblema:

$$\beta_t = \min \left\{ |S| - 1 - \sum_{e \in S} x_e - \sum_{e \in R \cap (E_0 \cup E_f)} \beta_e x_e - \sum_{e \in R \cap E_1} \beta_e (1 - x_e) : x \in X(t) \right\} \quad (5.16)$$

onde

$$X(t) = \left\{ x \in X_{WMST} : x_e = 1, e \in E_1 \setminus (R \cup \{t\}), x_e = 0, e \in E_0 \setminus R, x_t = 0 \right\}, \quad (5.17)$$

$t \in E_1 \setminus R$;

$$X(t) = \left\{ x \in X_{WMST} : x_e = 1, e \in E_1 \setminus R, x_e = 0, e \in E_0 \setminus (R \cup \{t\}), x_t = 1 \right\}, \quad (5.18)$$

$t \in E_0 \setminus R$;

$$X(t) = \left\{ x \in X_{WMST} : x_e = 1, e \in E_1 \setminus R, x_e = 0, e \in E_0 \setminus R, x_t = 1 \right\}, \quad (5.19)$$

$t \in E_f \setminus R$.

Nestes casos, se $X(t) = \emptyset$ atribuímos ao coeficiente β_t o valor zero ($\beta_t = 0$). Para efetuar o chamado *down-lifting*, ou seja, o levantamento das variáveis fixas a um, usamos o conjunto definido em (5.17). Neste caso, estamos a atribuir o valor zero à variável $x_t, t \in E_1 \setminus R$ da qual se está a fazer o *down-lifting*. As variáveis de E_1 ($E_1 \setminus (R \cup \{t\})$) das quais ainda não efetuámos o levantamento tomam o valor um e as variáveis fixas a zero das quais ainda não efetuámos o levantamento ($E_0 \setminus R$) tomam o valor zero.

Para efetuar o levantamento das variáveis fixas a zero, isto é, o chamado *up-lifting* usamos o conjunto definido em (5.18). Neste caso, estamos a atribuir o valor um à variável $x_t, t \in E_0 \setminus R$ da qual se está a fazer o *up-lifting*. Às restantes variáveis fixas a zero das quais ainda não efetuámos o levantamento ($E_0 \setminus (R \cup \{t\})$) atribuímos o valor zero e às variáveis fixas a um das quais ainda não efetuámos o levantamento ($E_1 \setminus R$) atribuímos o valor um.

Para efetuar o levantamento das variáveis de E_f , usamos o conjunto definido em (5.19). Neste caso, estamos a atribuir o valor um à variável $x_t, t \in E_f \setminus R$ da qual se está a fazer o levantamento. Às variáveis fixas a um das quais ainda não efetuámos o levantamento ($E_1 \setminus R$) atribuímos o valor um, e finalmente às variáveis fixas a zero das quais ainda não efetuámos o levantamento ($E_0 \setminus R$) atribuímos o valor zero.

Após o levantamento de todas as variáveis do conjunto R obtemos uma Desigualdade Generalizada de Cobertura Implícita Levantada.

Proposição 5.12.

Dada a DCI $\sum_{e \in S} x_e \leq |S| - 1$ para o conjunto restrito X^U , $E_f = E \setminus (S \cup E_0 \cup E_1)$ e $R \subseteq E_0 \cup E_1 \cup E_f$, a seguinte desigualdade válida

$$\sum_{e \in S} x_e + \sum_{e \in R \cap (E_0 \cup E_f)} \beta_e x_e + \sum_{e \in R \cap E_1} \beta_e (1 - x_e) \leq |S| - 1.$$

é chamada de Desigualdade Generalizada de Cobertura Implícita Levantada (DGCIL) e é válida no conjunto das soluções admissíveis X_{WMST} , onde os coeficientes β_e são obtidos sequencialmente resolvendo o subproblema (5.16).

A prova desta proposição é análoga à prova da Proposição 5.5.

As Desigualdades de Cobertura Implícita Levantada por *Down-lifting* e por *Up-lifting* são casos particulares das Desigualdades Generalizadas de Cobertura Implícita Levantada.

- As DCILs obtêm-se quando $E_0 \cup E_1 = \emptyset$ ou se todos os coeficientes das variáveis fixas a zero e a um forem nulos.
- As DCILULs obtêm-se quando $E_0 = \emptyset$ ou quando todos os coeficientes das variáveis fixas a zero forem nulos.
- E finalmente as DCILDLs obtêm-se quando $E_1 = \emptyset$ ou quando todos os coeficientes das variáveis fixas a um forem nulos.

Exemplo 5.15.

Considere-se a DCI $x_{04} \leq 0$ válida para o conjunto restrito

$$X^U = X_{WMST} \cap \{x_e : x_{01} = x_{12} = 1, x_{02} = x_{23} = x_{14} = 0\},$$

obtida no Exemplo 5.14. Para efetuar o levantamento das variáveis de E_1 podemos começar por calcular os coeficientes associados a cada uma das variáveis fixas a um e obtemos

$$\begin{aligned}\beta_{01} &= \min\{-x_{04} : x \in X_{WMST}, x_{01} = 0, x_{12} = 1, x_{02} = x_{23} = x_{14} = 0\} = -1 \\ \beta_{12} &= \min\{1 - x_{01} - x_{04} : x \in X_{WMST}, x_{12} = 0, x_{02} = x_{23} = x_{14} = 0\} = -1.\end{aligned}$$

De seguida efetua-se o levantamento das variáveis de E_0 , ou seja, calculam-se, sequencialmente, os coeficientes destas variáveis da seguinte forma

$$\begin{aligned}\beta_{23} &= \min\{2 - x_{01} - x_{04} - x_{12} : x \in X_{WMST}, x_{23} = 1, x_{02} = x_{04} = 0\} = 0 \\ \beta_{02} &= \min\{2 - x_{01} - x_{04} - x_{12} : x \in X_{WMST}, x_{02} = 1, x_{14} = 0\} = 0 \\ \beta_{14} &= \min\{2 - x_{01} - x_{04} - x_{12} : x \in X_{WMST}, x_{14} = 1\} = 0\end{aligned}$$

Após o cálculo dos coeficientes das variáveis fixas a zero e a um obtém-se a seguinte desigualdade

$$\mathbf{x}_{01} + x_{04} + \mathbf{x}_{12} \leq 2.$$

De seguida para tornar mais forte a desigualdade anterior efetua-se o levantamento das variáveis de $E_f = E \setminus (S \cup E_0 \cup E_1) = \{\{0, 3\}, \{1, 3\}, \{2, 4\}, \{3, 4\}\}$ usando o levantamento sequencial usual e os coeficientes calculam-se da seguinte forma:

$$\begin{aligned}\beta_{34} &= \min\{2 - x_{01} - x_{04} - x_{12} : x \in X_{WMST}, x_{34} = 1\} = 0 \\ \beta_{13} &= \min\{2 - x_{01} - x_{04} - x_{12} : x \in X_{WMST}, x_{13} = 1\} = 1 \\ \beta_{03} &= \min\{2 - x_{01} - x_{04} - x_{12} - x_{13} : x \in X_{WMST}, x_{03} = 1\} = 0 \\ \beta_{24} &= \min\{2 - x_{01} - x_{04} - x_{12} - x_{13} : x \in X_{WMST}, x_{24} = 1\} = 0\end{aligned}$$

A DGCIL é dada por

$$x_{01} + x_{04} + x_{12} + \mathbf{x}_{13} \leq 2,$$

que é válida para o conjunto das soluções admissíveis X_{WMST} e corta a solução fracionária $(2, 2 > 2)$.

Em Anexo apresentam-se todas as facetas de P_{WMST} para o Exemplo 2.1 obtidas através do *software* PORTA (*Polyhedron Representation Transformation Algorithm*) [14]. Para cada uma das facetas obtidas pelo *software* foi feita a respetiva caracterização.

Capítulo 6

Algoritmos de Separação

Com o propósito de usar classes de desigualdades válidas como planos de corte e de tornar este processo eficiente necessitamos de algoritmos de separação, isto é, precisamos de ter um conjunto de rotinas para gerar desigualdades válidas e detetar quando é que elas são violadas. Kaparis e Letchford [42, 43] apresentam um estudo muito completo sobre separação de desigualdades válidas para o Problema Saco-mochila Binário.

Infelizmente os problemas de separação para Desigualdades de Cobertura, Desigualdades de Cobertura Levantada e Desigualdades Generalizadas de Cobertura Levantada são NP-difíceis (ver [17, 30, 44]).

Neste capítulo com o intuito de fortalecer as formulações apresentadas no Capítulo 3, descrevemos algoritmos de separação para as classes de desigualdades válidas abordadas no Capítulo 5.

Na Secção 6.1 começamos por apresentar dois algoritmos heurísticos de separação genéricos. Num dos algoritmos é introduzida uma desigualdade válida, em cada iteração, enquanto que no outro podem ser introduzidas várias desigualdades válidas numa iteração. Em cada uma das subsecções seguintes apresentamos as particularidades dos algoritmos de separação que é necessário especificar para introduzir as desigualdades válidas descritas no Capítulo 5. Na Secção 6.1 apresentamos também um algoritmo heurístico para determinação dos coeficientes das variáveis a efetuar levantamento. Na Secção 6.2 começamos por apresentar as experiências computacionais realizadas com os algoritmos de separação propostos, depois comparamos os vários algoritmos apresentados. De seguida apresentamos os resultados computacionais dos melhores algoritmos de separação e por fim apresentamos uma síntese dos resultados computacionais obtidos neste capítulo.

6.1 Algoritmos de Separação Genéricos

Vamos começar por apresentar dois algoritmos de separação genéricos, os quais vão servir como base a todos os algoritmos de separação propostos ao longo deste capítulo.

Algoritmo de Separação Genérico 1 (Sep1)

Passo 1: *Resolver a relaxação linear*

Seja x^* a solução da relaxação linear de uma formulação para o Problema WMST.

Se x^* for inteira, então STOP.

Seja $E_{RL} = \{e : x_e^* > 0\}$ o conjunto formado pelas arestas da solução da relaxação linear com valor positivo.

Seja $ORD = \{O_1, \dots, O_{nord}\}$ um conjunto de $nord$ ordenações predefinidas.

Considerar a primeira ordenação O_1 e fazer $k = 1$.

Passo 2: *Ordenar as arestas da relaxação linear*

Seja E_{RLOrd_k} a lista ordenada de arestas da solução da relaxação linear de acordo com a ordenação de arestas k .

Passo 3: *Obter uma desigualdade válida*

Construir, caso seja possível, uma desigualdade válida (DV).

Passo 4: *Decisão sobre a inserção da desigualdade válida*

Se a DV corta a solução fracionária x^* , então

inserir a DV no modelo e voltar ao **Passo 1**.

Caso contrário,

Fazer $k = k + 1$ (passar à ordenação seguinte).

Se $k > nord$, então

STOP.

Caso contrário,

voltar ao **Passo 2**.

No Passo 1 resolvemos a relaxação linear do Procedimento P-WMTZ+C descrito na Secção 3.3. Se a solução x^* não for inteira obtemos um conjunto que denominamos de E_{RL} , constituído pelas arestas com $x_e > 0$ na solução da relaxação linear, impomos um número fixo de ordenações *nord* e estabelecemos a primeira ordenação de arestas ($k = 1$). Em seguida, no Passo 2 ordenamos as arestas desse conjunto de acordo com a ordenação indicada k , formando a lista de arestas $E_{RLOrd k}$. No Passo 3 temos de construir uma desigualdade válida, caso seja possível. Finalmente, no Passo 4 é avaliado se a desigualdade válida formada no Passo 3 corta a solução fracionária x^* (solução da relaxação linear obtida no Passo 1). Se a desigualdade válida cortar a solução fracionária x^* , será introduzida no modelo e voltamos ao Passo 1, onde resolvemos a nova relaxação linear. Caso a desigualdade válida não corte a solução fracionária x^* , ou não seja possível formar uma desigualdade válida no Passo 3, temos ainda de averiguar se já foram utilizadas todas as ordenações definidas inicialmente por *nord*. Em caso afirmativo, o algoritmo termina. No caso de existirem outras ordenações para efetuar, voltamos ao Passo 2 e ordenamos as arestas do conjunto E_{RL} por outra ordenação k . Observe-se que outros critérios de paragem podem ser impostos, por exemplo, podemos terminar o algoritmo quando se encontrarem um certo número de relaxações lineares seguidas com valor igual ou impor a introdução de um certo número de cortes.

De seguida propomos algumas formas de ordenar as arestas da relaxação linear com $x_e > 0$ para formar a lista de arestas $E_{RLOrd k}$.

1. **OrdW**: Ordenar por ordem decrescente de peso, w_e .
2. **OrdWX**: Ordenar por ordem decrescente de valor de $w_e \times x_e$.
3. **OrdX**: Ordenar por ordem decrescente de valor de x_e .
4. **OrdKnap**: Ordenar por ordem decrescente de valor de $\frac{1 - x_e}{w_e}$.
5. **OrdFixW**: Começar por ordenar por ordem decrescente de peso, w_e as arestas com $x_e = 1$ e de seguida, ordenar as restantes arestas ($0 < x_e < 1$) também por ordem decrescente de peso, w_e .
6. **OrdFixWX**: Começar por ordenar por ordem decrescente de valor de $w_e \times x_e$ as arestas com $x_e = 1$ e de seguida, ordenar as restantes arestas ($0 < x_e < 1$) também por ordem decrescente de $w_e \times x_e$.

No Algoritmo Sep1, iterativamente, inserimos uma desigualdade válida no modelo e resolvemos uma nova relaxação linear, mas podemos querer introduzir não uma, mas várias desigualdades no modelo (se for possível) e só depois resolver a relaxação linear. Assim no novo algoritmo proposto apenas alteramos os Passos 3 e 4.

Algoritmo de Separação Genérico 2 (Sep2)

Passo 3: *Obter uma desigualdade válida*

Construir, caso seja possível, uma desigualdade válida (DV).

Fazer $n_{DV} = 0$.

Passo 4: *Decisão sobre a inserção da desigualdade válida*

Fazer $k = k + 1$ (passar à ordenação seguinte).

Se a DV corta a solução fracionária x^* , então

inserir a DV no modelo.

$$n_{DV} = n_{DV} + 1.$$

Se $k > nord$, então

Se $n_{DV} > 0$, então

voltar ao **Passo 1**.

Caso contrário,

STOP.

Caso contrário,

voltar ao **Passo 2**.

No Passo 3 para além de construirmos, caso seja possível, uma desigualdade válida, vamos inicializar o número de desigualdades válidas ($n_{DV} = 0$). No Passo 4 do Algoritmo Sep2 é avaliado se a desigualdade válida formada no Passo 3 corta a solução fracionária x^* (solução da relaxação linear obtida no Passo 1). Se a desigualdade válida cortar a solução fracionária x^* , será introduzida no modelo e aumenta-se em uma unidade o número de desigualdades válidas a introduzir no modelo. No caso de ainda não se terem testado todas as ordenações definidas inicialmente por $nord$, então voltamos

ao Passo 2 e ordenamos as arestas do conjunto E_{RL} por outra ordenação k . Caso não existam mais ordenações disponíveis e existam desigualdades válidas a inserir no modelo ($n_{DV} > 0$), então volta-se ao Passo 1 para resolver a nova relaxação linear. O algoritmo termina se não existirem mais ordenações disponíveis nem desigualdades válidas a inserir no modelo. Tal como referimos anteriormente para o Algoritmo Sep1 podemos impor outros critérios de paragem. Por exemplo, terminar o algoritmo quando se encontrarem um certo número de relaxações lineares seguidas com valor igual ou quando um certo número de cortes forem introduzidos no modelo.

Nas secções seguintes descrevemos para cada classe de desigualdades válidas apresentadas no Capítulo 5 os procedimentos a adotar no Passo 3 para a formação de uma desigualdade válida.

6.1.1 Algoritmo de Separação para as DC

Para descrever um algoritmo de separação para as Desigualdades de Cobertura (DCs) precisamos de encontrar as sucessivas DCs que é possível obter no Passo 3 dos Algoritmos Sep1 e Sep2. Assim, para descrever os algoritmos para introdução de DCs será necessário especificar o Passo 3.

Passo 3: *Obter uma Desigualdade de Cobertura*

Seja $S = \{\}$.

Fazer $p = 1$ (inserir a primeira aresta de E_{RLOrdk} no conjunto S).

Passo 3.1: *Construir o conjunto S*

Atualizar o conjunto S com as primeiras p arestas da lista E_{RLOrdk} .

Passo 3.2: *Verificar se S é Cobertura*

Calcular $W_S = \sum_{e \in S} w_e$.

Se $W_S > W$, então

S é uma Cobertura.

DC: $\sum_{e \in S} x_e \leq |S| - 1$.

Caso contrário,

Se $p < \#E_{RLOrdk}$, então

$$p = p + 1.$$

Voltar ao **Passo 3.1**.

Caso contrário,

não é possível formar uma cobertura.

No Passo 3, para formar uma DC, temos de formar uma Cobertura S , para tal, adicionamos, sucessivamente, uma aresta da lista E_{RLOrdk} ao conjunto S , até que a soma dos pesos das arestas contidas em S seja superior a W .

Note-se que no Passo 3.2, $\#E_{RLOrdk}$ representa o número de arestas da lista E_{RLOrdk} .

Para exemplificar os algoritmos de separação propostos para as DCs usamos as seis ordenações propostas ($nord = 6$) e a instância de 5 nodos do Exemplo 2.1.

Exemplo 6.1.

Consideremos o Exemplo 2.1. De seguida vamos descrever em detalhe a primeira iteração do Algoritmo Sep1 para o caso das DCs e os principais resultados das restantes iterações encontram-se na Tabela 6.1.

Iteração 1:

Passo 1: O conjunto de arestas com $x_e > 0$ na solução da relaxação linear do Procedimento P-WMTZ+C é dado por $E_{RL} = \{\{0, 1\}, \{0, 4\}, \{1, 2\}, \{1, 3\}, \{3, 4\}\}$, onde $x_{01} = 1$; $x_{04} = 0,2$; $x_{12} = 1$; $x_{13} = 0,8$; $x_{34} = 0,8$ e $x_{43} = 0,2$ (restantes variáveis com valor nulo).

Passo 2: Ordenando as arestas do conjunto E_{RL} , por exemplo, por ordem decrescente de peso (OrdW) obtemos a seguinte lista de arestas ordenadas

$$E_{RLOrdW} = [\{0, 4\}, \{0, 1\}, \{1, 2\}, \{3, 4\}, \{1, 3\}]. \quad (6.1)$$

Passo 3: Para formar a Cobertura S vamos adicionar, sucessivamente, uma aresta de E_{RLOrdW} ao conjunto S da seguinte forma,

$$S = \{\{0, 4\}\} \rightsquigarrow W_S = w_{04} = 7;$$

$$S = \{\{0, 1\}, \{0, 4\}\} \rightsquigarrow W_S = w_{04} + w_{01} = 7 + 6 = 13;$$

$$S = \{\{0, 1\}, \{0, 4\}, \{1, 2\}\} \rightsquigarrow W_S = w_{04} + w_{01} + w_{12} = 13 + 6 = 19;$$

$$S = \{\{0, 1\}, \{0, 4\}, \{1, 2\}, \{3, 4\}\} \rightsquigarrow W_S = w_{04} + w_{01} + w_{12} + w_{34} = 19 + 5 = 24 > 20.$$

Como $W_S = 24 > 20 = W$, S é uma Cobertura e podemos construir a correspondente DC

$$x_{01} + x_{04} + x_{12} + x_{34} \leq 3. \quad (6.2)$$

Passo 4: Como a DC obtida corta a solução fracionária $(3, 2 > 3)$, então é introduzida no modelo e volta-se ao Passo 1.

A informação respeitante às restantes iterações encontra-se resumida na Tabela 6.1.

Iter.	Sep1-DC	
2	Passo 1	$x^*: x_{01} = 1; x_{04} = 0,5; x_{12} = 1; x_{13} = 1$ e $x_{34} = 0,5$ Custo = 25,5 e peso = 20
	Passo 2, 3 e 4	Ordenação: OrdW DC: $x_{01} + x_{04} + x_{12} + x_{34} \leq 3$. Não corta x^* ($3 \not\geq 3$).
	Passo 2, 3 e 4	Ordenação: OrdWX DC: $x_{01} + x_{04} + x_{12} + x_{34} \leq 3$. Não corta x^* ($3 \not\geq 3$).
	Passo 2, 3 e 4	Ordenação: OrdX DC: $x_{01} + x_{04} + x_{12} + x_{13} + x_{34} \leq 4$. Não corta x^* ($4 \not\geq 4$).
	Passo 2, 3 e 4	Ordenação: OrdFixWX DC: $x_{01} + x_{04} + x_{12} + x_{13} \leq 3 \rightsquigarrow$ Inserir, corta x^* ($3,5 > 3$).
3	Passo 1	$x^*: x_{01} = 1; x_{03} = 0,125; x_{04} = 0,125; x_{12} = 1; x_{13} = 0,875$ e $x_{34} = 0,875$ Custo = 25,75 e peso = 20
	Passo 2, 3 e 4	Ordenação: OrdW DC: $x_{01} + x_{03} + x_{04} \leq 2$. Não corta x^* ($1,25 \not\geq 2$).
	Passo 2, 3 e 4	Ordenação: OrdWX DC: $x_{01} + x_{03} + x_{12} + x_{13} + x_{34} \leq 4$. Não corta x^* ($3,875 \not\geq 4$).
	Passo 2, 3 e 4	Ordenação: OrdX DC: $x_{01} + x_{03} + x_{12} + x_{13} + x_{34} \leq 4$. Não corta x^* ($3,875 \not\geq 4$).
	Passo 2, 3 e 4	Ordenação: OrdFixWX DC: $x_{01} + x_{03} + x_{12} + x_{13} + x_{34} \leq 4$. Não corta x^* ($3,875 \not\geq 4$).
	Passo 2, 3 e 4	Ordenação: OrdFixW DC: $x_{01} + x_{03} + x_{04} + x_{12} \leq 3$. Não corta x^* ($2,25 \not\geq 3$).
	Passo 2, 3 e 4	Ordenação: OrdKnap DC: $x_{03} + x_{04} + x_{13} + x_{34} \leq 3$. Não corta x^* ($2 \not\geq 4$).

Tabela 6.1: Principais resultados por iteração do Algoritmo Sep1 para as DCs aplicado ao Exemplo 2.1.

Como podemos observar neste algoritmo, são introduzidas no modelo duas DCs, uma na Iteração 1 usando a ordenação OrdW e outra na Iteração 2 usando a ordenação OrdFixWX. Na Iteração 3 não é possível obter nenhuma DC e dado que se percorreram todas as ordenações, o algoritmo termina e a solução obtida tem custo 25,75 e peso 20.

Na Tabela 6.2 podemos observar as principais etapas do Algoritmo Sep2 para as DCs aplicado ao Exemplo 2.1. Na referida tabela apresentamos apenas as desigualdades que cortam a solução x^* .

Iter.	Sep2-DC	
1	Passo 1	x^* : $x_{01} = 1$; $x_{04} = 0,2$; $x_{12} = 1$; $x_{13} = 0,8$; $x_{34} = 0,8$ e $x_{43} = 0,2$ Custo = 25,4 e peso = 20
	Passo 2, 3 e 4	Ordenação: OrdW e OrdFixW DC: $x_{01} + x_{04} + x_{12} + x_{34} \leq 3 \rightsquigarrow$ Inserir, corta x^* ($3,2 > 3$).
2	Passo 1	x^* : $x_{01} = 1$; $x_{04} = 0,5$; $x_{12} = 1$; $x_{13} = 1$ e $x_{34} = 0,5$ Custo = 25,5 e peso = 20
	Passo 2, 3 e 4	Ordenação: OrdFixW e OrdFixWX DC: $x_{01} + x_{04} + x_{12} + x_{13} \leq 3 \rightsquigarrow$ Inserir, corta x^* ($3,5 > 3$).
3	Passo 1	x^* : $x_{01} = 1$; $x_{03} = 0,125$; $x_{04} = 0,125$; $x_{12} = 1$; $x_{13} = 0,875$ e $x_{34} = 0,875$ Custo = 25,75 e peso = 20

Tabela 6.2: Principais resultados por iteração do Algoritmo Sep2 para as DCs aplicado ao Exemplo 2.1.

Neste exemplo, com a aplicação do Algoritmo Sep2 são adicionadas também duas DCs, sendo estas as introduzidas no Algoritmo Sep1. Nas Iteração 1 e 2 foram omitidos os passos correspondentes às ordenações que não formam nenhuma DC.

Uma desvantagem da aplicação do Algoritmo Sep2 poderá ser o consumo de tempo em obter desigualdades já obtidas, pois na Tabela 6.2 nas Iterações 1 e 2 são obtidas a mesma DC por dois modos de ordenação de arestas diferentes.

A solução obtida neste algoritmo tem custo 25,75 e peso 20 e é igual à obtida pelo Algoritmo Sep1.

Podemos observar também que as coberturas correspondentes às DCs obtidas nos Algoritmos Sep1 e Sep2 são minimais.

Proposição 6.1.

Se S é uma Cobertura formada pelos Algoritmos Sep1 e Sep2 usando a ordenação OrdW, então S é Cobertura Minimal.

Demonstração.

Seja S uma Cobertura, com $|S| = s$ e $E_{OrdW} = [e_1, e_2, \dots, e_{s-1}, e_s, \dots, e_t]$ a lista de arestas ordenadas por ordem decrescente de peso (OrdW), onde t é o número de arestas da relaxação linear com solução positiva. Os pesos das arestas de E_{OrdW} verificam,

$$w_{e_1} \geq w_{e_2} \geq \dots \geq w_{e_{s-1}} \geq w_{e_s} \geq \dots \geq w_{e_t}.$$

Se considerarmos a sucessiva inclusão de arestas em S obtemos:

$$S = \{e_1\} \rightsquigarrow w_{e_1} < W$$

$$S = \{e_1, e_2\} \rightsquigarrow w_{e_1} + w_{e_2} < W$$

$$\vdots$$

$$S = \{e_1, e_2, \dots, e_{s-1}\} \rightsquigarrow w_{e_1} + w_{e_2} + \dots + w_{e_{s-1}} \leq W.$$

Como ainda não temos a cobertura formada, pois só temos $s - 1$ arestas incluídas na cobertura então adicionamos a próxima aresta da lista E_{OrdW} e obtemos

$$w_{e_1} + w_{e_2} + \dots + w_{e_{s-1}} + w_{e_s} > W,$$

o que é equivalente a ter $\sum_{e \in S} w_e > W$, pelo que S é uma Cobertura.

Se removermos a aresta de menor peso, e_s , da cobertura ficamos com

$$\sum_{e \in S - \{e_s\}} w_e \leq W, \tag{6.3}$$

e se removermos qualquer outra aresta e_i , com $i = \{1, 2, \dots, s - 1\}$, sabemos que $w_{e_i} \geq w_{e_s}$, logo a desigualdade (6.3) é sempre verificada.

Assim S é Cobertura Minimal.

□

Quando se utiliza uma das outras cinco ordenações propostas a cobertura formada pode não ser minimal.

Por exemplo, com a ordenação de arestas OrdWX,

$$E_{RLOrdWX} = [\{0, 1\}, \{1, 2\}, \{3, 4\}, \{1, 3\}, \{0, 4\}],$$

obtemos a cobertura $S = \{\{0, 1\}, \{1, 2\}, \{3, 4\}, \{1, 3\}, \{0, 4\}\}$, a qual não é uma cobertura minimal, pois se removermos a aresta $\{1, 3\}$ do conjunto S o peso das restantes arestas é $W_S = 25 > 20$.

Note-se ainda que com diferentes sequências de ordenações ou utilizando um maior ou menor número de ordenações podemos obter DCs diferentes e, consequentemente, levar à obtenção de diferentes limites inferiores para o valor ótimo.

6.1.2 Algoritmo de Separação para as DCI

Para aplicar os Algoritmos Sep1 e Sep2 ao caso particular das Desigualdades de Cobertura Implícita (DCIs), temos de efetuar algumas alterações ao Passo 3 descrito na Subsecção 6.1.1, pois as CI são baseadas numa estrutura em árvore.

Passo 3: *Obter uma Desigualdade de Cobertura Implícita*

Seja $S = \{\}$.

Fazer $p = 1$ (inserir a primeira aresta de E_{RLOrdk} no conjunto S).

Passo 3.1: *Construir o conjunto S (sem ciclos)*

Atualizar o conjunto S com as primeiras p arestas da lista E_{RLOrdk} , desde que não formem ciclos.

Passo 3.2: *Verificar se S é Cobertura Implícita*

Obter a árvore de suporte de peso mínimo T , que inclui as p primeiras arestas de S .

Se $W(T) > W$, então

S é uma CI.

DCI: $\sum_{e \in S} x_e \leq |S| - 1$.

Caso contrário,

Se $p < \#E_{RLOrk}$, então

$p = p + 1$.

Voltar ao **Passo 3.1**.

Caso contrário,

não é possível formar uma CI.

Para obter uma DCI, no Passo 3.1 atualizamos, sucessivamente, o conjunto S com as arestas de E_{RLOrk} , desde que não formem ciclos. Inicialmente o conjunto S começa com uma aresta ($p = 1$). A formação da DCI é realizada no Passo 3.2, onde formamos uma árvore de suporte de peso mínimo com as arestas de S . Enquanto a árvore obtida verificar a restrição de peso e existirem arestas de E_{RLOrk} para inserir em S , voltamos ao Passo 3.1 e atualizamos o conjunto S adicionando mais uma aresta da lista E_{RLOrk} . No caso de não existirem mais arestas para inserir em S , então não é possível formar uma CI. Caso a restrição de peso não seja verificada, S é uma CI e temos a correspondente DCI.

De seguida exemplificamos os Algoritmos Sep1 e Sep2 para as DCIs usando as seis ordenações propostas ($nord = 6$) e a instância de 5 nodos do Exemplo 2.1.

Exemplo 6.2.

Consideremos o Exemplo 2.1. De seguida apresentamos a explicação detalhada da Iteração 1 do Algoritmo Sep1 para as DCIs e na Tabela 6.3 encontram-se os principais passos das restantes iterações.

Iteração 1:

Passo 1: O conjunto de arestas com $x_e > 0$ na solução da relaxação linear do Procedimento P-WMTZ+C é dado por $E_{RL} = \{\{0, 1\}, \{0, 4\}, \{1, 2\}, \{1, 3\}, \{3, 4\}\}$, onde $x_{01} = 1$; $x_{04} = 0,2$; $x_{12} = 1$; $x_{13} = 0,8$; $x_{34} = 0,8$ e $x_{43} = 0,2$ (restantes variáveis com valor nulo).

Passo 2: De seguida ordenamos as arestas do conjunto E_{RL} , por exemplo, por ordem decrescente de peso e obtemos a seguinte lista de arestas ordenadas

$$E_{RLOrdW} = [\{0, 4\}, \{0, 1\}, \{1, 2\}, \{3, 4\}, \{1, 3\}].$$

Passo 3: Considerando a primeira aresta da lista ordenada E_{RLOrdW} , formamos no Passo 3.1 o conjunto $S = \{\{0, 4\}\}$ ($p = 1$).

No Passo 3.2 construímos a árvore de suporte de peso mínimo que inclui esta aresta e obtemos a solução admissível da Figura 6.1 com peso 14.

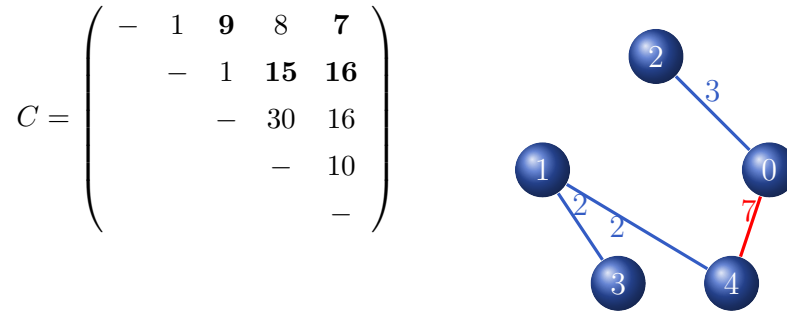


Figura 6.1: Árvore de suporte de peso mínimo incluindo a aresta $\{0, 4\}$ com custo 47 e peso 14.

Como a restrição de peso da árvore da Figura 6.1 não é violada, voltamos ao Passo 3.1 e aumentamos para $p = 2$ o número de arestas no conjunto $S = \{\{0, 1\}, \{0, 4\}\}$ e obtemos a seguinte árvore admissível com peso 18.

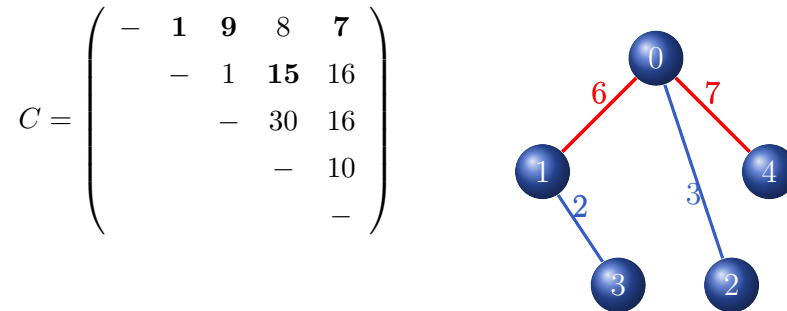


Figura 6.2: Árvore de suporte de peso mínimo incluindo as arestas $\{0, 1\}$ e $\{0, 4\}$ com custo 32 e peso 18.

A restrição de peso continua a ser verificada (ver Figura 6.2), o que nos leva a aumentar para $p = 3$ o número de arestas em S . Desta forma, $S = \{\{0, 1\}, \{0, 4\}, \{1, 2\}\}$ e obtemos uma solução não admissível com peso 21 que se encontra representada na figura seguinte.

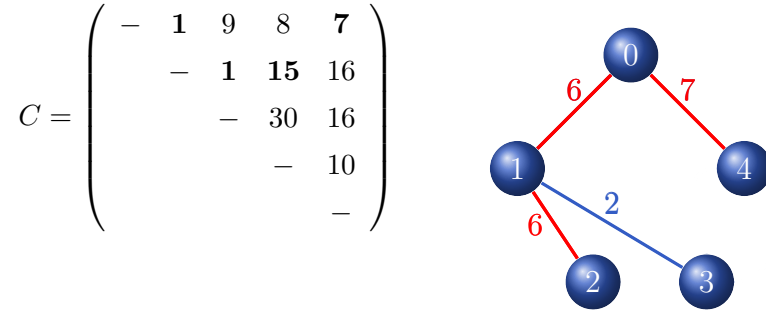


Figura 6.3: Árvore de suporte de peso mínimo incluindo as arestas $\{0, 1\}$, $\{0, 4\}$ e $\{1, 2\}$ com custo 24 e peso 21.

A restrição de peso não é verificada ($W(T) = 21 > 20 = W$), o que significa que foi encontrada a CI. Podemos assim construir a correspondente DCI,

$$x_{01} + x_{04} + x_{12} \leq 2.$$

Passo 4: Como a DCI obtida corta a solução fracionária x^* ($2,2 > 2$) é introduzida no modelo.

A informação respeitante às restantes iterações encontra-se resumida na Tabela 6.3.

Iter.	Sep1-DCI	
2	Passo 1	x^* : $x_{01} = 1$; $x_{03} \approx 0,1667$; $x_{12} = 1$; $x_{13} \approx 0,8333$ e $x_{34} = 1$ Custo = 25,8333 e peso = 20
	Passo 2, 3 e 4	Ordenação: OrdW DCI: $x_{01} + x_{03} + x_{12} \leq 2 \rightsquigarrow$ Inserir, corta x^* ($2,1667 > 2$).
3	Passo 1	x^* : $x_{01} = 0,875$; $x_{02} = 0,125$; $x_{03} = 0,125$; $x_{04} = 0,125$; $x_{12} = 0,875$; $x_{13} = 0,75$; $x_{21} = 0,125$; $x_{34} = 0,875$ e $x_{43} = 0,125$ Custo = 26,125 e peso = 20
	Passo 2, 3 e 4	Ordenação: OrdW DCI: $x_{01} + x_{03} + x_{04} + x_{12} \leq 3$. Não corta x^* ($2,125 \not\geq 3$).
	Passo 2, 3 e 4	Ordenação: OrdFixW DCI: $x_{02} + x_{12} + x_{13} + x_{34} \leq 3$. Não corta x^* ($2,875 \not\geq 3$).

Tabela 6.3: Principais resultados por iteração do Algoritmo Sep1 para as DCIs aplicado ao Exemplo 2.1.

Neste algoritmo são introduzidas no modelo duas DCIs, uma na Iteração 1 e uma na Iteração 2 ambas utilizando a ordenação OrdW. Como não é possível obter mais nenhuma DCI na Iteração 3, dado que se percorreram todas as ordenações, então o algoritmo termina e a solução obtida tem custo 26,125 e peso 20. Note-se que na Iteração 3 foram omitidos os passos correspondentes às ordenações que não formam nenhuma DCI.

Na Tabela 6.4 podemos observar as principais etapas do Algoritmo Sep2 para as DCIs aplicado ao Exemplo 2.1.

Iter.	Sep2-DCI	
1	Passo 1	x^* : $x_{01} = 1$; $x_{04} = 0,2$; $x_{12} = 1$; $x_{13} = 0,8$; $x_{34} = 0,8$ e $x_{43} = 0,2$ Custo = 25,4 e peso = 20
	Passo 2, 3 e 4	Ordenação: OrdW e OrdFixW DCI: $x_{01} + x_{04} + x_{12} \leq 2 \rightsquigarrow$ Inserir, corta x^* ($2,2 > 2$).
2	Passo 1	x^* : $x_{01} = 1$; $x_{03} \approx 0,1667$; $x_{12} = 1$; $x_{13} \approx 0,8333$ e $x_{34} = 1$ Custo = 25,8333 e peso = 20
	Passo 2, 3 e 4	Ordenação: OrdW DCI: $x_{01} + x_{03} + x_{12} \leq 2 \rightsquigarrow$ Inserir, corta x^* ($2,1667 > 2$).
	Passo 2, 3 e 4	Ordenação: OrdFixW DCI: $x_{01} + x_{03} + x_{12} + x_{34} \leq 3 \rightsquigarrow$ Inserir, corta x^* ($3,1667 > 3$).
3	Passo 1	x^* : $x_{01} = 0,875$; $x_{02} = 0,125$; $x_{03} = 0,125$; $x_{04} = 0,125$; $x_{12} = 0,875$; $x_{13} = 0,75$; $x_{21} = 0,125$; $x_{34} = 0,875$ e $x_{43} = 0,125$ Custo = 26,125 e peso = 20

Tabela 6.4: Principais resultados por iteração do Algoritmo Sep2 para as DCIs aplicado ao Exemplo 2.1.

No Algoritmo Sep2 são adicionadas três DCIs, sendo duas delas as introduzidas no Algoritmo Sep1.

Note-se que nas Iterações 1 e 2 foram omitidos os passos correspondentes às ordenações que não formam nenhuma DCI.

A solução obtida no Algoritmo Sep2 tem custo 26,125 e peso 20, sendo a mesma da obtida no Algoritmo Sep1.

6.1.3 Algoritmo de Separação para as DCIE

Para obter Desigualdades de Cobertura Implícita Estendida (DCIEs) temos de completar o Passo 3, para que quando se obtenha uma DCI se passe a obter uma correspondente DCIE.

Passo 3: Obter uma Desigualdade de Cobertura Implícita Estendida

Passo 3.1: Obter uma DCI

Seja $S = \{\}$.

Fazer $p = 1$ (inserir a primeira aresta de E_{RLOrk} no conjunto S).

Passo 3.1.1: Construir o conjunto S (sem ciclos)

Atualizar o conjunto S com as primeiras p arestas da lista E_{RLOrk} , desde que não formem ciclos.

Passo 3.1.2: Verificar se S é Cobertura Implícita

Obter a árvore de suporte de peso mínimo T , que inclui as p primeiras arestas de S .

Se $W(T) > W$, então

S é uma CI.

DCI: $\sum_{e \in S} x_e \leq |S| - 1$. Ir para o **Passo 3.2**.

Caso contrário,

Se $p < \#E_{RLOrk}$, então

$p = p + 1$.

Voltar ao **Passo 3.1.1**.

Caso contrário,

não é possível formar uma CI.

Passo 3.2: Obter uma DCIE

Calcular $w_{max} = \max\{w_f : f \in S\}$.

Obter o conjunto $S' = \{e \in E \setminus S : w_e \geq w_{max} \text{ e } S \cup \{e\} \text{ forma ciclo}\}$.

DCIE: $\sum_{e \in S} x_e + \sum_{e \in S'} x_e \leq |S| - 1$.

Quando no Passo 3.1 é obtida uma DCI, no Passo 3.2 temos apenas de obter o conjunto S' . Uma aresta pertence ao conjunto S' se não pertencer a S , se o seu peso for superior ou igual ao peso máximo das arestas do conjunto S e, ainda, se formar ciclo com as arestas do conjunto S . Desta forma obtemos uma DCIE.

No caso de aplicarmos os Algoritmos Sep1 e Sep2 para as DCIEs ao Exemplo 2.1 não é possível obter nenhuma DCIE. Assim para exemplificar estes algoritmos no caso das DCIEs temos de recorrer a um outro exemplo.

Nas Tabelas 6.5 e 6.6 podemos observar as principais etapas dos Algoritmos Sep1 e Sep2 para as DCIEs aplicado ao Exemplo 5.4.

Iter.	Sep1-DCIE	
1	Passo 1	x^* : $x_{01} = 1$; $x_{04} = 0,2$; $x_{12} = 1$; $x_{13} = 0,8$; $x_{34} = 0,8$ e $x_{43} = 0,2$ Custo = 25,4 e peso = 20
	Passo 2, 3 e 4	Ordenação: OrdW DCI: $x_{01} + x_{04} + x_{12} \leq 2$ DCIE: $x_{01} + x_{04} + x_{12} + x_{02} \leq 2 \rightsquigarrow$ Inserir, corta x^* ($2,2 > 2$).
2	Passo 1	x^* : $x_{01} = 1$; $x_{03} \approx 0,1667$; $x_{12} = 1$; $x_{13} \approx 0,8333$ e $x_{34} = 1$ Custo $\approx 25,8333$ e peso = 20
	Passo 2, 3 e 4	Ordenação: OrdW DCI: $x_{01} + x_{03} + x_{12} \leq 2 \rightsquigarrow$ Inserir, corta x^* ($2,1667 > 2$).
3	Passo 1	x^* : $x_{01} \approx 0,8571$; $x_{03} \approx 0,1429$; $x_{04} \approx 0,1429$; $x_{12} = 1$; $x_{13} \approx 0,7143$; $x_{31} \approx 0,1429$; $x_{34} \approx 0,8571$ e $x_{43} \approx 0,1429$ Custo $\approx 26,8571$ e peso = 20
	Passo 2, 3 e 4	Ordenação: OrdW DCI: $x_{03} + x_{04} \leq 1$. Não corta x^* ($0,2857 \not> 1$).
	Passo 2, 3 e 4	Ordenação: OrdFixW DCI: $x_{03} + x_{04} + x_{12} \leq 2$. Não corta x^* ($1,2857 \not> 2$).
	Passo 2, 3 e 4	Ordenação: OrdKnap DCI: $x_{03} + x_{04} + x_{13} + x_{34} \leq 3$. Não corta x^* ($2,1429 \not> 3$).

Tabela 6.5: Principais resultados por iteração do Algoritmo Sep1 para as DCIEs aplicado ao Exemplo 5.4.

Neste algoritmo são introduzidas duas desigualdades válidas, uma DCIE na Iteração 1 e uma DCI na Iteração 2, ambas obtidas usando a ordenação OrdW. Para a DCI obtida na Iteração 2 não foi possível obter uma DCIE. Como não é possível na Iteração 3 obter mais nenhuma DCI ou DCIE, dado que se percorreram todas as ordenações, então o algoritmo termina e a solução obtida tem custo 26,8571 e peso 20. Note-se que na Iteração 3 foram omitidos os passos correspondentes às ordenações que não formam

nenhuma DCI e, consequentemente, nenhuma DCIE.

Iter.	Sep2-DCIE	
1	Passo 1	x^* : $x_{01} = 1$; $x_{04} = 0,2$; $x_{12} = 1$; $x_{13} = 0,8$; $x_{34} = 0,8$ e $x_{43} = 0,2$ Custo = 25,4 e peso = 20
	Passo 2, 3 e 4	Ordenação: OrdW e OrdFixW DCI: $x_{01} + x_{04} + x_{12} \leq 2$ DCIE: $x_{01} + x_{04} + x_{12} + x_{02} \leq 2 \rightsquigarrow$ Inserir, corta x^* ($2,2 > 2$).
2	Passo 1	x^* : $x_{01} = 1$; $x_{03} \approx 0,1667$; $x_{12} = 1$; $x_{13} \approx 0,8333$ e $x_{34} = 1$ Custo $\approx 25,8333$ e peso = 20
	Passo 2, 3 e 4	Ordenação: OrdW DCI: $x_{01} + x_{03} + x_{12} \leq 2 \rightsquigarrow$ Inserir, corta x^* ($2,1667 > 2$).
	Passo 2, 3 e 4	Ordenação: OrdFixW DCI: $x_{01} + x_{03} + x_{12} + x_{34} \leq 3 \rightsquigarrow$ Inserir, corta x^* ($3,1667 > 3$).
	Passo 2, 3 e 4	Ordenação: OrdKnap DCI: $x_{03} + x_{12} + x_{13} + x_{34} \leq 3$. Não Inserir ($3 \not\leq 3$).
3	Passo 1	x^* : $x_{01} \approx 0,8571$; $x_{03} \approx 0,1429$; $x_{04} \approx 0,1429$; $x_{12} = 1$; $x_{13} \approx 0,7143$; $x_{31} \approx 0,1429$; $x_{34} \approx 0,8571$ e $x_{43} \approx 0,1429$ Custo $\approx 26,8571$ e peso = 20
	Passo 2, 3 e 4	Ordenação: OrdW DCI: $x_{03} + x_{04} \leq 1$. Não Inserir ($0,2857 \not\leq 1$).
	Passo 2, 3 e 4	Ordenação: OrdFixW DCI: $x_{03} + x_{04} + x_{12} \leq 2$. Não Inserir ($1,2857 \not\leq 2$).
	Passo 2, 3 e 4	Ordenação: OrdKnap DCI: $x_{03} + x_{04} + x_{13} + x_{34} \leq 3$. Não Inserir ($2,1429 \not\leq 3$).

Tabela 6.6: Principais resultados por iteração do Algoritmo Sep2 para as DCIEs aplicado ao Exemplo 5.4.

No Algoritmo Sep2 são adicionadas três desigualdades válidas, sendo uma delas uma DCIE e duas delas DCIs. Note-se que nas Iterações 1, 2 e 3 foram omitidos os passos correspondentes às ordenações que não formam nenhuma DCI e, consequentemente, nenhuma DCIE.

A solução obtida neste algoritmo tem custo 26,8571 e peso 20, sendo a mesma da obtida no Algoritmo Sep1.

6.1.4 Algoritmos de Separação para as DCIL

Para obter Desigualdades de Cobertura Implícita Levantada (DCILs) temos de especificar o Passo 3 de modo que depois de se obter uma DCI se venha a encontrar uma correspondente DCIL.

Passo 3: *Obter uma Desigualdade de Cobertura Implícita Levantada*

Passo 3.1: *Obter uma DCI*

Este passo é igual ao Passo 3.1 do Algoritmo para as DCIEs.

Passo 3.2: *Obter uma DCIL*

Calcular os coeficientes β_e , $e \in R \subseteq E \setminus S$.

$$\beta_e = \min \left\{ |S| - 1 - \sum_{f \in S} x_f - \sum_{f \in R} \beta_f x_f : x \in X_{WMST}, x_e = 1 \right\}.$$

$$\text{DCIL: } \sum_{e \in S} x_e + \sum_{e \in R} \beta_e x_e \leq |S| - 1.$$

No Passo 3.1, se for possível obter uma árvore de suporte de peso mínimo que contenha as arestas de S e não verifique a restrição de peso, obtemos uma DCI e no Passo 3.2 efetuamos o cálculo dos coeficientes para fazer o levantamento das variáveis do conjunto R de arestas que ainda não foram levantadas e que se pretendem levantar de modo a obter uma DCIL. Assim, para aplicar os algoritmos de separação ao caso das DCILs temos que ter em conta apenas a ordem de determinação dos coeficientes para efetuar o levantamento das variáveis. Recorde-se que as variáveis com $x_e > 0$, que estão na solução da relaxação linear encontram-se ordenadas pelas ordens propostas na Secção 6.1. As restantes variáveis com $x_e = 0$ na solução da relaxação linear encontram-se por uma ordem arbitrária.

Ordem de Determinação dos Coeficientes no Levantamento das Variáveis

- **Levantamento das Variáveis 1 (LV1)**

Efetuar o levantamento apenas das variáveis com $x_e > 0$, que estão na solução da relaxação linear de acordo com as ordenações propostas na Secção 6.1.

- **Levantamento das Variáveis 2 (LV2)**

Efetuar o levantamento de todas as variáveis. Sendo as variáveis com $x_e > 0$, que estão na solução da relaxação linear levantadas em primeiro lugar de acordo com as ordenações propostas na Secção 6.1. As restantes variáveis são levantadas numa ordem arbitrária.

- **Levantamento das Variáveis 3 (LV3)**

Efetuar o levantamento de todas as variáveis. Sendo as variáveis com $x_e > 0$, que estão na solução da relaxação linear levantadas em primeiro lugar de acordo com as ordenações propostas na Secção 6.1. As restantes variáveis são levantadas por ordem decrescente de peso.

Observe-se que de entre as $\frac{n(n+1)}{2}$ variáveis, podemos efetuar o levantamento de todas as variáveis que não pertençam à Cobertura S , isto é, necessitamos de determinar no máximo $\frac{n(n+1)}{2} - |S|$ coeficientes de variáveis para efetuar o seu levantamento. Para o cálculo de forma sequencial dos coeficientes existem $\left(\frac{n(n+1)}{2} - |S|\right)!$ maneiras diferentes de efetuar o levantamento das variáveis. No caso do Exemplo 2.1, em que temos 5 nodos e uma cobertura com $|S| = 3$ (Exemplo 6.2), existem $7! = 5040$ modos diferentes de efetuar o levantamento das variáveis, o que significa que dependendo da ordem de levantamento das variáveis também podem ser obtidas diferentes DCILs.

Note-se que no caso dos coeficientes de todas as variáveis a efetuar o levantamento serem nulos, não temos uma DCIL, mas temos uma DCI, a qual poderá ser inserida no modelo, caso corte a solução fracionária, x^* .

Caso a DCI contenha apenas uma aresta, a DCI correspondente é da forma $x_e \leq 0$. Neste caso não faz sentido efetuar o levantamento das variáveis, uma vez que o cálculo dos coeficientes β_f será efetuado resolvendo o seguinte problema,

$$\beta_f = \min\{-x_e : x \in X_{WMST}, x_f = 1\}.$$

Os possíveis valores destes coeficientes são -1 ou 0 , mas como $\beta_f \geq 0$, então neste caso não é necessário determinar os coeficientes, pois estes são todos nulos.

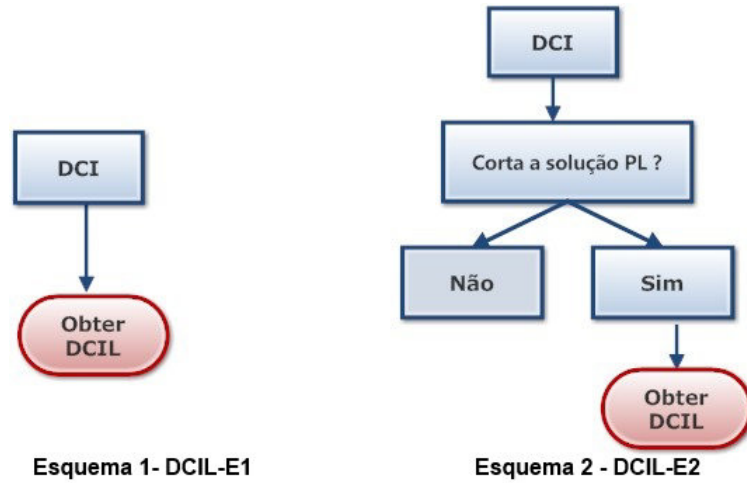


Figura 6.4: Esquemas de obtenção de uma DCIL dada uma DCI.

Na Figura 6.4 podemos observar duas estratégias para a formação de DCILs. No Passo 3 do algoritmo de separação proposto para as DCILs, gera-se uma DCIL sempre que no Passo 3.1 é obtida uma DCI (Esquema 1 - DCIL-E1 da Figura 6.4). Evidenciamos no Esquema 2- DCIL-E2 da Figura 6.4 que podemos só gerar uma DCIL apenas no caso da DCI obtida no Passo 3.1 cortar a solução fracionária, x^* . Caso esta não corte, não é obtida nenhuma DCIL. Este facto pode ser vantajoso reduzindo o tempo de execução do algoritmo de separação.

Se o cálculo dos vários coeficientes das variáveis a fazer levantamento for efetuado usando um algoritmo exato pode demorar muito tempo, dado que é necessária a resolução de no máximo $\frac{n(n+1)}{2} - |S|$ subproblemas, no caso de se efetuar o levantamento de todas as variáveis.

Para diminuir o número de subproblemas, não vamos efetuar o levantamento usual de variáveis nos seguintes casos:

- Quando a aresta correspondente à variável a efetuar levantamento pertencer à árvore de suporte de peso mínimo que contém as arestas do conjunto S ;
- Quando a variável a efetuar levantamento não verificar a seguinte restrição,

$$w_{ij} + (n - 2)p_{min} \leq W, \quad (6.4)$$

onde p_{min} é o peso mínimo de entre todos os pesos dos arcos $(i, j) \in A$.

Para explicar a razão da condição dada por (6.4), podemos pensar que nos vários subproblemas a solução obtida corresponde a uma árvore de suporte que verifica a restrição de peso, isto é, $\sum_{e \in E_T} w_e x_e \leq W$. Como no caso do levantamento usual a aresta $\{i, j\}$ correspondente à variável que se pretende efetuar o levantamento tem de estar na árvore, então a variável x_{ij} toma o valor um, pelo que obtemos $w_{ij} + \sum_{e \in E_T \setminus \{\{i, j\}\}} w_e x_e \leq W$. Assim ficam a faltar introduzir na árvore $n - 2$ arestas e admitindo que estas arestas a introduzir têm peso mínimo p_{min} obtemos a seguinte restrição $w_{ij} + (n - 2)p_{min} \leq W$.

Para determinar os coeficientes β_e de forma sequencial vamos descrever um algoritmo aproximado, baseado em relaxação Lagrangeana que é adaptado do Algoritmo Alg2 descrito na Secção 4.4. Aproveitamos para descrever um algoritmo genérico que para além de efetuar o levantamento usual de variáveis também efetua os chamados *down-lifting* e *up-lifting*.

Cálculo dos coeficientes das variáveis a efetuar levantamento

Vamos considerar que pretendemos efetuar o levantamento da variável x_t . Seja $R \subseteq E_0 \cup E_1 \cup E_f$ o conjunto de variáveis que já foram sujeitas a levantamento, onde E_1 é o conjunto de variáveis fixas a um, E_0 é o conjunto de variáveis fixas a zero e $E_f = E \setminus (E_0 \cup E_1 \cup S)$. Para calcular o coeficiente β_t necessitamos de um algoritmo genérico aproximado que resolva o seguinte problema,

$$\beta_t = \left\{ |S| - 1 - \sum_{e \in S} x_e - \sum_{e \in R \cap (E_0 \cup E_f)} \beta_e x_e - \sum_{e \in R \cap E_1} \beta_e (1 - x_e) : x \in X(t) \right\}, \quad (6.5)$$

onde

$$X(t) = \left\{ x \in X_{WMST} : x_e = 1, e \in E_1 \setminus R, x_e = 0, e \in E_0 \setminus R, x_t = 1 \right\}, \quad (6.6)$$

$t \in E_f \setminus R$;

$$X(t) = \left\{ x \in X_{WMST} : x_e = 1, e \in E_1 \setminus (R \cup \{t\}), x_e = 0, e \in E_0 \setminus R, x_t = 0 \right\}, \quad (6.7)$$

$t \in E_1 \setminus R$;

$$X(t) = \left\{ x \in X_{WMST} : x_e = 1, e \in E_1 \setminus R, x_e = 0, e \in E_0 \setminus (R \cup \{t\}), x_t = 1 \right\}, \quad (6.8)$$

$t \in E_0 \setminus R$.

Recordamos que caso se utilize o conjunto definido por (6.6) estamos a efetuar o levantamento usual de variáveis, no caso de utilizarmos o conjunto definido por (6.7) estamos a efetuar o chamado *down-lifting* e no caso de utilizarmos o conjunto definido por (6.8) estamos a efetuar o chamado *up-lifting*. Para resolver o problema (6.5) temos que minimizar a função objetivo $|S| - 1 - \sum_{e \in S} x_e - \sum_{e \in R \cap (E_0 \cup E_f)} \beta_e x_e - \sum_{e \in R \cap E_1} \beta_e (1 - x_e)$. Tendo por base esta função objetivo vamos considerar uma matriz de custos modificados. Denotamos os custos modificados de cada arco $(i, j) \in A$ por \tilde{c}_{ij} . Como os custos dos arcos $(i, j) \in A$ são iguais aos custos das arestas $\{i, j\} \in E$, para simplificar a notação, definimos apenas os custos modificados das arestas $e = \{i, j\}$ do seguinte modo

$$\tilde{c}_e = \tilde{c}_{ij} = \begin{cases} |S| - 2 - \sum_{f \in R \cap E_1} \beta_f, & \text{se } e \in S; \\ |S| - 1 - \beta_e - \sum_{f \in R \cap E_1} \beta_f, & \text{se } e \in R \cap (E_0 \cup E_f); \\ |S| - 1 - \sum_{f \in (R \cap E_1) \setminus \{e\}} \beta_f, & \text{se } e \in R \cap E_1. \end{cases}$$

Inicialmente quando todos os coeficientes β_e das variáveis a efetuar levantamento forem nulos, pretende-se que o maior número de arestas de S esteja na solução, pois queremos minimizar a função objetivo. Os custos modificados vão sendo sucessivamente alterados uma vez que dependem dos valores dos coeficientes das variáveis já levantadas. Sempre que se determina um novo coeficiente os custos são alterados e têm em conta o novo valor obtido.

Para resolver o problema dado em (6.5) vamos usar a relaxação Lagrangeana. Assim, associamos à restrição de peso um multiplicador de Lagrange λ ($\lambda \geq 0$) e adicionando o produto do multiplicador pela restrição de peso à função objetivo do problema (6.5) e obtemos o seguinte problema relaxado

$$\beta_t = \min \{h(S, R, \lambda, \beta) : x \in X_L(t)\}, \quad (6.9)$$

onde

$$h(S, R, \lambda, \beta) = -\lambda W + |S| - 1 - \sum_{e \in S} x_e - \sum_{e \in R \cap (E_0 \cup E_f)} \beta_e x_e - \sum_{e \in R \cap E_1} \beta_e (1 - x_e) + \sum_{e \in E} \lambda w_e x_e,$$

e o conjunto $X_L(t)$ é definido por um dos três conjuntos (6.6), (6.7) ou (6.8) com apenas uma alteração, em vez de $x \in X_{WMST} = X_T \cap X_K$, apenas se tem de verificar que as soluções pertencem ao conjunto formado pelas árvores de suporte, isto é, $x \in X_T$.

Os valores ponderados para cada arco $(i, j) \in A$ podem ser definidos do seguinte modo

$$p_{ij} = \tilde{c}_{ij} + \lambda w_{ij},$$

onde o multiplicador de Lagrange, λ ($\lambda \geq 0$) é dado por

$$\lambda = \frac{\tilde{C}_M(T_\alpha) - \tilde{C}_M(T_\mu)}{W(T_\mu) - W(T_\alpha)}, \quad (6.10)$$

sendo T_α uma árvore de suporte admissível e T_μ uma árvore de suporte não admissível para o Problema WMST.

Antes da apresentação do algoritmo heurístico para calcular o coeficiente das variáveis a efetuar levantamento, temos de ter em conta que, dependendo do tipo de levantamento que se está a efetuar (levantamento usual, *down-lifting* ou *up-lifting*), é necessário definir alguns critérios para a construção das árvores de suporte.

Critérios para obter árvores

- Critério de levantamento usual

A árvore de suporte tem de conter a aresta t .

- Critério de *down-lifting*

A árvore de suporte não pode conter a aresta t , mas tem de conter as arestas de $E_1 \setminus R$.

- Critério de *up-lifting*

A árvore de suporte tem de conter a aresta t e não conter as arestas de $E_0 \setminus R$.

Deste modo a seguir encontra-se, de forma resumida, um algoritmo heurístico para calcular os coeficientes das variáveis a efetuar o levantamento. Notemos que de cada vez que a execução deste algoritmo é chamada encontramos o valor de um coeficiente β_t .

Algoritmo heurístico para o cálculo dos coeficientes das variáveis a efetuar levantamento (HCoef)

Passo 1: *Inicializações*

Passo 1.1: *Obter um limite superior para β_t*

Obter, se possível, a árvore de suporte de peso mínimo, T_{w_I} , que verifica uma das três condições seguintes:

Se $t \in E_f$, então T_{w_I} verifica o Critério de levantamento usual.

Se $t \in E_1$, então T_{w_I} verifica o Critério de *down-lifting*.

Se $t \in E_0$, então T_{w_I} verifica o Critério de *up-lifting*.

Se T_{w_I} existir, então

Se $W(T_{w_I}) > W$, então

$\beta_t = 0$, STOP.

Caso contrário,

$T_\alpha = T_{w_I}$.

Se $t \in E_f$ e $\tilde{C}_M(T_\alpha) \leq 0$, então

$\beta_t = 0$, STOP.

Caso contrário,

$\beta_t = 0$, STOP.

Passo 1.2: Obter um limite inferior para β_t

Obter, se possível, a árvore de suporte de custo modificado mínimo, T_{c_I} , que verifica uma das três condições seguintes:

Se $t \in E_f$, então T_{c_I} verifica o Critério de levantamento usual.

Se $t \in E_1$, então T_{c_I} verifica o Critério de *down-lifting*.

Se $t \in E_0$, então T_{c_I} verifica o Critério de *up-lifting*.

Se T_{c_I} existir, então

Se $W(T_{c_I}) \leq W$, então

T_{c_I} , é a árvore correspondente à solução ótima.

$\beta_t = \tilde{C}_M(T_{c_I})$, STOP.

Caso contrário,

$T_\mu = T_{c_I}$.

Caso contrário,

$\beta_t = 0$, STOP.

Passo 2: Obter uma nova árvore

Calcular os valores ponderados $p_{ij} = \tilde{c}_{ij} + \lambda w_{ij}$ para cada arco $(i, j) \in A$.

Obter, se possível, a árvore de suporte ponderada mínima, T_{p_I} , que verifica uma das três condições seguintes:

Se $t \in E_f$, então T_{p_I} verifica o Critério de levantamento usual.

Se $t \in E_1$, então T_{p_I} verifica o Critério de *down-lifting*.

Se $t \in E_0$, então T_{p_I} verifica o Critério de *up-lifting*.

Calcular $\tilde{C}_M(T_{p_I})$, $W(T_{p_I})$ e $P(T_{p_I})$.

Se T_{p_I} não existir, então

$\beta_t = 0$, STOP.

Passo 3: Atualização de limites

Se $W(T_{p_I}) \leq W$, então

atualizar o LS, isto é, se $\tilde{C}_M(T_{p_I}) \leq \tilde{C}_M(T_\alpha)$ substituir T_α por T_{p_I} .

Se $t \in E_f$ e $\tilde{C}_M(T_\alpha) \leq 0$, então

$\beta_t = 0$, STOP.

Caso contrário,

atualizar o LI, isto é, se $\tilde{C}_M(T_{p_I}) \geq \tilde{C}_M(T_\mu)$ substituir T_μ por T_{p_I} .

Se $\tilde{C}_M(T_\alpha) = \tilde{C}_M(T_\mu)$, então

$\beta_t = \tilde{C}_M(T_\alpha)$, STOP.

Passo 4: Critério de paragem

Se $|P(T_\alpha) - P(T_{p_I})| \leq tol$, então

T_α é a árvore correspondente à solução aproximada.

Calcular β_t , usando a relaxação Lagrangeana dada por (6.9), STOP.

Se $t \in E_f$ e $\beta_t < 0$, então $\beta_t = 0$.

Caso contrário,

ir para o **Passo 2**.

Notamos que inicialmente todos os coeficientes β_e são nulos. No Passo 1.1 é obtida a árvore de suporte de peso mínimo que verifica determinado critério dependendo do procedimento de levantamento que se está a efetuar (levantamento usual, *down-lifting* ou *up-lifting*), a qual se denota por T_{w_I} , onde $\tilde{C}_M(T_{w_I})$ e $W(T_{w_I})$ são, respetivamente, o custo modificado e o peso da árvore T_{w_I} . Se esta árvore existir e se não verificar a restrição de peso, então não existe solução e o algoritmo termina, sendo $\beta_t = 0$. Caso a restrição de peso seja verificada, então foi encontrado um limite superior (LS) para o custo modificado e $T_\alpha = T_{w_I}$. Neste caso, se estivermos a efetuar o levantamento usual ($t \in E_f$) e o custo modificado da árvore T_α for inferior ou igual a zero, então $\beta_t = 0$ e o algoritmo termina. No caso de não ser possível formar uma árvore T_{w_I} o algoritmo termina, sendo $\beta_t = 0$.

No Passo 1.2 obtém-se a árvore de suporte de custo modificado mínimo que verifica determinado critério dependendo do procedimento de levantamento que se está a efetuar (levantamento usual, *down-lifting* ou *up-lifting*), a qual é designada por T_{c_I} , onde $\tilde{C}_M(T_{c_I})$ e $W(T_{c_I})$ são, respetivamente, o custo modificado e o peso da árvore T_{c_I} . Se esta árvore não verificar a restrição de peso, então foi encontrado um limite inferior (LI) para o custo modificado e $T_\mu = T_{c_I}$. No caso de verificação da restrição de peso significa que foi encontrada a solução ótima e o algoritmo termina, sendo $\beta_t = \tilde{C}_M(T_{c_I})$. No caso de não ser possível formar uma árvore T_{c_I} o algoritmo termina, sendo $\beta_t = 0$.

No Passo 2 calculam-se os valores ponderados p_{ij} para cada arco $(i, j) \in A$. De seguida é obtida a árvore de suporte ponderada mínima que verifica determinado critério dependendo do procedimento de levantamento que se está a efetuar (levantamento usual, *down-lifting* ou *up-lifting*), a qual é denotada por T_{p_I} , onde $\tilde{C}_M(T_{p_I})$ e $W(T_{p_I})$ são, respetivamente, o custo modificado e o peso da árvore T_{p_I} . Assim podemos definir os valores ponderados $P(T_{p_I}) = \tilde{C}_M(T_{p_I}) + \lambda W(T_{p_I})$ para um $\lambda \geq 0$. No caso de não ser possível formar uma árvore T_{p_I} o algoritmo termina, sendo $\beta_t = 0$.

A árvore T_{p_I} é atualizada no Passo 3. No caso da restrição de peso ser verificada e o custo modificado da árvore T_{p_I} não ser superior ao custo modificado da árvore T_α , então atualizamos o limite superior (LS) e substituímos T_α por T_{p_I} , nesta situação temos ainda de averiguar no caso de estarmos a efetuar o levantamento usual ($t \in E_f$) se o custo modificado que atualizámos é inferior ou igual a zero e neste caso o algoritmo termina com $\beta_t = 0$. Caso a restrição de peso seja violada e o custo modificado da árvore T_{p_I} não for inferior ao custo modificado da árvore T_μ , então atualizamos o limite inferior (LI) e substituímos T_μ por T_{p_I} . Após se ter efetuado a atualização dos limites podemos verificar se as duas árvores T_α e T_μ têm os mesmos custos modificados e em

caso afirmativo o algoritmo termina com $\beta_t = \tilde{C}_M(T_\alpha)$.

Os Passos 2 e 3 são executados até que se verifique $|P(T_\alpha) - P(T_{p_I})| \leq tol$, e assim o algoritmo termina, sendo T_α a árvore correspondente à solução aproximada e β_t é obtido usando a relaxação Lagrangeana dada por (6.9). No caso do coeficiente β_t ser um número fracionário arredonda-se o seu valor para o inteiro mais próximo. Se estivermos a efetuar o levantamento usual de variáveis e se o coeficiente β_t for negativo, então considera-se $\beta_t = 0$.

No exemplo que se segue aplicamos o Algoritmo Sep1 para as DCILs ao Exemplo 2.1 e exemplificamos também como se efetua o cálculo de alguns coeficientes das variáveis a efetuar levantamento usando o Algoritmo HCoef.

Exemplo 6.3.

Dada a DCI, $x_{01} + x_{04} + x_{12} \leq 2$ obtida na Iteração 1 do Exemplo 6.2. Podemos efetuar o levantamento das restantes variáveis recorrendo à ordenação que está a ser utilizada. Assim a lista de arestas da solução da relaxação linear é dada por $E_{RLOrdW} = [\{0, 4\}, \{0, 1\}, \{1, 2\}, \{3, 4\}, \{1, 3\}]$. Como as três primeiras arestas já se encontram na DCI, então vamos fazer o levantamento das restantes arestas de E_{RLOrdW} utilizando o Algoritmo HCoef para o cálculo dos coeficientes das variáveis a efetuar levantamento.

Para exemplificar o funcionamento do Algoritmo HCoef vamos calcular, detalhadamente, e de forma sequencial dois coeficientes. No levantamento da variável x_{34} calcula-se o coeficiente β_{34} do seguinte modo,

$$\beta_{34} = \min\{2 - x_{01} - x_{04} - x_{12} : x \in X_{WMST}, x_{34} = 1\}.$$

No Passo 1 do Algoritmo HCoef são obtidas as seguintes árvores,

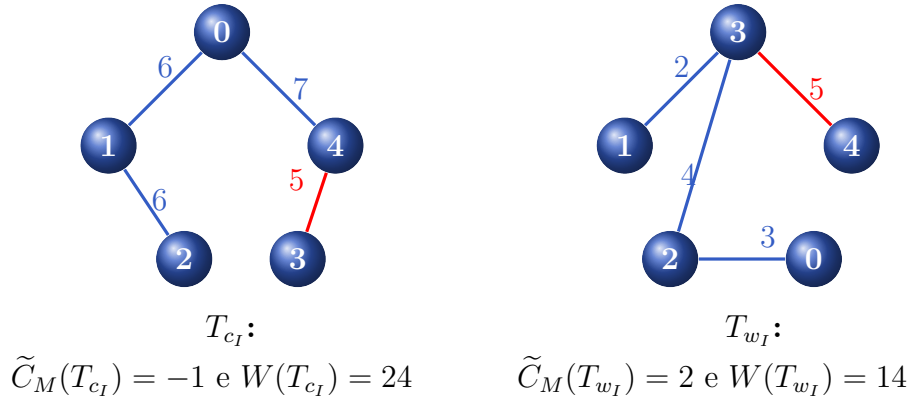


Figura 6.5: Árvores de suporte T_{c_I} e T_{w_I} obtidas no Passo 1 do Algoritmo HCoef no cálculo do coeficiente de levantamento usual β_{34} .

Ambas as árvores da Figura 6.5 contêm a aresta $\{3, 4\}$ correspondente à variável da qual se pretende efetuar o levantamento. A árvore T_{w_I} foi obtida no Passo 1.1 e como tem peso 14, verifica a restrição de peso, assim $T_\alpha = T_{w_I}$. No Passo 1.2 é obtida a árvore T_{c_I} que não verifica a restrição de peso, então $T_\mu = T_{c_I}$. Deste modo obtemos um limite inferior e um limite superior para o coeficiente β_{34} , ou seja,

$$-1 \leq \beta_{34} \leq 2.$$

No Passo 2 é obtida a árvore de suporte ponderada mínima que contém a aresta $\{3, 4\}$ (ver Figura 6.6).

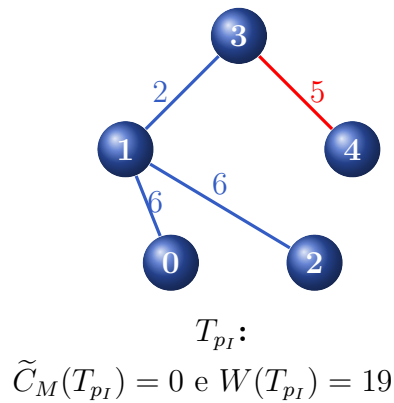


Figura 6.6: Árvore de suporte T_{p_I} obtida no Passo 2 do Algoritmo HCoef no cálculo do coeficiente de levantamento usual β_{34} .

No Passo 3 é atualizado o LS, pois $\tilde{C}_M(T_{p_I}) = 0 \leq 2 = \tilde{C}_M(T_\alpha)$, mas como $\tilde{C}_M(T_\alpha) = 0$, então $\beta_{34} = 0$.

Para calcular o coeficiente β_{13} resolve-se o seguinte problema

$$\beta_{13} = \min\{2 - x_{01} - x_{04} - x_{12} : x \in X_{W_{MST}}, x_{13} = 1\}.$$

Como a aresta $\{1, 3\}$ pertence à árvore de suporte de peso mínimo que contém as arestas de S (Figura 6.3), então $\beta_{13} = 0$.

Note-se que, no caso de aplicarmos o Levantamento de Variáveis LV1, não se consegue obter nenhuma DCIL.

Para calcular os restantes coeficientes das variáveis que não pertencem à solução da relaxação linear temos de as ordenar, por exemplo, por ordem decrescente de peso (Levantamento de Variáveis LV3). Assim, como $w_{03} > w_{24} > w_{23} > w_{02} > w_{14}$ vamos começar por determinar o coeficiente da variável x_{03} resolvendo o seguinte problema

$$\beta_{03} = \min\{2 - x_{01} - x_{04} - x_{12} : x \in X_{W_{MST}}, x_{03} = 1\}.$$

Para determinar o coeficiente β_{03} , no Passo 1 do Algoritmo HCoef são obtidas as seguintes árvores,

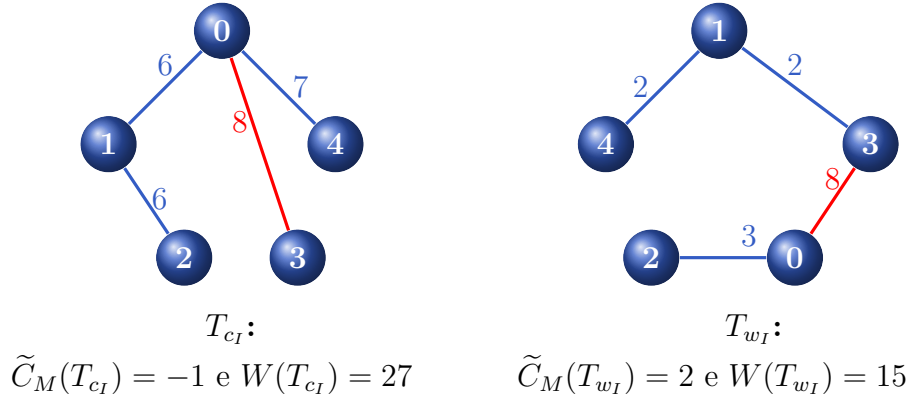


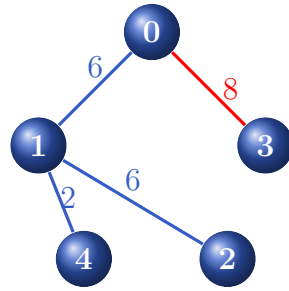
Figura 6.7: Árvores de suporte T_{c_I} e T_{w_I} obtidas no Passo 1 do Algoritmo HCoef no cálculo do coeficiente de levantamento usual β_{03} .

Ambas as árvores da Figura 6.7 contêm a aresta $\{0, 3\}$ correspondente à variável da qual se pretende efetuar o levantamento. A árvore T_{w_I} foi obtida no Passo 1.1 e como tem peso 15, verifica a restrição de peso, assim $T_\alpha = T_{w_I}$. No Passo 1.2 é obtida

a árvore T_{c_I} que não verifica a restrição de peso, então $T_\mu = T_{c_I}$. Deste modo obtemos um limite inferior e um limite superior para o coeficiente β_{03} , ou seja,

$$-1 \leq \beta_{03} \leq 2.$$

No Passo 2 é obtida a árvore de suporte ponderada mínima que contém a aresta $\{0, 3\}$ (ver Figura 6.8).



T_{p_I} :

$$\tilde{C}_M(T_{p_I}) = 0 \text{ e } W(T_{p_I}) = 22$$

Figura 6.8: Primeira árvore de suporte T_{p_I} obtida no Passo 2 do Algoritmo HCoef no cálculo do coeficiente de levantamento usual β_{03} .

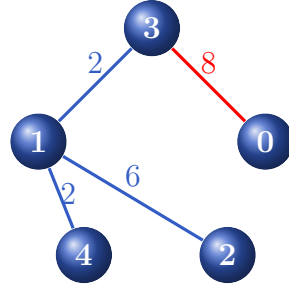
No Passo 3 é atualizado o LI, pois

$$W(T_{p_I}) = 22 > 20 = W \text{ e } \tilde{C}_M(T_{p_I}) = 0 \geq -1 = \tilde{C}_M(T_\mu).$$

Assim temos que

$$0 \leq \beta_{03} \leq 2.$$

Volta-se ao Passo 2, onde é obtida a nova árvore de suporte ponderada mínima que contém a aresta $\{0, 3\}$ e que se encontra representada na figura seguinte,



T_{p_I} :

$$\tilde{C}_M(T_{p_I}) = 1 \text{ e } W(T_{p_I}) = 18$$

Figura 6.9: Segunda árvore de suporte T_{p_I} obtida no Passo 2 do Algoritmo HCoef no cálculo do coeficiente de levantamento usual β_{03} .

Agora no Passo 3 é atualizado o LS, pois

$$W(T_{p_I}) = 18 \leq 20 = W \text{ e } \tilde{C}_M(T_{p_I}) = 1 \leq 2 = \tilde{C}_M(T_\alpha).$$

Assim temos que

$$0 \leq \beta_{03} \leq 1.$$

Voltando ao Passo 3 obtém-se a mesma árvore de suporte da Figura 6.9. Como o critério de paragem do algoritmo é verificado, então

$$\beta_{03} = 0,25 \times 20 + 1 + 0,25 \times 18 = 0,5.$$

Efetuando o arredondamento para o inteiro mais próximo obtemos

$$\beta_{03} = 1.$$

De seguida calculam-se os restantes coeficientes de forma sequencial usando também o Algoritmo HCoef.

$$\beta_{24} = \min\{2 - x_{01} - x_{04} - x_{03} - x_{12} : x \in X_{WMST}, x_{24} = 1\} = 0;$$

$$\beta_{23} = \min\{2 - x_{01} - x_{04} - x_{03} - x_{12} : x \in X_{WMST}, x_{23} = 1\} = 0;$$

$$\beta_{02} = \min\{2 - x_{01} - x_{04} - x_{03} - x_{12} : x \in X_{WMST}, x_{02} = 1\} = 0;$$

$$\beta_{14} = \min\{2 - x_{01} - x_{04} - x_{03} - x_{12} : x \in X_{WMST}, x_{14} = 1\} = 0.$$

Logo, a DCIL obtida é dada por:

$$x_{01} + x_{04} + x_{12} + \mathbf{x_{03}} \leq 2.$$

Esta desigualdade corta a solução fracionária ($2,2 > 2$). Após a introdução da DCIL no modelo obtém-se a nova solução da relaxação linear do Procedimento P-WMTZ+C $x_{01} = 1$; $x_{12} = 1$; $x_{13} = 1$ e $x_{34} = 1$, com custo de 27 e peso 20. Como esta solução é inteira obtivemos a solução ótima da instância com apenas a introdução de um corte.

Se aplicarmos o Algoritmo Sep2 ao Exemplo de 5 nodos apenas se forma a DCIL obtida no Algoritmo Sep1.

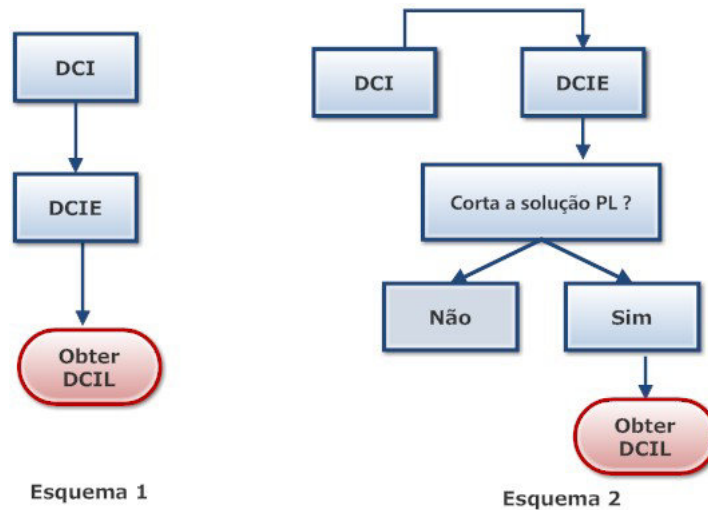


Figura 6.10: Esquemas de obtenção de uma DCIL dada uma DCIE.

Em vez de efetuarmos o levantamento das variáveis de uma DCI podemos obter a DCIE e depois efetuar o levantamento da DCIE (Esquema 1 da Figura 6.10). No Esquema 2 podemos observar que após a obtenção da DCI obtemos uma DCIE e se esta cortar a solução fracionária, x^* , então efetuamos o levantamento das restantes variáveis e obtemos a DCIL. No caso da DCIE não cortar a solução fracionária, não obtemos nenhuma DCIL. O Esquema 2 poderá ser mais vantajoso em termos de tempo de execução.

6.1.5 Algoritmo de Separação para as DCILDL

Para obter Desigualdades de Cobertura Implícita Levantada por *Down-Lifting* (DCILDLs) temos de alterar o Passo 3 de modo a obter uma CI válida para o conjunto restrito $X^{E_1} = X_{W_{MST}} \cap \{x : x_f = 1, f \in E_1\}$ e a correspondente DCILDL.

Passo 3: Obter uma Desigualdade de Cobertura Implícita Levantada por *Down-Lifting*

Seja E_1 o conjunto de variáveis fixas a um.

Passo 3.1: Obter uma DCI válida para o conjunto restrito X^{E_1}

Seja $S = \{\}$.

Fazer $p = 1$ (inserir em S , a primeira aresta de E_{RLOrdk} tal que $e \notin E_1$).

Passo 3.1.1: Construir o conjunto S (sem ciclos)

Atualizar o conjunto S com as primeiras p arestas de E_{RLOrdk} , desde que não formem ciclos e que não pertençam ao conjunto E_1 .

Passo 3.1.2: Verificar se S é CI para o conjunto restrito X^{E_1}

Obter a árvore de suporte de peso mínimo T , que inclui as p primeiras arestas de S e as arestas de E_1 .

Se $W(T) > W$, então

S é uma CI para o conjunto restrito X^{E_1} .

DCI: $\sum_{e \in S} x_e \leq |S| - 1$. Ir para o **Passo 3.2**.

Caso contrário,

Se $p < \#E_{RLOrdk}$, então

$p = p + 1$.

Voltar ao **Passo 3.1.1**.

Caso contrário,

não é possível formar uma CI.

Passo 3.2: Obter uma DCILDL

Calcular os coeficientes β_e , $e \in R \subseteq E_1 \cup E_f$.

$$\beta_e = \min \left\{ |S| - 1 - \sum_{t \in S} x_t - \sum_{t \in R \cap E_f} \beta_t x_t - \sum_{t \in R \cap E_1} \beta_t (1 - x_t) : x \in X(e) \right\},$$

- $e \in E_f \setminus R$

$$X(e) = \left\{ x \in X_{WMST} : x_t = 1, t \in E_1 \setminus R, x_e = 1 \right\};$$

- $e \in E_1 \setminus R$

$$X(e) = \left\{ x \in X_{WMST} : x_t = 1, t \in E_1 \setminus (R \cup \{e\}), x_e = 0 \right\}.$$

$$\text{DCILDL: } \sum_{e \in S} x_e + \sum_{e \in R \cap E_f} \beta_e x_e + \sum_{e \in R \cap E_1} \beta_e (1 - x_e) \leq |S| - 1.$$

Inicialmente no Passo 3 consideramos um conjunto de variáveis fixas a um, que designamos por E_1 , depois no Passo 3.1 obtemos uma DCI válida para o conjunto restrito X^{E_1} e no Passo 3.2 efetuamos o cálculo dos coeficientes para fazer o levantamento das variáveis de E_1 e de $E_f = E \setminus (S \cup E_1)$ e obter a DCILDL. Assim, para aplicar os algoritmos de separação ao caso das DCILDLs temos que ter em conta os três fatores que se seguem:

- A escolha do conjunto de variáveis a fixar a um (conjunto E_1);
- A ordem dos procedimentos a adotar para efetuar o levantamento das variáveis;
- A ordem de determinação dos coeficientes para efetuar o levantamento das variáveis.

Escolha do Conjunto de Variáveis a Fixar a um, E_1

Quando fixamos variáveis a um estamos a fazer com que essas variáveis façam parte da solução. Segundo Gu et. al. [29] devem-se selecionar as variáveis com valores mais elevados na solução da relaxação linear. Assim, propomos os seguintes modos de escolha do conjunto E_1 :

- **Fixação de Variáveis D1 (FVD1)**

Fixar a um as variáveis que têm valor um na solução da relaxação linear.

- **Fixação de Variáveis D2 (FVD2)**

Fixar a um as variáveis que têm valor um na solução da relaxação linear e que não pertencem à árvore de suporte de peso mínimo, T_w .

- **Fixação de Variáveis D3 (FVD3)**

Fixar a um uma única variável, a variável de menor peso com valor um na solução da relaxação linear e que não pertence à árvore de suporte de peso mínimo, T_w .

Ordem de Determinação dos Coeficientes no Levantamento das Variáveis

- **Procedimento de Levantamento LD (PL-LD)**

Primeiro efetuamos o levantamento sequencial usual e depois o *down-lifting*.

- **Procedimento de Levantamento DL (PL-DL)**

Primeiro efetuamos o *down-lifting* e depois o levantamento sequencial usual.

Ordem de Determinação dos Coeficientes no Levantamento das Variáveis

Para efetuar o levantamento sequencial usual podemos usar as mesmas ordens propostas na Subsecção 6.1.4. No *down-lifting* ordenamos as variáveis por ordem crescente de peso. Podemos efetuar primeiro o levantamento das variáveis com menor peso ou com maior peso.

Note-se que:

- Quando o número de variáveis a fixar a um for nulo ($E_1 = \emptyset$), não se aplica o procedimento *down-lifting* e aplica-se apenas o procedimento de levantamento sequencial usual, se a DCI válida para o conjunto restrito X^{E_1} cortar a solução fracionária x^* ;
- No caso da DCI válida para o conjunto restrito X^{E_1} conter apenas uma variável, então faz-se apenas o *down-lifting*, uma vez que se efetuarmos o levantamento sequencial usual os coeficientes são todos nulos.

Para exemplificar o Algoritmo HCoef vamos calcular um coeficiente *down-lifting* para o Exemplo 2.1.

Exemplo 6.4.

Dada a DCI, $x_{04} \leq 0$ válida para o conjunto restrito $X^{E_1} = X_{WMST} \cap \{x_e : x_{01} = x_{12} = 1\}$. Para fazer o levantamento da variável x_{12} e calcular o coeficiente β_{12} temos de resolver o seguinte problema

$$\beta_{12} = \min\{-x_{04} : x \in X_{WMST}, x_{12} = 0, x_{01} = 1\}.$$

Para determinar este coeficiente, no Passo 1 do Algoritmo HCoef são obtidas duas árvores, as quais contêm a aresta $\{0, 1\}$ e não contêm a aresta $\{1, 2\}$ (ver Figura 6.11).

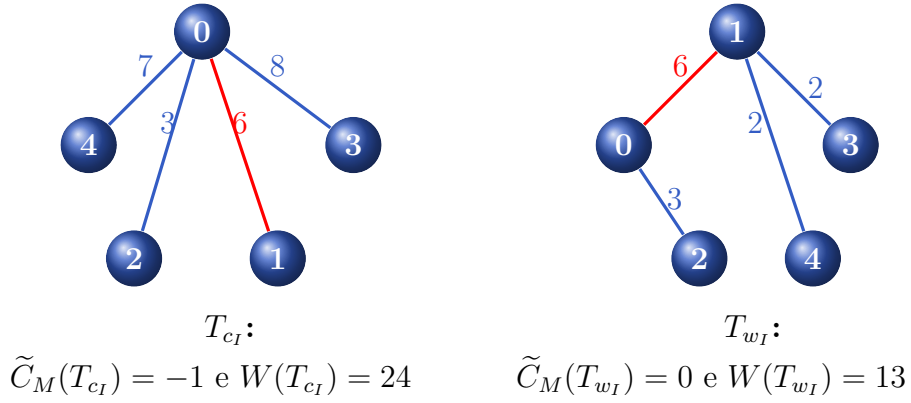
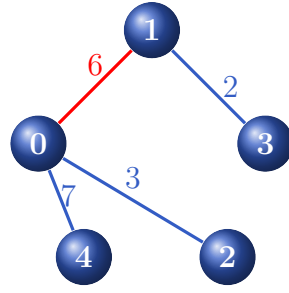


Figura 6.11: Árvores de suporte T_{c_I} e T_{w_I} obtidas no Passo 1 do Algoritmo HCoef no cálculo do coeficiente *down-lifting* β_{12} .

A árvore T_{w_I} foi obtida no Passo 1.1 e como tem peso 13, verifica a restrição de peso, assim $T_\alpha = T_{w_I}$. No Passo 1.2 é obtida a árvore T_{c_I} que não verifica a restrição de peso, então $T_\mu = T_{c_I}$. Deste modo obtemos um limite inferior e um limite superior para o coeficiente β_{12} , ou seja,

$$-1 \leq \beta_{12} \leq 0.$$

No Passo 2 é obtida a árvore de suporte ponderada mínima que contém a aresta $\{0, 1\}$ e não contém a aresta $\{1, 2\}$ (ver Figura 6.12).



T_{p_I} :

$$\tilde{C}_M(T_{p_I}) = -1 \text{ e } W(T_{p_I}) = 18$$

Figura 6.12: Árvore de suporte T_{p_I} obtida no Passo 2 do Algoritmo HCoef no cálculo do coeficiente *down-lifting* β_{12} .

No Passo 3 é atualizado o LS, pois

$$W(T_{p_I}) = 18 \leq 20 = W \text{ e } \tilde{C}_M(T_{p_I}) = -1 = \tilde{C}_M(T_\alpha).$$

Assim temos que

$$\beta_{12} = \tilde{C}_M(T_\alpha) = -1.$$

De seguida calculam-se os restantes coeficientes de forma sequencial usando também o Algoritmo HCoef.

Nas Tabelas 6.7 e 6.8 podemos observar as principais etapas do Algoritmo Sep1 para as DCILDLs aplicado ao Exemplo 2.1, onde são escolhidos dois conjuntos diferentes de fixação de variáveis.

Iter.	Sep1-DCILDL, FVD3	
1	Passo 1	x^* : $x_{01} = 1$; $x_{04} = 0,2$; $x_{12} = 1$; $x_{13} = 0,8$; $x_{34} = 0,8$ e $x_{43} = 0,2$ Custo = 25,4 e peso = 20
	Passo 2, 3 e 4	Ordenação: OrdW $E_1 = \{\{1, 2\}\}$ DCI: $x_{01} + x_{04} \leq 1$ DCILDL: $x_{01} + x_{03} + x_{04} + x_{12} \leq 2 \rightsquigarrow$ Inserir, corta x^* ($2,2 > 2$).
2	Passo 1	x^* : $x_{01} = 1$; $x_{12} = 1$; $x_{13} = 1$ e $x_{34} = 1$ Custo = 27 e peso = 20

Tabela 6.7: Principais resultados por iteração do Algoritmo Sep1 para as DCILDLs com fixação de variáveis FVD3 aplicado ao Exemplo 2.1.

Iter.	Sep1-DCILDL, FVD2	
1	Passo 1	x^* : $x_{01} = 1$; $x_{04} = 0,2$; $x_{12} = 1$; $x_{13} = 0,8$; $x_{34} = 0,8$ e $x_{43} = 0,2$ Custo = 25,4 e peso = 20
	Passo 2, 3 e 4	Ordenação: OrdW $E_1 = \{\{0,1\}; \{1,2\}\}$ DCI: $x_{04} \leq 0$ DCILDL: $x_{01} + x_{04} + x_{12} \leq 2 \rightsquigarrow$ Inserir, corta x^* ($2,2 > 2$).
2	Passo 1	x^* : $x_{01} = 1$; $x_{03} \approx 0,1667$; $x_{12} = 1$; $x_{13} \approx 0,8333$ e $x_{34} = 1$ Custo = 25,8333 e peso = 20
	Passo 2, 3 e 4	Ordenação: OrdW $E_1 = \{\{0,1\}; \{1,2\}; \{3,4\}\}$ DCI: $x_{03} \leq 0$ DCILDL: $x_{01} + x_{03} + x_{12} \leq 2 \rightsquigarrow$ Inserir, corta x^* ($2,1667 > 2$).
3	Passo 1	x^* : $x_{01} = 0,875$; $x_{02} = 0,125$; $x_{03} = 0,125$ $x_{04} = 0,125$; $x_{12} = 0,875$; $x_{13} = 0,75$; $x_{21} = 0,125$; $x_{34} = 0,875$ e $x_{43} = 0,125$ Custo = 26,125 e peso = 20
	Passo 2, 3 e 4	Ordenação: OrdW $E_1 = \{\}$ DCI: $x_{01} + x_{03} + x_{04} \leq 2$. Não corta x^* ($1,125 \not> 2$).

Tabela 6.8: Principais resultados por iteração do Algoritmo Sep1 para as DCILDLs com fixação de variáveis FVD2 aplicado ao Exemplo 2.1.

Da observação das Tabelas 6.7 e 6.8 podemos concluir que a utilização de diferentes formas de escolher o conjunto E_1 podem levar à obtenção de diferentes DCILDLs. Nestes dois exemplos foi utilizado o mesmo procedimento de levantamento das variáveis, PL-LD. Utilizando a fixação de variáveis FVD2 (Tabela 6.8) foram introduzidas no modelo duas DCILDLs e não foi obtida a solução ótima, enquanto que utilizando a fixação de variáveis FVD3 (Tabela 6.7) só foi introduzida no modelo uma DCILDL, a qual levou à obtenção da solução ótima com custo 27 e peso 19.

Caso se utilize o procedimento de levantamento das variáveis PL-DL e as fixações de variáveis FVD2 e FVD3 obtêm-se os resultados dados na Tabela 6.7. Assim também se pode observar que a utilização de diferentes procedimentos de levantamento de variáveis pode levar à obtenção de diferentes DCILDLs.

Se aplicarmos o Algoritmo Sep2 ao Exemplo de 5 nodos apenas se forma a DCILDL obtida na Iteração 1 do Algoritmo Sep1 que se encontra na Tabela 6.7.

6.1.6 Algoritmo de Separação para as DCILUL

No caso de pretendermos obter Desigualdades de Cobertura Implícita Levantada por *Up-Lifting* (DCILULs) temos de alterar o Passo 3, da Secção 6.1.5 para agora obter uma CI válida para o conjunto restrito $X^{E_0} = X_{WMST} \cap \{x : x_f = 0, f \in E_0\}$ e a correspondente DCILUL. A principal diferença no Passo 3.1 deste algoritmo relativamente ao algoritmo para as DCILDLs é a alteração do conjunto E_1 para o conjunto E_0 . Deste modo temos de obter uma DCI válida para o conjunto restrito X^{E_0} e no Passo 3.1.2 a árvore de suporte de peso mínimo tem de conter as arestas de S , mas não pode conter as arestas do conjunto E_0 .

Passo 3: *Obter uma Desigualdade de Cobertura Implícita Levantada por Up-Lifting*

Seja E_0 o conjunto de variáveis fixas a zero.

Passo 3.1: *Obter uma DCI válida para o conjunto restrito X^{E_0}*

Seja $S = \{\}$.

Fazer $p = 1$ (inserir em S , a primeira aresta de E_{RLOrk} tal que $e \notin E_0$).

Passo 3.1.1: *Construir o conjunto S (sem ciclos)*

Atualizar o conjunto S com as primeiras p arestas de E_{RLOrk} , desde que não formem ciclos e que não pertençam ao conjunto E_0 .

Passo 3.1.2: *Verificar se S é CI para o conjunto restrito X^{E_0}*

Obter a árvore de suporte de peso mínimo T , que inclui as p primeiras arestas de S e cujas arestas de E_0 não pertençam à árvore.

Se $W(T) > W$, então

S é uma CI para o conjunto restrito X^{E_0} .

DCI: $\sum_{e \in S} x_e \leq |S| - 1$. Ir para o **Passo 3.2**.

Caso contrário,

Se $p < \#E_{RLOrk}$, então

$p = p + 1$.

Voltar ao **Passo 3.1.1**.

Caso contrário,

não é possível formar uma CI.

Passo 3.2: Obter uma DCILUL

Calcular os coeficientes β_e , $e \in R \subseteq E_0 \cup E_f$.

$$\beta_e = \min \left\{ |S| - 1 - \sum_{t \in S} x_t - \sum_{t \in R \cap (E_0 \cup E_f)} \beta_t x_t : x \in X(e) \right\},$$

- $e \in E_f \setminus R$

$$X(e) = \left\{ x \in X_{WMST} : x_t = 0, t \in E_0 \setminus R, x_e = 1 \right\};$$

- $e \in E_0 \setminus R$

$$X(e) = \left\{ x \in X_{WMST} : x_t = 0, t \in E_0 \setminus (R \cup \{e\}), x_e = 1 \right\}.$$

$$\text{DCILUL: } \sum_{e \in S} x_e + \sum_{e \in R \cap (E_0 \cup E_f)} \beta_e x_e \leq |S| - 1.$$

Inicialmente no Passo 3 consideramos um conjunto de variáveis fixas a zero que designamos por E_0 , depois no Passo 3.1 obtemos uma DCI válida para o conjunto restrito X^{E_0} e no Passo 3.2 efetuamos o cálculo dos coeficientes para fazer o levantamento das variáveis de E_0 e de $E_f = E \setminus (S \cup E_0)$ e obter a DCILUL. Assim, para aplicar os algoritmos de separação ao caso das DCILULs temos que ter em conta os três fatores que se seguem:

- A escolha do conjunto de variáveis a fixar a zero (conjunto E_0);
- A ordem dos procedimentos a adotar para efetuar o levantamento das variáveis;
- A ordem de determinação dos coeficientes para efetuar o levantamento das variáveis.

Escolha do Conjunto das Variáveis a Fixar a zero, E_0

Quando fixamos variáveis a zero estamos a eliminar a possibilidade dessas variáveis estarem na solução.

• Fixação de Variáveis U1 (FVU1)

Fixar a zero todas as variáveis cujas correspondentes arestas pertencem à árvore de suporte de peso mínimo T_w , e não tomam o valor um (tomam um valor inferior a um) na solução da relaxação linear.

- **Fixação de Variáveis U2 (FVU2)**

Fixar a zero as variáveis cujas correspondentes arestas pertencem à árvore de suporte de peso mínimo T_w , e têm valor zero na solução da relaxação linear.

- **Fixação de Variáveis U3 (FVU3)**

Fixar a zero uma única variável, a de maior peso que tem valor inferior a um na solução da relaxação linear e cuja correspondente aresta pertence à árvore de suporte de peso mínimo T_w .

- **Fixação de Variáveis U4 (FVU4)**

Fixar a zero uma única variável, a de maior peso que tem valor zero na solução da relaxação linear e cuja correspondente aresta pertence à árvore de suporte de peso mínimo T_w .

Procedimentos a adotar para levantamento das variáveis

- **Procedimento de Levantamento LU (PL-LU)**

Primeiro efetuamos o levantamento sequencial usual e depois o *up-lifting*.

- **Procedimento de Levantamento UL (PL-UL)**

Primeiro efetuamos o *up-lifting* e depois o levantamento sequencial usual.

Ordem de Determinação dos Coeficientes no Levantamento das Variáveis

Para efetuar o levantamento sequencial usual podemos usar as mesmas ordens propostas na Subsecção 6.1.4. No *up-lifting* ordenamos as variáveis por ordem crescente de peso. Podemos testar se efetuamos primeiro o levantamento das variáveis com menor peso ou com maior peso.

Note-se que:

- Quando o número de variáveis a fixar a zero for nulo, ou seja, $E_0 = \emptyset$ não se aplica o procedimento *up-lifting* e aplica-se apenas o procedimento de levantamento sequencial usual, se a DCI válida para o conjunto restrito X^{E_0} cortar a solução fracionária x^* .

- No caso da DCI válida para o conjunto restrito X^{E_0} conter apenas uma variável, então faz-se apenas o *up-lifting*, uma vez que, se efetuarmos o levantamento sequencial usual os coeficientes são todos nulos.

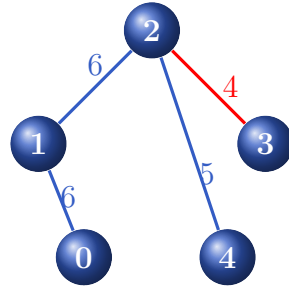
Para exemplificar o Algoritmo HCoef vamos calcular dois coeficientes *up-lifting* para o Exemplo 2.1.

Exemplo 6.5.

Dada a DCI, $x_{04} \leq 0$ válida para o conjunto restrito $X^{E_0} = X_{WMST} \cap \{x_e : x_{02} = x_{13} = x_{14} = x_{23} = 0\}$. Para fazer o levantamento da variável x_{23} e calcular o coeficiente β_{23} temos de resolver o seguinte problema

$$\beta_{23} = \min\{-x_{04} : x \in X_{WMST}, x_{23} = 1, x_{02} = x_{13} = x_{14} = 0\}.$$

Para determinar este coeficiente, no Passo 1.1 do Algoritmo HCoef é obtida a árvore de suporte de peso mínimo que contém a aresta $\{2, 3\}$ e não contém as restantes arestas de E_0 (ver Figura 6.13).



T_{w_I} :

$$\tilde{C}_M(T_{w_I}) = 0 \text{ e } W(T_{w_I}) = 21$$

Figura 6.13: Árvore de suporte T_{w_I} obtida no Passo 1.1 do Algoritmo HCoef no cálculo do coeficiente *up-lifting* β_{23} .

Como $W(T_{w_I}) = 21 > 20 = W$, então $\beta_{23} = 0$.

Para calcular o coeficiente β_{02} temos de resolver o seguinte problema

$$\beta_{02} = \min\{-x_{04} : x \in X_{WMST}, x_{02} = 1, x_{13} = x_{14} = 0\}.$$

Para determinar este coeficiente, no Passo 1.1 do Algoritmo HCoef são obtidas duas árvores de suporte que contêm a aresta $\{0, 2\}$ e não contêm as arestas $\{1, 3\}$ e $\{1, 4\}$ (ver Figura 6.14).

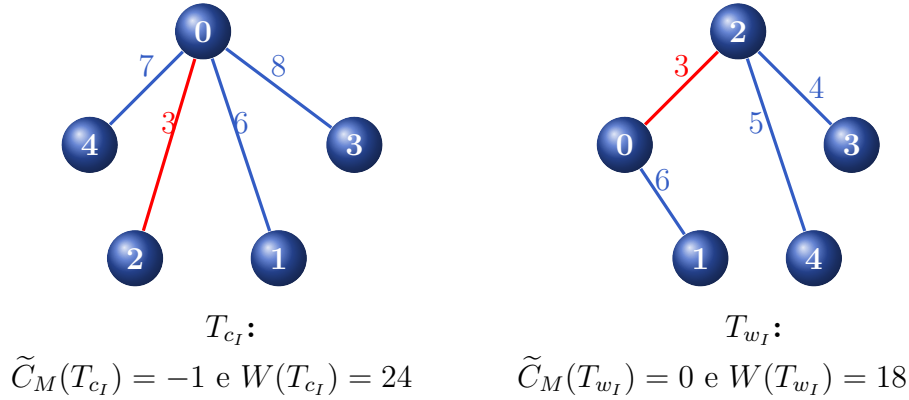


Figura 6.14: Árvores de suporte T_{c_I} e T_{w_I} obtidas no Passo 1 do Algoritmo HCoef no cálculo do coeficiente *up-lifting* β_{02} .

A árvore T_{w_I} foi obtida no Passo 1.1 e como tem peso 18, verifica a restrição de peso, assim $T_\alpha = T_{w_I}$. No Passo 1.2 é obtida a árvore T_{c_I} que não verifica a restrição de peso, então $T_\mu = T_{c_I}$. Deste modo obtemos um limite inferior e um limite superior para o coeficiente β_{02} , ou seja,

$$-1 \leq \beta_{02} \leq 0.$$

No Passo 2 é obtida a árvore de suporte ponderada mínima que contém a aresta $\{0, 2\}$ e não contém as arestas $\{1, 3\}$ e $\{1, 4\}$ (ver Figura 6.15).

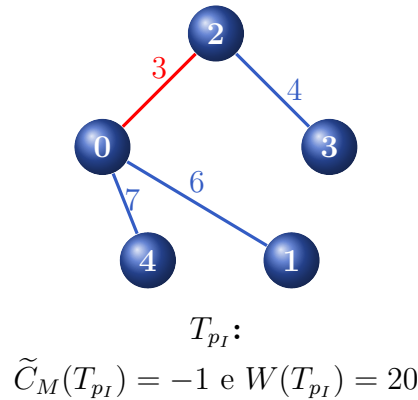


Figura 6.15: Árvore de suporte T_{p_I} obtida no Passo 2 do Algoritmo HCoef no cálculo do coeficiente *up-lifting* β_{02} .

No Passo 3 é atualizado o LS, pois

$$W(T_{p_I}) = 20 = W \text{ e } \tilde{C}_M(T_{p_I}) = -1 = \tilde{C}_M(T_\alpha).$$

Assim temos que

$$\beta_{02} = \tilde{C}_M(T_\alpha) = -1.$$

Os restantes coeficientes são também calculados de forma sequencial através do Algoritmo HCoef.

Nas Tabelas 6.9 e 6.10 podemos observar as principais etapas do Algoritmo Sep1 para as DCILULs aplicado ao Exemplo 2.1, onde são escolhidos dois procedimentos diferentes de levantamento de variáveis.

Iter.	Sep1-DCILUL, PL-UL	
1	Passo 1	x^* : $x_{01} = 1$; $x_{04} = 0,2$; $x_{12} = 1$; $x_{13} = 0,8$; $x_{34} = 0,8$ e $x_{43} = 0,2$ Custo = 25,4 e peso = 20
	Passo 2, 3 e 4	Ordenação: OrdW $E_0 = \{\{0, 2\}; \{1, 3\}; \{1, 4\}; \{2, 3\}\}$ DCI: $x_{01} + x_{04} + x_{12} \leq 2$ DCILUL: $x_{01} + x_{03} + x_{04} + x_{12} \leq 2 \rightsquigarrow$ Inserir, corta x^* ($2,2 > 2$).
2	Passo 1	x^* : $x_{01} = 1$; $x_{12} = 1$; $x_{13} = 1$ e $x_{34} = 1$ Custo = 27 e peso = 20

Tabela 6.9: Principais resultados por iteração do Algoritmo Sep1 para as DCILULs utilizando o procedimento de levantamento de variáveis PL-UL aplicado ao Exemplo 2.1.

Da observação das Tabelas 6.9 e 6.10 podemos concluir que a utilização de diferentes procedimentos de levantamento das variáveis podem levar à obtenção de diferentes DCILULs. Nestes dois exemplos foram utilizadas as mesmas formas de escolher o conjunto E_0 . Utilizando o procedimento de levantamento das variáveis PL-LU (Tabela 6.10) foram introduzidas no modelo duas DCILULs e não foi obtida a solução ótima, enquanto que utilizando o procedimento de levantamento das variáveis PL-UL (Tabela 6.9) só foi introduzida no modelo uma DCILUL, a qual levou à obtenção da solução ótima com custo 27 e peso 19. Note-se que tal como verificámos para o caso das DCILDLs, a utilização de diferentes formas de escolher o conjunto E_0 pode levar à obtenção de diferentes DCILULs.

Iter.	Sep1-DCILUL, PL-LU	
1	Passo 1	x^* : $x_{01} = 1$; $x_{04} = 0,2$; $x_{12} = 1$; $x_{13} = 0,8$; $x_{34} = 0,8$ e $x_{43} = 0,2$ Custo = 25,4 e peso = 20
	Passo 2, 3 e 4	Ordenação: OrdW $E_0 = \{\{0,2\}; \{1,3\}; \{1,4\}; \{2,3\}\}$ DCI: $x_{04} \leq 0$ DCILUL: $-x_{02} + x_{04} - x_{13} - x_{14} \leq 0$. Não corta x^* ($-0,6 \not\geq 0$).
	Passo 2, 3 e 4	Ordenação: OrdWX $E_0 = \{\{0,2\}; \{1,3\}; \{1,4\}; \{2,3\}\}$ DCI: $x_{01} \leq 0$ DCILUL: $x_{01} - x_{02} - x_{13} - x_{14} \leq 0 \rightsquigarrow$ Inserir, corta x^* ($0,2 > 0$).
2	Passo 1	x^* : $x_{01} = 1$; $x_{04} = 0,5$; $x_{12} = 1$; $x_{13} = 1$ e $x_{34} = 0,5$ Custo = 25,5 e peso = 20
	Passo 2, 3 e 4	Ordenação: OrdW $E_0 = \{\{0,1\}; \{1,2\}; \{2,3\}\}$ DCI: $x_{01} + x_{04} + x_{12} \leq 2$ DCILUL: $x_{01} - x_{02} + x_{03} + x_{04} + x_{12} - x_{14} \leq 2 \rightsquigarrow$ Inserir, corta x^* ($2,5 > 2$).
3	Passo 1	x^* : $x_{01} \approx 0,8571$; $x_{02} \approx 0,1429$; $x_{04} \approx 0,2857$ $x_{12} \approx 0,8571$; $x_{13} \approx 0,7143$; $x_{21} \approx 0,1429$; $x_{34} \approx 0,7143$; e $x_{43} = 0,2857$ Custo = 25,8571 e peso = 20
	Passo 2, 3 e 4	Ordenação: OrdW $E_0 = \{\{0,2\}; \{1,3\}; \{1,4\}; \{2,3\}\}$ DCI: $x_{04} \leq 0$ DCILUL: $-x_{02} + x_{04} - x_{13} - x_{14} \leq 0$. Não corta x^* ($-0,5714 \not\geq 0$).
	Passo 2, 3 e 4	Ordenação: OrdWX e OrdX $E_0 = \{\{0,2\}; \{1,3\}; \{1,4\}; \{2,3\}\}$ DCI: $x_{01} \leq 0$ DCILUL: $x_{01} - x_{02} - x_{13} - x_{14} \leq 0$. Não corta x^* ($0 \not\geq 0$).

Tabela 6.10: Principais resultados por iteração do Algoritmo Sep1 para as DCILULs utilizando o procedimento de levantamento de variáveis PL-LU aplicado ao Exemplo 2.1.

Se aplicarmos o Algoritmo Sep2 ao Exemplo de 5 nodos apenas se forma a DCILUL obtida na Iteração 1 do Algoritmo Sep1 que se encontra na Tabela 6.9.

6.1.7 Algoritmo de Separação para as DGCIL

Para obter Desigualdades Generalizadas de Cobertura Implícita Levantada (DGCILs) temos de alterar o Passo 3, dado nas Secções 6.1.5 e 6.1.6 de modo a usarmos dois conjuntos de variáveis fixas e obter uma DCI válida para o conjunto restrito $X^U = X_{WMST} \cap \{x : x_e = 0, e \in E_0, x_e = 1, e \in E_1\}$, onde $U = E_0 \cup E_1$ e a correspondente DGCIL. A principal diferença no Passo 3.1 deste algoritmo relativamente aos algoritmos aplicados às DCILDLs e às DCILULs é a alteração do conjunto de variáveis fixas, que neste caso passamos a ter dois conjuntos de variáveis fixas, o conjunto E_0 de variáveis fixas a zero e o conjunto E_1 de variáveis fixas a um. Deste modo temos que obter uma DCI válida para o conjunto restrito X^U , onde no Passo 3.1.2 a árvore de suporte de peso mínimo tem de conter as arestas de S e as arestas do conjunto E_1 , mas não pode conter as arestas do conjunto E_0 .

Passo 3: *Obter uma Desigualdade Generalizada de Cobertura Implícita Levantada*

Seja E_0 o conjunto de variáveis fixas a zero e E_1 o conjunto de variáveis fixas a um.

Passo 3.1: *Obter uma DCI válida para o conjunto restrito X^U*

Seja $S = \{\}$.

Fazer $p = 1$ (inserir em S , a primeira aresta de E_{RLOrk} tal que $e \notin E_0 \cup E_1$).

Passo 3.1.1: *Construir o conjunto S (sem ciclos)*

Atualizar o conjunto S com as primeiras p arestas de E_{RLOrk} , desde que não formem ciclos e que não pertençam ao conjunto E_0 nem ao conjunto E_1 .

Passo 3.1.2: *Verificar se S é CI para o conjunto restrito X^U*

Obter a árvore de suporte de peso mínimo T , que inclui as p primeiras arestas de S e as arestas de E_1 , mas cujas arestas de E_0 não pertençam à árvore.

Se $W(T) > W$, então

S é uma CI para o conjunto restrito X^U .

DCI: $\sum_{e \in S} x_e \leq |S| - 1$. Ir para o **Passo 3.2**.

Caso contrário,

Se $p < \#E_{RLOrdk}$, então

$$p = p + 1.$$

Voltar ao **Passo 3.1.1**.

Caso contrário,

não é possível formar uma CI.

Passo 3.2: Obter uma DGCIL

Calcular os coeficientes β_e , $e \in R \subseteq E_0 \cup E_1 \cup E_f$.

$$\beta_e = \min \left\{ |S| - 1 - \sum_{t \in S} x_t - \sum_{t \in R \cap (E_0 \cup E_f)} \beta_t x_t - \sum_{t \in R \cap E_1} \beta_t (1 - x_t) : x \in X(e) \right\},$$

- $e \in E_f \setminus R$

$$X(e) = \left\{ x \in X_{WMST} : x_t = 1, t \in E_1 \setminus R, x_t = 0, t \in E_0 \setminus R, x_e = 1 \right\};$$

- $e \in E_0 \setminus R$

$$X(e) = \left\{ x \in X_{WMST} : x_t = 1, t \in E_1 \setminus R, x_t = 0, t \in E_0 \setminus (R \cup \{e\}), x_e = 1 \right\};$$

- $e \in E_1 \setminus R$

$$X(e) = \left\{ x \in X_{WMST} : x_t = 1, t \in E_1 \setminus (R \cup \{e\}), x_t = 0, t \in E_0 \setminus R, x_e = 0 \right\}.$$

$$\text{DGCIL: } \sum_{e \in S} x_e + \sum_{e \in R \cap (E_0 \cup E_f)} \beta_e x_e + \sum_{e \in R \cap E_1} \beta_e (1 - x_e) \leq |S| - 1.$$

Inicialmente no Passo 3 consideramos dois conjuntos: um conjunto de variáveis fixas a zero que designamos por E_0 , e outro de variáveis fixas a um que designamos por E_1 . No Passo 3.1 obtemos uma DCI válida para o conjunto restrito X^U e no Passo 3.2 efetuamos o cálculo dos coeficientes para fazer o levantamento das variáveis de E_0 , de E_1 e de $E_f = E \setminus (S \cup E_0 \cup E_1)$ e obter a DGCIL. Assim, para aplicar os algoritmos de separação ao caso das DGCILs temos que ter em conta os três fatores que se seguem:

- A escolha dos conjuntos de variáveis a fixar a zero (conjunto E_0) e a fixar a um (conjunto E_1);

- A ordem dos procedimentos a adotar para efetuar o levantamento das variáveis;
- A ordem de determinação dos coeficientes para efetuar o levantamento das variáveis.

Escolha dos Conjuntos de Variáveis a Fixar a zero e a um, E_0 e E_1

Como estamos a fixar, simultaneamente, dois conjuntos de variáveis, estamos a fazer com que as variáveis fixas a um façam parte da solução e as variáveis fixas a zero não façam parte da solução.

Para escolha dos conjuntos E_0 e E_1 propomos efetuar uma combinação das escolhas feitas nas duas subsecções anteriores.

Procedimentos a adotar para levantamento das variáveis

- **Procedimento de Levantamento DUL (PL-DUL)**

Primeiro efetuamos o *down-lifting*, depois o *up-lifting* e por fim o levantamento sequencial usual.

- **Procedimento de Levantamento UDL (PL-UDL)**

Primeiro efetuamos o *down-lifting*, depois o o levantamento sequencial usual e por fim o *up-lifting*.

- **Procedimento de Levantamento ULD (PL-ULD)**

Primeiro efetuamos o *up-lifting*, depois o levantamento sequencial usual e por fim o *down-lifting*.

- **Procedimento de Levantamento DLU (PL-DLU)**

Primeiro efetuamos o *up-lifting*, depois o *down-lifting* e por fim o levantamento sequencial usual.

- **Procedimento de Levantamento LDU (PL-LDU)**

Primeiro efetuamos o levantamento sequencial usual, depois o *down-lifting* e por fim o *up-lifting*.

- **Procedimento de Levantamento LUD (PL-LUD)**

Primeiro efetuamos o levantamento sequencial usual, depois o *up-lifting* e por fim o *down-lifting*.

Ordem de Determinação dos Coeficientes no Levantamento das Variáveis

Para efetuar o levantamento usual podemos usar as mesmas ordens propostas na Subsecção 6.1.4. No *up-lifting* e no *down-lifting* ordenamos as variáveis de cada conjunto E_0 e E_1 por ordem crescente de peso. Podemos testar se efetuamos primeiro o levantamento das variáveis com menor peso ou com maior peso em cada um dos conjuntos.

Note-se que:

- Quando o número de variáveis a fixar a zero for nulo, ou seja, $E_0 = \emptyset$ não se aplica o procedimento *up-lifting* e aplicam-se apenas os procedimentos de levantamento sequencial usual e de *down-lifting*. Se o número de variáveis a fixar a um for nulo ($E_1 = \emptyset$) aplicam-se apenas os procedimentos *up-lifting* e levantamento sequencial usual;
- Caso os dois conjuntos de variáveis sejam nulos só se aplica o procedimento de levantamento sequencial usual;
- No caso da DCI válida para o conjunto restrito X^U conter apenas uma variável, então faz-se *up-lifting* e *down-lifting*, uma vez que, se efetuarmos o levantamento sequencial usual em primeiro lugar os coeficientes são todos nulos.

Para calcular os coeficientes das variáveis a efetuar levantamento usa-se o algoritmo aproximado HCoef, descrito na Subsecção 6.1.4.

Exemplo 6.6.

Nas Tabelas 6.11 e 6.12 podemos observar as principais etapas do Algoritmo Sep1 para as DGCILs aplicado ao Exemplo 2.1, onde são escolhidos dois procedimentos diferentes de levantamento de variáveis.

Iter.	Sep1-DGCIL, PL-DLU	
1	Passo 1	x^* : $x_{01} = 1$; $x_{04} = 0,2$; $x_{12} = 1$; $x_{13} = 0,8$; $x_{34} = 0,8$ e $x_{43} = 0,2$ Custo = 25,4 e peso = 20
	Passo 2, 3 e 4	Ordenação: OrdW $E_0 = \{\{0, 2\}; \{1, 4\}; \{2, 3\}\}$ $E_1 = \{\{0, 1\}; \{1, 2\}\}$ DCI: $x_{04} \leq 0$ DGCIL: $x_{01} - x_{02} + 2x_{03} + x_{04} + x_{12} - x_{14} \leq 2 \rightsquigarrow$ Inserir, corta x^* ($0,2 > 0$).
2	Passo 1	x^* : $x_{01} = 1$; $x_{04} = 0,2$; $x_{12} = 1$; $x_{13} = 0,6$; $x_{14} = 0,2$; $x_{34} = 0,6$ e $x_{43} = 0,4$ Custo = 25,6 e peso = 20
	Passo 2, 3 e 4	Ordenação: OrdW $E_0 = \{\{0, 2\}; \{2, 3\}\}$ $E_1 = \{\{0, 1\}; \{1, 2\}\}$ DCI: $x_{04} \leq 0$ DGCIL: $x_{01} + x_{03} + x_{04} + x_{12} \leq 2 \rightsquigarrow$ Inserir, corta x^* ($2,2 > 2$).
3	Passo 1	x^* : $x_{01} \approx 0,8333$; $x_{02} \approx 0,1667$; $x_{04} \approx 0,1667$ $x_{12} \approx 0,8333$; $x_{13} \approx 0,8333$; $x_{21} \approx 0,1667$; $x_{34} \approx 0,8333$; e $x_{43} = 0,1667$ Custo = 27 e peso = 19,33

Tabela 6.11: Principais resultados por iteração do Algoritmo Sep1 para as DGCILs utilizando o procedimento de levantamento de variáveis PL-DLU aplicado ao Exemplo 2.1.

Iter.	Sep1-DGCIL, PL-DUL	
1	Passo 1	x^* : $x_{01} = 1$; $x_{04} = 0,2$; $x_{12} = 1$; $x_{13} = 0,8$; $x_{34} = 0,8$ e $x_{43} = 0,2$ Custo = 25,4 e peso = 20
	Passo 2, 3 e 4	Ordenação: OrdW $E_0 = \{\{0, 2\}; \{1, 4\}; \{2, 3\}\}$ $E_1 = \{\{0, 1\}; \{1, 2\}\}$ DCI: $x_{04} \leq 0$ DGCIL: $x_{01} + x_{03} + x_{04} + x_{12} \leq 2 \rightsquigarrow$ Inserir, corta x^* ($2,2 > 2$).
2	Passo 1	x^* : $x_{01} = 1$; $x_{12} = 1$; $x_{13} = 1$ e $x_{34} = 1$ Custo = 27 e peso = 20

Tabela 6.12: Principais resultados por iteração do Algoritmo Sep1 para as DGCILs utilizando o procedimento de levantamento de variáveis PL-DLU aplicado ao Exemplo 2.1.

Da observação das Tabelas 6.11 e 6.12 podemos concluir, tal como visto para as DCILDLs e para as DCILULs, que utilizando diferentes procedimentos de levantamento de variáveis podemos encontrar diferentes DGCILs. Nestes dois exemplos foram utilizadas as mesmas formas de escolher os conjuntos E_0 e E_1 . Utilizando o procedimento de levantamento das variáveis PL-DLU (Tabela 6.11) foram introduzidas no modelo duas DGCILs e não foi obtida a solução ótima, enquanto que utilizando o procedimento de levantamento das variáveis PL-DUL (Tabela 6.12) só foi introduzida no modelo uma DGCIL, a qual levou à obtenção da solução ótima com custo 27 e peso 19.

Note-se que tal como verificámos para o caso das DCILDLs e das DCILULs, utilizando diferentes formas de escolher os conjuntos E_0 e E_1 pode levar à obtenção de diferentes DGCILs.

Se aplicarmos o Algoritmo Sep2 ao Exemplo de 5 nodos apenas se forma a DGCIL obtida na Iteração 1 do Algoritmo Sep1 que se encontra na Tabela 6.12.

Note-se que a desigualdade válida

$$x_{01} + x_{03} + x_{04} + x_{12} \leq 2 \quad (6.11)$$

que corta a solução da relaxação linear e que com a sua introdução no modelo se consegue obter a solução ótima foi obtida em todas as desigualdades derivadas de levantamento de variáveis.

Para obter a desigualdade (6.11) como uma DCIL primeiro obteve-se a DCI, $x_{01} + x_{04} + x_{12} \leq 2$ e o coeficiente $\beta_{03} = 1$ foi obtido efetuando o levantamento usual.

Ao fixar o conjunto $E_1 = \{\{1, 2\}\}$ obtemos também a desigualdade (6.11) como uma DCILDL, onde a DCI, $x_{01} + x_{04} \leq 1$ é válida para o conjunto restrito X^{E_1} e o coeficiente $\beta_{12} = -1$ é obtido efetuando o *down-lifting* e o coeficiente $\beta_{03} = 1$ efetuando o levantamento usual.

Ao fixar o conjunto $E_0 = \{\{0, 2\}; \{1, 3\}; \{1, 4\}; \{2, 3\}\}$ obtemos a mesma desigualdade (6.11) como uma DCILUL, onde a DCI, $x_{01} + x_{04} + x_{12} \leq 2$ é válida para o conjunto restrito X^{E_0} e o coeficiente $\beta_{03} = 1$ é obtido efetuando o levantamento usual.

No caso das DGCILs podemos obter a desigualdade (6.11) fixando por exemplo os conjuntos $E_0 = \{\{0, 2\}; \{1, 4\}; \{2, 3\}\}$ e $E_1 = \{\{0, 1\}; \{1, 2\}\}$, a DCI, $x_{04} \leq 0$ é válida para o conjunto restrito $X^{E_0 \cup E_1}$ e os coeficientes $\beta_{12} = -1$ e $\beta_{01} = -1$ são obtidos efetuando o *down-lifting* e o coeficiente $\beta_{03} = 1$ efetuando o levantamento usual.

6.2 Experiências e Resultados Computacionais

Nesta secção descrevemos algumas experiências computacionais realizadas e apresentamos os resultados computacionais dos algoritmos heurísticos de separação apresentados ao longo deste capítulo. Efetuamos também uma comparação entre os algoritmos de separação com o objetivo de avaliar qual o mais eficiente. Neste sentido pretendemos avaliar a qualidade do valor do limite inferior obtido pela respetiva relaxação linear. Assim, para avaliar o limite inferior obtido em cada um dos algoritmos calculamos o $gap = \frac{OPT - LI}{OPT} \times 100$, onde LI é o valor do limite inferior obtido para o valor do custo e OPT é o valor ótimo ($\vartheta(WMST)$) ou o melhor valor obtido pelo Procedimento P-WMTZ+C do Capítulo 3.

Relativamente aos tempos de execução vamos analisar em termos médios os tempos usados para a obtenção do valor ótimo ou de um limite superior para o valor do custo (tempos de execução da relaxação linear conjuntamente com os tempos de execução do Algoritmo *Branch and Bound*).

Os algoritmos heurísticos de separação foram implementados usando o *software Xpress 7.3* (*Xpress-Optimizer 23.01.03* e *Xpress-Mosel 3.4.0*) [1] e todos os testes foram efetuados num Intel(R) Core(TM)2 Duo CPU (T7100) 2.00 GHz processador e 4Gb de RAM.

6.2.1 Experiências realizadas com as DCs e as DCIs

Nas experiências computacionais realizadas com as DCs e com as DCIs foram utilizadas todas as ordenações propostas na Secção 6.1. No caso de se obterem dois valores da relaxação linear iguais com a mesma ordenação de arestas mudamos de ordenação.

Na Tabela 6.13 são apresentados os *gaps* médios, em percentagem, do limite inferior obtidos pelo Procedimento P-WMTZ+C_L e por cada um dos Algoritmos Sep1 e Sep2 para as DCs e para as DCIs e os respetivos desvios padrões. Note-se que os referidos *gaps* correspondem aos valores obtidos pelas respetivas relaxações lineares. Os tempos médios, em segundos, apresentados na Tabela 6.13, correspondem aos tempos de execução até obter o valor ótimo ou um limite superior para o valor do custo (tempo de execução da relaxação linear + tempo de execução do Algoritmo *Branch and Bound*). Na mesma tabela encontram-se os desvios padrões correspondentes aos tempos.

		P-WMTZ+C _L	Sep1-DCs	Sep2-DCs	Sep1-DCIs	Sep2-DCIs
<i>Gap</i> Médio (%)	10	22,292	19,747	19,747	19,451	19,467
	20	9,874	9,627	9,627	9,552	9,562
	40	6,362	6,337	6,337	6,331	6,332
	60	4,830	4,815	4,815	4,810	4,812
σ_{gap}	10	7,061	7,217	7,217	7,324	7,335
	20	4,380	4,388	4,388	4,431	4,427
	40	0,840	0,838	0,838	0,844	0,843
	60	0,863	0,863	0,863	0,861	0,861
Tempo Médio (s)	10	0,109	0,154	0,148	0,151	0,169
	20	1,350	1,274	1,287	1,372	1,956
	40	41,756	36,886	42,449	35,787	40,408
	60	517,422	553,105	932,913	1513,102	875,270
σ_{tempo}	10	0,042	0,034	0,043	0,045	0,039
	20	0,867	0,513	0,372	0,947	1,488
	40	29,749	15,673	23,380	22,001	24,511
	60	338,821	407,218	1480,329	2300,852	992,761

Tabela 6.13: Comparação do Procedimento P-WMTZ+C_L e dos Algoritmos Sep1 e Sep2 nas DCs e nas DCIs nas instâncias QC de 10 a 60 nodos.

Em termos de tempos médios de execução, o algoritmo de separação mais rápido

parece ser o Algoritmo Sep1, nas DCs e nas DCIs. No caso das DCIs, apenas em instâncias de 60 nodos o tempo médio do Algoritmo Sep1 é superior ao tempo médio do Algoritmo Sep2.

Tanto nas DCs como nas DCIs os *gaps* médios obtidos são muito idênticos nos dois Algoritmos Sep1 e Sep2. Os *gaps* médios mais baixos são obtidos pelo Algoritmo Sep1 com as DCIs. Em qualquer um dos algoritmos os *gaps* médios são inferiores aos obtidos com o Procedimento P-WMTZ+C_L. Os limites inferiores obtidos usando estes algoritmos de separação aumentam em 39 das 40 instâncias (de 10 a 60 nodos). Apenas numa das instâncias (QC60-9) o limite inferior obtido é igual ao obtido com o Procedimento P-WMTZ+C_L. Os algoritmos para as DCIs são os que obtêm melhores limites inferiores. Deste modo, podemos afirmar que faz sentido fazer o levantamento de uma DCI para obtenção de desigualdades derivadas através de procedimentos de levantamento usual.

Em termos médios, não se observa uma redução do tempo quando se aplicam os algoritmos heurísticos de separação comparativamente ao Procedimento P-WMTZ+C_L. Podemos observar que há melhorias de tempo em 32,5% das instâncias de 10 a 60 nodos (13 instâncias em 40) no Algoritmo Sep1 para as DCIs.

6.2.2 Experiências Realizadas com as DCIEs

Nas experiências computacionais realizadas com as DCIEs também foram utilizadas todas as ordenações propostas na Secção 6.1. No caso de se obterem dois valores da relaxação linear iguais com a mesma ordenação de arestas mudamos de ordenação.

Na Tabela 6.14 são apresentados os *gaps* médios, em percentagem, do limite inferior obtidos pelo Procedimento P-WMTZ+C_L e por cada um dos Algoritmos Sep1 e Sep2 para as DCIEs e os respetivos desvios padrões. Note-se que os referidos *gaps* correspondem aos valores obtidos pelas respetivas relaxações lineares. Os tempos médios, em segundos, apresentados na Tabela 6.14, correspondem aos tempos de execução até obter o valor ótimo ou um limite superior para o valor do custo (tempo de execução da relaxação linear + tempo de execução do Algoritmo *Branch and Bound*). Na mesma tabela encontram-se os desvios padrões correspondentes aos tempos.

		P-WMTZ+C _L	Sep1-DCIEs	Sep2-DCIEs
Gap Médio(%)	10	22,292	17,885	13,181
	20	9,874	9,511	8,995
	40	6,362	6,327	6,249
	60	4,830	4,802	4,782
σ_{gap}	10	7,061	7,595	3,667
	20	4,380	4,445	4,001
	40	0,840	0,843	0,864
	60	0,863	0,853	0,845
Tempo Médio (s)	10	0,109	0,170	0,205
	20	1,350	1,601	1,669
	40	41,756	36,206	34,171
	60	517,422	897,147	657,020
σ_{tempo}	10	0,042	0,056	0,065
	20	0,867	1,090	0,857
	40	29,749	21,568	19,530
	60	338,821	1337,665	584,175

Tabela 6.14: Comparação do Procedimento P-WMTZ+C_L e dos Algoritmos Sep1 e Sep2 nas DCIEs nas instâncias QC de 10 a 60 nodos.

Em termos de tempos médios, nas instâncias de 40 e 60 nodos o algoritmo de separação mais rápido para as DCIEs parece ser o Algoritmo Sep2, mas nas instâncias de 10 e 20 nodos o tempo médio do Algoritmo Sep2 é superior ao tempo médio do Algoritmo Sep1.

Quanto aos *gaps* médios obtidos são inferiores no Algoritmo Sep2. Em qualquer um dos algoritmos os *gaps* médios são inferiores aos obtidos com o Procedimento P-WMTZ+C_L. Os limites inferiores obtidos usando estes algoritmos heurísticos de separação aumentam em todas as instâncias (de 10 a 60 nodos).

Em termos de tempos médios de execução verifica-se uma redução nas instâncias de 40 nodos, tanto no Algoritmo Sep1 como no Algoritmo Sep2 quando comparados com o Procedimento P-WMTZ+C_L. Podemos observar que há melhorias de tempo em 32,5% das instâncias de 10 a 60 nodos (13 instâncias em 40) no Algoritmo Sep1 para as DCIEs e em 37,5% (15 instâncias em 40) no Algoritmo Sep2 para as DCIEs.

6.2.3 Experiências Realizadas com as DCILs

De experiências computacionais realizadas com as DCILs, verificamos que se utilizarmos todas as ordenações propostas na Secção 6.1, pode-se tornar dispendioso do ponto vista computacional. Das várias experiências computacionais efetuadas com combinações das diferentes ordenações verificamos que com algumas delas, em geral, obtemos as mesmas desigualdades. Assim, decidimos utilizar apenas três ordenações: OrdW, OrdWX e OrdFixW. Começamos por obter uma desigualdade usando a ordenação de arestas OrdW, depois usamos a ordenação OrdWX e por fim a ordenação OrdFixW. No caso de se obterem dois valores da relaxação linear iguais com a mesma ordenação de arestas mudamos de ordenação.

Quando tentamos encontrar uma CI formam-se várias árvores de suporte. Sempre que estas árvores verifiquem a restrição de peso podemos atualizar o limite superior para o custo (LSC) do Problema WMST. Quando se aplica o Algoritmo HCoef para determinar os vários coeficientes das variáveis a efetuar levantamento também se encontram várias árvores de suporte que verificam a restrição de peso, sendo assim também podemos atualizar o limite superior para o custo do Problema WMST, desde que o custo obtido seja inferior ou igual a LSC. Assim os Algoritmos Sep1 e Sep2 podem terminar no caso do valor obtido na relaxação linear ser igual ao valor do LSC.

Os resultados computacionais que se apresentam para as duas estratégias apresentadas na Secção 6.1.4 usam o levantamento das variáveis LV2, dado que após a realização de várias experiências com as diferentes ordens de levantamento apresentadas na Secção 6.1.6 este foi o que apresentou os melhores resultados.

Na Tabela 6.15 são apresentados os *gaps* médios (em percentagem) do limite inferior obtidos pelo Procedimento P-WMTZ+ C_L e por cada um dos Algoritmos Sep1 e Sep2 para as DCILs e os respetivos desvios padrões. Note-se que os referidos *gaps* correspondem aos valores obtidos pelas respetivas relaxações lineares. Os tempos médios, em segundos, apresentados na Tabela 6.15, correspondem aos tempos de execução até obter o valor ótimo ou um limite superior para o valor do custo (tempo de execução da relaxação linear + tempo de execução do Algoritmo *Branch and Bound*). Na mesma tabela encontram-se os desvios padrões correspondentes aos tempos.

Os resultados computacionais apresentados na Tabela 6.15 para o Algoritmo Sep2 foram obtidos usando apenas a Estratégia DCIL-E2, uma vez que esta estratégia foi a que apresentou tempos médios mais baixos para o Algoritmo Sep1.

		Sep1			Sep2
		P-WMTZ+C _L	DCIL-E1	DCIL-E2	DCIL-E2
Gap Médio (%)	10	22,292	6,671	9,632	3,910
	20	9,874	0,088	0,096	0,084
	40	6,362	1,515	1,650	2,141
	60	4,830	0,546	0,596	0,984
σ_{gap}	10	7,061	5,210	5,690	4,376
	20	4,380	0,277	0,276	0,264
	40	0,840	2,541	2,735	2,825
	60	0,863	1,725	1,715	2,034
Tempo Médio (s)	10	0,109	0,228	0,189	0,220
	20	1,350	1,579	1,087	2,000
	40	41,756	55,764	29,259	48,773
	60	517,422	209,591	154,743	375,455
σ_{tempo}	10	0,042	0,086	0,057	0,041
	20	0,867	0,947	0,824	1,570
	40	29,749	50,803	27,386	37,153
	60	338,821	213,496	293,342	546,707

Tabela 6.15: Comparação do Procedimento P-WMTZ+C_L e dos Algoritmos Sep1 e Sep2 nas DCILs nas instâncias QC de 10 a 60 nodos.

Relativamente ao Algoritmo Sep1, a Estratégia DCIL-E1 é a que apresenta *gap* mais baixo, mas em relação ao tempo de execução este é superior ao da Estratégia DCIL-E2 (quase o dobro). Quando se forma uma DCI que não corta a solução da relaxação linear, obtemos na mesma a DCIL, logo estamos constantemente a resolver subproblemas para obter os coeficientes de levantamento e por vezes a DCIL encontrada não corta a solução da relaxação linear. Como podemos observar no Algoritmo Sep2, os *gaps* em instâncias de 10 e 20 nodos são melhores, mas depois em instâncias de 40 e 60 pioram comparativamente ao Algoritmo Sep1. Usando a Estratégia DCIL-E2, os tempos médios de execução do Algoritmo Sep2 são sempre mais elevados do que no Algoritmo Sep1.

Em qualquer um dos algoritmos, os *gaps* médios são inferiores aos obtidos com o Procedimento P-WMTZ+C_L. Os limites inferiores obtidos usando estes algoritmos heurísticos de separação aumentam em todas as instâncias (de 10 a 60 nodos).

Em termos de tempos médios verifica-se uma redução em instâncias com 20 ou

mais nodos no Algoritmo Sep1 usando a Estratégia DCIL-E2 quando comparado com o Procedimento P-WMTZ+C_L. Podemos encontrar 60% das instâncias de 10 a 60 nodos (24 instâncias em 40) em que os tempos de execução do Algoritmo Sep1 usando a Estratégia DCIL-E2 são inferiores aos tempos de execução obtidos com o Procedimento P-WMTZ+C_L.

Após a realização de experiências computacionais verificámos que as estratégias propostas na Secção 6.1.6 para levantamento de uma DCIE não vêm trazer melhorias nem nos *gaps* nem nos tempos médios de execução.

6.2.4 Experiências Realizadas com as DCILDLs

Se pensarmos em fixar a um o conjunto de variáveis que têm valor um na solução da relaxação linear do Procedimento P-WMTZ+C (fixação de variáveis FVD1 da Subsecção 6.1.5), em alguns exemplos, o número de variáveis a fixar a um pode ser muito grande (quase o mesmo número que o das arestas da árvore). Por exemplo, em instâncias de 10 nodos fixam-se a um cerca de 8 variáveis, em instâncias de 20 nodos cerca de 18 variáveis, em instâncias de 40 nodos cerca de 38 variáveis e em instâncias de 60 nodos cerca de 56 variáveis. Para evitar que o conjunto E_1 seja demasiado grande vamos apresentar os resultados das quatro estratégias seguintes.

Estratégia *Down-Lifting* 1 (DL-E1)

Escolha do conjunto E_1 : FVD2

Procedimento de levantamento das variáveis: PL-LD

Estratégia *Down-Lifting* 2 (DL-E2):

Escolha do conjunto E_1 : FVD3.

Procedimento de levantamento das variáveis: PL-LD.

Estratégia *Down-Lifting* 3 (DL-E3)

Escolha do conjunto E_1 : FVD2

Procedimento de levantamento das variáveis: PL-DL

Estratégia *Down-Lifting* 4 (DL-E4):

Escolha do conjunto E_1 : FVD3

Procedimento de levantamento das variáveis: PL-DL

As estratégias apresentadas para o Algoritmo Sep1 diferem na escolha do conjunto de variáveis a fixar a um e na ordem dos procedimentos para levantamento das variáveis.

Na Tabela 6.16 são apresentados os *gaps* médios (em percentagem) do limite inferior obtidos pelo Procedimento P-WMTZ+C_L e por cada um dos Algoritmos Sep1 e Sep2 para as DCILDLs e os respetivos desvios padrões. Note-se que os referidos *gaps* correspondem aos valores obtidos pelas respetivas relaxações lineares. Os tempos médios, em segundos, apresentados na Tabela 6.16, correspondem aos tempos de execução até obter o valor ótimo ou um limite superior para o valor do custo (tempo de execução da relaxação linear + tempo de execução do Algoritmo *Branch and Bound*). Na mesma tabela encontram-se os desvios padrões correspondentes aos tempos.

Os resultados computacionais apresentados na Tabela 6.16 para o Algoritmo Sep2 foram obtidos usando apenas a Estratégia DL-E3, uma vez que esta estratégia se mostrou a mais eficiente em termos de tempos médios e de *gaps* médios para o Algoritmo Sep1.

		Sep1					Sep2
		P-WMTZ+C _L	DL-E1	DL-E2	DL-E3	DL-E4	DL-E3
<i>Gap</i> Médio (%)	10	22,292	10,617	10,699	8,891	8,891	5,622
	20	9,874	3,359	2,524	1,304	1,300	0,565
	40	6,362	5,713	4,114	0,940	1,263	1,035
	60	4,830	4,047	2,859	0,382	0,409	0,695
σ_{gap}	10	7,061	6,964	6,274	6,151	6,151	5,773
	20	4,380	3,877	3,111	2,604	2,602	1,005
	40	0,840	1,480	1,471	2,217	2,295	2,254
	60	0,863	1,470	1,039	1,096	1,097	1,450
Tempo Médio (s)	10	0,109	0,208	0,220	0,195	0,198	0,251
	20	1,350	1,504	1,467	1,157	1,148	2,073
	40	41,756	51,654	65,114	27,474	30,176	40,378
	60	517,422	481,450	193,296	73,554	78,639	204,174
σ_{tempo}	10	0,042	0,078	0,086	0,070	0,069	0,055
	20	0,867	1,506	1,469	0,969	1,073	1,677
	40	29,749	36,275	65,476	34,029	35,130	45,521
	60	338,821	552,349	388,098	89,572	93,186	243,529

Tabela 6.16: Comparação do Procedimento P-WMTZ+C_L e dos Algoritmos Sep1 e Sep2 nas DCILDLs nas instâncias QC de 10 a 60 nodos.

No Algoritmo Sep1 as Estratégias DL-E3 e DL-E4 são as mais rápidas e estas são comparáveis em termos de tempo médio de execução. Quanto à qualidade das soluções obtidas é preferível utilizar a Estratégia DL-D3, pois para instâncias de 40 e 60 nodos os *gaps* médios são inferiores aos obtidos através da Estratégia DL-E4. Os *gaps* médios obtidos pelo Algoritmo Sep2 para instâncias de 10 e 20 nodos são mais baixos que os obtidos pelo Algoritmo Sep1 usando a mesma estratégia, mas tendem a piorar em instâncias de 40 e 60 nodos. Em termos de tempo de execução a Estratégia DL-E3 do Algoritmo Sep1 apresenta melhores resultados quando comparada com a mesma estratégia do Algoritmo Sep2.

Em qualquer um dos algoritmos os *gaps* médios são inferiores aos obtidos com o Procedimento P-WMTZ+C_L. Os limites inferiores obtidos usando estes algoritmos heurísticos de separação aumentam em todas as instâncias (de 10 a 60 nodos).

Em termos de tempos médios verifica-se uma redução em instâncias com 20 ou mais nodos no Algoritmo Sep1 usando a Estratégia DL-E3 quando comparado com o Procedimento P-WMTZ+C_L. Podemos encontrar 65% das instâncias de 10 a 60 nodos (26 instâncias em 40) em que os tempos de execução do Algoritmo Sep1 usando a Estratégia DL-E3 são inferiores aos tempos de execução obtidos com o Procedimento P-WMTZ+C_L.

6.2.5 Experiências Realizadas com as DCILULs

Para encontrar DCILULs realizaram-se várias experiências efetuando combinações das diferentes formas de escolher o conjunto E_0 com os vários procedimentos de levantamento das variáveis propostos na Subsecção 6.1.6, das quais apresentamos as quatro melhores:

Estratégia *Up-Lifting* 1 (UL-E1)

Escolha do conjunto E_0 : FVU3

Procedimento de levantamento das variáveis: PL-LU

Estratégia *Up-Lifting* 2 (UL-E2):

Escolha do conjunto E_0 : FVU4

Procedimento de levantamento das variáveis: PL-LU

Estratégia *Up-Lifting* 3 (UL-E3):

Escolha do conjunto E_0 : FVU2

Procedimento de levantamento das variáveis: PL-UL

Estratégia *Up-Lifting* 4 (UL-E4):

Escolha do conjunto E_0 : FVU4

Procedimento de levantamento das variáveis: PL-UL

As estratégias apresentadas para o Algoritmo Sep1 diferem na escolha do conjunto de variáveis a fixar a zero e na ordem dos procedimentos de levantamento das variáveis.

Na Tabela 6.17 são apresentados os *gaps* médios (em percentagem) do limite inferior obtidos pelo Procedimento P-WMTZ+C_L e por cada um dos Algoritmos Sep1 e Sep2 para as DCILULs e os respectivos desvios padrões. Note-se que os referidos *gaps* correspondem aos valores obtidos pelas respectivas relaxações lineares. Os tempos médios, em segundos, apresentados na Tabela 6.17, correspondem aos tempos de execução até obter o valor ótimo ou um limite superior para o valor do custo (tempo de execução da relaxação linear + tempo de execução do Algoritmo *Branch and Bound*). Na mesma tabela encontram-se os desvios padrões correspondentes aos tempos.

Os resultados computacionais apresentados na Tabela 6.17 para o Algoritmo Sep2 foram obtidos usando apenas a Estratégia UL-E4, uma vez que esta estratégia se mostrou a mais eficiente em termos de tempos médios e de *gaps* médios para o Algoritmo Sep1.

		Sep1					Sep2
		P-WMTZ+C _L	UL-E1	UL-E2	UL-E3	UL-E4	UL-E4
<i>Gap</i> Médio (%)	10	22,292	13,788	7,694	9,652	9,280	4,844
	20	9,874	6,791	6,489	0,902	1,025	0,978
	40	6,362	4,103	3,701	2,119	2,101	2,041
	60	4,830	2,691	2,866	1,453	0,773	1,267
σ_{gap}	10	7,061	3,621	4,400	3,889	4,275	4,475
	20	4,380	5,685	4,993	2,098	2,476	2,331
	40	0,840	2,128	1,885	2,807	2,782	2,715
	60	0,863	1,556	1,514	2,359	1,690	2,097
Tempo Médio (s)	10	0,109	0,203	0,229	0,289	0,262	0,273
	20	1,350	2,231	2,003	2,511	1,861	2,984
	40	41,756	89,639	114,625	72,127	54,137	72,101
	60	517,422	659,180	427,924	248,016	190,024	351,448
σ_{tempo}	10	0,042	0,093	0,053	0,078	0,091	0,100
	20	0,867	1,108	1,418	3,456	1,483	2,848
	40	29,749	88,168	175,842	86,058	43,831	55,188
	60	338,821	1093,039	526,810	254,598	239,052	330,918

Tabela 6.17: Comparação do Procedimento P-WMTZ+C_L e dos Algoritmos Sep1 e Sep2 nas DCI-LULs nas instâncias QC de 10 a 60 nodos.

No Algoritmo Sep1, as Estratégias UL-E3 e UL-E4, em geral, são as que apresentam *gaps* mais baixos e os menores tempos médios de execução.

No Algoritmo Sep2, os *gaps* médios obtidos para instâncias até 40 nodos são menores do que os obtidos pelo Algoritmo Sep1 usando a mesma estratégia, mas tendem a piorar em instâncias de 60 nodos. A nível de tempo de execução é preferível aplicar a Estratégia UL-E4 do Algoritmo Sep1.

Em qualquer um dos algoritmos os *gaps* médios são inferiores aos obtidos com o Procedimento P-WMTZ+C_L. Os limites inferiores obtidos usando estes algoritmos heurísticos de separação aumentam em todas as instâncias (de 10 a 60 nodos).

Em termos de tempos médios verifica-se uma redução em instâncias com 20 ou mais nodos no Algoritmo Sep1 usando a Estratégia UL-E4 quando comparado com o Procedimento P-WMTZ+C_L. Podemos encontrar 42,5% das instâncias de 10 a 60 nodos (17 instâncias em 40) em que os tempos de execução do Algoritmo Sep1 usando a Estratégia UL-E4 são inferiores aos tempos de execução obtidos com o Procedimento P-WMTZ+C_L.

6.2.6 Experiências Realizadas com as DGCILs

Depois de realizadas várias experiências computacionais com as diferentes combinações possíveis entre as escolhas dos conjuntos E_0 e E_1 e os diferentes procedimentos de levantamento das variáveis escolheram-se as três que apresentam melhores resultados para as DGCILs, as quais se encontram descritas a seguir.

Estratégia Generalizada 1 (GL-E1)

Escolha do conjunto E_1 : FVD2

Escolha do conjunto E_0 : FVU3

Procedimento de levantamento das variáveis: PL-DUL

Estratégia Generalizada 2 (GL-E2):

Escolha do conjunto E_1 : FVD2

Escolha do conjunto E_0 : FVU3

Procedimento de levantamento das variáveis: PL-UDL

Estratégia Generalizada 3 (GL-E3)

Escolha do conjunto E_1 : FVD2

Escolha do conjunto E_0 : FVU3

Procedimento de levantamento das variáveis: PL-ULD

As estratégias apresentadas para o Algoritmo Sep1 diferem na escolha dos conjuntos de variáveis a fixar a zero e a um e na ordem dos procedimentos de levantamento de variáveis.

Na Tabela 6.18 são apresentados os *gaps* médios (em percentagem) do limite inferior obtidos pelo Procedimento P-WMTZ+ C_L e por cada um dos Algoritmos Sep1 e Sep2 para as DGCILs e os respectivos desvios padrões. Note-se que os referidos *gaps* correspondem aos valores obtidos pelas respectivas relaxações lineares. Os tempos médios, em segundos, apresentados na Tabela 6.18, correspondem aos tempos de execução até obter o valor ótimo ou um limite superior para o valor do custo (tempo de execução da relaxação linear + tempo de execução do Algoritmo *Branch and Bound*). Na mesma tabela encontram-se os desvios padrões correspondentes aos tempos.

Os resultados computacionais apresentados na Tabela 6.18 para o Algoritmo Sep2 foram obtidos usando apenas a Estratégia GL-E2, uma vez que esta estratégia se

mostrou a mais eficiente em termos de tempos médios de execução para o Algoritmo Sep1.

		Sep1				Sep2
		P-WMTZ+C _L	GL-E1	GL-E2	GL-E3	GL-E2
<i>Gap</i> Médio (%)	10	22,292	7,533	9,477	8,009	6,493
	20	9,874	1,130	1,201	1,595	1,202
	40	6,362	1,940	2,119	3,570	2,020
	60	4,830	0,846	0,949	1,649	0,921
σ_{gap}	10	7,061	5,907	5,853	4,665	3,939
	20	4,380	2,134	2,333	2,479	2,338
	40	0,840	2,713	2,771	1,730	2,659
	60	0,863	1,872	2,027	1,984	1,964
Tempo Médio (s)	10	0,109	0,374	0,292	0,389	0,299
	20	1,350	2,178	2,052	2,086	2,828
	40	41,756	76,569	68,194	137,023	100,902
	60	517,422	456,713	339,516	445,974	369,289
σ_{tempo}	10	0,042	0,133	0,122	0,174	0,107
	20	0,867	2,766	3,211	3,091	2,650
	40	29,749	60,440	81,144	154,134	111,226
	60	338,821	532,450	473,387	495,796	410,294

Tabela 6.18: Comparação do Procedimento P-WMTZ+C_L e dos Algoritmos Sep1 e Sep2 nas DG-CILs nas instâncias QC de 10 a 60 nodos.

No Algoritmo Sep1, a Estratégia GL-E2 é a mais rápida e a Estratégia GL-E1 é a que obtém *gaps* mais baixos.

No Algoritmo Sep2, os *gaps* médios obtidos para instâncias de 10, 40 e 60 nodos são menores do que os obtidos pelo Algoritmo Sep1 usando a mesma estratégia. A nível de tempo de execução é preferível aplicar a Estratégia GL-E2 do Algoritmo Sep1.

Em qualquer um dos algoritmos os *gaps* médios são inferiores aos obtidos com o Procedimento P-WMTZ+C_L. Os limites inferiores obtidos usando estes algoritmos heurísticos de separação aumentam em todas as instâncias (de 10 a 60 nodos).

Em termos de tempos médios verifica-se uma redução apenas em instâncias de 60 nodos no Algoritmo Sep1 usando a Estratégia GL-E2 quando comparado com o Procedimento P-WMTZ+C_L. Podemos encontrar 50% das instâncias de 10 a 60 nodos (20 instâncias em 40) em que os tempos de execução do Algoritmo Sep1 usando a

Estratégia GL-E2 são inferiores aos tempos de execução obtidos com o Procedimento P-WMTZ+C_L.

6.2.7 Comparação dos Algoritmos de Separação

Com o objetivo de comparar as melhores estratégias de separação obtidas nas subseções anteriores elaborámos os gráficos das Figuras 6.16, 6.17 e 6.18. No gráfico da Figura 6.16 comparamos os tempos de execução na obtenção do valor ótimo ou de um limite superior para o valor do custo (tempo de execução da relaxação linear + tempo de execução do Algoritmo *Branch and Bound*) entre o Procedimento P-WMTZ+C e as melhores estratégias heurísticas de separação. No gráfico da Figura 6.17 apresentam-se as percentagens de soluções ótimas obtidas com as melhores estratégias de separação. Finalmente, no gráfico da Figura 6.18 comparamos o número de nodos usado no Algoritmo *Branch and Bound* entre o Procedimento P-WMTZ+C e as melhores estratégias heurísticas de separação.

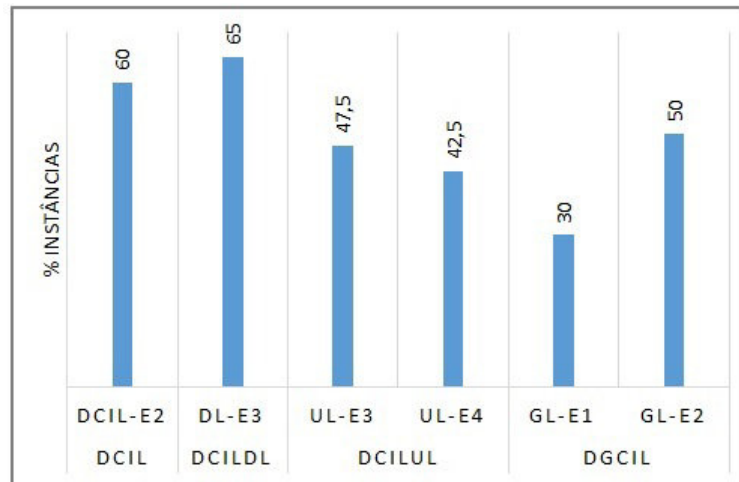


Figura 6.16: Percentagem de instâncias, em cada uma das melhores estratégias heurísticas de separação com tempo inferior ao tempo do Procedimento P-WMTZ+C_L em instâncias QC de 10 a 60 nodos.

As Estratégias DCIL-E2 e DL-E3 são as que apresentam a maior percentagem de instâncias com tempo inferior ao tempo obtido na execução do Procedimento P-WMTZ+C.

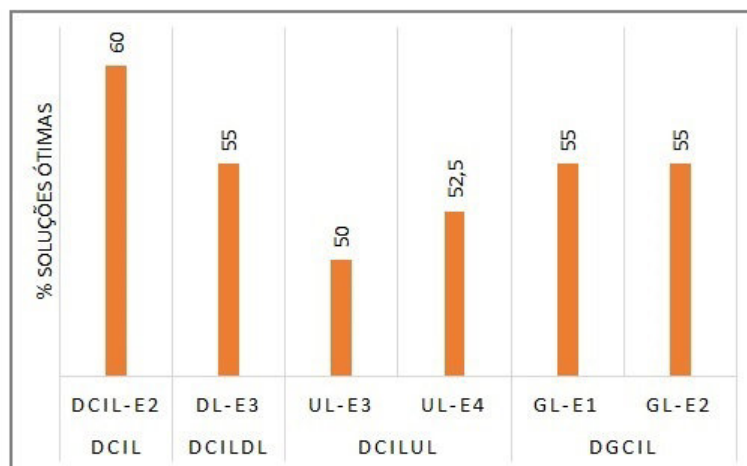


Figura 6.17: Percentagem de soluções ótimas obtidas com as melhores estratégias heurísticas de separação em instâncias QC de 10 a 60 nodos.

Em todas as estratégias heurísticas de separação apresentadas no gráfico da Figura 6.17, a percentagem de soluções ótimas é superior ou igual a 50%.

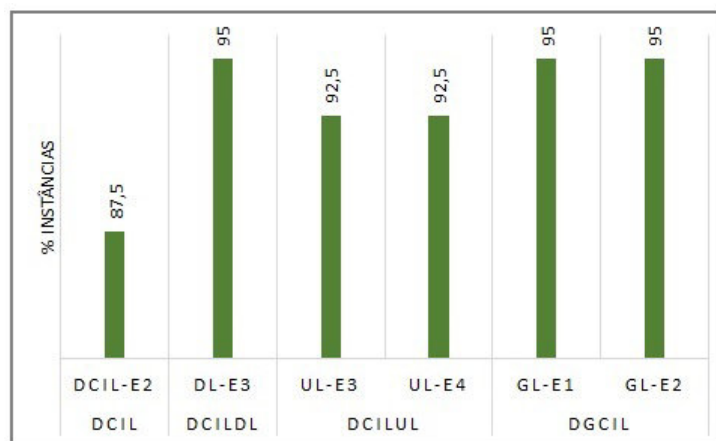


Figura 6.18: Percentagem de instâncias, em cada uma das melhores estratégias heurísticas de separação com o número de nodos no *Branch and Bound* inferior ao número usado pelo Procedimento P-WMTZ+C em instâncias QC de 10 a 60 nodos.

Podemos observar no gráfico da Figura 6.18 que as estratégias heurísticas de separação apresentadas usam um número de nodos no *Branch and Bound* inferior ao número de nodos no *Branch and Bound* usado pelo Procedimento P-WMTZ+C. As Estratégias DL-E3, GL-E1 e GL-E2 são as que apresentam uma maior percentagem de instâncias em que o número de nodos no *Branch and Bound* é inferior ao número de nodos usados pelo *Branch and Bound* do Procedimento P-WMTZ+C.

Na Tabela 6.19 resumem-se os resultados computacionais mais relevantes referentes ao Procedimento P-WMTZ+C_L e aos melhores algoritmos heurísticos de separação. Apresentam-se os *gaps* médios (em percentagem) do limite inferior e os respetivos desvios padrões. Note-se que os referidos *gaps* correspondem aos valores obtidos pelas respetivas relaxações lineares.

Os tempos médios, em segundos, apresentados na Tabela 6.19, correspondem aos tempos de execução até obter o valor ótimo ou um limite superior para o valor do custo (tempo de execução da relaxação linear + tempo de execução do Algoritmo *Branch and Bound*). Na mesma tabela encontram-se os desvios padrões correspondentes aos tempos.

			Algoritmo Sep1					
			DCILs	DCILDLs	DCILULs		DGCILs	
		P-WMTZ+C _L	DCIL-E2	DL-E3	UL-E3	UL-E4	GL-E1	GL-E2
<i>Gap</i> Médio (%)	10	22,292	9,632	8,891	9,652	9,280	7,533	9,477
	20	9,874	0,096	1,304	0,902	1,025	1,130	1,201
	40	6,362	1,650	0,940	2,119	2,101	1,940	2,119
	60	4,830	0,596	0,382	1,453	0,773	0,846	0,949
σ_{gap}	10	7,061	5,690	6,151	3,889	4,275	5,907	5,853
	20	4,380	0,276	2,604	2,098	2,476	2,134	2,333
	40	0,840	2,735	2,217	2,807	2,782	2,713	2,771
	60	0,863	1,715	1,096	2,359	1,690	1,872	2,027
Tempo Médio (s)	10	0,109	0,189	0,195	0,289	0,262	0,374	0,292
	20	1,350	1,087	1,157	2,511	1,861	2,178	2,052
	40	41,756	29,259	27,474	72,127	54,137	76,569	68,194
	60	517,422	154,743	73,554	248,016	190,024	456,713	339,516
σ_{tempo}	10	0,042	0,057	0,070	0,078	0,091	0,133	0,122
	20	0,867	0,824	0,969	3,456	1,483	2,766	3,211
	40	29,749	27,386	34,029	86,058	43,831	60,440	81,144
	60	338,821	293,342	89,572	254,598	239,052	532,450	473,387
N.º Médio de Nodos no B&B	10	60	28	21	21	15	13	18
	20	1158	8	289	595	514	136	304
	40	28775	4828	2603	7874	6396	5763	7439
	60	171415	32697	2744	23751	7228	12236	14536

Tabela 6.19: Comparação do Procedimento P-WMTZ+C_L e dos melhores algoritmos heurísticos de separação nas instâncias QC de 10 a 60 nodos.

Em relação ao *gap* médio mais baixo, podemos observar que nas instâncias de 10 nodos é obtido usando a Estratégia GL-E1, nas instâncias de 20 nodos é obtido usando a Estratégia DCIL-E2 e para 40 e 60 nodos usando a Estratégia DL-E3.

As Estratégias DCIL-E2 e DL-E3 são as mais rápidas. Em relação à Estratégia DCIL-E2 apresenta tempos médios inferiores em instâncias de 10 e 20 nodos. À medida que o número de nodos aumenta a Estratégia DL-E3 torna-se mais rápida (instâncias de 40 e 60 nodos).

Em relação ao número de nodos no *Branch and Bound* também se obtém melhores resultados usando a Estratégia DL-E3.

6.2.8 Resultados Computacionais

Nesta subsecção vamos apresentar os resultados computacionais das duas melhores estratégias heurísticas de separação, isto é, das Estratégias DCIL-E2 e DL-E3 para instâncias Quase Caminhos de 80, 100 e 150 nodos. Para estas estratégias heurísticas de separação e para o Procedimento P-WMTZ+C_L apresentamos na Tabela 6.20 os *gaps* médios (em percentagem) do limite inferior e os respectivos desvios padrões. Note-se que os referidos *gaps* correspondem aos valores obtidos pelas respectivas relaxações lineares.

Os tempos médios, em segundos, apresentados na Tabela 6.20, correspondem aos tempos de execução até obter o valor ótimo ou um limite superior para o valor do custo (tempo de execução da relaxação linear + tempo de execução do Algoritmo *Branch and Bound*). Na mesma tabela encontram-se os desvios padrões correspondentes aos tempos.

		P-WMTZ+C _L	DCIL-E2	DL-E3
<i>Gap</i> Médio (%)	80	3,778	0,643	0
	100	3,192	0,254	0
	150	1,680	1,103	0,737
σ_{gap}	80	0,884	1,356	0
	100	0,274	0,802	0
	150	0,360	0,775	0,994
Tempo Médio (<i>s</i>)	80	2535,069	422,553	74,436
	100	10596,601	1322,044	234,924
	150	10804,610	7437,602	4838,626
σ_{tempo}	80	2095,415	641,694	25,973
	100	645,980	3333,550	119,356
	150	1,819	4137,739	4669,236
N.º Médio de Nodos no B&B	80	426456	47044	12
	100	1020861	35802	1
	150	322897	201438	116934

Tabela 6.20: Comparação do Procedimento P-WMTZ+C_L e dos dois melhores algoritmos heurísticos de separação nas instâncias QC de 80 a 150 nodos.

Em instâncias Quase Caminhos de 80 a 150 nodos, quando se aplica a Estratégia DCIL-E2 do Algoritmo Sep1 obtém-se 66,7% de instâncias com solução ótima (20 em 30 instâncias). Com a aplicação da Estratégia DL-E3 do Algoritmo Sep1 obtemos 86,67% (26 em 30 instâncias). Os *gaps* médios mais baixos e os tempos médios mais baixos são obtidos usando a Estratégia DL-E3. A Estratégia DL-E3 reduz significativamente o número de nodos no *Branch and Bound*. Assim, a Estratégia DL-E3 parece ser a mais eficiente.

Instâncias Aleatórias e Euclidianas

Para instâncias do grupo Aleatórias e Euclidianas, o Procedimento P-WMTZ+C é relativamente rápido na obtenção do valor ótimo para instâncias com menos de 300 nodos no caso das instâncias Aleatórias e com menos de 200 nodos no caso das instâncias Euclidianas. Em instâncias de maiores dimensões a relaxação linear do Procedimento P-WMTZ+C é mais demorada do que a obtenção do valor ótimo depois de entrar em *Branch and Bound*. Deste modo, em termos de tempo não temos

vantagens em usar a introdução de desigualdades como planos de corte usando as estratégias heurísticas de separação. Foram efetuadas algumas experiências com algumas das instâncias Aleatórias e Euclidianas e verifica-se que quando se aplica a Estratégia DL-E3 das DCILDLs, o número de nodos no *Branch and Bound*, em geral, diminui comparativamente ao número de nodos no *Branch and Bound* obtido pelo Procedimento P-WMTZ+C.

6.2.9 Síntese dos Resultados Computacionais

Os melhores resultados são obtidos usando o Algoritmo Sep1, onde em cada iteração é introduzida apenas uma desigualdade válida.

A percentagem de soluções ótimas obtida em todas as instâncias é de 62,86% (44 em 70 instâncias) usando a Estratégia DCIL-E2 das DCILs e de 68,57% (48 em 70 instâncias) usando a Estratégia DL-E3 das DCILDLs. Para instâncias Quase Caminhos de 10 a 60 nodos, de entre as várias estratégias usadas ao aplicar o Algoritmo Sep1, as duas melhores estratégias de separação são a Estratégia DCIL-E2 e a Estratégia DL-E3, as quais obtêm as soluções de melhor qualidade no menor tempo. Para instâncias de 80 a 150 nodos concluímos que a Estratégia DL-E3 das DCILDLs é a que apresenta os melhores resultados.

Capítulo 7

Método *Feasibility Pump*

A Heurística *Feasibility Pump* (FP) é uma sequência de arredondamentos de soluções fracionárias obtidas através da relaxação linear que conduz, no caso do Problema WMST, a uma solução inteira admissível (caso exista alguma), em que o seu custo é o valor ótimo ou um limite superior para o valor ótimo.

Quando se usam formulações, os *solvers* dos vários pacotes de Otimização tais como, o do *software Xpress*, em algumas instâncias difíceis gastam muito tempo a determinar a primeira solução admissível. Por esse motivo, o desenvolvimento de métodos heurísticos genéricos que interagem com pacotes de resolução têm mostrado grande importância, do ponto de vista prático, para encontrar uma primeira solução admissível em problemas difíceis.

O Método *Feasibility Pump* é um esquema heurístico bastante utilizado devido à sua rapidez e também à qualidade da primeira solução admissível encontrada. Foi proposto por Fischetti, Glover e Lodi [18] com o objetivo de encontrar soluções admissíveis (se existir alguma) para problemas genéricos de Programação Linear Inteira Mista (MIP - *Mixed-Integer Linear Programming*). Posteriormente, o método foi melhorado por Fischetti, Bertacco e Lodi [12] e por Achterberg e Berthold [2].

Neste capítulo apresentamos alguns conceitos importantes para a aplicação do Método *Feasibility Pump* e descrevemos uma Heurística *Feasibility Pump* básica aplicada ao Problema WMST na Secção 7.1. Com o objetivo de melhorar a qualidade das soluções obtidas apresentamos, na Secção 7.2, uma versão onde se altera a função objetivo na Heurística *Feasibility Pump* aplicada ao Problema WMST. Na Secção 7.3 apresentamos algumas extensões do método com o objetivo de melhorar o seu desempenho quando aplicado ao Problema WMST. De seguida, na Secção 7.4 apresentamos

o comportamento da Heurística *Feasibility Pump* do ponto de vista geométrico, no caso geral e no caso particular do problema em estudo. Na última secção começamos por descrever e discutir algumas experiências computacionais realizadas. Posteriormente, apresentamos os resultados obtidos usando quatro estratégias Heurísticas *Feasibility Pump* e, por fim, comparamos as estratégias heurísticas propostas, em termos de qualidade da solução admissível determinada e de tempo de execução, com a Heurística do *software Xpress* aplicada ao nodo raiz utilizando a Formulação WMTZ e também com a primeira solução inteira admissível obtida através da Formulação WMTZ quando usa o procedimento *Branch and Bound*. Para terminar o capítulo apresentamos uma síntese dos resultados computacionais obtidos.

7.1 Heurística *Feasibility Pump* Básica Aplicada ao Problema WMST

Para aplicar a Heurística *Feasibility Pump* Básica ao Problema WMST, começamos por obter a solução ótima da relaxação linear de uma formulação para o Problema WMST, a qual designamos por $x^* \in X_L$. Denotamos por $C(x^*)$, o custo da solução x^* e por $W(x^*)$, o peso da solução x^* . Caso a solução x^* seja inteira, então a solução $x^* \in X_{WMST}$ corresponde a árvore de suporte T_{x^*} .

Designamos por \tilde{x} o ponto que é obtido por arredondamento do ponto x da seguinte forma: $\tilde{x}_{ij} = [x_{ij}]$, para todo o arco $(i, j) \in A$, sendo $[.]$ o operador escalar de arredondamento. No caso do operador de arredondamento para o inteiro mais próximo, o arredondamento de cada variável seria efetuado da seguinte forma:

$$\tilde{x}_{ij} = \begin{cases} 1, & \text{se } x_{ij} \geq 0,5 \\ 0, & \text{caso contrário} \end{cases}, (i, j) \in A.$$

Deste modo, todas as componentes \tilde{x}_{ij} , $\forall (i, j) \in A$, de \tilde{x} são inteiras e este ponto pode ser considerado uma solução inteira. Denotamos por $C(\tilde{x})$, o custo da solução \tilde{x} e por $W(\tilde{x})$, o peso da solução \tilde{x} . Em geral, a referida solução não é admissível, embora em alguns casos possa ser uma solução inteira admissível. Caso a solução \tilde{x} corresponda a uma árvore de suporte esta será designada por $T_{\tilde{x}}$, e o custo será denotado por $C(T_{\tilde{x}})$ e o peso por $W(T_{\tilde{x}})$.

Exemplo 7.1.

Consideramos as variáveis com solução positiva na solução da relaxação linear (Exemplo 3.2),

$$x^* : x_{01} = 1; x_{04} = 0,2; x_{12} = 1; x_{13} = 0,8; x_{34} = 0,8 \text{ e } x_{43} = 0,2;$$

relativa à instância de 5 nodos obtida no Exemplo 2.1. Se efetuarmos o arredondamento para o inteiro mais próximo de todas as variáveis (com solução positiva), obtemos a solução,

$$\tilde{x} = [x^*] : \tilde{x}_{01} = 1; \tilde{x}_{04} = 0; \tilde{x}_{12} = 1; \tilde{x}_{13} = 1; \tilde{x}_{34} = 1 \text{ e } \tilde{x}_{43} = 0.$$

A esta solução \tilde{x} corresponde a árvore de suporte $T_{\tilde{x}}$ que se encontra representada na Figura 7.1.

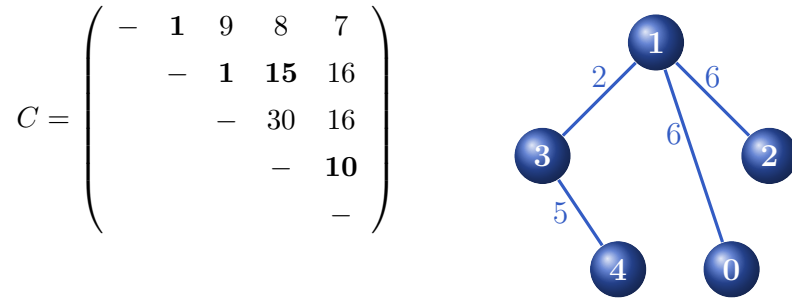


Figura 7.1: Árvore de suporte $T_{\tilde{x}}$ com custo 27 e peso 19.

O peso desta árvore é $W(T_{\tilde{x}}) = 19 < 20$ e o custo é $C(T_{\tilde{x}}) = 27$. Logo, neste caso, a solução \tilde{x} é uma solução inteira admissível e corresponde à solução ótima desta instância de 5 nodos.

Suponhamos que obtemos o seguinte arredondamento,

$$\tilde{x} = [x^*] : \tilde{x}_{01} = 1; \tilde{x}_{04} = 1; \tilde{x}_{12} = 1; \tilde{x}_{13} = 1; \tilde{x}_{34} = 0 \text{ e } \tilde{x}_{43} = 0.$$

A esta solução \tilde{x} corresponde a árvore de suporte $T_{\tilde{x}}$ que se encontra representada na Figura 7.2.

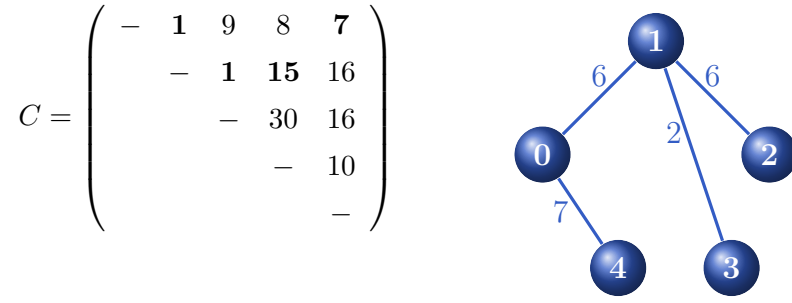


Figura 7.2: Árvore de suporte $T_{\tilde{x}}$ com custo 24 e peso 21.

O peso desta árvore é $W(T_{\tilde{x}}) = 21 > 20$ e o custo é $C(T_{\tilde{x}}) = 24$. Logo, neste caso, \tilde{x} é uma solução inteira não admissível, pois não verifica a restrição de peso.

Na primeira situação do Exemplo 7.1 a solução \tilde{x} é inteira admissível ($\tilde{x} \in X_{WMST}$), pois é uma árvore de suporte que verifica a restrição de peso, isto é, $W(T_{\tilde{x}}) \leq W$. Na segunda situação do mesmo exemplo, a solução \tilde{x} é inteira e corresponde a uma árvore de suporte, no entanto esta árvore não verifica a restrição de peso, isto é, $W(T_{\tilde{x}}) > W$, pelo que nesta situação dizemos que \tilde{x} é uma solução inteira não admissível. Além destes dois casos que observámos no Exemplo 7.1, também podem ocorrer mais dois casos, o caso em que a restrição de peso é verificada, mas a solução \tilde{x} não corresponde a uma árvore de suporte e o caso em que não se verifica a restrição de peso e \tilde{x} não corresponde a uma árvore de suporte. Nestes dois últimos casos a solução \tilde{x} é inteira não admissível. Em resumo, a solução \tilde{x} é sempre inteira, e pode ou não ser admissível dependendo se satisfaz ou não as restantes restrições (ser árvore de suporte e verificar a restrição de peso).

A função distância entre um ponto $x \in X_L$ e um ponto inteiro \tilde{x} pode ser definida pela norma L_1 tal que,

$$\Delta(x, \tilde{x}) = \sum_{(i,j) \in A} |x_{ij} - \tilde{x}_{ij}|, \quad (7.1)$$

e pode ser reescrita da seguinte forma

$$\Delta(x, \tilde{x}) = \sum_{(i,j) \in A: \tilde{x}_{ij}=0} x_{ij} + \sum_{(i,j) \in A: \tilde{x}_{ij}=1} (1 - x_{ij}). \quad (7.2)$$

Como tal, a função não linear definida por (7.1) pode ser transformada na função linear definida por (7.2).

Dado um ponto inteiro \tilde{x} , o ponto $x^* \in X_L$ mais próximo de \tilde{x} pode ser determinado através da minimização de um novo problema, cujas restrições são as mesmas do Problema WMST e apenas a função objetivo é alterada para

$$\vartheta(x^*) = \min \Delta(x^*, \tilde{x}).$$

$$s.a. x^* \in X_L.$$

Se a distância for nula, isto é, se $\vartheta(x^*) = 0$, então x^* é uma solução inteira admissível obtida para o Problema WMST. Por outro lado, se $\vartheta(x^*) > 0$ e for obtido um ponto $x^* \in X_L$, então o novo ponto inteiro \tilde{x} mais próximo de x^* é facilmente obtido por arredondamento de x^* . Assim, a Heurística *Feasibility Pump* sugere que para encontrar uma solução inteira admissível para o Problema WMST se atualize, iterativamente, o par de pontos (x^*, \tilde{x}) com o objetivo de reduzir, tanto quanto possível, a sua distância $\Delta(x^*, \tilde{x})$.

De seguida apresentamos uma descrição resumida da Heurística *Feasibility Pump* Básica Aplicada ao Problema WMST, posteriormente, explicamos de forma mais detalhada o seu funcionamento. Denote-se por TL o tempo limite imposto para executar a heurística e *maxIter* o número máximo de iterações.

Heurística *Feasibility Pump* Básica Aplicada ao Problema WMST

Passo 1: *Obter a solução da relaxação linear*

Seja x^* a solução da relaxação linear de uma formulação para o Problema WMST.

Se x^* é inteira, então

x^* é a solução ótima. STOP.

Caso contrário,

ir para o **Passo 2**.

Passo 2: *Arredondar os valores da solução da relaxação linear*

Seja $\tilde{x}^0 := \lfloor x^* \rfloor$.

Passo 3: *Critério de paragem e pumping cycle*

$k := 0$.

Enquanto (tempo < TL e $k < \text{maxIter}$) fazer

$k := k + 1$.

Determinar x^* resolvendo a relaxação linear:

$$\vartheta(x^*) = \min \Delta(x^*, \tilde{x}^{k-1})$$

$$\text{s.a. } x^* \in X_L.$$

Se $\vartheta(x^*) = 0$, então

x^* é uma solução inteira admissível, STOP.

Caso contrário,

Se $\exists(i, j) \in A : [x_{ij}^*] \neq \tilde{x}_{ij}^{k-1}$, então

atualizar $\tilde{x}^k := [x^*]$.

Caso contrário,

aplicar Mecanismo de Perturbação¹.

A heurística inicializa no Passo 1 com um ponto inicial x^* que é, em geral, a solução da relaxação linear de uma formulação para o Problema WMST. Caso a solução x^* seja inteira ($x^* \in X_{WMST}$), então foi obtida a solução ótima e a heurística termina. Caso contrário, definimos no Passo 2 uma solução inteira \tilde{x} obtida através do arredondamento da solução x^* .

No Passo 3, enquanto o tempo limite TL e o número máximo de iterações maxIter não forem excedidos, executamos uma iteração do Método *Feasibility Pump*, chamada *pumping cycle*. Em cada *pumping cycle* consideramos a solução corrente \tilde{x} e, através da relaxação linear, encontramos o ponto $x^* \in X_L$ que está mais próximo da solução corrente \tilde{x} , minimizando assim a distância $\Delta(x^*, \tilde{x})$.

Se $\vartheta(x^*) = 0$, isso significa que x^* é uma solução inteira admissível ($x^* \in X_{WMST}$) e a heurística termina. Caso contrário, se pelo menos uma das componentes $[x_{ij}^*]$ for diferente da respetiva componente \tilde{x}_{ij} obtida na iteração anterior, substituímos \tilde{x} pelo arredondamento de x^* de modo a reduzir ainda mais a distância $\Delta(x^*, \tilde{x})$.

¹Este Mecanismo de Perturbação será explicado detalhadamente mais adiante.

O principal problema na aplicação deste método está na possibilidade de depois de um certo número de iterações o método poder entrar em ciclo. A ocorrência de ciclo deve-se ao facto de se encontrarem várias vezes a mesma sequência de pontos x^* e \tilde{x} , o que leva a que a distância $\Delta(x^*, \tilde{x})$ não seja reduzida. De modo a contornar este problema, introduzimos um simples mecanismo de perturbação (explicado mais adiante) na solução corrente, fazendo com que o processo de busca e convergência seja reiniciado a partir de uma nova solução x^* .

O processo termina quando uma solução inteira for encontrada, ou quando o tempo limite ou o número máximo de iterações impostos forem excedidos.

Se o tempo limite ou o número máximo de iterações forem excedidos e não houve sucesso em obter uma solução inteira admissível, a Heurística *Feasibility Pump* termina com falha.

7.2 Alteração da Função Objetivo na Heurística *Feasibility Pump* Aplicada ao Problema WMST

Com o objetivo de obter soluções de melhor qualidade os autores Achterberg e Berthold [2] propõem uma modificação na Heurística *Feasibility Pump* original para os Problemas MIP. Tendo por base as ideias de Achterberg e Berthold [2] podemos adaptar esta pequena modificação para aplicar ao Problema WMST.

Na Heurística *Feasibility Pump* Básica para o Problema WMST apresentada na Secção 7.1, apenas se utiliza a função objetivo do Problema WMST na obtenção da primeira solução admissível $x^* \in X_L$. Nas restantes iterações utiliza-se como função objetivo a função dada por (7.2). Nesta heurística modificada a primeira solução é obtida como na heurística anterior e nas restantes iterações vamos usar uma função objetivo modificada que contenha também informação da função objetivo do Problema WMST.

Assim, a função objetivo (7.2) é substituída por uma combinação linear convexa entre a função objetivo do Problema WMST $\sum_{(i,j) \in A} c_{ij}x_{ij} = c^T x$ e a função distância $\Delta(x, \tilde{x})$, e passa a ser definida por,

$$\Delta_\alpha(x, \tilde{x}) := (1 - \alpha)\Delta(x, \tilde{x}) + \alpha \frac{\sqrt{|A|}}{\|c\|} c^T x, \quad (7.3)$$

onde $\|c\|$ é a norma Euclideana do vetor dos custos e $\alpha \in [0, 1]$. Deste modo, à medida que o número de iterações aumenta, a influência da função objetivo do Problema WMST vai sendo, progressivamente, reduzida dando lugar à função objetivo (7.2) quando o valor de $\alpha = 0$ é atingido. Deste modo, aumenta-se a probabilidade de convergência para soluções inteiras admissíveis de melhor qualidade.

O Passo 3 da heurística descrita na Secção 7.1 é alterado para:

Passo 3: *Critério de paragem e pumping cycle*

$k := 0$.

Atribuir um valor a α_k e a φ .

Enquanto (tempo < TL e $k < \maxIter$) fazer

$k := k + 1$.

Determinar x^* resolvendo a relaxação linear:

$$\vartheta_{\alpha_k}(x^*) = \min \Delta_{\alpha_k}(x^*, \tilde{x}^{k-1})$$

$$s.a. x^* \in X_L.$$

Se $\vartheta_{\alpha_k}(x^*) = 0$, então

x^* é uma solução inteira admissível, STOP.

Caso contrário,

Se $\exists(i, j) \in A : [x_{ij}^*] \neq \tilde{x}_{ij}^{k-1}$, então

atualizar $\tilde{x}^k := [x^*]$.

Caso contrário,

Se $\tilde{x}^k = \tilde{x}^{k-1}$ e $\alpha_k - \alpha_{k-1} \leq \delta_\alpha$, então

aplicar Mecanismo de Perturbação¹.

Guardar o par (\tilde{x}^k, α_k) .

Atualizar $\alpha_k = \varphi \alpha_{k-1}$.

Os Passos 1 e 2 da heurística apresentada na Secção 7.1 mantêm-se inalterados.

No Passo 3, o valor de α é inicializado com $\alpha_0 \in [0, 1]$ e é atribuído um valor fixo a $\varphi \in]0, 1[$. A solução x^* é determinada usando a função objetivo (7.3) em vez da (7.2).

¹Este Mecanismo de Perturbação será explicado detalhadamente mais adiante.

O valor de α é geometricamente reduzido em cada iteração segundo o parâmetro fixo φ , ou seja, $\alpha_k = \varphi\alpha_{k-1}$.

Note-se que se $\alpha_k = 0$, então $\Delta_{\alpha_k}(x, \tilde{x}) := \Delta(x, \tilde{x})$, o que corresponde à função objetivo utilizada na Heurística *Feasibility Pump* Básica Aplicada ao Problema WMST.

Em cada iteração, o par (\tilde{x}^k, α_k) é guardado e o método pode entrar em ciclo se existirem duas iterações seguidas $k-1$ e k , tais que, os correspondentes pares $(\tilde{x}^{k-1}, \alpha_{k-1})$ e (\tilde{x}^k, α_k) satisfazem $\tilde{x}^k = \tilde{x}^{k-1}$ e $\alpha_k - \alpha_{k-1} \leq \delta_\alpha$, para um dado valor $\delta_\alpha \in [0, 1]$. No caso do método entrar em ciclo também podemos aplicar os mecanismos de perturbação que serão descritos mais adiante na Secção 7.3.

7.3 Extensões da Heurística *Feasibility Pump* para o Problema WMST

O interesse ao aplicarmos a Heurística *Feasibility Pump* ao Problema WMST é o de obter soluções inteiras admissíveis de boa qualidade e de forma rápida. Assim, temos de ter em conta o modo de obtenção das sucessivas soluções $x^* \in X_L$, os critérios a adotar para efetuar o arredondamento das variáveis e os mecanismos de perturbação a aplicar.

7.3.1 Modo de Obtenção das Soluções x^*

A solução inicial pode ser obtida através da solução da relaxação linear do Procedimento P-WMTZ+C apresentado na Secção 3.3, pois foi aquele que apresentou os melhores resultados em termos de tempo de execução para todos os grupos de instâncias em teste.

Em instâncias grandes Aleatórias e Euclidianas a relaxação linear do Procedimento P-WMTZ+C demora muito tempo na obtenção de uma solução admissível (ver resultados computacionais do Capítulo 3), então se utilizarmos este procedimento a Heurística *Feasibility Pump* iria levar muito tempo para a obtenção de uma solução inteira admissível. Assim, para contornar este problema e reduzir o tempo de execução podemos

aplicar as seguintes estratégias:

1. Todas as soluções $x^* \in X_L$ são obtidas recorrendo à relaxação linear da Formulação WMTZ.
2. A primeira solução $x^* \in X_L$ é obtida através da relaxação linear da Formulação WMTZ e durante cada *pumping cycle* usamos a relaxação linear do Procedimento P-WMTZ+C para obter as sucessivas soluções $x^* \in X_L$.
3. A primeira solução $x^* \in X_L$ é obtida através da relaxação linear do Procedimento P-WMTZ+C com um tempo limite na introdução de cortes e durante cada *pumping cycle* usamos a relaxação linear da Formulação WMTZ para obter as sucessivas soluções $x^* \in X_L$.
4. Todas as soluções $x^* \in X_L$ são obtidas recorrendo à relaxação linear do Procedimento P-WMTZ+C com um tempo limite na introdução dos cortes.

Na primeira estratégia obtemos uma solução $x^* \in X_L$ de forma rápida, mas fraca. Usando o Procedimento P-WMTZ+C a solução $x^* \in X_L$ encontrada é de melhor qualidade, mas é obtida de forma mais lenta. Para tentar diminuir o tempo de execução, na terceira estratégia usamos o Procedimento P-WMTZ+C com tempo limitado na introdução dos cortes para obtenção da solução inicial $x^* \in X_L$ e em cada *pumping cycle* usamos a Formulação WMTZ para obter as sucessivas soluções $x^* \in X_L$. Na última estratégia usamos sempre o Procedimento P-WMTZ+C com limite de tempo na introdução dos cortes.

7.3.2 Critérios de Arredondamento das Variáveis

Observamos que um dos aspetos a ter em conta, em relação ao arredondamento das variáveis, é o facto de que se não efetuamos o arredondamento a algumas variáveis, nem para zero nem para um, ficando elas com um valor fracionário, a heurística termina quando obtiver uma solução admissível (não necessariamente inteira). O que leva a concluir que para obter uma solução inteira admissível temos de efetuar o arredondamento a todas as variáveis. Notamos que se não efetuarmos o arredondamento a todas

as variáveis o valor obtido para o custo pode corresponder tanto a um limite superior como a um limite inferior.

Para o arredondamento das variáveis podemos estabelecer diferentes critérios. De seguida apresentamos alguns critérios gerais que se podem utilizar para efetuar o arredondamento das variáveis.

- **Critério de Arredondamento 1 (A1)**

$$\tilde{x}_{ij} = \begin{cases} 1, & \text{se } x_{ij} \geq a \\ 0, & \text{caso contrário} \end{cases}, (i, j) \in A,$$

onde a constante a pode tomar valores no intervalo $]0, 1[$. Note-se que no caso de $a = 0,5$ temos o chamado arredondamento simétrico.

- **Critério de Arredondamento 2 (A2)**

$$\tilde{x}_{ij} = \begin{cases} 1, & \text{se } x_{ij} \geq b \\ 0, & \text{se } x_{ij} \leq a \\ p, & \text{caso contrário} \end{cases}, (i, j) \in A,$$

onde a constante a pode tomar valores no intervalo $]0; 0,5[$ e a constante b no intervalo $]0,5; 1[$. A constante p pode tomar apenas dois valores, isto é, $p \in \{0, 1\}$. Para obter o valor de p , geramos aleatoriamente um valor $p_{ij} \in [0, 1]$. Se $p_{ij} \geq 0,5$, então $p = 1$, caso contrário $p = 0$.

- **Critério de Arredondamento 3 (A3)**

$$\tilde{x}_{ij} = \begin{cases} 1, & \text{se } x_{ij} \geq a \text{ ou } (i, j) \in T \\ 0, & \text{caso contrário} \end{cases}, (i, j) \in A,$$

onde a constante a pode tomar valores no intervalo $]0, 1[$ e T é uma árvore de suporte. Esta árvore pode ser a árvore de suporte de custo mínimo T_c , a árvore de suporte de peso mínimo T_w , ou uma árvore de suporte T_p que verifique a restrição de peso.

7.3.3 Mecanismos de Perturbação

Os mecanismos de perturbação são utilizados sempre que todas as componentes \tilde{x}_{ij}^k $((i, j) \in A)$ do novo \tilde{x}^k coincidam com as componentes de \tilde{x}^{k-1} , isto é, se em duas iterações consecutivas as soluções arredondadas são iguais. No caso do valor de $\vartheta(x^*)$ ser o mesmo em três iterações seguidas, também é utilizado o mecanismo de perturbação. Nestas duas situações temos de modificar algumas componentes \tilde{x}_{ij} $((i, j) \in A)$, mesmo que aumente o valor da distância $\Delta(x^*, \tilde{x})$, de modo que o processo de busca e convergência seja reiniciado a partir de uma nova solução x^* .

Neste sentido, podemos utilizar os dois mecanismos que se seguem e que são idênticos aos utilizados em [18]:

- **Mecanismo de Perturbação 1 (P1):**

Modificar apenas as variáveis \tilde{x}_{ij} $((i, j) \in A)$, tais que $|x_{ij} - \tilde{x}_{ij}| > \sigma$.

No caso de $x_{ij} - \tilde{x}_{ij} > \sigma$, então modificar o valor de $\tilde{x}_{ij} = 0$ para $\tilde{x}_{ij} = 1$. Caso contrário, modificar para $\tilde{x}_{ij} = 0$.

- **Mecanismo de Perturbação 2 (P2):**

Para cada $(i, j) \in A$, gerar um valor aleatório $\rho_{ij} \in [-0,3; 0,7]$ e modificar apenas as variáveis \tilde{x}_{ij} $((i, j) \in A)$, tais que $|x_{ij} - \tilde{x}_{ij}| + \max\{\rho_{ij}, 0\} > \delta$.

No caso de $x_{ij} - \tilde{x}_{ij} + \max\{\rho_{ij}, 0\} > \delta$, então modificar o valor de $\tilde{x}_{ij} = 0$ para $\tilde{x}_{ij} = 1$. Caso contrário, modificar para $\tilde{x}_{ij} = 0$.

O Mecanismo P1 pode ser considerado fraco pois, quando temos $|x_{ij} - \tilde{x}_{ij}| = 0$ $((i, j) \in A)$ essas variáveis não são modificadas. Quanto ao Mecanismo P2, este pode ser considerado forte, uma vez que dá a possibilidade também às variáveis tais que $|x_{ij} - \tilde{x}_{ij}| = 0$ $((i, j) \in A)$ de serem modificadas, desde que $\max\{\rho_{ij}, 0\} > \delta$. Assim, o conjunto de variáveis que podem ser trocadas no Mecanismo P2 é maior do que no Mecanismo P1.

Com a aplicação dos mecanismos de perturbação podem ser efetuadas muitas trocas e alterado o valor de todas as variáveis da solução. Assim, para evitar que tal aconteça, é necessário limitar também o número de trocas. De acordo com [18], podemos efetuar um número inteiro de trocas pertencente ao intervalo $\left[\frac{M}{2}, \frac{3M}{2}\right]$, onde M

é um parâmetro fixo. Para encontrar este valor geramos um número inteiro aleatório no intervalo considerado.

7.4 Comportamento da Heurística *Feasibility Pump*

Do ponto de vista geométrico o comportamento da Heurística *Feasibility Pump* gera duas trajetórias de pontos x^* e \tilde{x} , que se esperam ser convergentes. Uma trajetória satisfaz as restrições da relaxação linear e pode ou não satisfazer as restrições de integralidade. A outra trajetória satisfaz as restrições de integralidade, podendo ou não satisfazer as restantes restrições dadas por (3.2) e/ou a restrição de peso, isto é, uma trajetória é formada por uma sucessão de pontos $x^* \in X_L$ (que podem ser ou não inteiros), enquanto que a outra trajetória é formada por uma sucessão de pontos \tilde{x} inteiros (que podem ser ou não admissíveis para o Problema WMST).

A distância $\Delta(x^*, \tilde{x})$ pode ser interpretada como a diferença entre os pontos $x^* \in X_L$ e \tilde{x} que tentamos reduzir efetuando o *pumping cycle*.

Como a cada ponto x^* e \tilde{x} se faz corresponder um custo e um peso, também se pode interpretar a Heurística *Feasibility Pump* como duas trajetórias de custos e duas trajetórias de pesos que se esperam convergentes para um mesmo custo e um mesmo peso,

$$C(T_{x^*}) = C(T_{\tilde{x}}) \text{ e } W(T_{x^*}) = W(T_{\tilde{x}}).$$

A heurística termina quando as duas sequências convergem ou quando o tempo ou o número de iterações imposto forem excedidos.

Nos três gráficos que se seguem podemos observar geometricamente o comportamento da instância 80-7 do grupo de instâncias Quase Caminhos com valor $W = 1000$.

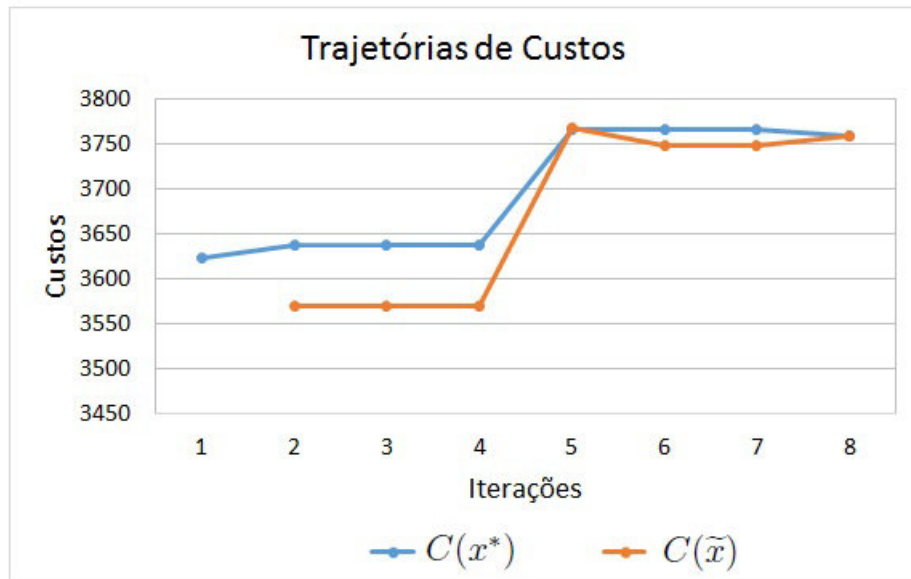


Figura 7.3: Representação geométrica das duas trajetórias de custos (instância QC80-7).

Na Figura 7.3 encontram-se representadas as sucessões de pontos relativos aos custos das duas seqüências de soluções x^* e \tilde{x} . Observamos que as duas trajetórias de custos convergem após oito iterações para o mesmo custo 3759.

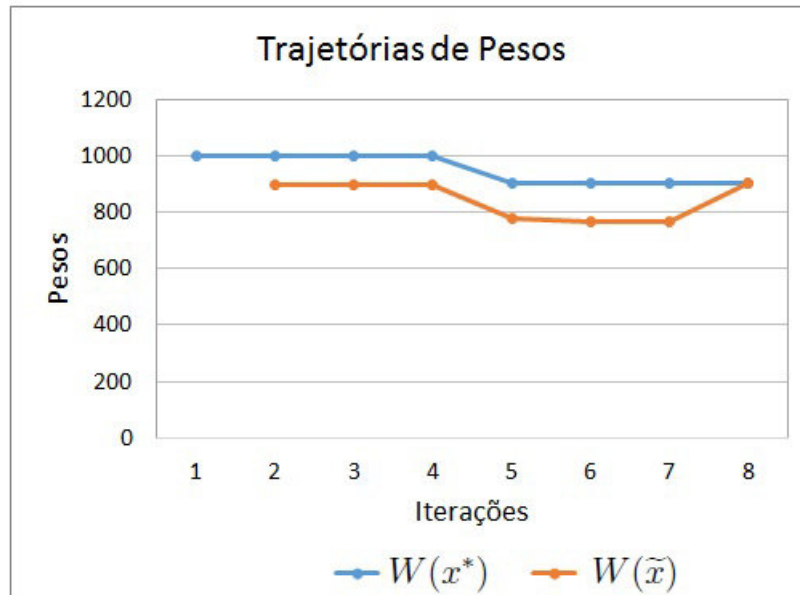


Figura 7.4: Representação geométrica das duas trajetórias de pesos (instância QC80-7).

Na Figura 7.4 encontram-se representadas as sucessões de pontos relativos aos pesos das duas seqüências de soluções x^* e \tilde{x} . Observamos que as duas trajetórias de pesos convergem após 8 iterações para o mesmo peso 904.

Note-se que na Iteração 1 apenas existe custo e peso para a solução x^* , pois é a primeira solução obtida que corresponde à solução ótima da relaxação linear e ainda não foi efetuado o primeiro arredondamento para obter \tilde{x} .

Neste gráfico podemos observar ainda se as soluções \tilde{x} verificam a restrição de peso. Neste exemplo, o peso das sucessivas soluções \tilde{x} é sempre inferior ou igual a 1000. Observe-se que em outros exemplos as soluções \tilde{x} podem não verificar a restrição de peso.

Notemos também que na Iteração 5 observamos no gráfico da Figura 7.3 que $C(x^*) = C(\tilde{x})$, mas através da observação do gráfico da Figura 7.4 podemos verificar que as soluções não são as mesmas, dado que $W(x^*) \neq W(\tilde{x})$.

Entre cada um dos pontos x^* e \tilde{x} está associada uma distância $\Delta(x^*, \tilde{x})$, que se pretende minimizar, isto é, que se tenta reduzir a zero. No seguinte gráfico podemos visualizar o comportamento da função distância ao longo das oito iterações na instância 80-7 do grupo de instâncias Quase Caminhos.

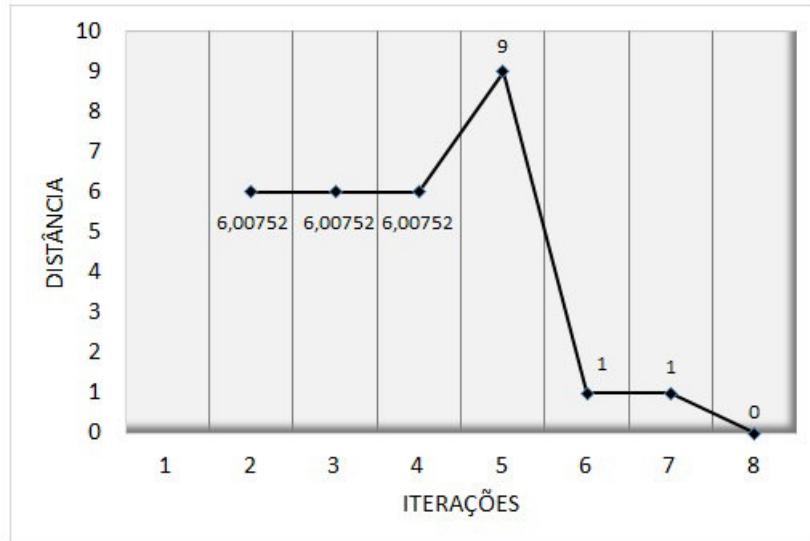


Figura 7.5: Evolução da distância $\Delta(x^*, \tilde{x})$ ao longo das iterações (instância QC80-7).

No gráfico da Figura 7.5 observamos que a função distância, iterativamente, se reduz

a zero, ao fim de oito iterações. Note-se que só a partir da Iteração 2 é que se minimiza a distância. Neste exemplo, observamos a existência de três iterações seguidas com a mesma distância 6,00752 (Iterações 2, 3 e 4), o que leva a efetuarmos modificações nas variáveis através da introdução de um mecanismo de perturbação e, devido a este facto, existe um crescimento do valor da função distância para 9 (Iteração 5).

Nesta instância, a solução inteira admissível obtida é uma árvore de suporte de custo 3759 e peso 904, que não corresponde à solução ótima da instância considerada, mas apresenta um *gap* de 0,21%.

7.5 Experiências e Resultados Computacionais

Nesta secção, começamos por descrever e discutir algumas experiências computacionais realizadas com as duas heurísticas propostas nas Secções 7.1 e 7.2. De seguida, propomos quatro estratégias heurísticas e apresentamos um exemplo comparativo entre as quatro estratégias heurísticas propostas. Por fim, os resultados computacionais são comparados entre si, com a Heurística do *software Xpress* aplicada ao nodo raiz utilizando a Formulação WMTZ e com a primeira solução inteira admissível obtida pela Formulação WMTZ quando usa o procedimento de *Branch and Bound*.

Todos os testes foram efetuados num Intel(R) Core(TM)2 Duo CPU (T7100) 2.00 GHz processador e 4Gb de RAM e usou-se o *software Xpress* 7.3 (*Xpress-Optimizer 23.01.03* e *Xpress-Mosel 3.4.0*) [1] para implementar todas as estratégias heurísticas.

Em todas as heurísticas é obtido um limite superior para o valor do custo, que pode ou não ser o valor ótimo. O *gap* correspondente é dado por $gap = \frac{LS - OPT}{OPT} \times 100$, onde *LS* é o valor do limite superior obtido para o valor do custo através da heurística considerada e *OPT* é o valor ótimo ($\vartheta(WMST)$) ou o melhor valor encontrado até ao momento.

7.5.1 Descrição das Experiências Computacionais Realizadas

Das experiências computacionais realizadas verificámos que as três primeiras estratégias de obtenção das soluções $x^* \in X_L$ (Subsecção 7.3.1) não apresentaram bons resultados nas instâncias Aleatórias e Euclideanas, pois ocorreram muitos casos em que as Heurísticas *Feasibility Pump* terminaram com falhas devido ao tempo limite imposto ou ao número máximo de iterações ter sido excedido. É de referir ainda que nas instâncias Quase Caminhos qualquer uma destas estratégias apresentou bons resultados.

No arredondamento das variáveis, em relação ao Critério A1 foram efetuadas experiências com vários valores de a ($a = 0,5$; $a = 0,6$; $a = 0,75$ e outros) e verificámos que em todos os grupos de instâncias, os melhores resultados se obtêm com um valor de $a = 0,5$, ou seja, efetuando o arredondamento para o inteiro mais próximo. Com base nas experiências efetuadas observámos que quanto mais próximo do valor um estiver o valor de a , maior é o tempo de execução e o *gap*. Este facto pode ser devido a que com o aumento da constante a , o número de variáveis que se arredondam para o valor um diminui.

Após experiências com vários valores de a e de b ($a \in]0; 0,5[$ e $b \in]0,5; 1[$), o Critério A2 apresentou bons resultados com $a = 0,4$ e $b = 0,6$ para instâncias Quase Caminhos. Nos grupos de instâncias Aleatórias e Euclideanas ocorrem muitos casos em que as Heurísticas *Feasibility Pump* terminaram com falhas devido ao tempo limite imposto ou ao número máximo de iterações ter sido excedido ou as soluções encontradas não serem de boa qualidade.

Em relação ao Critério A3 verificámos que é importante a escolha de uma árvore com custo próximo do custo da solução ótima. Os melhores resultados obtêm-se utilizando a árvore de suporte que se designou por T_p , sendo esta árvore obtida através do Algoritmo Lagrangeano 2 da Subsecção 4.4.

Em relação à aplicação dos mecanismos de perturbação foram testadas as seguintes estratégias:

- utilizar apenas um mecanismo de perturbação, ou o Mecanismo P1 ou o Mecanismo P2;

- utilizar os dois mecanismos de perturbação de forma intercalada, isto é, em iterações ímpares utilizar o Mecanismo P1 e em iterações pares o Mecanismo P2;
- utilizar os dois mecanismos de perturbação de forma aleatória, isto é, gerar um número aleatório p , entre 0 e 1. Se $p \geq 0,5$, utilizar o Mecanismo P2, caso contrário, utilizar o Mecanismo P1.

A aplicação de qualquer uma destas combinações dos mecanismos de perturbação leva à obtenção de resultados satisfatórios, embora a utilização dos dois mecanismos P1 e P2 de forma aleatória seja melhor em termos de tempo de execução. Para utilizarmos estes dois mecanismos temos de escolher valores para os parâmetros σ e δ . Das experiências computacionais realizadas com σ e δ no intervalo $[0; 0,5]$, os parâmetros $\sigma = 0,01$ e $\delta = 0,4$, apresentam os melhores resultados. Na Heurística FP com a função objetivo alterada, depois de efetuadas experiências com vários valores de α_0 , φ e δ_α ($\alpha_0, \delta_\alpha \in]0, 1[$ e $\varphi \in [0, 1]$), decidimos utilizar $\alpha_0 = \varphi = 0,1$ e $\delta_\alpha = 0,05$, para evitar a ocorrência de muitas falhas na heurística e tentar que o tempo não aumente muito em relação à Heurística FP Básica.

Para a obtenção dos resultados computacionais em todas as heurísticas apresentadas, impusemos um limite de tempo $TL = 10000$ segundos e um número máximo de iterações $maxIter = 50$. Para os restantes parâmetros decidimos utilizar $\sigma = 0,01$; $\delta = 0,4$; $\alpha_0 = \varphi = 0,1$ e $\delta_\alpha = 0,05$.

Tendo por base as experiências computacionais descritas anteriormente, apresentamos de seguida quatro esquemas heurísticos cujos resultados computacionais serão apresentados na subsecção seguinte.

Heurística *Feasibility Pump* Básica 1 (FPB1)

- A **obtenção das soluções** x^* é efetuada recorrendo à relaxação linear do Procedimento P-WMTZ+C com um limite de tempo igual a 100 segundos na introdução dos cortes.
- O **critério de arredondamento** utilizado é o Critério A1 com $a = 0,5$.

- O **mecanismo de perturbação** utilizado é o que combina os dois Mecanismos P1 e P2 de forma aleatória com os parâmetros $\sigma = 0,01$ e $\delta = 0,4$. Só é permitido efetuar um número inteiro de trocas pertencente ao intervalo $\left[\frac{M}{2}, \frac{3M}{2}\right]$, com $M = 20$.

Heurística *Feasibility Pump* Básica 2 (FPB2)

- A **obtenção das soluções** x^* é efetuada recorrendo à relaxação linear da Formulação WMTZ.
- O **critério de arredondamento** utilizado é o Critério A3 com $a = 0,9$ e a utilização da árvore de suporte $T = T_p$ obtida através do Algoritmo Lagrangeano 2 da Subsecção 4.4.
- O **mecanismo de perturbação** utilizado é o mesmo que na Heurística FPB1.

Heurística *Feasibility Pump* com Alteração da Função Objetivo 1 (FPO1)

Aplicar a Heurística FP com alteração da função objetivo proposta para o Problema WMST com os mesmos critérios adotados na Heurística FPB1.

Heurística *Feasibility Pump* com Alteração da Função Objetivo 2 (FPO2)

Aplicar a Heurística FP com alteração da função objetivo proposta para o Problema WMST com os mesmos critérios adotados na Heurística FPB2.

7.5.2 Exemplos Comparativos das Heurísticas *Feasibility Pump*

De seguida apresentamos alguns gráficos (Figuras 7.6, 7.7 e 7.8), onde podemos observar o comportamento da função distância ao longo das várias iterações usando as quatro estratégias heurísticas propostas anteriormente. O gráfico da Figura 7.6 é relativo à instância 80-7 do grupo de instâncias Quase Caminhos (QC80-7), o da Figura 7.7 é relativo à instância 80-5 do grupo de instâncias Aleatórias (R80-5) e o gráfico da Figura 7.8 refere-se à instância 80-3 do grupo de instâncias Euclidianas (E80-3).

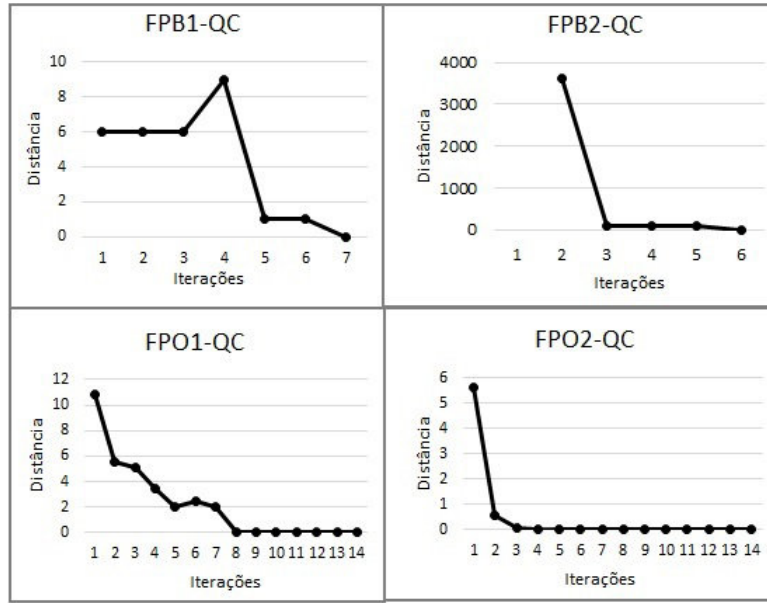


Figura 7.6: Evolução da distância $\Delta(x^*, \tilde{x})$ ao longo das iterações na instância QC80-7 nas quatro estratégias heurísticas propostas.

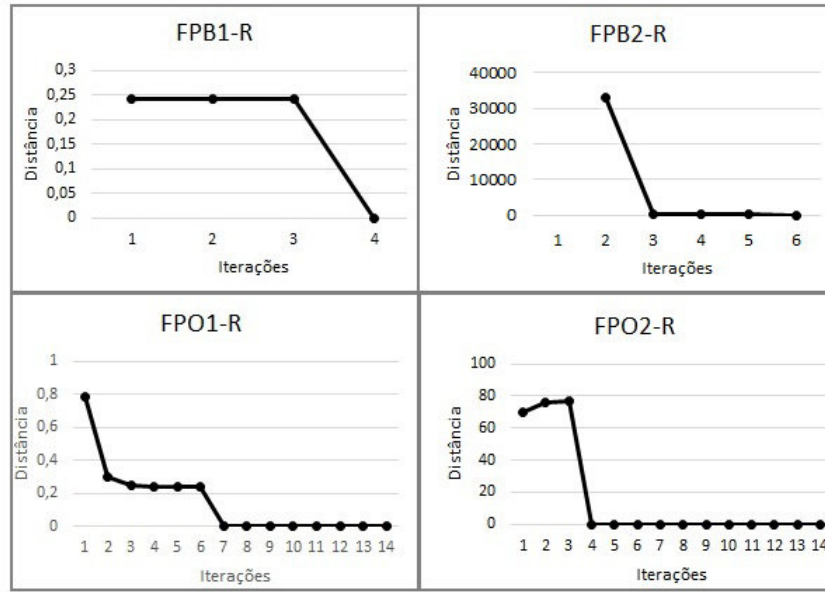


Figura 7.7: Evolução da distância $\Delta(x^*, \tilde{x})$ ao longo das iterações na instância R80-5 nas quatro estratégias heurísticas propostas.

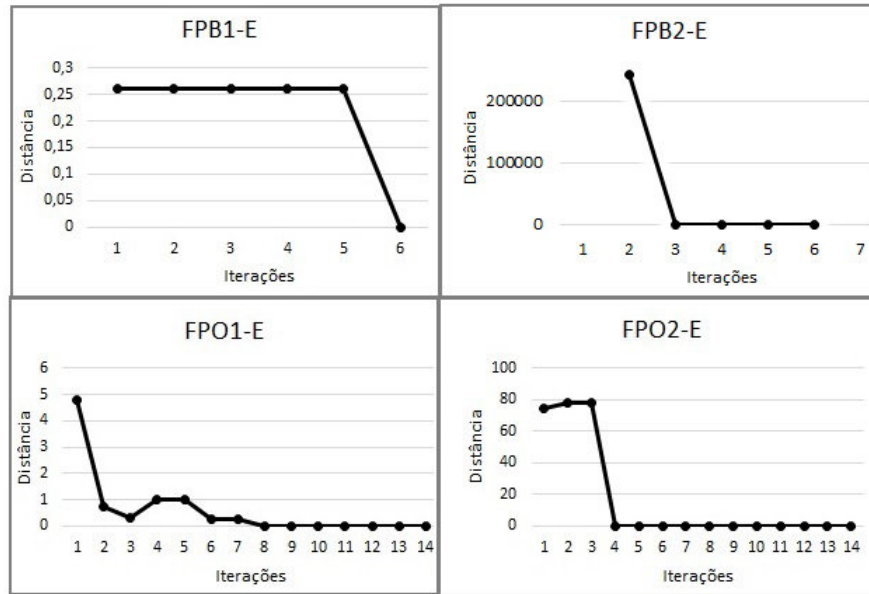


Figura 7.8: Evolução da distância $\Delta(x^*, \tilde{x})$ ao longo das iterações na instância E80-3 nas quatro estratégias heurísticas propostas.

Em qualquer um dos três grupos de instâncias, Quase Caminhos, Aleatórias e Euclídeas, as Heurísticas FPO1 e FPO2 necessitam de mais iterações para obter uma solução inteira admissível.

Na Tabela 7.1 podemos consultar os valores obtidos para o custo e peso para as instâncias QC80-7, R80-5 e E80-3 em cada uma das Heurísticas FP propostas e ainda o respetivo *gap*.

		QC80-7	R80-5	E80-3
FPB1	Custo	3769	34421	271898
	Peso	904	96953	70556
	<i>Gap</i>	0,48	3,33	3,59
FPB2	Custo	3751	33926	265002
	Peso	905	98144	70252
	<i>Gap</i>	0	1,84	0,96
FPO1	Custo	3759	34421	271898
	Peso	904	96953	70556
	<i>Gap</i>	0,21	3,33	3,59
FPO2	Custo	3751	33926	265002
	Peso	905	98144	70968
	<i>Gap</i>	0	1,84	0,96

Tabela 7.1: Comparação dos valores obtidos usando as quatro Heurísticas FP propostas.

Como podemos observar a negrito na Tabela 7.1, as Heurísticas FP onde se obtêm as soluções de melhor qualidade são as Heurísticas FPB2 e FPO2.

Na instância QC80-7, com a aplicação das Heurísticas FPB2 e FPO2 é obtida a solução ótima com custo 3751 e peso 905.

Quanto às instâncias R80-5 e E80-3, em nenhuma das Heurísticas FP se consegue obter a solução ótima. Na instância R80-5, a solução mais próxima da ótima é obtida quando se aplicam as Heurísticas FPB2 e FPO2 com custo 33926 e peso 98144 e *gap* 1,84%. Na instância E80-3, o *gap* mais baixo (0,96%) também é obtido através das Heurísticas FPB2 e FPO2 com custo 265002 e peso 70968.

7.5.3 Resultados Computacionais

Nesta secção apresentamos os resultados computacionais das quatro estratégias Heurísticas *Feasibility Pump* descritas na secção anterior.

Para podermos efetuar uma comparação da qualidade da solução obtida e do tempo de execução, apresentamos também os resultados obtidos pela Heurística do *software Xpress* aplicada ao nodo raiz utilizando a Formulação WMTZ (HXNR) e da primeira solução inteira admissível obtida pelo *solver* do *software Xpress* através da Formulação WMTZ, quando usa o procedimento *Branch and Bound* (PSIABB).

Os testes são realizados para as 215 instâncias geradas (ver Secção 2.3).

De modo a facilitar a análise, em cada grupo de instâncias, estas são separadas em 12 subgrupos, agrupadas por dimensão. Para estudar a qualidade das soluções obtidas são apresentados os *gaps* médios (em percentagem) e os respetivos desvios padrões e para estudar a rapidez são apresentados os tempos médios de execução (em segundos) e os respetivos desvios padrões.

Instâncias Quase Caminhos

Nas Heurísticas FPB2, FPO2 e HXNR é possível encontrar solução inteira admissível em todas as instâncias Quase Caminhos (95 instâncias). No caso da Heurística HXNR, a referida solução é sempre encontrada no nodo raiz.

Usando as Heurísticas FPB1 e FPO1 é possível encontrar solução inteira admissível em 93 das instâncias, o que corresponde a 97,89% dos casos. Estas heurísticas falham apenas em 2 instâncias (na Heurística FPB1 as instâncias são de 150 nodos e na Heurística FPO1 uma instância é de 60 e a outra de 300 nodos). As falhas referidas devem-se ao número máximo de iterações ter sido excedido.

Quanto à Heurística PSIABB, não foi possível obter solução inteira admissível em 5 instâncias (uma de 500 nodos e 4 de 1000 nodos), dado que excedeu o tempo limite imposto (10000 segundos).

No gráfico da Figura 7.9 encontram-se representadas as percentagens de *gaps* nulos obtidas no grupo de instâncias Quase Caminhos para as várias estratégias heurísticas propostas.

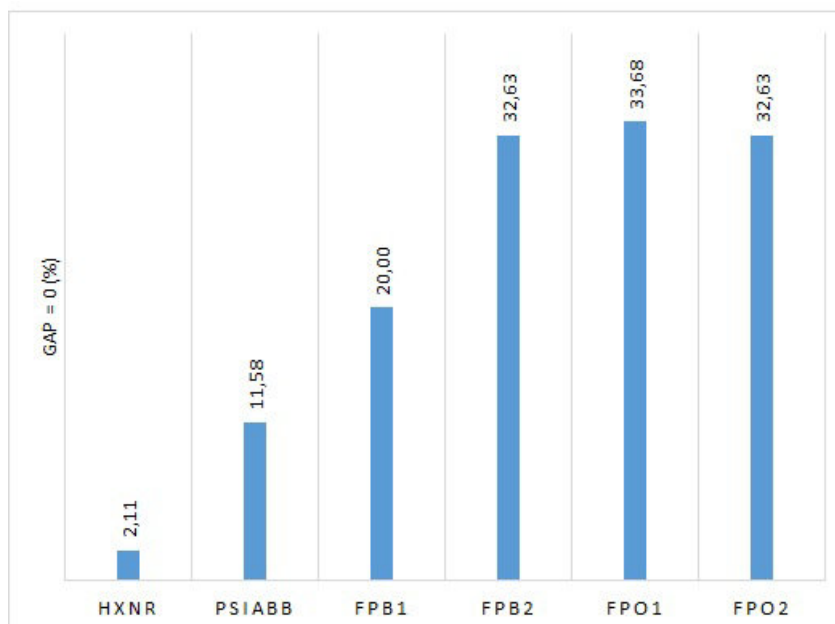


Figura 7.9: Percentagem de *gaps* nulos nas diferentes heurísticas propostas nas instâncias QC.

Das estratégias heurísticas apresentadas, a que tem maior percentagem de *gaps* nulos é a Heurística FPO1 com 33,68% dos casos (32 em 95 instâncias), depois temos as Heurísticas FPB2 e FPO2 com 32,63% dos casos (31 em 95 instâncias), a seguir com 20% dos casos (19 em 95 instâncias) temos a Heurística FPB1, depois temos a Heurística PSIABB onde é possível encontrar 11,58% de *gaps* nulos (11 em 95 instâncias) e por fim, através da Heurística HXNR temos 2,11% dos casos (2 em 95 instâncias). Portanto, qualquer uma das estratégias Heurísticas FP propostas apresenta maior percentagem de *gaps* nulos quando comparada com as Heurísticas HXNR e PSIABB.

Na Tabela 7.2 podemos observar os *gaps* médios, em percentagem, obtidos pelas Heurísticas HXNR, PSIABB, FPB1, FPB2, FPO1 e FPO2 e na Tabela 7.3 encontram-se os correspondentes desvios padrões.

N.º Nodos	HXNR	PSIABB	FPB1	FPB2	FPO1	FPO2
10	46,949	15,080	15,311	7,138	11,723	7,138
20	23,121	3,591	1,293	0,574	1,293	0,574
40	23,925	3,479	1,687	0,579	0,675	0,579
60	12,949	2,683	0,135	0,022	0,150	0,022
80	9,798	1,491	0,207	0,058	0,134	0,058
100	7,147	1,048	0,117	0,665	0,077	0,665
150	8,684	1,574	1,151	0,773	0,518	0,773
200	3,777	0,457	0,564	1,180	0,076	1,180
300	3,050	0,185	0,607	1,204	0	1,204
400	8,974	0,427	0,663	0,895	0,143	0,895
500	3,894	0,343	0,845	1,063	0,264	1,063
1000	1,414	0,004^a	0,459	1,414	0,030	1,414

Tabela 7.2: *Gaps* médios, em percentagem, obtidos pelas heurísticas propostas nas instâncias QC.

N.º Nodos	HXNR	PSIABB	FPB1	FPB2	FPO1	FPO2
10	41,805	30,180	11,157	8,075	7,344	8,075
20	13,448	4,228	1,539	0,610	1,539	0,610
40	5,751	3,036	2,150	0,614	0,696	0,614
60	2,721	3,228	0,247	0,029	0,257	0,029
80	1,740	1,344	0,220	0,051	0,123	0,051
100	0,602	1,209	0,252	1,398	0,159	1,398
150	2,767	1,200	1,361	1,271	0,953	1,271
200	0,833	0,386	0,570	1,425	0,151	1,425
300	0,923	0,341	0,350	0,884	0	0,884
400	10,702	0,686	0,699	0,727	0,199	0,727
500	1,049	0,642	0,612	1,325	0,354	1,325
1000	0,430	— ^a	0,208	0,430	0,068	0,430

Tabela 7.3: Desvios Padrões dos *gap* obtidos pelas heurísticas propostas nas instâncias QC.

^a *Gap* apenas da instância QC1000-5, pelo que, não foi possível calcular desvio padrão.

A fim de comparar a qualidade das soluções obtidas entre as Heurísticas FP propostas e a Heurística HXNR no grupo de instâncias Quase Caminhos, apresentamos o seguinte gráfico.

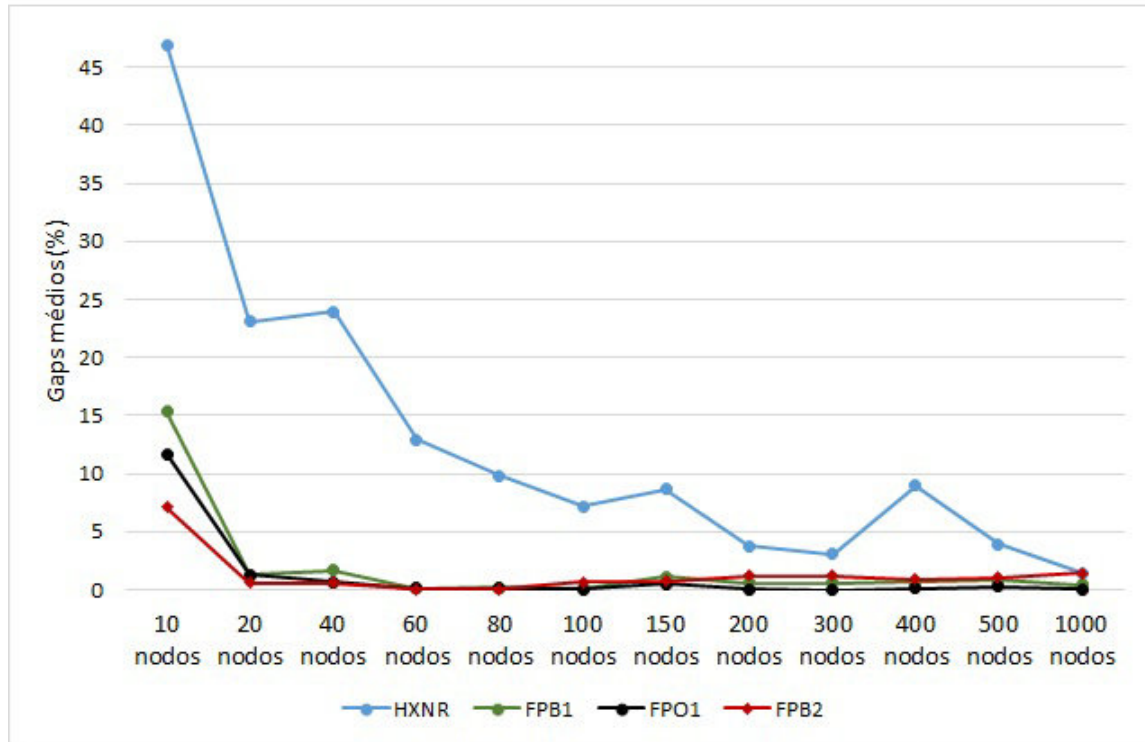


Figura 7.10: Comparação dos *gaps* médios, em percentagem, entre as Heurísticas FP propostas e a Heurística HXNR nas instâncias QC.

Como as Heurísticas FPB2 e FPO2 apresentam os mesmos *gaps* médios, no gráfico acima apenas se apresentam os *gaps* médios correspondentes à Heurística FPB2. Através do gráfico da Figura 7.10 e das Tabelas 7.2 e 7.3 podemos observar que qualquer das Heurísticas FP apresenta soluções de melhor qualidade quando comparada com a Heurística HXNR. É possível visualizar ainda, a formação de dois grupos: um formado pelas instâncias com menos de 100 nodos, onde as Heurísticas FPB2 e FPO1 encontram as soluções de melhor qualidade e o outro formado pelas instâncias de 100 ou mais nodos, onde a Heurística FPO1 apresenta soluções de melhor qualidade. Estes dois grupos podem ser denominados por grupo de instâncias pequenas e por grupo de instâncias grandes. Assim, à medida que aumenta a dimensão do problema, a Heurística FPO1 é a estratégia que apresenta soluções de melhor qualidade nas instâncias Quase Caminhos.

Na Tabela 7.4 podemos observar os tempos médios de execução (em segundos) das várias heurísticas propostas e na Tabela 7.5 encontram-se os respectivos desvios padrões.

N.º Nodos	HXNR	PSIABB	FPB1	FPB2	FPO1	FPO2
10	0,010	0,090	0,028	0,011	0,066	0,019
20	0,018	0,603	0,138	0,031	0,329	0,084
40	0,062	3,623	0,506	0,117	1,495	0,291
60	0,145	4,413	1,165	0,290	6,114	0,722
80	0,300	13,504	3,070	0,554	8,251	1,505
100	0,629	27,912	5,936	0,970	14,475	2,511
150	2,070	68,741	68,904	3,008	45,251	7,520
200	10,331	121,789	53,847	7,894	104,922	18,130
300	11,420	388,644	297,539	40,857	641,356	69,166
400	50,293	1340,455	483,319	90,462	840,883	173,079
500	51,274	4594,168	823,763	268,782	1656,248	371,326
1000	498,721	9434,064	1707,616	1014,109	6244,978	1712,898

Tabela 7.4: Tempos médios, em segundos, obtidos pelas heurísticas propostas nas instâncias QC.

N.º Nodos	HXNR	PSIBB	FPB1	FPB2	FPO1	FPO2
10	0,002	0,017	0,011	0,011	0,004	0,001
20	0,002	0,188	0,029	0	0,013	0,003
40	0,015	0,990	0,087	0,020	0,098	0,014
60	0,003	0,739	0,456	0,015	7,285	0,015
80	0,007	6,638	1,309	0,023	0,228	0,076
100	0,201	11,936	3,779	0,057	0,415	0,063
150	1,051	58,716	108,248	0,151	2,030	0,702
200	12,701	61,637	12,845	1,415	9,759	0,802
300	0,134	135,503	245,209	7,360	623,046	2,122
400	53,274	554,393	536,410	4,307	25,410	5,140
500	0,529	3561,055	401,203	75,996	45,706	11,036
1000	29,572	1952,226	533,252	88,882	782,108	369,810

Tabela 7.5: Desvios padrões dos tempos obtidos pelas heurísticas propostas nas instâncias QC.

Em termos de tempo de execução, das quatro Heurísticas FP apresentadas a Heurística FPB2 é a mais rápida, depois temos a Heurística FPO2, a seguir a Heurística FPB1 e a mais lenta é a que apresenta soluções de melhor qualidade para instâncias de

100 a 500 nodos. Nas instâncias de 400 e 500 nodos, o tempo de execução da Heurística FPO1 é superior a quatro vezes o tempo de execução das Heurísticas FPB2 e FPO2 e superior ao dobro do tempo de execução da Heurística FPB1.

Neste grupo de instâncias, a Heurística HXNR é a mais rápida, embora apresente soluções de pior qualidade.

Atendendo a que a Heurística FPB2 é a mais rápida das quatro estratégias Heurísticas FP propostas, vamos compará-la com a Heurística HXNR (Figura 7.11) e com a Heurística PSIABB (Figura 7.12).

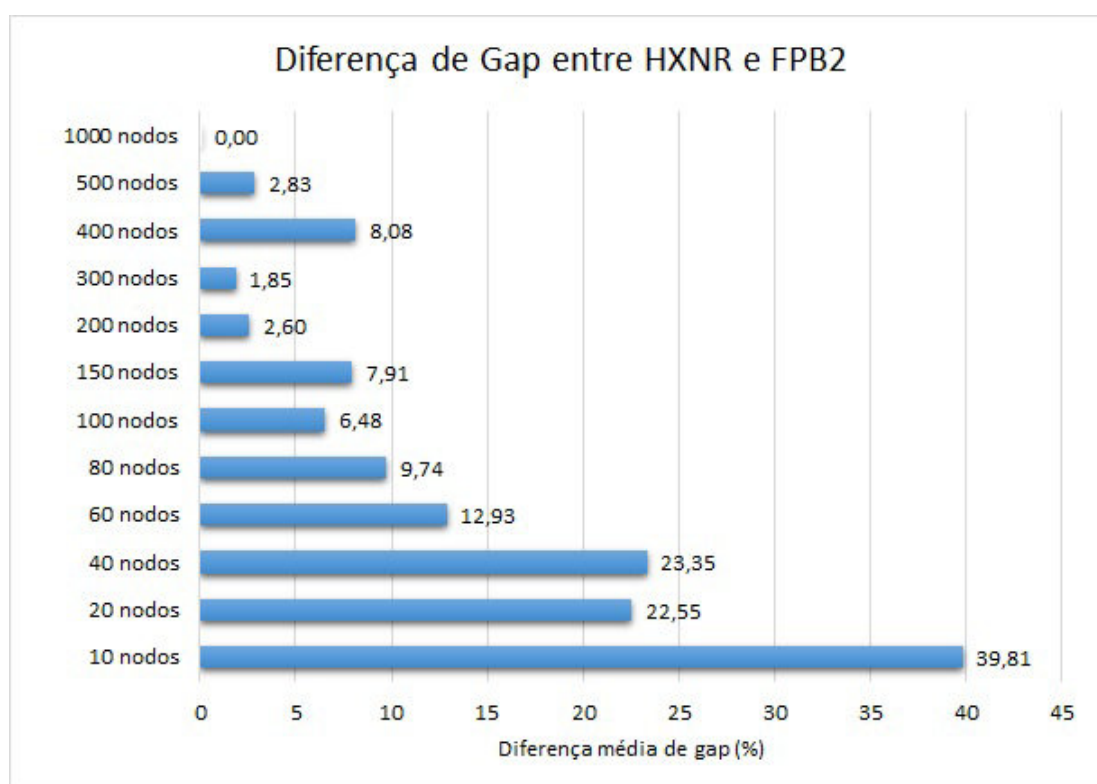


Figura 7.11: Diferenças médias de *gaps*, em percentagem, entre as Heurísticas HXNR e FPB2 nas instâncias QC.

Do gráfico da Figura 7.11 podemos observar que a diferença média de *gap* existente entre as Heurísticas HXNR e FPB2 é bastante significativa, o que leva a inferir que, apesar da Heurística FPB2 gastar mais tempo de execução, consegue obter soluções de melhor qualidade quando comparada com a Heurística HXNR. É de notar que apenas numa instância (QC10-8) a Heurística HXNR encontra uma solução de melhor

qualidade comparativamente à Heurística FPB2.

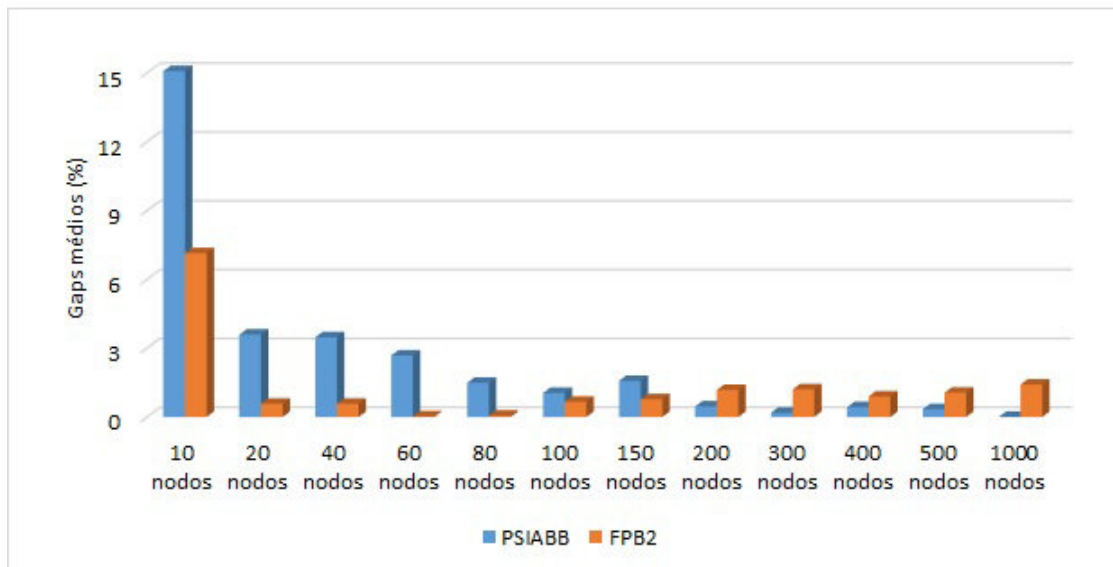


Figura 7.12: Comparação dos *gaps* médios, em percentagem, entre as Heurísticas FPB2 e a PSIABB nas instâncias QC.

Dado que os resultados das Tabelas 7.4 e 7.5 provam que a Heurística FPB2 é mais rápida do que a Heurística PSIABB e atendendo também a que, em termos de *gap*, se pode observar (Figura 7.12 e Tabelas 7.2 e 7.3) que, em instâncias de 200 ou mais nodos, a qualidade das soluções da Heurística FPB2 diminui, faz todo o sentido usar esta heurística num esquema enumerativo para melhorar a solução inteira admissível obtida. A Heurística FPB2 apresenta melhores soluções do que a Heurística PSIABB em 68,42% dos casos (65 em 95 instâncias) e também apresenta uma maior percentagem de *gaps* nulos (32,63% > 2,11%).

É de notar que em relação às instâncias de 1000 nodos, o *gap* médio da Heurística PSIABB é o mais baixo, mas esse valor só corresponde à solução da instância QC1000-5, pois para as restantes instâncias de 1000 nodos e algumas instâncias de 500 nodos não foi possível obter solução admissível no tempo imposto (10000 segundos).

Instâncias Aleatórias

Nas Heurísticas FPB2 e FPO2 é possível encontrar solução inteira admissível em

todas as instâncias Aleatórias (60 instâncias). No caso da Heurística HXNR, em 2 instâncias de 1000 nodos não é possível, no tempo estabelecido, encontrar solução inteira admissível e em 56 instâncias (93,33%) a referida solução inteira admissível é encontrada no nodo raiz. Apenas em 2 instâncias de 500 nodos é obtida a solução num nodo diferente do nodo raiz. Na Heurística FPO1 é possível encontrar solução inteira admissível em 52 das instâncias Aleatórias em teste, o que corresponde a 86,67% dos casos. Por exemplo, em todas as instâncias de 1000 nodos ocorre falha e esta deve-se ao tempo limite ter excedido. Na Heurística FPB1 ocorrem 7 falhas, sendo assim, nesta heurística é possível encontrar solução inteira admissível em 88,33% dos casos (53 em 60 instâncias).

Quanto à Heurística PSIABB não foi possível obter solução inteira admissível em 5 instâncias, sendo todas elas de 1000 nodos. O motivo da não obtenção de solução inteira admissível é devido ao limite de tempo imposto exceder os 10000 segundos.

No gráfico da Figura 7.13 encontram-se representadas as percentagens de *gaps* nulos obtidas no grupo de instâncias Aleatórias para as várias estratégias heurísticas propostas.

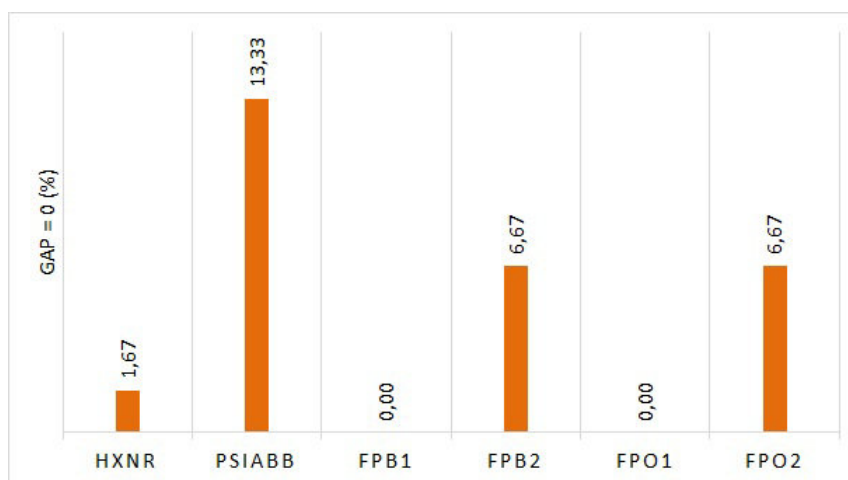


Figura 7.13: Percentagem de *gaps* nulos nas diferentes heurísticas propostas nas instâncias R.

Das quatro Heurísticas FP estudadas, a que apresenta a maior percentagem de *gaps* nulos é a Heurística PSIABB, com 13,33% (8 em 60 instâncias). Com a aplicação das Heurísticas FPB2 e FPO2 encontram-se 6,67% de *gaps* nulos (4 em 60 instâncias). Através das Heurísticas FPB1 e FPO1 não foi possível obter *gaps* nulos em nenhuma instância do grupo de instâncias Aleatórias. Com a Heurística HXNR apenas se encontram 1,67% de *gaps* nulos (1 em 60 instâncias).

Na Tabela 7.6 podemos observar os *gaps* médios, em percentagem, obtidos pelas Heurísticas HXNR, PSIABB, FPB1, FPB2, FPO1 e FPO2 e na Tabela 7.7 encontram-se os correspondentes desvios padrões.

N.º Nodos	HXNR	PSIABB	FPB1	FPB2	FPO1	FPO2
10	86,769	4,723	10,798	3,840	10,798	3,840
20	51,009	9,647	15,113	5,285	15,113	5,285
40	324,918	35,041	12,448	2,743	12,448	2,743
60	537,160	7,289	10,238	1,312	10,238	1,312
80	522,234	8,817	1,802	1,505	1,802	1,505
100	639,715	44,725	10,514	0,547	5,119	0,547
150	1486,381	16,399	8,337	0,787	2,055	0,787
200	290,743	19,828	2,307	0,533	0,731	0,533
300	1061,546	14,129	10,043	0,239	6,512	0,239
400	759,847	12,353	5,747	0,145	2,751	0,145
500	261,209	26,000	15,072	0,220	14,056	0,220
1000	7235,841	<i>-b</i>	<i>-b</i>	0,071	<i>-b</i>	0,071

Tabela 7.6: *Gaps* médios, em percentagem, obtidos pelas heurísticas propostas nas instâncias R.

N.º Nodos	HXNR	PSIABB	FPB1	FPB2	FPO1	FPO2
10	83,293	9,340	7,596	3,430	7,596	3,430
20	90,885	12,317	5,720	5,006	5,720	5,006
40	504,830	44,644	9,647	1,676	9,647	1,676
60	751,221	10,084	11,366	1,074	11,366	1,074
80	938,157	11,919	1,560	1,320	1,560	1,320
100	1043,844	34,460	16,675	0,264	5,262	0,264
150	2766,229	10,241	14,132	0,684	1,782	0,684
200	286,631	26,807	3,576	0,379	0,809	0,379
300	770,713	7,302	7,541	0,621	0,499	0,621
400	497,208	4,803	4,986	0,098	2,017	0,098
500	371,192	52,175	10,800	0,136	15,155	0,136
1000	11549,122	<i>-b</i>	<i>-b</i>	0,081	<i>-b</i>	0,081

Tabela 7.7: Desvios Padrões dos *gaps* obtidos pelas heurísticas propostas nas instâncias R.

^bNão foi possível calcular o *gap* médio nem o desvio padrão dado que ocorreu falha para todas as instâncias de 1000 nodos.

O gráfico da Figura 7.14 apresenta os *gaps* médios (em percentagem) obtidos pelas Heurísticas FP propostas.

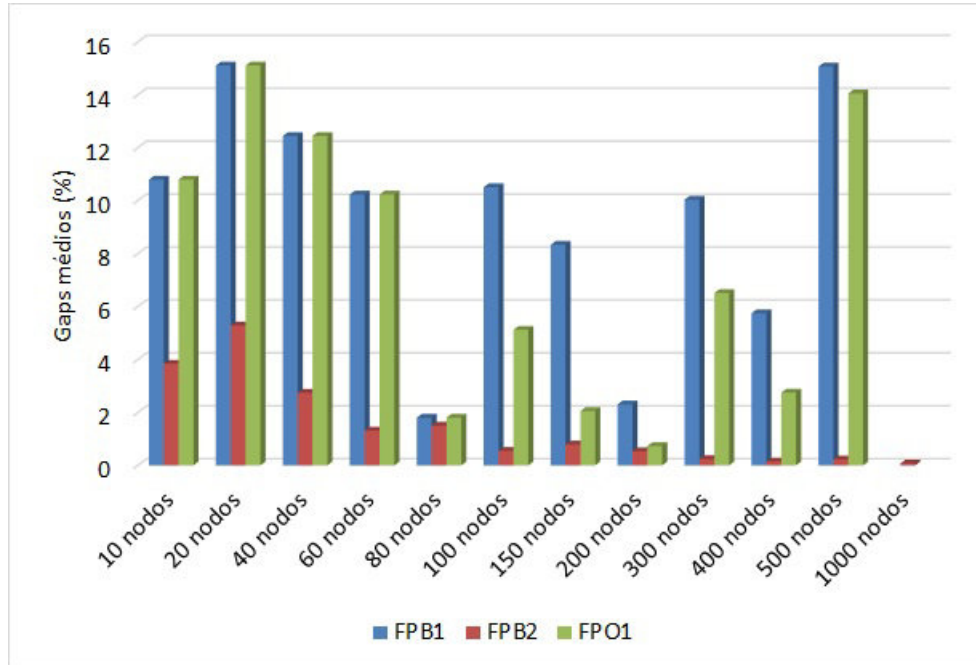


Figura 7.14: Comparação dos *gaps* médios, em percentagem, entre as Heurísticas FP propostas nas instâncias R.

Notamos que no gráfico acima não foram apresentados os *gaps* médios obtidos através da Heurística FPO2 dado que são iguais aos apresentados pela Heurística FPB2. Através do gráfico da Figura 7.14 e das Tabelas 7.6 e 7.7 podemos concluir que as soluções de melhor qualidade são obtidas utilizando as Heurísticas FPB2 e FPO2. Note-se ainda que para as instâncias de 1000 nós apenas foi apresentada no gráfico a barra correspondente à Heurística FPB2, pois devido à ocorrência de falha (exceder o tempo imposto) não foi possível obter um valor para o *gap* médio das Heurísticas FPB1 e FPO1.

Qualquer uma das Heurísticas FP propostas apresenta *gap* de melhor qualidade quando comparada com a Heurística HXNR, sendo que as soluções obtidas através da Heurística HXNR apresentam *gap* extremamente elevados.

Os tempos médios de execução (em segundos), das várias heurísticas propostas encontram-se registados na Tabela 7.8 e os respetivos desvios padrões na Tabela 7.9.

N.º Nodos	HXNR	PSIABB	FPB1	FPB2	FPO1	FPO2
10	0,025	0,265	0,118	0,009	0,228	0,025
20	0,075	0,484	0,184	0,031	0,390	0,059
40	0,177	3,054	0,515	0,115	1,872	0,249
60	0,556	3,888	1,370	0,274	4,493	0,624
80	1,389	8,867	1,735	0,543	9,246	1,186
100	2,868	19,438	6,253	1,211	19,741	2,649
150	10,592	59,744	41,051	4,979	132,372	10,034
200	18,483	127,362	87,119	12,521	194,213	23,432
300	62,829	310,566	809,338	39,647	1108,573	72,916
400	151,223	1181,846	633,165	92,527	1450,826	155,788
500	1313,775	1952,868	2965,244	311,440	3913,382	532,116
1000	5427,520	10042,940	10618,800	1724,168	11771,360	1463,854

Tabela 7.8: Tempos médios, em segundos, obtidos pelas heurísticas propostas nas instâncias R.

N.º Nodos	HXNR	PSIABB	FPB1	FPB2	FPO1	FPO2
10	0,021	0,091	0,187	0,009	0,214	0,008
20	0,089	0,094	0,169	0,001	0,076	0,007
40	0,181	0,818	0,178	0,008	0,191	0,011
60	0,233	1,328	0,982	0,014	0,167	0,025
80	1,042	3,352	0,954	0,045	0,340	0,047
100	1,152	11,567	2,284	0,352	2,305	0,904
150	4,695	30,560	20,621	1,175	119,473	2,670
200	5,497	92,635	51,493	2,555	55,060	6,311
300	19,978	166,714	1012,314	15,762	939,541	32,523
400	51,623	459,940	192,898	45,278	454,522	98,084
500	1482,344	1810,176	1083,116	31,912	1581,267	30,537
1000	4214,302	20,804	525,087	264,514	1056,622	159,162

Tabela 7.9: Desvios padrões dos tempos obtidos pelas heurísticas propostas nas instâncias R.

Em termos de tempo médio de execução podemos afirmar que das quatro Heurísticas FP apresentadas, a Heurística FPB2 é a mais rápida, depois temos a Heurística FPO2, a seguir a Heurística FPB1 e a mais lenta é a Heurística FPO1.

Ao compararmos os tempos de execução da Heurística FPB2 com os tempos de execução da Heurística HXNR (Tabelas 7.8 e 7.9) observamos que a Heurística FPB2 apresenta tempos de execução inferiores.

Nas instâncias Aleatórias a Heurística FP que é mais rápida e obtém soluções de melhor qualidade é, sem dúvida, a Heurística FPB2.

Ao compararmos a Heurística FPB2 com a Heurística PSIABB, verificamos que esta apresenta soluções de melhor qualidade em menor tempo, o que leva a concluir que a Heurística FPB2 pode ser usada num esquema enumerativo de modo a melhorar a solução obtida.

Instâncias Euclidianas

Na Heurística HXNR é possível encontrar solução inteira admissível no nodo raiz em 60% dos casos (36 em 60 instâncias), nos restantes casos, em 3 instâncias de 1000 nodos não é possível, no tempo estabelecido, encontrar solução inteira admissível e em 21 instâncias obtém-se uma solução inteira admissível num outro nodo diferente do nodo raiz.

Nas Heurísticas FPB2 e FPO2 é possível encontrar solução inteira admissível em todas as 60 instâncias Euclidianas, mas nas Heurísticas, FPB1 e FPO1, isso não é possível em algumas instâncias, sendo de realçar que em nenhuma delas se encontram soluções inteiras admissíveis para as instâncias de 500 e 1000 nodos, devido a exceder o tempo limite imposto. A Heurística FPB1 apresenta 80% de soluções inteiras admissíveis (48 em 60 instâncias) e a Heurística FPO1 75% (45 em 60 instâncias).

Quanto à Heurística PSIABB foi possível obter solução inteira admissível em 55 das 60 instâncias, isto é, em 91,67% dos casos. Ocorrem falhas em todas as instâncias de 1000 nodos devido ao limite de tempo imposto exceder os 10000 segundos.

No gráfico da Figura 7.15 encontram-se representadas as percentagens de *gaps* nulos obtidas no grupo de instâncias Euclidianas para as várias estratégias heurísticas propostas.

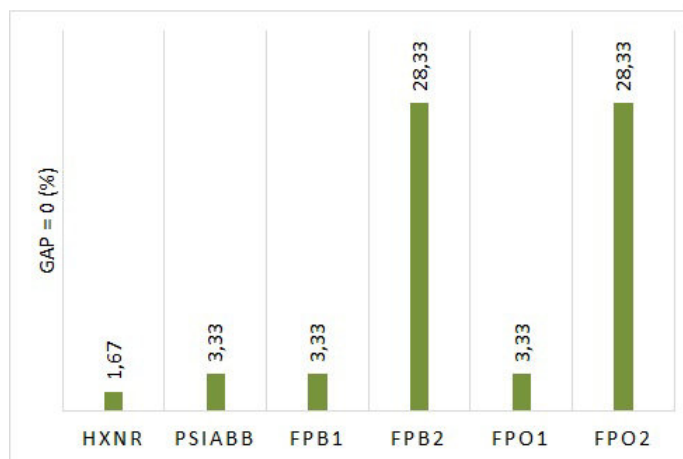


Figura 7.15: Percentagem de *gaps* nulos nas diferentes heurísticas propostas nas instâncias E.

Através do gráfico da Figura 7.15 podemos observar que das várias heurísticas propostas, as que apresentam a maior percentagem de *gaps* nulos são as Heurísticas FPB2 e FPO2, com 28,33% dos casos (17 em 60 instâncias), depois temos as Heurísticas PSIABB, FPB1 e FPO1 com 3,33% dos casos (2 em 60 instâncias) e por fim, temos a Heurística HXNR, onde apenas se encontra 1,67% de *gaps* nulos (1 em 60 instâncias).

Nas Tabelas 7.10 e 7.11 encontram-se os *gaps* médios, em percentagem, e os respectivos desvios padrões das Heurísticas HXNR, PSIABB, FPB1, FPB2, FPO1 e FPO2.

N.º Nodos	HXNR	PSIABB	FPB1	FPB2	FPO1	FPO2
10	8,403	7,812	5,805	5,274	5,805	5,274
20	10,604	17,306	4,095	2,250	4,095	2,250
40	35,935	24,242	3,629	1,959	3,629	1,959
60	34,519	38,492	2,612	0,747	2,612	0,747
80	24,208	24,109	1,382	0,576	1,382	0,576
100	61,786	23,687	0,964	0,563	0,964	0,563
150	233,198	30,507	25,211	0,284	2,897	0,284
200	305,504	34,327	41,024	0,402	5,910	0,402
300	704,702	19,144	62,551	0,219	39,983	0,219
400	525,941	16,987	98,556	0	133,266	0
500	573,380	25,729	— ^c	0	— ^c	0
1000	1960,172	— ^c	— ^c	0	— ^c	0

Tabela 7.10: *Gaps* médios, em percentagem, obtidos pelas heurísticas propostas nas instâncias E.

N.º Nodos	HXNR	PSIABB	FPB1	FPB2	FPO1	FPO2
10	8,221	9,465	3,329	3,493	3,329	3,493
20	14,241	19,424	4,028	2,685	4,028	2,685
40	45,006	14,753	2,539	0,390	2,539	0,390
60	15,882	13,610	2,256	0,708	2,256	0,708
80	12,182	12,210	1,394	0,428	1,394	0,428
100	86,493	11,118	1,062	0,662	1,062	0,662
150	279,136	16,628	17,206	0,194	2,266	0,194
200	315,907	27,172	18,747	0,195	1,867	0,195
300	483,672	4,060	22,101	0,163	25,940	0,163
400	466,805	5,088	11,418	0	12,070	0
500	745,813	16,562	— ^c	0	— ^c	0
1000	14,540	— ^c	— ^c	0	— ^c	0

Tabela 7.11: Desvios Padrões dos *gaps* obtidos pelas heurísticas propostas nas instâncias E.

^cNão foi possível calcular o *gap* médio nem o desvio padrão dado que ocorreu falha para todas as instâncias.

O gráfico da Figura 7.16 apresenta os *gaps* médios (em percentagem) obtidos pelas Heurísticas FP propostas.

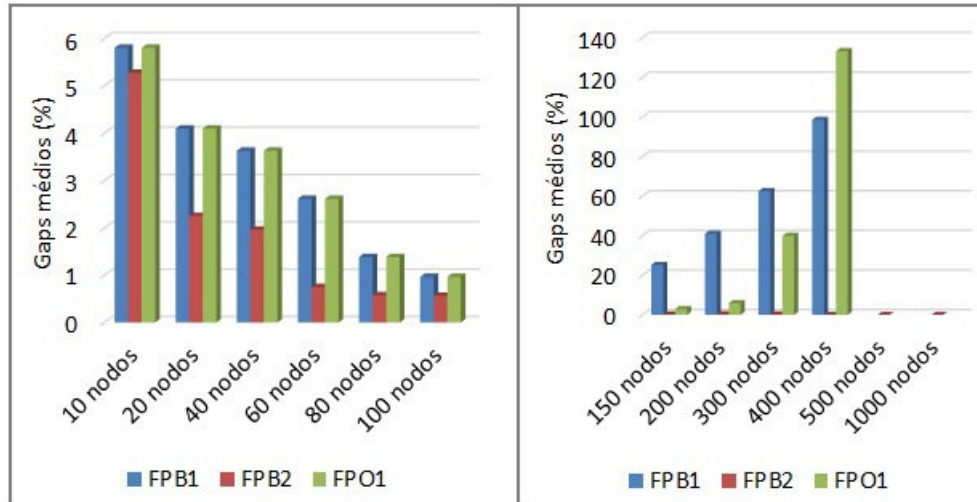


Figura 7.16: Comparação dos *gaps* médios, em percentagem, entre as Heurísticas FP propostas nas instâncias E.

Através do gráfico da Figura 7.16 e das Tabelas 7.10 e 7.11 podemos verificar que as soluções de melhor qualidade são obtidas utilizando as Heurísticas FPB2 e FPO2. Observamos também que como os *gaps* médios obtidos através das Heurísticas FPB2 e FPO2 são iguais, apenas apresentamos no gráfico acima uma destas heurísticas.

Qualquer uma das Heurísticas FP propostas apresenta *gaps* de melhor qualidade quando comparada com a Heurística HXNR. As soluções obtidas através da Heurística HXNR apresentam *gaps* extremamente elevados.

Os tempos médios de execução (em segundos) das várias heurísticas propostas encontram-se registados na Tabela 7.12 e os desvios padrões na Tabela 7.13.

N.º Nodos	HXNR	PSIABB	FPB1	FPB2	FPO1	FPO2
10	0,081	0,889	0,037	0,013	0,078	0,016
20	1,086	3,014	0,119	0,031	0,452	0,072
40	5,213	9,083	0,880	0,115	2,746	0,281
60	10,611	12,982	3,320	0,287	7,604	0,664
80	152,247	156,859	12,439	0,540	21,497	1,283
100	74,225	104,205	42,819	1,102	59,754	2,695
150	11,260	110,573	240,824	2,299	246,808	5,370
200	77,320	252,359	444,323	4,945	507,809	11,707
300	151,588	1064,143	1480,574	18,866	1991,436	33,212
400	384,974	3403,522	6608,878	39,502	6556,432	77,788
500	1237,238	6089,540	10701,340	98,892	10422,740	150,658
1000	6556,780	10049,880	11551,760	1171,194	11315,420	1249,820

Tabela 7.12: Tempos médios, em segundos, obtidos pelas heurísticas propostas nas instâncias E.

N.º Nodos	HXNR	PSIABB	FPB1	FPB2	FPO1	FPO2
10	0,069	0,213	0,018	0,007	0,011	0
20	0,844	0,898	0,067	0,011	0,065	0,008
40	3,805	5,176	0,501	0,008	0,572	0,011
60	8,182	6,948	1,347	0,009	1,503	0,028
80	169,931	167,887	4,064	0,050	3,427	0,049
100	128,850	123,519	14,519	0,265	14,035	0,735
150	3,916	33,553	129,518	0,109	130,900	0,474
200	118,815	87,049	71,833	0,406	149,765	0,646
300	239,359	420,261	334,313	3,469	884,028	1,793
400	512,596	1663,865	2307,711	1,545	3520,799	8,077
500	1839,296	2653,828	293,686	20,967	326,293	7,509
1000	4868,197	48,077	674,723	136,915	991,950	145,128

Tabela 7.13: Desvios padrões dos tempos obtidos pelas heurísticas propostas nas instâncias E.

Em termos de tempo médio de execução podemos afirmar que das quatro Heurísticas FP apresentadas, a FPB2 é a mais rápida, depois temos a Heurística FPO2 e as Heurísticas FPB1 e FPO1 são as mais lentas.

Ao compararmos os tempos de execução das Heurística FPB2 e HXNR (Tabelas 7.12 e 7.13) observamos que estes são inferiores na Heurística FPB2. É de notar que o tempo de execução da Heurística HXNR é superior ao dobro do tempo de execução da Heurística FPB2.

No grupo de instâncias Euclidianas as Heurísticas FP que apresentam soluções de melhor qualidade são sem dúvida as Heurísticas FPB2 e FPO2, sendo a Heurística FPB2 a que apresenta menores tempos de execução.

Comparando a Heurística FPB2 com a Heurística PSIABB, esta apresenta soluções de melhor qualidade em menor tempo (a Heurística PSIABB apresenta tempos superiores a mais de 8 vezes o tempo de execução da Heurística FPB2), o que leva a concluir que a Heurística FPB2 pode ser usada num esquema enumerativo de modo a melhorar a solução obtida.

7.5.4 Síntese dos Resultados Computacionais

Em todos os grupos de instâncias verificámos que a Heurística FPB2 encontra soluções admissíveis de boa qualidade quando comparada com as restantes Heurísticas *Feasibility Pump* propostas. Esta também apresenta uma boa qualidade quando comparada com a Heurística HXNR e com a Heurística PSIABB. Em termos de tempo computacional, nas instâncias Aleatórias e Euclidianas a Heurística FPB2 é relativamente rápida quando comparada com qualquer heurística testada. No caso das instâncias Quase Caminhos, a mais rápida é a Heurística HXNR, mas em termos de qualidade das soluções encontradas, esta não apresenta bons resultados quando comparada com qualquer uma das Heurísticas FP.

Podemos também observar que dado que a Heurística FPB2 é mais rápida do que a Heurística PSIABB em todos os grupos de instâncias, então pode ser usada num esquema enumerativo de modo a melhorar a solução inteira admissível obtida.

Capítulo 8

Método *Local Branching*

No caso do Problema WMST, o Método *Local Branching* é um esquema enumerativo que constrói uma sequência de soluções inteiras admissíveis com valor decrescente de custos. O custo da última solução inteira admissível da sequência corresponde ao valor ótimo ou a um limite superior para o valor ótimo.

O método começa com uma solução inicial inteira admissível, depois é definida uma vizinhança dessa solução através da introdução no modelo de uma inequação linear chamada de restrição *local branching*. A exploração da vizinhança é feita através da utilização do *solver* de um *software* de Otimização, como por exemplo, o *Mosel* do *software Xpress*. Caso a nova solução obtida tenha menor custo, então é atualizada a solução e o procedimento de definição e exploração é executado novamente até que não existam melhoramentos no valor do custo das soluções obtidas.

O Método *Local Branching* foi proposto por Fischetti e Lodi [19] para resolver Problemas de Programação Linear Inteira Mista (MIP - *Mixed-Integer Linear Program*) e é considerado um método de melhoramento bastante eficiente na resolução de problemas de grande dimensão.

Segundo Hassen et al. [34], o *Local Branching* é uma técnica MIP que foi planeada para ser um método exato, mas que atua como um método heurístico se um limite de tempo for definido e atingido antes que a solução ótima do problema seja encontrada.

Assim, o *Local Branching* pode ser considerado também um método heurístico que permite encontrar soluções inteiras admissíveis de boa qualidade e de forma rápida, mas com a desvantagem de não garantir a obtenção da solução ótima.

Neste capítulo mostramos a aplicação do Método *Local Branching* ao Problema WMST. Na Secção 8.1 começamos por introduzir alguns conceitos importantes para aplicação do Método *Local Branching* e apresentamos um Algoritmo *Local Branching*

Clássico para o Problema WMST. Na Secção 8.2 apresentamos algumas extensões do método com o objetivo de melhorar o seu desempenho quando aplicado ao problema em estudo.

Na Secção 8.3 referimos as diferenças existentes entre o Método *Feasibility Pump* e o Método *Local Branching* e fazemos uma abordagem geométrica do comportamento do Método *Local Branching*. Na Secção 8.4 descrevemos vários algoritmos derivados do Algoritmo *Local Branching Clássico* descrito na Secção 8.1. Por fim, na Secção 8.5, apresentamos e discutimos os resultados computacionais obtidos através da aplicação dos vários algoritmos de *Local Branching* propostos e efetuamos um estudo comparativo entre o algoritmo *Local Branching* que apresentou os melhores resultados, para um grupo de instâncias teste e o Algoritmo *Branch and Bound* do Procedimento P-WMTZ+C. Para finalizar o capítulo apresentamos uma síntese dos resultados computacionais obtidos.

8.1 *Local Branching* Aplicado ao Problema WMST

Para aplicar o Método *Local Branching* ao Problema WMST, começamos por obter uma primeira solução inteira admissível que se denomina de solução de referência e que designamos por $\bar{x} \in X_{WMST}$, a qual pode ser obtida usando o *solver* de um *software* de Otimização ou utilizando um método heurístico. À solução inteira admissível $\bar{x} \in X_{WMST}$ corresponde uma árvore de suporte, com custo $C(T_{\bar{x}})$ e peso $W(T_{\bar{x}})$.

Tendo por base a solução de referência \bar{x} e o facto das variáveis serem binárias, definem-se dois conjuntos de variáveis $\bar{S} = \{(i, j) \in A : \bar{x}_{ij} = 1\}$, designado de suporte binário de \bar{x} , o qual é constituído pelas variáveis que tomam o valor um na solução de referência e o seu complementar $\bar{S}^C = \{(i, j) \in A : \bar{x}_{ij} = 0\}$, que é constituído pelas variáveis que tomam o valor zero na solução de referência.

O parâmetro k é um valor inteiro e positivo que indica o número máximo de variáveis binárias que podem alterar o valor. Dado um valor para o parâmetro k define-se a vizinhança k -OPT de \bar{x} , denotada por $\mathcal{N}(\bar{x}, k)$, da seguinte maneira,

$$\mathcal{N}(\bar{x}, k) = \left\{ x \in X_{WMST} : \Delta(x, \bar{x}) = \sum_{(i,j) \in \bar{S}} (1 - x_{ij}) + \sum_{(i,j) \in \bar{S}^C} x_{ij} \leq k \right\},$$

isto é, $\mathcal{N}(\bar{x}, k)$ é o conjunto de soluções inteiras admissíveis ($\bar{x} \in X_{WMST}$) que satisfazem a restrição adicional de *local branching*

$$\Delta(x, \bar{x}) := \sum_{(i,j) \in \bar{S}} (1 - x_{ij}) + \sum_{(i,j) \in \bar{S}^C} x_{ij} \leq k. \quad (8.1)$$

Nesta restrição linear, os dois termos do lado esquerdo contam o número de variáveis binárias que trocam de valores (relativamente a \bar{x}), do valor um para o valor zero e do valor zero para o valor um, respetivamente, esse número total de trocas define uma vizinhança de tamanho k .

No Problema WMST, a cardinalidade do suporte binário \bar{S} de qualquer solução inteira admissível é sempre constante e igual ao número de variáveis unitárias da solução \bar{x} , isto é, igual ao número de arestas da correspondente árvore de suporte $T_{\bar{x}}$.

Sempre que se efetua uma troca podemos, por exemplo, atribuir o valor zero a uma variável que antes tinha valor um, mas para que o $|\bar{S}|$ seja constante é necessário também atribuir o valor um a uma variável que antes tinha valor zero, o que resulta numa contabilização dupla.

Para simplificar a restrição (8.1), podemos pensar em contabilizar apenas metade das trocas efetuadas, isto é, contabilizar apenas as variáveis com valor um na solução de referência. Deste modo a restrição (8.1) pode ser escrita da seguinte forma,

$$\Delta(x, \bar{x}) := \sum_{(i,j) \in \bar{S}} (1 - x_{ij}) \leq k' \left(= \frac{k}{2} \right). \quad (8.2)$$

Um outro problema conhecido da literatura onde a cardinalidade do suporte binário é constante é o Problema do Caixeiro Viajante (TSP - *Traveling Salesman Problem*), onde a restrição (8.2) permite que no máximo k' arestas sejam trocadas de rota de referência \bar{x} .

Como o próprio nome sugere, a restrição *local branching* pode ser usada como um critério de ramificação num esquema de enumeração do tipo *Branch and Bound*.

Dada uma solução de referência \bar{x} , o espaço de soluções associado ao nó corrente pode ser particionado em duas ramificações:

$$(i) \Delta(x, \bar{x}) \leq k \text{ (ramo esquerdo)} \quad \text{ou} \quad (ii) \Delta(x, \bar{x}) \geq k + 1 \text{ (ramo direito)}.$$

A escolha do tamanho das vizinhanças, dado pelo parâmetro k , torna-se difícil, pois pode depender do tamanho e da estrutura das instâncias utilizadas. O valor de

k deverá ser suficientemente grande para que a vizinhança $\mathcal{N}(\bar{x}, k)$ contenha soluções melhores que \bar{x} mas, por outro lado, deverá ser pequeno o suficiente de forma a garantir que a vizinhança $\mathcal{N}(\bar{x}, k)$ seja explorada rapidamente. Em [19] são utilizados valores de k no intervalo $[10, 20]$, os quais têm mostrado ser eficientes para Problemas MIP.

No esquema que se segue apresentamos de forma resumida as principais etapas da implementação do Método *Local Branching* Clássico aplicado ao Problema WMST.

Algoritmo *Local Branching* Clássico (ALBC) Aplicado ao Problema WMST

Passo 1: *Obter uma solução inteira admissível inicial*

Fazer $t := 1$. Seja $\bar{x} = \bar{x}^t$ a solução de referência obtida através de um *solver*.

Passo 2: *Introduzir a restrição local branching*

$t := t + 1$.

Definir $\bar{S} = \{(i, j) \in A : \bar{x}_{ij} = 1\}$ e um valor de k inteiro e positivo.

Introduzir a restrição *local branching* $\Delta(x, \bar{x}) \leq k$.

Obter a solução ótima \bar{x}^t da vizinhança $\mathcal{N}(\bar{x}, k)$ utilizando um *solver*.

Passo 3: *Critério de paragem e obtenção de novas soluções inteiras admissíveis*

Se *status* de $\mathcal{N}(\bar{x}, k) = \text{“Optimum found”}$, então

Se $C(T_{\bar{x}^t}) < C(T_{\bar{x}})$, então

atualizar $\bar{x} = \bar{x}^t$ e voltar a efetuar o **Passo 2**.

Caso contrário,

introduzir $\Delta(x, \bar{x}) \geq k + 1$ e utilizar um *solver* para obter \bar{x}^t .

Se $C(T_{\bar{x}^t}) < C(T_{\bar{x}})$, então

atualizar $\bar{x} = \bar{x}^t$.

STOP.

No Passo 1, o *solver* de um *software* de Otimização é chamado e a execução do Algoritmo *Branch and Bound* de uma formulação para o Problema WMST é interrompida assim que seja encontrada uma primeira solução inteira admissível $\bar{x}^t \in X_{WMST}$, a qual é tomada como solução de referência $\bar{x} = \bar{x}^t$.

No Passo 2, é inserida uma nova restrição, $\Delta(x, \bar{x}^t) \leq k$, para reduzir o espaço de soluções X_{WMST} e assim formar o novo espaço de soluções $X_t = X_{WMST} \cap \mathcal{N}(\bar{x}^t, k)$. O *solver* é então chamado para encontrar a solução ótima da vizinhança $\mathcal{N}(\bar{x}^t, k)$. No Passo 3, se a nova solução encontrada \bar{x}^{t+1} for melhor do que a solução \bar{x}^t , isto é, se $C(T_{\bar{x}^{t+1}}) < C(T_{\bar{x}^t})$, então volta-se ao Passo 2, onde é inserida a nova restrição $\Delta(x, \bar{x}^{t+1}) \leq k$ tendo como base a nova solução de referência \bar{x}^{t+1} . O novo espaço de soluções para este subproblema passa a ser $X_{t+1} = (X_{WMST} \setminus X_t) \cap \mathcal{N}(\bar{x}^{t+1}, k)$ e o *solver* é chamado novamente para encontrar a solução ótima da vizinhança $\mathcal{N}(\bar{x}^{t+1}, k)$.

No caso de continuar a existir melhoria no custo das soluções obtidas, o procedimento de ramificação continua a ser executado. Após ℓ iterações foram inseridas ℓ restrições *local branching*, sendo o novo espaço de soluções para o subproblema em questão $X_\ell = (X_{WMST} \setminus X_1 \setminus X_2 \setminus \dots \setminus X_{\ell-1}) \cap \mathcal{N}(\bar{x}^\ell, k)$.

Se não for encontrada uma solução de custo inferior ao custo da solução de referência, então não é possível aplicar mais o método, sendo o *solver* chamado para explorar o restante espaço de pesquisa correspondente à ramificação direita $\Delta(x, \bar{x}) \geq k + 1$. Se o custo da solução encontrada nesta exploração local for inferior ao custo da solução de referência, então é atualizada a solução de referência e conclui-se a enumeração.

Exemplo 8.1.

Como exemplo de aplicação do Algoritmo ALBC ao Problema WMST apresentamos um esquema em árvore na Figura 8.1, onde é usado o Algoritmo ALBC na instância 10-8 do grupo de instâncias Quase Caminhos com $W = 1000$. Para a restrição *local branching* considerou-se a desigualdade (8.2) com $k' = 5$ e utilizou-se a Formulação WMTZ.

No nodo 1 é obtida através do *solver* do *software Xpress*, a solução de referência $\bar{x} = \bar{x}^1$, com custo 659 e peso 991. Após a introdução da restrição *local branching*, $\Delta(x, \bar{x}^1) \leq 5$, obteve-se no nodo 2 a solução \bar{x}^2 . Como $C(T_{\bar{x}^2}) = 605 < 659 = C(T_{\bar{x}^1})$, então a solução de referência \bar{x} é atualizada para \bar{x}^2 . No nodo 3 é adicionada a ramificação esquerda correspondente à introdução da restrição $\Delta(x, \bar{x}^2) \leq 5$ e obtida

no nodo 4 a mesma solução obtida no nodo 2. Como esta solução não tem custo inferior ao da solução de referência, então voltamos ao nodo 3 e inserimos a ramificação direita correspondente à restrição $\Delta(x, \bar{x}^2) \geq 6$ para explorar o restante espaço de pesquisa e assim concluir a enumeração no nodo 5 com uma solução que é atualizada, pois $C(T_{\bar{x}^4}) = 577 < 605$.

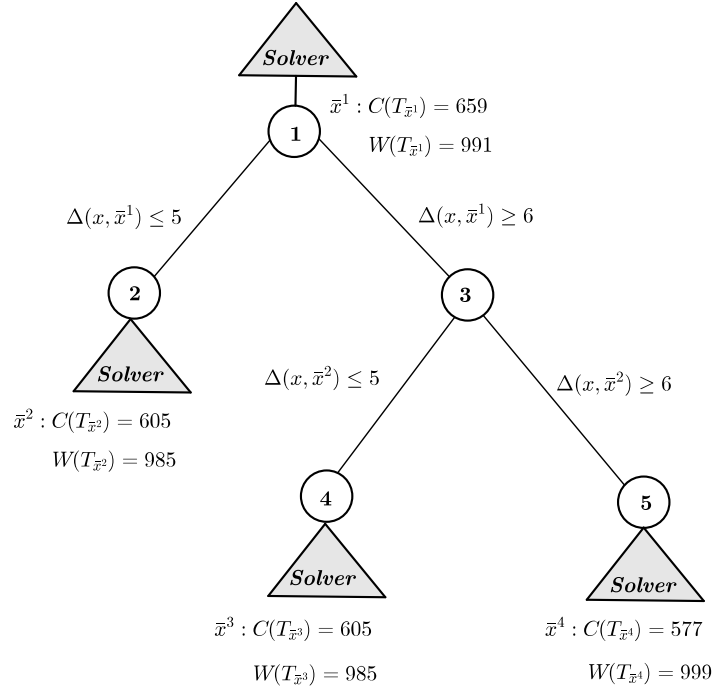


Figura 8.1: Esquema de ramificação obtido usando o Algoritmo ALBC na instância QC10-8.

Assim, $\bar{x} = \bar{x}^4$ é a solução obtida e corresponde a uma árvore de suporte $T_{\bar{x}}$ que contém as arestas $\{0, 1\}$; $\{0, 7\}$; $\{1, 2\}$; $\{2, 3\}$; $\{2, 5\}$; $\{4, 6\}$; $\{6, 7\}$; $\{7, 8\}$ e $\{8, 9\}$. A árvore $T_{\bar{x}}$ tem custo $C(T_{\bar{x}}) = 577$ e peso $W(T_{\bar{x}}) = 999$ e corresponde à solução ótima desta instância.

8.2 Extensões do Método *Local Branching*

Com os métodos exatos há a garantia de encontrar a solução ótima global. No entanto, à medida que a dimensão dos problemas aumenta, os tempos de execução também aumentam, revelando-se necessário o uso de métodos heurísticos.

Uma vantagem do uso de métodos heurísticos comparativamente ao uso de métodos exatos, reside no facto de ser possível obter soluções admissíveis muito rapidamente, mas por outro lado apesar de se obterem soluções de forma rápida existe a desvantagem de não ser garantida a obtenção da solução ótima.

Ao usarmos o Método *Local Branching* como um método exato, existe uma maior integração com o *solver* de um *software* de Otimização, mas temos a desvantagem de desperdiçar muito tempo computacional, particularmente, quando nenhuma solução melhor do que a solução de referência é encontrada. Se for imposto um limite de tempo no Algoritmo *Local Branching* Clássico aplicado ao Problema WMST este passa a comportar-se como um método heurístico.

Apresentamos, de seguida, algumas formas de melhorar o desempenho da Heurística *Local Branching*.

8.2.1 Solução de Referência Inicial

A escolha da solução de referência inicial é muito importante. É usual considerar uma solução inteira admissível inicial fornecida pelo *solver*. Contudo, em algumas instâncias, obter uma solução inteira admissível inicial pode demorar muito tempo, pelo que podemos usar como solução de referência:

1. A solução correspondente à árvore de suporte de peso mínimo T_w ;
2. A solução obtida através de um esquema Heurístico *Feasibility Pump*;
3. A solução correspondente a uma árvore de suporte que verifique a restrição de peso.

Enquanto nas situações 1 e 3, as soluções obtidas são inteiras admissíveis, na situação 2, por vezes, não é possível encontrar uma solução inteira admissível. Este facto é justificado pela falha da Heurística *Feasibility Pump*. Na eventualidade de tal falha ocorrer, podemos utilizar o *solver* para obter uma solução inteira admissível inicial.

8.2.2 Mecanismos de Intensificação e de Diversificação

Estes dois mecanismos de intensificação e de diversificação utilizam-se quando a solução não melhora. No que segue, convém referir, que $\lfloor p \rfloor$ converte o número p no maior número inteiro menor ou igual a p e $\lceil p \rceil$ converte o número p no menor número inteiro maior ou igual a p .

- **Mecanismo de Intensificação**

Este mecanismo tem por objetivo reduzir o tamanho da vizinhança a ser explorada e, conseqüentemente, o tempo de pesquisa.

O lado direito da restrição $\Delta(x, \bar{x}) \leq k$ é reduzido para $\lfloor \frac{k}{2} \rfloor$, ou seja, introduz-se a restrição $\Delta(x, \bar{x}) \leq \lfloor \frac{k}{2} \rfloor$ e utiliza-se novamente o *solver* para encontrar uma nova solução.

- **Mecanismos de Diversificação**

Estes mecanismos têm por objetivo aumentar o tamanho da vizinhança a ser explorada e, conseqüentemente, o tempo de pesquisa.

Começamos por aplicar um mecanismo de diversificação fraca e caso nenhuma solução melhor seja encontrada, aplicamos um mecanismo de diversificação mais forte.

- **Mecanismo de Diversificação Fraca**

O lado direito da restrição $\Delta(x, \bar{x}) \leq k$ é aumentado em $\lceil \frac{k}{2} \rceil$, ou seja, é introduzida a restrição $\Delta(x, \bar{x}) \leq k + \lceil \frac{k}{2} \rceil$ e utilizamos novamente o *solver* para encontrar uma nova solução.

- **Mecanismo de Diversificação Forte**

O lado direito da restrição $\Delta(x, \bar{x}) \leq k$ é aumentado em $2 \times \lceil \frac{k}{2} \rceil$, isto é, é introduzida a restrição $\Delta(x, \bar{x}) \leq k + 2\lceil \frac{k}{2} \rceil$ e utilizamos novamente o *solver* para encontrar uma nova solução. Este mecanismo de diversificação é inspirado na técnica *Variable Neighborhood Search* [47].

Ao aplicar este último mecanismo, geralmente encontra-se uma solução pior que a melhor solução conhecida, mas não muito distante dela.

Note-se que o mecanismo de diversificação forte só é executado se o mecanismo de diversificação fraca foi executado sem sucesso na iteração anterior.

8.2.3 Critérios de Paragem

Em alguns casos, e dependendo do valor do parâmetro k , o tempo de processamento do ramo esquerdo até encontrar a solução ótima na vizinhança $\mathcal{N}(\bar{x}, k)$ é muito elevado. Assim, do ponto de vista heurístico é razoável impor à partida um critério de paragem no *solver*. Pode-se optar, por exemplo, pelos dois critérios seguintes:

- explorar as vizinhanças até que a primeira solução inteira seja obtida;
- impor um limite de tempo na exploração das vizinhanças.

Como são impostos critérios de paragem no *solver* quando se explora uma dada vizinhança pode acontecer uma das seguintes situações:

1. A solução obtida é ótima

O *solver* resolveu o subproblema até encontrar a solução ótima da vizinhança $\mathcal{N}(\bar{x}, k)$. Nesta situação podem ocorrer os seguintes casos:

- **A solução ótima obtida é melhor do que a solução de referência**
O *solver* conseguiu encontrar a solução ótima \bar{x}^t da vizinhança $\mathcal{N}(\bar{x}, k)$, tal que, $C(T_{\bar{x}^t}) < C(T_{\bar{x}})$, então atualizamos a solução de referência para a nova solução encontrada. Sem alterar o valor do parâmetro k , introduzimos a restrição $\Delta(x, \bar{x}) \leq k$.
- **A solução ótima obtida não é melhor do que a solução de referência**
O *solver* conseguiu encontrar a solução ótima \bar{x}^t da vizinhança $\mathcal{N}(\bar{x}, k)$, tal que, $C(T_{\bar{x}^t}) \geq C(T_{\bar{x}})$. Neste caso, o *solver* é chamado para explorar o restante espaço de pesquisa (introdução da restrição $\Delta(x, \bar{x}) \geq k + 1$ correspondente à ramificação direita) e assim concluir a enumeração.

2. A solução obtida não é admissível

O *solver* não conseguiu encontrar uma solução inteira admissível para o subproblema na vizinhança $\mathcal{N}(\bar{x}, k)$.

3. A solução obtida é admissível

O *solver* resolveu o subproblema e encontrou uma solução inteira admissível, mas não corresponde à solução ótima da vizinhança $\mathcal{N}(\bar{x}, k)$. Nesta situação podem ainda ocorrer dois casos:

- **A solução obtida é melhor do que a solução de referência**

O *solver* conseguiu encontrar na vizinhança $\mathcal{N}(\bar{x}, k)$ uma solução inteira admissível \bar{x}^t , tal que, $C(T_{\bar{x}^t}) < C(T_{\bar{x}})$, então atualizamos a solução de referência para a nova solução encontrada. Sem alterar o valor do parâmetro k , introduzimos a restrição $\Delta(x, \bar{x}) \leq k$.

- **A solução obtida não é melhor do que a solução de referência**

O *solver* conseguiu encontrar na vizinhança $\mathcal{N}(\bar{x}, k)$ uma solução inteira admissível \bar{x}^t , tal que, $C(T_{\bar{x}^t}) \geq C(T_{\bar{x}})$. Neste caso podemos aplicar os mecanismos de intensificação ou diversificação de modo a reduzir ou a aumentar a vizinhança.

8.3 *Local Branching versus Feasibility Pump*

Na Heurística *Feasibility Pump* (FP), para a obtenção da solução inicial é necessário recorrer ao *solver* de um *software* de Otimização para obter x^* , a solução ótima da relaxação linear de uma formulação para o Problema WMST, sendo esta depois arredondada de acordo com determinado critério e obtida assim uma solução inteira \tilde{x} , que pode ou não ser admissível. Na Heurística *Local Branching* (LB) podemos recorrer inicialmente ao *solver* para resolver o Algoritmo *Branch and Bound* de uma formulação para o Problema WMST até que seja obtida uma primeira solução inteira admissível, \bar{x} , ou optar por uma das soluções de referência proposta na Subsecção 8.2.1. Enquanto que na Heurística LB a solução inicial \bar{x} é inteira e admissível, na Heurística FP a solução inicial \tilde{x} é inteira, mas pode ou não ser admissível.

Em ambos os métodos utilizamos a função distância $\Delta(x, \bar{x})$ e pretendemos que esta seja pequena, isto é, na Heurística FP a função distância é utilizada como função objetivo que se pretende minimizar, enquanto que na Heurística LB é usada como restrição, onde se pretende explorar uma vizinhança pequena (não permitindo que se efetuem muitas trocas) de tal forma que a vizinhança seja explorada rapidamente.

Quando em duas iterações consecutivas as soluções forem iguais, isto é, não houver melhoramento no custo das soluções, na Heurística FP aplicam-se mecanismos de perturbação e na Heurística LB aplicam-se mecanismos de intensificação e de diversificação.

Geometricamente, a Heurística FP define duas trajetórias de pontos que se esperam convergentes (ver Secção 7.4), enquanto que a Heurística LB define apenas uma trajetória de pontos inteiros e admissíveis que se espera convergente para um ponto. A Heurística LB também se pode interpretar como uma trajetória de custos e uma trajetória de pesos que convergem para um custo $C(T_{\bar{x}})$ e um peso $W(T_{\bar{x}})$.

Nos gráficos das Figuras 8.2 e 8.3 podemos observar as trajetórias de custos e de pesos da instância 10-8 do grupo de instâncias Quase Caminhos.



Figura 8.2: Representação da trajetória de custos da instância QC10-8.



Figura 8.3: Representação da trajetória de pesos da instância QC10-8.

A trajetória de custos converge para o custo 577 e a trajetória de pesos converge para o peso 999. Assim é obtida a solução ótima que corresponde a uma árvore de suporte com custo $C(T_{\bar{x}}) = 577$ e peso $W(T_{\bar{x}}) = 999$.

Note-se que no caso da Heurística LB a sequência de custos obtida é decrescente, pois só são atualizadas as soluções no caso de existir melhoramento. Na Heurística FP as duas sequências de custos podem não ter qualquer monotonia.

8.4 Descrição dos Algoritmos *Local Branching* para o WMST

Tendo por base o Algoritmo ALBC aplicado ao Problema WMST apresentado e descrito detalhadamente na Secção 8.1, propomos cinco versões para testar e comparar.

Algoritmo *Local Branching* 1 (ALB1)

Este algoritmo é o apresentado na Secção 8.1 com ligeiras modificações. Atendendo a que à medida que a dimensão do problema aumenta, o tempo de execução também

aumenta, há a necessidade de impor um limite de tempo para a execução do algoritmo (TL). O facto de impor um limite de tempo, leva a que o Passo 3 do Algoritmo ALBC aplicado ao Problema WMST seja modificado, tomando a seguinte forma:

Passo 3: Critério de paragem e obtenção de novas soluções inteiras admissíveis

Se *status* de $\mathcal{N}(\bar{x}, k) = \text{“Optimum found”}$, então

Se $C(T_{\bar{x}^t}) < C(T_{\bar{x}})$, então

atualizar $\bar{x} = \bar{x}^t$ e voltar a efetuar o **Passo 2**.

Caso contrário,

introduzir $\Delta(x, \bar{x}) \geq k + 1$ e utilizar um *solver* para obter \bar{x}^t .

Se $C(T_{\bar{x}^t}) < C(T_{\bar{x}})$, então

atualizar $\bar{x} = \bar{x}^t$.

STOP.

Caso contrário,

Se *status* de $\mathcal{N}(\bar{x}, k) = \text{“Unfeasible”}$, então

STOP.

Caso contrário,

Se *status* de $\mathcal{N}(\bar{x}, k) = \text{“Unfinished”}$, então

Se $C(T_{\bar{x}^t}) < C(T_{\bar{x}})$, então

atualizar $\bar{x} = \bar{x}^t$ e voltar a efetuar o **Passo 2**.

Caso contrário,

introduzir $\Delta(x, \bar{x}) \geq k + 1$ e utilizar um *solver* para obter \bar{x}^t .

Se $C(T_{\bar{x}^t}) < C(T_{\bar{x}})$, então

atualizar $\bar{x} = \bar{x}^t$.

STOP.

Se um limite total de tempo é imposto para a execução do algoritmo, significa que na última vizinhança a ser explorada pode acontecer que o *solver* tenha de parar

antes que uma solução inteira admissível seja encontrada ou antes de ser provada a otimalidade na vizinhança explorada. Estes casos que podem ocorrer encontram-se descritos na Subsecção 8.2.3.

Algoritmo *Local Branching* 2 (ALB2)

Neste algoritmo é também imposto um limite de tempo (igual ao do Algoritmo ALB1). São introduzidos mecanismos de intensificação e de diversificação e o *solver* é interrompido assim que se obtenha uma solução inteira admissível no Algoritmo *Branch and Bound* de uma formulação para o Problema WMST. Assim, é necessário modificar os Passo 2 e 3 do Algoritmo ALBC descrito na Secção 8.1 que passam a ter a seguinte forma:

Passo 2: *Introduzir a restrição local branching*

$t := t + 1$.

Definir $\bar{S} = \{(i, j) \in A : \bar{x}_{ij} = 1\}$ e um valor de k inteiro e positivo.

Introduzir a restrição *local branching* $\Delta(x, \bar{x}) \leq k$.

Obter uma solução inteira admissível \bar{x}^t da vizinhança $\mathcal{N}(\bar{x}, k)$ utilizando o *solver*.

Passo 3: *Critério de paragem e obtenção de novas soluções inteiras admissíveis*

Se *status* de $\mathcal{N}(\bar{x}, k) = \text{“Optimum found”}$, então

Se $C(T_{\bar{x}^t}) < C(T_{\bar{x}})$, então

atualizar $\bar{x} = \bar{x}^t$ e voltar a efetuar o **Passo 2**.

Caso contrário,

introduzir $\Delta(x, \bar{x}) \geq k + 1$ e utilizar um *solver* para obter \bar{x}^t .

Se $C(T_{\bar{x}^t}) < C(T_{\bar{x}})$, então

atualizar $\bar{x} = \bar{x}^t$.

STOP.

Caso contrário,

Se *status* de $\mathcal{N}(\bar{x}, k) = \text{“Unfeasible”}$, então

aumentar $k := k + 1$ e aplicar o Mecanismo de Diversificação Forte.

Caso contrário,

Se *status* de $\mathcal{N}(\bar{x}, k) = \text{“Unfinished”}$, então

Se $C(T_{\bar{x}^t}) < C(T_{\bar{x}})$, então

atualizar $\bar{x} = \bar{x}^t$ e voltar a efetuar o **Passo 2**.

Caso contrário,

aplicar os Mecanismos de Intensificação e de Diversificação.

Enquanto que no Passo 2 do Algoritmo ALBC procuramos obter a solução ótima da vizinhança, no Passo 2 deste algoritmo apenas procuramos encontrar a primeira solução inteira admissível da vizinhança.

Os mecanismos de intensificação e de diversificação são introduzidos quando não é encontrada uma solução inteira admissível ou quando a solução obtida é inteira admissível e não é melhor do que a solução de referência. Os mecanismos de intensificação e de diversificação são aplicados como descrito na Subsecção 8.2.2. Primeiro tentamos aplicar o mecanismo de intensificação, para reduzir o tempo de pesquisa. Se não se conseguir uma solução melhor, então aplicamos a diversificação fraca. O mecanismo de diversificação forte só é aplicado no caso de na iteração anterior se ter aplicado sem sucesso um mecanismo de diversificação fraca.

Algoritmo *Local Branching* 3 (ALB3)

Os Passos 2 e 3 foram implementados da mesma forma que os Passos 2 e 3 do Algoritmo ALB2. A solução de referência obtida no Passo 1 corresponde à primeira solução inteira admissível obtida pelo *solver* quando usa o procedimento *Branch and Bound*. Na obtenção das restantes soluções inteiras admissíveis é imposto um limite de tempo na exploração das vizinhanças (nodos de ramificação esquerdos).

Uma vantagem deste algoritmo em relação ao anterior é o facto de durante o tempo imposto para a exploração ser possível encontrar soluções inteiras admissíveis de melhor qualidade, dado que cada vez que se chama o *solver* este não é interrompido assim que se obtém a primeira solução inteira admissível, mas sim quando exceder o limite de

tempo imposto na exploração da vizinhança. Tal como nos Algoritmos ALB1 e ALB2 é ainda imposto um tempo limite total para a execução do algoritmo (TL).

Exemplo 8.2.

Para exemplificar a introdução dos mecanismos de intensificação e de diversificação vamos aplicar o Algoritmo ALB3 à instância 10-10 do grupo de instâncias Quase Caminhos com $W = 1000$. Para a restrição *local branching* considerou-se a desigualdade (8.2) com $k' = 5$ e utilizou-se a Formulação WMTZ.

No esquema em árvore da Figura 8.4 podemos observar que no nodo 1 é obtida a solução de referência $\bar{x} = \bar{x}^1$, com custo 782 e peso 872. Após se ter introduzido a restrição $\Delta(x, \bar{x}^1) \leq 5$, obteve-se no nodo 2 a solução \bar{x}^2 . Como o custo desta solução é inferior ao custo da solução de referência ($C(T_{\bar{x}^2}) = 662 < 782 = C(T_{\bar{x}^1})$), então atualiza-se $\bar{x} = \bar{x}^2$. No nodo 3 é adicionada a ramificação esquerda $\Delta(x, \bar{x}^2) \leq 5$ e a solução obtida no nodo 4 não melhora, o que leva a introduzir um mecanismo de intensificação. Então volta-se ao nodo 3 e adiciona-se a ramificação correspondente a $\Delta(x, \bar{x}^2) \leq 2$. Como a solução obtida no nodo 5 continua a não melhorar, então aplica-se um mecanismo de diversificação fraca que corresponde a voltar ao nodo 3 e introduzir mais uma ramificação com a restrição $\Delta(x, \bar{x}^2) \leq 7$. Como a solução obtida no nodo 6 tem custo inferior a $C(T_{\bar{x}}) = 662$, então atualiza-se a solução de referência para \bar{x}^5 . No ramo que liga o nodo 7 ao nodo 9 volta-se a aplicar o mecanismo de intensificação dado que a restrição que foi introduzida na ramificação do nodo 7 para o nodo 8 não originou melhorias. Finalmente, a ligação do nodo 7 ao nodo 10 explora o restante espaço de pesquisa. A solução tem custo 657 e peso 930.

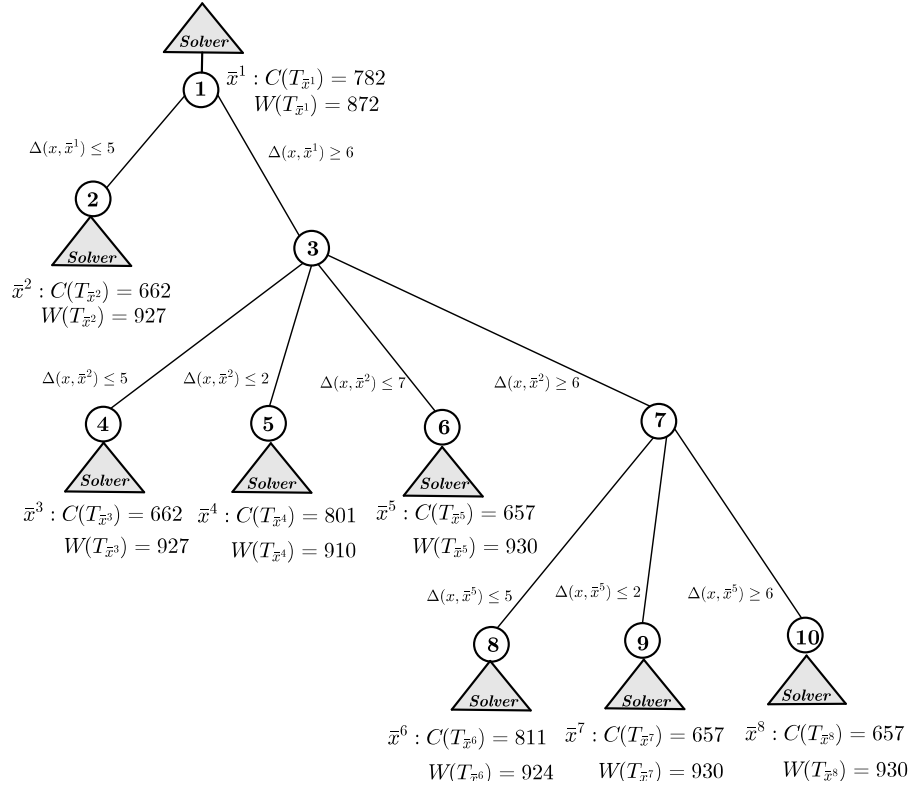


Figura 8.4: Esquema de ramificação obtido usando o Algoritmo ALB3 na instância QC10-10.

Os próximos algoritmos que vamos apresentar apenas diferem do Algoritmo ALB3 no Passo 1, onde é obtida uma solução de referência inicial sem recorrer ao *solver*.

Algoritmo *Local Branching* 4 (ALB4)

Para a obtenção da solução de referência inicial usamos a solução inteira admissível obtida através da Heurística FPB2 apresentada no Capítulo 7. A escolha desta heurística deve-se ao facto de, em geral, ter apresentado soluções de melhor qualidade num menor tempo (ver resultados computacionais do Capítulo 7). Com a aplicação deste algoritmo pretendemos melhorar a solução obtida através da Heurística FPB2. Em caso de falha da Heurística *Feasibility Pump* na obtenção de uma solução inteira admissível, usa-se

o *solver* para obter a primeira solução de referência.

Algoritmo *Local Branching* 5 (ALB5)

A primeira solução de referência corresponde à árvore de suporte de peso mínimo T_w , a qual pode ser obtida através da aplicação do Algoritmo de Prim [51].

8.5 Experiências para Comparação dos Algoritmos *Local Branching* e Resultados Computacionais

Nesta secção apresentamos os resultados computacionais dos vários algoritmos *Local Branching* propostos para o Problema WMST. Começamos por efetuar um estudo comparativo entre os vários algoritmos *Local Branching* propostos com o objetivo de escolher o mais eficiente para o comparar com o algoritmo *Branch and Bound* do Procedimento P-WMTZ+C.

Todos os testes foram efetuados num Intel(R) Core(TM)2 Duo CPU (T7100) 2.00 GHz processador e 4Gb de RAM e usou-se o *software Xpress 7.3* (*Xpress-Optimizer 23.01.03* e *Xpress-Mosel 3.4.0*) [1] para implementar todos os algoritmos *Local Branching*.

Como em todos estes algoritmos é obtido um limite superior para o valor do custo, que pode ou não ser o valor ótimo, o *gap* correspondente é dado por

$$gap = \frac{LS - OPT}{OPT} \times 100,$$

onde LS é o valor do limite superior obtido para o valor do custo através da aplicação de um dos algoritmos *Local Branching* e OPT é o valor ótimo ($\vartheta(WMST)$) ou o melhor valor encontrado até ao momento.

8.5.1 Exemplo Comparativo dos Algoritmos *Local Branching*

Com o objetivo de comparar os diferentes algoritmos *Local Branching* descritos ao longo da Seção 8.4 elaborámos a Tabela 8.1, onde se apresentam os sucessivos custos e pesos obtidos em cada um dos algoritmos para a instância 10-8 do grupo de instâncias Quase Caminhos.

	ALB1		ALB2		ALB3		ALB4		ALB5	
Iter	Custo	Peso	Custo	Peso	Custo	Peso	Custo	Peso	Custo	Peso
0	659	991	659	991	659	991	637	821	793	650
1	605	985	665	985	605	985	629	872	577	999
2	605	985	609	981	605	985	605	985	577	999
3	577	999	653	875	577	999	605	985	577	999
4			605	985			577	999		
5			673	861						
6			605	985						
7			577	999						

Tabela 8.1: Comparação dos algoritmos *Local Branching* na instância QC10-8.

Os valores a negrito na Tabela 8.1 indicam que foi atualizada a solução de referência.

Na instância QC10-8 o valor do custo e do peso da solução de referência inicial é igual nos Algoritmos ALB1, ALB2 e ALB3, pois nestes três algoritmos a solução inicial corresponde à primeira solução inteira admissível obtida pelo *solver* através da Formulação WMTZ quando usa o procedimento de *Branch and Bound*.

No Algoritmo ALB4 o valor do custo da solução de referência inicial é melhor do que o obtido nos restantes algoritmos, dado que é obtido através da Heurística FPB2 apresentada no Capítulo 7, enquanto que o valor do custo da solução de referência inicial usada no Algoritmo ALB5 é o de pior qualidade, pois corresponde ao valor do custo da árvore de suporte de peso mínimo, mas logo na iteração seguinte consegue-se obter o valor ótimo. Nesta instância todos os algoritmos convergem para o mesmo valor de custo e de peso, sendo o custo 577 e o peso 999 que corresponde à solução ótima da instância. Neste exemplo, o algoritmo que realiza um maior número de iterações é o Algoritmo ALB2.

8.5.2 Estudo Comparativo dos Algoritmos *Local Branching*

Nesta subsecção, para comparar os diferentes algoritmos *Local Branching* propostos apenas apresentamos os resultados computacionais de instâncias até 100 nodos dos grupos de instâncias Quase Caminhos, Aleatórias e Euclidianas (ver Secção 2.3).

Para começar vamos descrever os parâmetros utilizados nos algoritmos. Em todos os algoritmos *Local Branching* propostos foi imposto um limite de tempo global na execução do algoritmo que corresponde a $TL = 10800$ segundos. Após se efetuarem algumas experiências computacionais decidimos utilizar a restrição *local branching* dada pela expressão (8.2). Quanto aos valores de k testaram-se alguns valores no intervalo $[10, 20]$. No seguinte gráfico encontram-se os tempos médios de execução (em segundos) para instâncias até 100 nodos de três valores de k' .

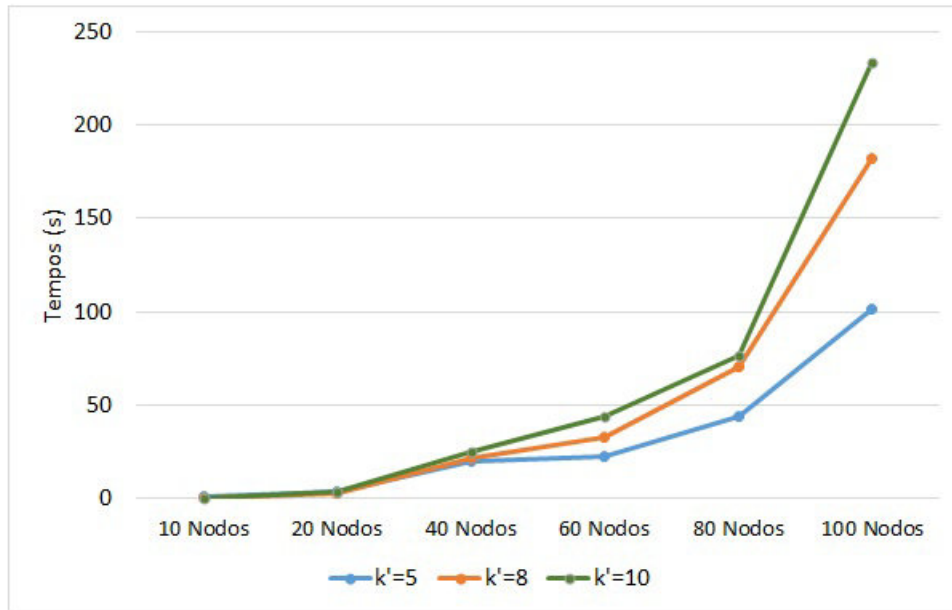


Figura 8.5: Tempos médios de execução, em segundos, obtidos usando o Algoritmo ALB4 para diferentes valores de k' .

Através do gráfico da Figura 8.5 podemos observar que à medida que o valor de k' aumenta, os tempos médios também aumentam para instâncias maiores ou iguais a 40 nodos. Realça-se ainda que se encontram soluções de melhor qualidade quando $k' = 5$. Assim, em todos os Algoritmos LB com exceção do Algoritmo ALB5 vamos utilizar o valor do parâmetro $k = 10$ ($k' = 5$). No Algoritmo ALB5 foi escolhido inicialmente

um $k' = \left\lceil \frac{n-1}{2} \right\rceil$. A escolha deste valor deve-se ao facto do custo da correspondente árvore de suporte de peso mínimo estar bastante afastada do valor ótimo e ao serem efetuadas poucas trocas pode não ser possível encontrar uma solução na vizinhança considerada $\mathcal{N}(\bar{x}, k)$.

Depois de realizadas algumas experiências, nos algoritmos onde se aplicam os mecanismos de intensificação e de diversificação o número de intensificações ou diversificações foi limitado a 5.

Nos Algoritmos ALB3, ALB4 e ALB5 foram impostos tempos (em segundos) para explorar as vizinhanças, os quais se encontram na seguinte tabela.

N.º Nodos		10	20	40	60	80	100
ALB3	QC e R	1	2	5	10	20	40
	E	1	2	10	30	200	350
ALB4 e ALB5	QC e R	1	2	5	10	20	40
	E	1	2	10	15	20	50

Tabela 8.2: Tempo, em segundos, imposto na exploração das vizinhanças.

De seguida comparamos os algoritmos *Local Branching* propostos em cada um dos grupos de instâncias em separado para averiguar qual dos algoritmos é o mais eficiente.

Instâncias Quase Caminhos

No gráfico da Figura 8.6 podemos observar a percentagem de soluções ótimas obtidas no grupo de instâncias Quase Caminhos em cada um dos algoritmos *Local Branching* propostos para instâncias até 100 nodos.

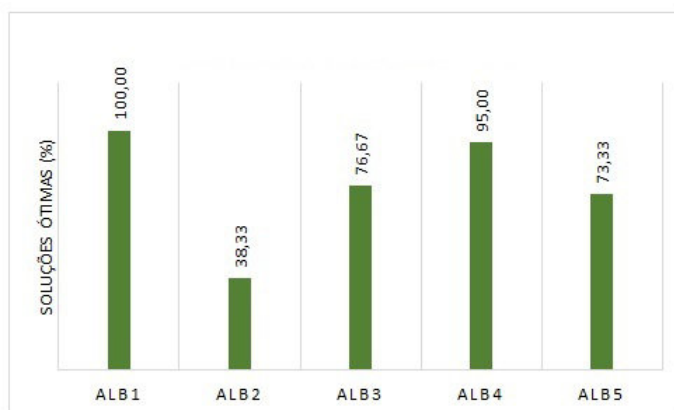


Figura 8.6: Percentagem de soluções ótimas obtidos pelos algoritmos *Local Branching* nas instâncias QC.

Em todos os algoritmos com exceção do Algoritmo ALB2 obtêm-se percentagens elevadas de soluções ótimas. Com a aplicação do Algoritmo ALB1 as 60 instâncias testadas apresentam *gap* nulo. O Algoritmo ALB4 é o segundo a ter maior percentagem de soluções ótimas (95% que corresponde a 57 em 60 instâncias). Depois, temos o Algoritmo ALB3 com soluções ótimas em 76,67% dos casos (46 em 60 instâncias). O Algoritmo ALB5 também apresenta uma boa percentagem de soluções ótimas, 73,33%, que corresponde a 44 instâncias. O algoritmo com menor percentagem de soluções ótimas é o Algoritmo ALB2 com apenas 38,33% dos casos (23 em 60 instâncias).

Na Tabela 8.3 podemos visualizar os *gaps* médios, em percentagem, e os correspondentes desvios padrões e também os tempos médios de execução, em segundos, e os respetivos desvios padrões obtidos para os cinco algoritmos *Local Branching* propostos.

	N.º Nodos	ALB1	ALB2	ALB3	ALB4	ALB5
<i>Gap</i> Médio (%)	10	0	0,338	0	0	0
	20	0	1,546	0,097	0	0
	40	0	1,063	0,222	0,174	0,181
	60	0	0,761	0,043	0	0
	80	0	0,156	0,037	0	0,015
	100	0	0,424	0,160	0	0
σ_{gap}	10	0	0,765	0	0	0
	20	0	3,752	0,306	0	0
	40	0	0,994	0,366	0,438	0,475
	60	0	0,640	0,092	0	0
	80	0	0,226	0,118	0	0,032
	100	0	0,551	0,280	0	0
Tempos Médio (s)	10	0,421	0,582	0,403	0,398	0,394
	20	3,750	4,663	3,491	2,368	3,035
	40	64,057	29,265	24,322	20,005	36,845
	60	2273,078	17,854	34,097	23,998	76,875
	80	3443,434	41,932	61,954	51,046	159,832
	100	9999,254	67,038	123,663	98,633	287,813
σ_{tempo}	10	0,092	0,212	0,053	0,143	0,191
	20	1,143	1,676	1,021	0,688	1,053
	40	35,763	6,668	5,040	5,844	3,450
	60	3554,363	3,349	14,049	5,315	6,752
	80	3593,114	8,988	17,306	10,495	24,908
	100	0,418	16,523	19,429	19,435	25,297

Tabela 8.3: Comparação dos algoritmos *Local Branching* em termos de *gaps* médios (em percentagem) e de tempo médio de execução (em segundos) nas instâncias QC.

Os Algoritmos ALB1, ALB4 e ALB5 são os que apresentam *gaps* de melhor qualidade. Quanto aos Algoritmos ALB2 e ALB3 são os que apresentam os *gaps* de pior qualidade, embora relativamente baixos.

Quanto ao tempo de execução podemos observar que os Algoritmo ALB2 e ALB4 são os mais rápidos. O Algoritmo ALB1 apesar de ter *gap* nulo para todas as instâncias apresenta tempos médios bastante elevados quando comparado com os restantes algoritmos.

Instâncias Aleatórias

No gráfico da Figura 8.7 podemos observar a percentagem de soluções ótimas obtidas no grupo de instâncias Aleatórias em cada um dos algoritmos *Local Branching* propostos para instâncias até 100 nodos.

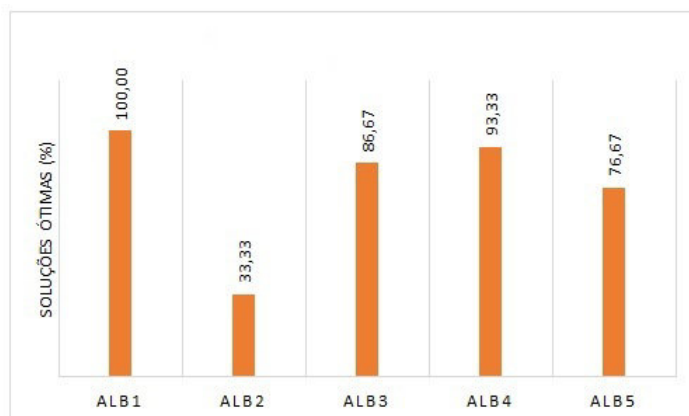


Figura 8.7: Percentagem de soluções ótimas obtidas pelos Algoritmos *Local Branching* nas instâncias R.

Da observação do gráfico, verifica-se que neste grupo de instâncias se encontram muitos *gaps* nulos. Com a aplicação do Algoritmo ALB1 as 30 instâncias testadas apresentam *gap* nulo. O Algoritmo ALB4 é o segundo a ter maior percentagem de soluções ótimas, 93,33% que corresponde a 28 instâncias. A seguir, temos o Algoritmo ALB3 com soluções ótimas em 86,67% das instâncias (26 em 30 instâncias). O Algoritmo ALB5 também apresenta uma boa percentagem de soluções ótimas, 76,67% que corresponde a 23 instâncias. O algoritmo com menor percentagem de soluções ótimas é o ALB2 com apenas 33,33% dos casos (10 em 30 instâncias).

Na Tabela 8.4 podemos observar os *gaps* médios, em percentagem obtidos para cada um dos algoritmos *Local Branching* e os respetivos desvios padrões. Na mesma tabela são também apresentados os tempos médios de execução, em segundos, e os correspondentes desvios padrões.

	N.º Nodos	ALB1	ALB2	ALB3	ALB4	ALB5
<i>Gap</i> Médio	10	0	0	0	0,501	0
	20	0	2,280	0,039	0	0
	40	0	1,705	0,369	0	24,433
	60	0	0,525	0	0	2,817
	80	0	1,137	0,007	0	0,021
	100	0	11,614	1,255	0,010	37,293
σ_{gap}	10	0	0	0	1,119	0
	20	0	3,535	0,087	0	0
	40	0	2,668	0,577	0	54,524
	60	0	0,637	0	0	6,300
	80	0	1,325	0,015	0	0,046
	100	0	12,036	1,225	0,023	74,585
Tempos Médio (s)	10	0,405	0,549	0,605	0,375	0,222
	20	2,683	4,733	2,818	2,346	2,209
	40	17,257	26,271	13,670	10,062	12,103
	60	18,891	20,321	20,037	13,473	15,001
	80	64,020	48,808	49,259	30,499	39,717
	100	1942,807	90,721	238,077	75,494	89,788
σ_{tempo}	10	0,164	0,373	0,343	0,166	0,089
	20	0,822	0,350	0,756	0,269	0,907
	40	8,272	6,820	4,726	2,595	10,435
	60	6,074	4,764	10,700	3,962	11,722
	80	54,930	10,903	54,326	12,240	32,372
	100	2690,993	45,009	189,743	40,712	122,236

Tabela 8.4: Comparação dos algoritmos *Local Branching* em termos de *gaps* médios (em percentagem) e de tempo médio de execução (em segundos) nas instâncias R.

Os Algoritmos ALB1, ALB3 e ALB4 além de apresentarem as maiores percentagens de soluções ótimas, também são os que apresentam os *gaps* de melhor qualidade. Quanto aos Algoritmos ALB2 e ALB5 são os que apresentam os *gaps* de pior qualidade.

O Algoritmo ALB4 apresenta tempos médios inferiores para instâncias com mais de 40 nodos e os Algoritmos ALB1 e ALB3 apresentam tempos médios mais elevados quando comparados com o Algoritmo ALB4.

Instâncias Euclidianas

No gráfico da Figura 8.8 apresentam-se as percentagens de soluções ótimas obtidas em cada um dos algoritmos *Local Branching* no grupo de instâncias Euclidianas para instâncias até 100 nodos.

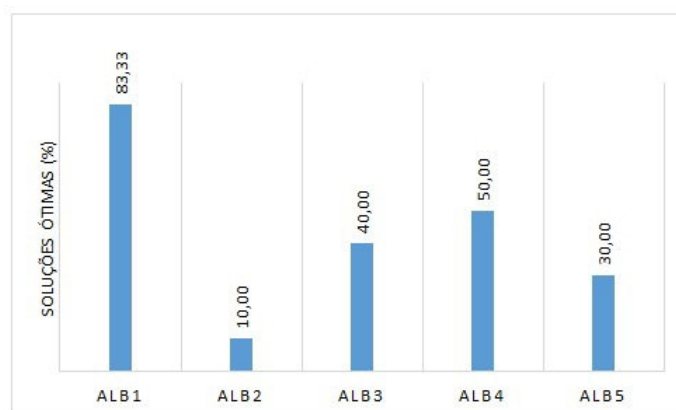


Figura 8.8: Percentagem de soluções ótimas obtidas pelos algoritmos *Local Branching* nas instâncias E.

Os algoritmos que apresentam maior percentagem de soluções ótimas são o Algoritmo ALB1 com 83,33% das instâncias (25 em 30 instâncias) e o Algoritmo ALB4 com 50% dos casos (16 em 30 instâncias). O Algoritmo ALB3 apresenta 40% de soluções ótimas que corresponde a 14 instâncias. No Algoritmo ALB5 obtém-se 30% de soluções ótimas, mas estas ocorrem apenas em instâncias de 10 e 20 nodos (9 em 30 instâncias) e no Algoritmo ALB2 apenas se consegue obter a solução ótima em 3 instâncias.

Na Tabela 8.5 são apresentados os *gaps* médios, em percentagem, obtidos para cada um dos algoritmos *Local Branching* propostos e os respetivos desvios padrões. Nesta tabela são também apresentados os tempos médios de execução, em segundos, acompanhados dos correspondentes desvios padrões.

	N.º Nodos	ALB1	ALB2	ALB3	ALB4	ALB5
<i>Gap</i> Médio (%)	10	0	2,180	0,489	0	0
	20	0	2,875	1,451	0,103	1,705
	40	0	4,416	1,975	0,121	2,234
	60	0	9,303	0,031	0,076	11,600
	80	0,281	9,154	0,460	0,085	34,397
	100	0,151	15,792	0,307	0,064	22,105
σ_{gap}	10	0	3,356	0,547	0	0
	20	0	3,191	3,045	0,230	3,813
	40	0	3,194	1,976	0,268	3,485
	60	0	6,355	0,059	0,126	8,665
	80	0,409	3,426	0,655	0,153	29,054
	100	0,207	3,422	0,330	0,069	26,790
Tempos Médio (s)	10	0,885	2,278	2,686	2,689	0,705
	20	6,520	14,917	13,407	6,845	6,739
	40	88,598	63,845	75,011	29,515	43,164
	60	3802,020	62,197	241,641	28,979	75,451
	80	8084,062	476,787	1927,288	44,585	123,633
	100	9102,888	319,507	3471,078	132,123	232,832
σ_{tempo}	10	0,396	1,204	1,770	1,782	0,478
	20	2,841	2,271	3,942	3,258	3,214
	40	71,630	10,628	23,181	7,639	36,337
	60	4238,789	28,713	116,432	6,311	41,659
	80	4285,257	384,109	1396,524	6,020	8,949
	100	2006,389	272,925	1169,519	48,589	118,691

Tabela 8.5: Comparação dos algoritmos *Local Branching* em termos de *gaps* médios (em percentagem) e de tempo médio de execução (em segundos) nas instâncias E.

Os *gaps* de melhor qualidade são obtidos nos Algoritmos ALB1, ALB3 e ALB4. Sendo os Algoritmos ALB2 e ALB5 os que apresentam os *gaps* de pior qualidade. À medida que o número de nodos aumenta, os *gaps* obtidos nos Algoritmos ALB2 e ALB5 parecem ser de pior qualidade.

O Algoritmo ALB4 é o mais rápido, pois apresenta tempos médios de execução inferiores a qualquer um dos outros algoritmos *Local Branching* para instâncias maiores do que 20 nodos. Os Algoritmos ALB1 e ALB3 apesar de terem *gaps* médios baixos apresentam tempos médios de execução bastante elevados quando comparados com os

tempos médios de execução do Algoritmo ALB4.

Síntese da Comparação dos vários Algoritmos *Local Branching*

Dos cinco algoritmos *Local Branching* apresentados podemos verificar que em todos os grupos de instâncias, o Algoritmo ALB4 se mostrou o mais eficiente, apresentando uma percentagem de soluções ótimas em 83,33% das instâncias testadas (100 em 120 instâncias) e soluções muito próximas das ótimas nos restantes casos. Este algoritmo foi também considerado o mais rápido.

8.5.3 Resultados Computacionais

Para avaliar o desempenho da Heurística *Local Branching* vamos comparar, em todas as instâncias, os resultados obtidos pelo Algoritmo ALB4 com os obtidos pelo Algoritmo *Branch and Bound* do procedimento de introdução de cortes P-WMTZ+C (Secção 3.3), o qual será designado por ABB-P-WMTZ+C.

Nas Tabelas 8.6, 8.7, 8.8, 8.9, 8.10, 8.11 e 8.12, a primeira coluna refere-se à designação da instância, as três colunas seguintes contêm o tempo de execução (t_{ABB} , em segundos), o valor ótimo ou o melhor valor do limite superior obtido (LS ABB) e o *gap* (em percentagem) e são referentes ao Algoritmo ABB-P-WMTZ+C. As cinco colunas seguintes contêm informação sobre o Algoritmo ALB4, uma refere-se à solução de referência obtida inicialmente pela Heurística FPB2 apresentada no Capítulo 7 (LS Inic.), a seguir temos a solução obtida após a aplicação do Algoritmo ALB4 (LS LB), depois temos o tempo de execução (t_{LB} , em segundos), a seguir o *gap* e para finalizar os dados referentes ao Algoritmo ALB4 temos o número de iterações (Iter.). Por fim, a última coluna apresenta a razão entre os tempos de execução do Algoritmo ABB-P-WMTZ+C e do Algoritmo ALB4. Em todas as tabelas, a sigla NFA significa que o limite superior nunca foi atualizado, e neste caso não foi obtida uma solução inteira admissível.

	ABB-P-WMTZ+C			ALB4					
Instância	t_{ABB}	LS ABB	Gap	LS Inic.	LS LB	t_{LB}	Gap	Iter.	t_{ABB}/t_{LB}
QC10-1	0,08	168	0	173	168	0,48	0	4	0,16
QC10-2	0,22	364	0	385	364	0,37	0	3	0,58
QC10-3	0,09	375	0	426	375	0,41	0	3	0,23
QC10-4	0,11	364	0	366	364	0,31	0	3	0,35
QC10-5	0,09	581	0	641	581	0,27	0	3	0,35
QC10-6	0,11	179	0	179	179	0,30	0	2	0,37
QC10-7	0,08	439	0	439	439	0,23	0	2	0,33
QC10-8	0,13	577	0	637	577	0,44	0	3	0,29
QC10-9	0,06	263	0	330	263	0,73	0	5	0,09
QC10-10	0,09	657	0	672	657	0,44	0	3	0,21
QC20-1	0,92	1581	0	1584	1581	1,98	0	2	0,46
QC20-2	1,08	1463	0	1473	1463	2,86	0	3	0,38
QC20-3	1,11	1364	0	1376	1364	2,90	0	3	0,38
QC20-4	0,70	674	0	678	674	1,92	0	3	0,37
QC20-5	0,61	547	0	547	547	1,59	0	2	0,38
QC20-6	1,42	930	0	936	930	2,18	0	2	0,65
QC20-7	3,57	1118	0	1118	1118	1,93	0	2	1,85
QC20-8	1,25	320	0	321	320	1,75	0	2	0,71
QC20-9	1,80	621	0	634	621	3,79	0	4	0,47
QC20-10	0,80	1446	0	1451	1446	2,78	0	3	0,29
QC40-1	29,78	1909	0	1940	1909	17,24	0	4	1,73
QC40-2	49,46	913	0	914	913	16,86	0	4	2,93
QC40-3	34,68	1155	0	1158	1156	29,89	0,09	6	1,16
QC40-4	88,17	776	0	778	778	13,93	0,26	3	6,33
QC40-5	25,79	2008	0	2009	2008	16,84	0	4	1,53
QC40-6	28,50	1436	0	1446	1436	17,90	0	4	1,59
QC40-7	101,52	1728	0	1731	1728	22,90	0	5	4,43
QC40-8	17,24	1922	0	1922	1922	13,88	0	3	1,24
QC40-9	14,57	1642	0	1662	1642	20,89	0	5	0,70
QC40-10	27,00	786	0	797	797	29,71	1,40	6	0,91

Tabela 8.6: Comparação entre os Algoritmos ALB4 e ABB-P-WMTZ+C nas instâncias QC de 10 a 40 nodos.

	ABB-P-WMTZ+C			ALB4					
Instância	t_{ABB}	LS ABB	Gap	LS Inic.	LS LB	t_{LB}	Gap	Iter.	t_{ABB}/t_{LB}
QC60-1	473,32	1619	0	1620	1619	30,91	0	4	15,31
QC60-2	610,76	2944	0	2944	2944	16,89	0	2	36,17
QC60-3	554,47	3462	0	3462	3462	20,89	0	3	26,54
QC60-4	1250,20	2144	0	2144	2144	20,84	0	3	59,98
QC60-5	214,40	2994	0	2994	2994	20,88	0	3	10,27
QC60-6	201,09	1785	0	1785	1785	19,95	0	3	10,08
QC60-7	783,40	2092	0	2093	2092	30,83	0	4	25,41
QC60-8	414,74	1486	0	1487	1486	26,99	0	3	15,37
QC60-9	603,67	2442	0	2443	2442	30,77	0	4	19,62
QC60-10	66,16	2138	0	2138	2138	21,04	0	3	3,14
QC80-1	4605,26	1769	0	1769	1769	36,71	0	2	125,44
QC80-2	3618,29	3318	0	3320	3318	51,92	0	3	69,69
QC80-3	2102,42	1661	0	1662	1661	54,86	0	4	38,32
QC80-4	731,77	4759	0	4759	4759	43,82	0	3	16,70
QC80-5	1423,68	1283	0	1284	1283	55,87	0	3	25,48
QC80-6	7280,10	2686	0	2690	2686	62,86	0	4	115,82
QC80-7	1915,78	3751	0	3751	3751	31,81	0	2	60,23
QC80-8	393,61	1959	0	1960	1959	53,18	0	3	7,40
QC80-9	1255,06	2382	0	2385	2382	63,62	0	4	19,73
QC80-10	2024,72	1962	0	1963	1962	55,81	0	3	36,28
QC100-1	10801,10	5410	0	5596	5410	126,88	0	4	85,13
QC100-2	10800,90	4539	0	4539	4539	86,74	0	3	124,52
QC100-3	10800,80	2840	0	2840	2840	86,13	0	3	125,40
QC100-4	10800,60	4153	0	4153	4153	89,20	0	3	121,08
QC100-5	10801,00	3723	0	3723	3723	86,24	0	3	125,25
QC100-6	10800,80	4666	0	4815	4666	126,72	0	4	85,23
QC100-7	10800,70	4981	0	4981	4981	85,87	0	3	125,78
QC100-8	10801,10	5213	0	5214	5213	126,69	0	4	85,26
QC100-9	10800,90	4240	0	4240	4240	85,91	0	3	125,73
QC100-10	8758,11	2086	0	2086	2086	85,95	0	3	101,90

Tabela 8.7: Comparação entre os Algoritmos ALB4 e ABB-P-WMTZ+C nas instâncias QC de 60 a 100 nodos.

	ABB-P-WMTZ+C			ALB4					
Instância	t_{ABB}	LS ABB	Gap	LS Inic.	LS LB	t_{LB}	Gap	Iter.	t_{ABB}/t_{LB}
QC150-1	10802,40	9251	0	9251	9251	141,78	0	3	76,19
QC150-2	10802,60	5365	0	5366	5365	223,62	0	5	48,31
QC150-3	10806,30	1987	0	2026	1987	211,99	0	4	50,98
QC150-4	10806,40	8173	0	8372	8173	212,55	0	4	50,84
QC150-5	10803,30	5768	0	5957	5768	202,20	0	4	53,43
QC150-6	10805,80	5616	0	5616	5616	138,78	0	3	77,87
QC150-7	10805,80	5138	0	5138	5138	141,19	0	3	76,53
QC150-8	10802,60	7146	0	7146	7146	138,82	0	3	77,82
QC150-9	10807,00	5801	0	5803	5801	204,03	0	4	52,97
QC150-10	10803,90	8094	0	8094	8094	146,13	0	3	73,94
QC200-1	10811,00	8949	0	8949	8949	197,84	0	3	54,64
QC200-2	10816,90	10462	0	10564	10462	292,60	0	4	36,97
QC200-3	10818,00	14814	0	15327	14814	481,79	0	7	22,45
QC200-4	10815,10	12639	0	12824	12639	597,70	0	7	18,09
QC200-5	10806,50	10437	0	10437	10437	208,33	0	3	51,87
QC300-1	10864,10	5562	0	5680	5562	442,19	0	4	24,57
QC300-2	10825,50	26842	0	27006	26842	944,77	0	9	11,46
QC300-3	10857,70	28018	0,02	28012	28012	679,32	0	6	15,98
QC300-4	10866,30	6602	0	6696	6602	686,95	0	6	15,82
QC300-5	10921,10	11383	0	11595	11595	689,94	1,86	6	15,83
QC400-1	10860,00	7138	0	7282	7138	1796,03	0	8	6,05
QC400-2	10938,60	16336	0	16475	16336	1398,10	0	6	7,82
QC400-3	10915,90	6623	0	6685	6623	1606,16	0	7	6,80
QC400-4	10873,30	7168	0	7216	7168	1409,06	0	6	7,72
QC400-5	11059,50	22095	0	22095	22095	1418,17	0	6	7,80
QC500-1	10924,50	14513	0	14667	14513	3429,02	0	6	3,19
QC500-2	11079,30	32723	0,45	32575	32575	4204,46	0	6	2,64
QC500-3	11270,60	25041	0	25876	25041	3501,10	0	7	3,22
QC500-4	11033,50	37829	0	38020	37829	3535,20	0	6	3,12
QC500-5	10979,10	29198	0,01	29315	29194	3451,37	0	6	3,18
QC1000-1	11690,10	NFA	—	77132	75948	9883,74	0	9	1,18
QC1000-2	12766,60	NFA	—	33224	32704	10002,10	0	9	1,28
QC1000-3	11872,10	NFA	—	51310	50127	9909,95	0	8	1,20
QC1000-4	12109,60	NFA	—	57844	57844	7963,73	0	6	1,52
QC1000-5	11505,10	96038	0	97438	96213	9509,04	0,18	8	1,21

Tabela 8.8: Comparação entre os Algoritmos ALB4 e ABB-P-WMTZ+C nas instâncias QC de 150 a 1000 nodos.

	ABB-P-WMTZ+C			ALB4					
Instância	t_{ABB}	LS ABB	Gap	LS Inic.	LS LB	t_{LB}	Gap	Iter.	t_{ABB}/t_{LB}
R10-1	0,062	19304	0	19949	19304	0,219	0	3	0,28
R10-2	0,234	20759	0	20759	20759	0,171	0	2	1,37
R10-3	0,046	15222	0	15603	15603	0,499	2,503	3	0,09
R10-4	0,219	19552	0	21378	19552	0,515	0	3	0,43
R10-5	0,374	29405	0	30586	29405	0,469	0	3	0,80
R20-1	1,014	32055	0	32705	32055	2,216	0	3	0,46
R20-2	1,919	32340	0	32655	32340	2,558	0	3	0,75
R20-3	0,452	33366	0	34872	33366	2,465	0	3	0,18
R20-4	0,657	31835	0	33500	31835	2,558	0	4	0,26
R20-5	1,623	30040	0	34149	30040	1,935	0	3	0,84
R40-1	4,509	27601	0	28230	27601	9,766	0	3	0,46
R40-2	3,681	39903	0	41165	39903	13,277	0	3	0,28
R40-3	5,725	28096	0	29522	28096	11,762	0	3	0,49
R40-4	2,573	28469	0	28588	28469	8,970	0	3	0,29
R40-5	4,18	32022	0	32912	32022	6,536	0	3	0,64
R60-1	4,509	39507	0	40078	39507	10,390	0	3	0,43
R60-2	6,069	30675	0	30956	30675	12,262	0	3	0,49
R60-3	6,928	31328	0	31399	31328	13,356	0	3	0,52
R60-4	7,27	44299	0	45659	44299	20,265	0	4	0,36
R60-5	5,026	33088	0	33387	33088	11,092	0	3	0,45
R80-1	15,194	33196	0	33571	33196	22,277	0	3	0,68
R80-2	16,177	45697	0	46030	45697	52,089	0	3	0,31
R80-3	12,416	39223	0	40640	39223	24,087	0	3	0,52
R80-4	11,169	49269	0	49374	49269	27,316	0	3	0,41
R80-5	16,288	33313	0	33926	33313	26,724	0	3	0,61
R100-1	12,9	26575	0	26742	26575	110,965	0	4	0,12
R100-2	28,799	1939	0	1951	1940	101,438	0,052	3	0,28
R100-3	23,103	2257	0	2273	2257	21,138	0	3	1,09
R100-4	16,803	10300	0	10372	10300	66,115	0	3	0,25
R100-5	21,812	3763	0	3766	3763	77,816	0	3	0,28
R150-1	120,09	2265	0	2281	2265	205,487	0	5	0,58
R150-2	97,233	32853	0	33070	32864	119,189	0,033	3	0,82
R150-3	49,048	32838	0	32873	32838	126,974	0	3	0,39
R150-4	67,835	29144	0	29708	29708	311,634	1,935	6	0,22
R150-5	141,153	13103	0	13172	13103	113,324	0	3	1,25

Tabela 8.9: Comparação entre os Algoritmos ALB4 e ABB-P-WMTZ+C nas instâncias R de 10 a 150 nodos.

	ABB-P-WMTZ+C			ALB4					
Instância	t_{ABB}	LS ABB	Gap	LS Inic.	LS LB	t_{LB}	Gap	Iter.	t_{ABB}/t_{LB}
R200-1	162,058	38935	0	39358	38935	190,902	0	3	0,85
R200-2	133,637	39161	0	39237	39190	335,644	0,074	5	0,40
R200-3	618,244	4216	0	4236	4217	322,948	0,024	5	1,91
R200-4	477,987	13503	0	13600	13503	355,444	0	5	1,34
R200-5	161,012	14638	0	14666	14638	291,675	0	4	0,55
R300-1	2696,66	40936	0	41062	40956	821,275	0,049	7	3,28
R300-2	2280,974	44684	0	44748	44748	714,641	0,143	6	3,19
R300-3	733,738	41486	0	41704	41704	705,510	0,525	6	1,04
R300-4	1007,47	14795	0	14826	14804	744,381	0,061	7	1,35
R300-5	1112,231	13656	0	13881	13661	651,648	0,037	6	1,71
R400-1	2607,21	100652	0	100754	100754	1496,020	0,101	6	1,74
R400-2	3506,99	14449	0	14486	14486	1367,210	0,256	6	2,57
R400-3	2249,23	16428	0	16457	16457	1350,430	0,177	6	1,67
R400-4	2231,39	14915	0	14915	14915	1305,650	0	6	1,71
R400-5	2547,54	42765	0	42846	42846	1512,790	0,189	6	1,68
R500-1	5425,43	46254	0	46421	46421	2969,630	0,361	6	1,83
R500-2	12598,78	48269	0	48385	48273	2004,380	0,008	4	6,29
R500-3	10318,92	53763	0	53944	53944	2998,650	0,337	6	3,44
R500-4	10075,06	46892	0	46942	46892	1583,560	0	3	6,36
R500-5	7708,52	47654	0	47681	47654	2232,740	0	5	3,45
R1000-1	20451,42	NFA	—	4812	4812	10999,400	0	5	1,86
R1000-2	13308,52	105880	0,150	105721	105721	10991,700	0	5	1,21
R1000-3	19643,35	29176	0	29191	29191	10075,100	0,051	3	1,95
R1000-4	18787,51	57917	0	57987	57987	10851,500	0,121	5	1,73
R1000-5	15992,69	101608	0	101796	101796	11125,800	0,185	5	1,44

Tabela 8.10: Comparação entre os Algoritmos ALB4 e ABB-P-WMTZ+C nas instâncias R de 200 a 1000 nodos.

	ABB-P-WMTZ+C			ALB4					
Instância	t_{ABB}	LS ABB	Gap	LS Inic.	LS LB	t_{LB}	Gap	Iter.	t_{ABB}/t_{LB}
E10-1	0,09	39983	0	43179	39983	2,64	0	4	0,04
E10-2	0,03	36919	0	39701	36919	1,00	0	4	0,03
E10-3	0,09	35208	0	35208	35208	5,66	0	6	0,02
E10-4	0,17	26287	0	28252	26287	1,73	0	4	0,10
E10-5	0,16	41242	0	42631	41242	2,42	0	3	0,06
E20-1	0,61	53456	0	56494	53456	7,89	0	5	0,08
E20-2	0,17	89486	0	89486	89486	12,04	0	6	0,01
E20-3	1,59	60017	0	62793	60017	5,29	0	4	0,30
E20-4	0,81	56543	0	56784	56543	5,18	0	5	0,16
E20-5	1,28	60426	0	60737	60737	3,82	0,515	3	0,33
E40-1	6,41	98041	0	99458	98041	22,23	0	4	0,29
E40-2	7,08	109750	0	111660	109750	29,22	0	3	0,24
E40-3	4,74	129062	0	131866	129068	22,90	0,005	3	0,21
E40-4	5,66	125203	0	128282	125955	32,39	0,601	5	0,17
E40-5	4,09	116579	0	118884	116579	40,84	0	6	0,10
E60-1	11,75	171935	0	175244	171944	34,45	0,005	4	0,34
E60-2	11,36	175895	0	176782	175895	36,21	0	3	0,31
E60-3	7,85	158736	0	158869	158869	20,81	0,084	3	0,38
E60-4	10,62	181475	0	182212	181475	27,02	0	4	0,39
E60-5	20,22	160581	0	161895	161052	26,41	0,293	4	0,77
E80-1	29,89	253121	0	255889	253187	36,71	0,026	4	0,81
E80-2	29,27	289283	0	289563	289312	50,36	0,010	4	0,58
E80-3	61,28	262485	0	265002	262531	48,03	0,018	4	1,28
E80-4	42,79	257653	0	258593	258575	48,20	0,358	4	0,89
E80-5	42,35	271514	0	272509	271549	39,62	0,013	4	1,07
E100-1	78,61	13353	0	13421	13361	113,07	0,060	4	0,70
E100-2	98,70	30933	0	30969	30933	93,96	0	4	1,05
E100-3	166,30	14035	0	14054	14044	85,66	0,064	4	1,94
E100-4	102,98	34434	0	34551	34495	190,69	0,177	6	0,54
E100-5	135,39	24768	0	25192	24773	177,23	0,020	5	0,76
E150-1	340,16	16386	0	16418	16388	145,13	0,012	4	2,34
E150-2	502,74	16589	0	16604	16590	168,75	0,006	4	2,98
E150-3	791,48	17313	0	17341	17313	178,85	0	4	4,43
E150-4	560,61	16725	0	16795	16734	207,46	0,054	5	2,70
E150-5	730,79	15709	0	15796	15710	205,33	0,006	5	3,56

Tabela 8.11: Comparação entre os Algoritmos ALB4 e ABB-P-WMTZ+C nas instâncias E de 10 a 150 nodos.

	ABB-P-WMTZ+C			ALB4					
Instância	t_{ABB}	LS ABB	Gap	LS Inic.	LS LB	t_{LB}	Gap	Iter.	t_{ABB}/t_{LB}
E200-1	1529,04	20425	0	20472	20439	402,12	0,069	6	3,80
E200-2	2801,89	18888	0	19027	18890	241,72	0,011	4	11,59
E200-3	1558,63	20911	0	20978	20925	288,27	0,067	5	5,41
E200-4	1078,38	20177	0	20245	20177	235,56	0	4	4,58
E200-5	2558,41	19592	0	19668	19593	402,56	0,005	6	6,36
E300-1	10040,98	24996	0	25077	24996	875,38	0	4	11,47
E300-2	16316,11	26276	0	26303	26290	1358,79	0,053	6	12,01
E300-3	9070,20	24802	0	24840	24812	1139,88	0,040	5	7,96
E300-4	14695,34	25781	0	25897	25784	1160,76	0,012	5	12,66
E300-5	11720,18	25241	0	25257	25244	1490,32	0,012	6	7,86
E400-1	27871,23	32985	11,128	29723	29682	1447,26	0	5	19,26
E400-2	68388,10	32577	13,933	28536	28536	2513,43	0,126	6	27,21
E400-3	21454,16	37056	21,094	30601	30601	2510,20	0	6	8,55
E400-4	19547,89	33652	12,196	29994	29994	2560,37	0	6	7,63
E400-5	18255,85	33376	9,530	30472	30472	2520,09	0	6	7,24
E500-1	30927,90	41192	24,681	33048	33038	4523,01	0	7	6,84
E500-2	30892,30	NFA	—	33619	33558	3872,91	0	6	7,98
E500-3	31087,70	NFA	—	33823	33823	4706,75	0	6	6,60
E500-4	14224,57	NFA	—	32382	32382	4765,20	0	6	2,99
E500-5	30922,60	NFA	—	34509	34509	4785,43	0	6	6,46
E1000-1	31034,00	NFA	—	49045	49045	10825,90	0	4	2,87
E1000-2	31048,30	NFA	—	47061	47061	10818,20	0	4	2,87
E1000-3	30967,00	NFA	—	48947	48947	10852,80	0	4	2,85
E1000-4	31112,40	NFA	—	48981	48981	10830,30	0	4	2,87
E1000-5	31194,80	NFA	—	48172	48172	10838,10	0	4	2,88

Tabela 8.12: Comparação entre os Algoritmos ALB4 e ABB-P-WMTZ+C nas instâncias E de 200 a 1000 nodos.

Instâncias Quase Caminhos

Nas instâncias Quase Caminhos com o Algoritmo ALB4 são obtidos 94,74% de *gaps* nulos, o que corresponde a 90 instâncias em 95 testadas. Em 7 instâncias, o valor do custo da solução obtida através do Algoritmo ALB4 é de melhor qualidade comparativamente ao valor do custo da solução obtida através do Algoritmo ABB-P-WMTZ+C (os custos dessas instâncias encontram-se a negrito na Tabela 8.8).

Com o Algoritmo ALB4 foi sempre possível encontrar um limite superior para o valor do custo, enquanto que através do Algoritmo ABB-P-WMTZ+C não foi possível, no tempo imposto, obter um limite superior para o valor do custo em quatro instâncias de 1000 nodos (QC1000-1, QC1000-2, QC1000-3 e QC1000-4).

Em 63,16% das instâncias (60 em 95 instâncias) através da aplicação do Algoritmo ALB4 é obtida uma solução de melhor qualidade do que a obtida através da Heurística FPB2. Das 35 instâncias onde não há melhoramento ao aplicar o Algoritmo ALB4, em 32 instâncias usando a Heurística FPB2 já era possível obter a solução ótima.

Os tempos de execução do Algoritmo ALB4 são inferiores aos tempos de execução do Algoritmo ABB-P-WMTZ+C em 77,89% das instâncias (74 instâncias em 95). Os tempos de execução do Algoritmo ALB4 são inferiores na instância QC20-7, em quase todas as instâncias de 40 nodos e em todas as instâncias com mais de 40 nodos.

Instâncias Aleatórias

Nas instâncias Aleatórias com o Algoritmo ALB4 são obtidos 65% de *gaps* nulos, o que corresponde a 39 instâncias em 60 testadas. Em apenas 2 instâncias, o valor do custo da solução obtida através do Algoritmo ALB4 é de melhor qualidade comparativamente ao valor do custo da solução obtida através do Algoritmo ABB-P-WMTZ+C (os custos dessas instâncias encontram-se a negrito na Tabela 8.10).

Com o Algoritmo ALB4 foi sempre possível encontrar um limite superior para o valor do custo, enquanto que através do Algoritmo ABB-P-WMTZ+C não foi possível, no tempo imposto, obter um limite superior para o valor do custo numa instância de 1000 nodos (R1000-1).

As soluções obtidas através da aplicação do Algoritmo ALB4 são de melhor qualidade comparativamente às soluções obtidas através da Heurística FPB2 em 73,33% das instâncias (44 em 60 instâncias). Nas instâncias onde não há melhoramento (17 instâncias) ao aplicar o Algoritmo ALB4, em 4 delas usando a Heurística FPB2 já era

possível obter a solução ótima (R10-2, R400-4, R1000-1 e R1000-2).

Os tempos de execução do Algoritmo ALB4 são inferiores aos tempos de execução do Algoritmo ABB-P-WMTZ+C em 41,67% das instâncias (25 em 60 instâncias). Os tempos de execução do Algoritmo ALB4 são inferiores nas instâncias R10-2, R100-3, R150-5, R200-3 e R200-4 e em todas as instâncias com mais de 200 nodos.

Instâncias Euclidianas

Nas instâncias Euclidianas com o Algoritmo ALB4 são obtidos 53,33% de *gaps* nulos, o que corresponde a 32 instâncias em 60 testadas. Podemos observar que em 15 instâncias, o valor do custo da solução obtida através do Algoritmo ALB4 é de melhor qualidade comparativamente ao valor do custo da solução obtida através do Algoritmo ABB-P-WMTZ+C (os custos dessas instâncias encontram-se a negrito na Tabela 8.12).

Com o Algoritmo ALB4 foi sempre possível encontrar um limite superior para o valor do custo, enquanto que através do Algoritmo ABB-P-WMTZ+C não foi possível, no tempo imposto, obter um limite superior para o valor do custo em 9 instâncias, sendo estas de 500 e 1000 nodos. Em todas as instâncias de 400 nodos e numa de 500 nodos os limites superiores obtidos através do Algoritmo ABB-P-WMTZ+C não são de boa qualidade dado que os *gaps* obtidos são bastante elevados.

As soluções obtidas através da aplicação do Algoritmo ALB4 são de melhor qualidade comparativamente às soluções obtidas através da Heurística FPB2 em 71,67% das instâncias (43 em 60 instâncias). Nas instâncias onde não há melhoramento (17 instâncias), ao aplicar o Algoritmo ALB4, em 12 delas usando a Heurística FPB2 já era possível obter a solução ótima.

Os tempos de execução do Algoritmo ALB4 são inferiores aos tempos de execução do Algoritmo ABB-P-WMTZ+C em 56,67% das instâncias (34 em 60 instâncias). Os tempos de execução do Algoritmo ALB4 são inferiores nas instâncias E80-3, R80-5, R100-2 e R100-3 e em todas as instâncias com mais de 100 nodos.

8.5.4 Síntese dos Resultados Computacionais

De todos os algoritmos *Local Branching* propostos verificámos através da Subsecção 8.5.2 que em todos os grupos de instâncias até 100 nodos, o Algoritmo ALB4 se mostrou

o mais eficiente.

Em 61,86% das instâncias (133 em 215 instâncias), os tempos obtidos através da utilização do Algoritmo ALB4 são inferiores aos tempos obtidos através do Algoritmo ABB-P-WMTZ+C. Verificámos também que para instâncias Quase Caminhos com mais do que 40 nodos, instâncias Aleatórias com mais do que 200 nodos e instâncias Euclideanas com mais do que 100 nodos, o Algoritmo ALB4 é mais rápido do que o Algoritmo ABB-P-WMTZ+C e as soluções encontradas são de boa qualidade, pois são obtidos 75,34% de *gaps* nulos (162 em 215 instâncias). Com a aplicação do Algoritmo ALB4 obtêm-se melhoramentos em 68,37% das instâncias (147 em 215 instâncias) comparativamente à aplicação da Heurística FPB2.

Capítulo 9

Considerações Finais

Ao longo desta tese abordámos vários métodos para resolver o Problema WMST, entre os quais o método *Branch and Bound* e *Branch and Cut* através do uso de formulações e procedimentos de introdução de cortes usando separação, os métodos de relaxação linear e Lagrangeana, métodos de geração e introdução de desigualdades válidas nas formulações usando separação e por fim os métodos *Feasibility Pump* e *Local Branching*.

No Capítulo 3 começámos por apresentar e comparar do ponto vista computacional várias formulações para o Problema WMST. Das várias comparações efetuadas resulta que, para os três grupos de instâncias em estudo: Quase Caminhos, Aleatórias e Euclidianas, o procedimento com introdução de cortes, que se aplica para fortalecer a formulação baseada nas desigualdades WMTZ (Procedimento P-WMTZ+C), é o que apresenta melhores resultados na obtenção da solução ótima ou de uma solução admissível para o valor ótimo. Para todas as formulações apresentadas no Capítulo 3, obtiveram-se também os valores da relaxação linear. Os limites inferiores de melhor qualidade são os obtidos através da relaxação linear do Procedimento P-WMTZ+C.

Uma outra forma de encontrar limites inferiores e superiores para o valor ótimo do Problema WMST, foi apresentada no Capítulo 4 com a implementação de algoritmos baseados em relaxação Lagrangeana. Os Algoritmos Alg2, Alg5, Alg6 e Alg7 mostraram-se bastante eficientes em todos os grupos de instâncias, pois com a aplicação destes algoritmos foi possível obter, rapidamente, algumas soluções ótimas e outras muito próximas das ótimas. Podemos dizer que os valores obtidos pela relaxação Lagrangeana dos Algoritmos Alg2, Alg5 e Alg6, em geral, são iguais aos valores obtidos

pela relaxação linear do Procedimento P-WMTZ+C apresentado no Capítulo 3. O Algoritmo Alg7, onde são incluídas desigualdades válidas é o que apresenta melhores limites inferiores.

No Capítulo 5 foram apresentadas apenas classes de desigualdades válidas e no Capítulo 6 foram apresentados algoritmos heurísticos de separação para gerar e introduzir as desigualdades válidas do Capítulo 5 nas formulações apresentadas no Capítulo 3 de modo a fortalecê-las. De experiências computacionais realizadas com os vários algoritmos heurísticos de separação, num grupo de instâncias teste, resulta que os algoritmos que utilizam a introdução das Desigualdades Implícita Levantada e das Desigualdades Implícita Levantada por *Down-Lifting* são os que apresentam os melhores resultados. Verificou-se ainda que ao testar instâncias de maior dimensão o procedimento de geração e introdução de Desigualdades Implícita Levantada por *Down-Lifting* é o melhor para utilização na prática.

Com o objetivo de obter soluções inteiras admissíveis para o Problema WMST apresentámos no Capítulos 7 heurísticas *Feasibility Pump*. Nas instâncias Aleatórias e Euclidianas, a Heurística FPB2 é a que encontra soluções admissíveis de melhor qualidade e mais rapidamente quando comparada com as restantes Heurísticas *Feasibility Pump* propostas e também quando comparada com a Heurística do *Xpress* aplicada ao nodo raiz utilizando a Formulação WMTZ (HXNR) e com a primeira solução inteira admissível obtida pelo *solver* do *software Xpress* através da Formulação WMTZ, quando usa o procedimento *Branch and Bound* (PSIABB). No caso das instâncias Quase Caminhos, a mais rápida é a Heurística HXNR, mas em termos de qualidade das soluções encontradas, esta não apresenta bons resultados quando comparada com a Heurística FPB2.

Um outro método heurístico tratado no Capítulo 8 e bastante eficiente para instâncias de maiores dimensões é baseado no Método *Local Branching*. De todos os algoritmos *Local Branching* propostos verificámos que em todos os grupos de instâncias, o Algoritmo que se mostrou o mais eficiente foi o ALB4. Este algoritmo utiliza como solução de referência inicial a solução inteira admissível obtida através da Heurística FPB2. Quando comparamos o Algoritmo ALB4 com o Algoritmo *Branch and Bound* do procedimento de introdução de cortes P-WMTZ+C verificamos que em mais de metade das instâncias (133 em 215 instâncias) os tempos obtidos através da utilização

do Algoritmo ALB4 são inferiores aos tempos obtidos através do Algoritmo *Branch and Bound* do procedimento de introdução de cortes P-WMTZ+C. Como a Heurística *Local Branching* é uma heurística de melhoramento, podemos observar que com a aplicação do Algoritmo ALB4 ocorrem melhoramentos em mais de metade das instâncias (147 em 215 instâncias) comparativamente à aplicação da Heurística FPB2.

É de observar que os métodos *Feasibility Pump* e *Local Branching* e os algoritmos Lagrangeanos são normalmente mais rápidos do que o uso de métodos *Branch and Bound* através das formulações e os métodos de planos de corte e daí mais adequados para resolver problemas de instâncias maiores, no entanto não garantem a obtenção da solução ótima e a qualidade das soluções, em geral, não é tão boa.

Em trabalhos futuros tencionamos desenvolver algoritmos heurísticos para obtenção dos coeficientes de levantamento usual, *down-lifting* e *up-lifting* mais rápidos, de modo a tornar os algoritmos heurísticos de separação mais rápidos. Muitos outros métodos de resolução do Problema WMST ficaram por aplicar e poderão ser aplicados futuramente.

Anexos

Facetas do conjunto P_{WMST} para uma instância de 5 nodos

Neste anexo apresentamos todas as facetas de P_{WMST} obtidas através do *software* PORTA para a instância de 5 nodos dada no Exemplo 2.1. Para cada uma das desigualdades também indicamos uma possível forma de obtenção.

$$(1) \quad x_{01} + x_{02} + x_{03} + x_{04} + x_{12} + x_{13} + x_{14} + x_{23} + x_{24} + x_{34} = 4$$

Restrição de cardinalidade.

$$(2) \quad -2x_{02} - x_{04} - x_{12} - 2x_{13} - 2x_{14} - 2x_{23} - x_{24} - x_{34} \leq -4$$

DCILUL (Fixação a zero).

Equivalente a: $x_{01} - x_{02} + x_{03} - x_{13} - x_{14} - x_{23} \leq 0$.

Escolha das variáveis a fixar: $x_{02} = x_{13} = x_{14} = x_{23} = 0$.

Estas arestas formam a árvore de suporte de peso mínimo.

DCI: $x_{01} \leq 0$.

Corta o PL. Valor de $PL = 25.5$.

$$(3) \quad -2x_{02} + x_{03} - 3x_{13} - 3x_{14} - 2x_{23} - x_{24} - x_{34} \leq -4$$

DGCIL (Fixação a zero e a um).

Escolha das variáveis a fixar: $x_{14} = x_{24} = 1$ e $x_{02} = x_{13} = x_{23} = x_{34} = 0$.

DCI: $x_{03} \leq 0$.

Corta o PL. Valor de $PL = 27 = OPT$.

$$(4) \quad -x_{02} - x_{03} - x_{04} - x_{12} - x_{13} - x_{14} - x_{23} - x_{24} - x_{34} \leq -3$$

Equivalente a: $x_{01} \leq 1$.

$$(5) \quad -x_{02} \quad -x_{04} \quad -x_{12} \quad -x_{13} \quad -2x_{14} \quad -x_{23} \quad -x_{24} \quad -x_{34} \leq -3$$

DCILUL (Fixação a zero).

Equivalente a: $x_{01} + x_{03} - x_{14} \leq 1$.

Escolha das variáveis a fixar: $x_{14} = 0$.

Esta aresta pertence à árvore de suporte de peso mínimo.

DCI: $x_{01} + x_{03} \leq 1$.

$$(6) \quad -x_{02} \quad -x_{04} \quad -2x_{13} \quad -2x_{14} \quad -x_{23} \quad -x_{24} \quad -x_{34} \leq -3$$

DCILUL (Fixação a zero).

Equivalente a: $x_{01} + x_{03} + x_{12} - x_{13} - x_{14} \leq 1$.

Escolha das variáveis a fixar: $x_{13} = x_{14} = 0$.

Estas arestas pertencem à árvore de suporte de peso mínimo.

DCI: $x_{01} + x_{03} \leq 1$.

Corta o PL. Valor de $PL = 25.5$.

$$(7) \quad -2x_{02} \quad -x_{04} \quad -x_{12} \quad -x_{13} \quad -x_{14} \quad -2x_{23} \quad -x_{24} \quad -x_{34} \leq -3$$

DCILUL (Fixação a zero).

Equivalente a: $x_{01} - x_{02} + x_{03} - x_{23} \leq 1$.

Escolha das variáveis a fixar: $x_{02} = x_{23} = 0$.

Estas arestas pertencem à árvore de suporte de peso mínimo.

DCI: $x_{01} + x_{03} \leq 1$.

$$(8) \quad -x_{02} \quad -2x_{13} \quad -2x_{14} \quad -2x_{23} \quad -x_{24} \quad -x_{34} \leq -3$$

DCILUL (Fixação a zero).

Equivalente a: $x_{01} + x_{03} + x_{04} + x_{12} - x_{13} - x_{14} - x_{23} \leq 1$.

Escolha das variáveis a fixar: $x_{13} = x_{14} = x_{23} = 0$.

Estas arestas pertencem todas à árvore de suporte de peso mínimo.

DCI: $x_{01} + x_{03} \leq 1$.

Corta o PL. O valor do $PL = 27 = OPT$.

$$(9) \quad -2x_{02} \quad +x_{03} \quad -2x_{13} \quad -2x_{14} \quad -2x_{23} \quad -x_{24} \quad -x_{34} \leq -3$$

DGCIL (Fixação a zero e a um).

Escolha das variáveis a fixar: $x_{14} = x_{24} = 1$ e $x_{02} = x_{13} = x_{23} = x_{34} = 0$.

DCI: $x_{03} \leq 0$.

Corta o PL. Valor de $PL = 27 = OPT$.

Note-se que nas desigualdades (2) e (8) se escolhem as mesmas variáveis para

se fixar a zero e a um, mas a ordem do cálculo dos coeficientes é feita de forma diferente.

$$(10) \quad -x_{02} + x_{03} - x_{04} - 2x_{13} - 3x_{14} - x_{23} - x_{24} - x_{34} \leq -3$$

DGCIL (Fixação a zero a um).

Equivalente a: $x_{01} + 2x_{03} + x_{12} - x_{13} - 2x_{14} \leq 1$.

Escolha das variáveis a fixar: $x_{12} = 1$ e $x_{03} = x_{13} = x_{14} = 0$.

DCI: $x_{01} \leq 0$.

Corta o PL. Valor de $PL = 25.5$.

$$(11) \quad -2x_{02} + x_{03} - 2x_{13} - 3x_{14} - x_{23} - x_{24} - x_{34} \leq -3$$

DGCIL (Fixação a zero e a um).

Escolha das variáveis a fixar: $x_{13} = x_{34} = 1$ e $x_{02} = x_{14} = x_{23} = x_{24} = 0$.

DCI: $x_{03} \leq 0$.

Corta o PL. Valor de $PL = 25.8$.

$$(12) \quad -x_{02} - x_{12} - x_{13} - x_{14} - x_{23} - x_{24} \leq -2$$

DCIE.

Equivalente a: $x_{01} + x_{03} + x_{04} + x_{34} \leq 2$.

DCI: $x_{01} + x_{04} + x_{34} \leq 2$.

Corta o PL. Valor de $PL = 25.5$.

$$(13) \quad -x_{02} - x_{13} - x_{14} - x_{23} - x_{24} - x_{34} \leq -2$$

DCIL.

Equivalente a: $x_{01} + x_{03} + x_{04} + x_{12} \leq 2$.

DCI: $x_{01} + x_{04} + x_{12} \leq 2$.

Corta o PL. Valor de $PL = 27 = OPT$.

$$(14) \quad -x_{02} - x_{03} - x_{12} - x_{13} - x_{23} - x_{24} - x_{34} \leq -2$$

Restrição de eliminação de subcircuito / Clique.

Equivalente a: $x_{01} + x_{04} + x_{14} \leq 2$.

$$(15) \quad -x_{02} - x_{04} - x_{12} - x_{13} - x_{14} - x_{23} - x_{34} \leq -2$$

DCI.

Equivalente a: $x_{01} + x_{03} + x_{24} \leq 2$.

$$(16) \quad -x_{02} \quad -x_{04} \quad -x_{12} \quad -x_{14} \quad -x_{23} \quad -x_{24} \quad -x_{34} \leq -2$$

Restrição de eliminação de subcircuito / Clique.

Equivalente a: $x_{01} + x_{03} + x_{13} \leq 2$.

$$(17) \quad -x_{03} \quad -x_{04} \quad -x_{13} \quad -x_{14} \quad -x_{23} \quad -x_{24} \quad -x_{34} \leq -2$$

Restrição de eliminação de subcircuito / Clique.

Equivalente a: $x_{01} + x_{02} + x_{12} \leq 2$.

$$(18) \quad -x_{02} \quad -2x_{13} \quad -x_{14} \quad -x_{23} \quad -x_{34} \leq -2$$

DCILUL (Fixação a zero).

Equivalente a: $x_{01} + x_{03} + x_{04} + x_{12} - x_{13} + x_{24} \leq 2$.

Escolha das variáveis a fixar: $x_{13} = 0$.

DCI: $x_{01} + x_{03} + x_{04} \leq 2$.

$$(19) \quad -x_{04} \quad -x_{13} \quad -2x_{14} \quad -x_{23} \quad -x_{24} \quad -x_{34} \leq -2$$

DGCIL (Fixação a zero e a um).

Equivalente a: $x_{01} + x_{02} + x_{03} + x_{12} - x_{14} \leq 2$.

Escolha das variáveis a fixar: $x_{12} = x_{02} = 1$ e $x_{14} = 0$.

DCI: $x_{03} \leq 0$.

$$(20) \quad -x_{02} \quad +x_{03} \quad -x_{04} \quad -x_{13} \quad -2x_{14} \quad -x_{23} \quad -x_{24} \quad -x_{34} \leq -2$$

DGCIL (Fixação a zero e a um).

Escolha das variáveis a fixar: $x_{14} = 1$ e $x_{02} = x_{04} = x_{13} = x_{23} = x_{24} = x_{34} = 0$.

DCI: $x_{03} \leq 0$.

$$(21) \quad -x_{02} \quad +x_{03} \quad -2x_{13} \quad -2x_{14} \quad -x_{23} \quad -x_{24} \leq -2$$

DGCIL (Fixação a zero e a um).

Escolha das variáveis a fixar: $x_{14} = 1$ e $x_{02} = x_{13} = x_{23} = x_{24} = 0$.

DCI: $x_{03} \leq 0$.

Corta o PL. Valor de $PL = 25.5$.

$$(22) \quad -x_{02} \quad +x_{03} \quad -2x_{13} \quad -2x_{14} \quad -x_{23} \quad -x_{34} \leq -2$$

DCILDL (Fixação a zero e a um).

Escolha das variáveis a fixar: $x_{14} = 1$ e $x_{02} = x_{13} = x_{23} = x_{34} = 0$.

DCI: $x_{03} \leq 0$.

$$(23) \quad -2x_{02} + x_{03} - x_{13} - 2x_{14} - x_{23} - x_{24} - x_{34} \leq -2$$

DCILUL (Fixação a zero).

Equivalente a: $x_{01} - x_{02} + 2x_{03} + x_{04} + x_{12} - x_{14} \leq 2$.

Escolha das variáveis a fixar: $x_{02} = x_{03} = x_{14} = 0$.

Apenas as arestas $\{0, 2\}$ e $\{1, 4\}$ pertencem à árvore de suporte de peso mínimo, a aresta $\{0, 3\}$ não pertence, mas pertence à solução admissível considerada inicialmente.

DCI: $x_{01} + x_{04} + x_{12} \leq 2$.

Corta PL. O valor do $PL = 25.6$.

$$(24) \quad -x_{02} - x_{13} - x_{14} \leq -1$$

DCE.

Equivalente a: $x_{01} + x_{03} + x_{04} + x_{12} + x_{23} + x_{24} + x_{34} \leq 3$.

DC: $x_{01} + x_{12} + x_{23} + x_{34} \leq 3$.

Corta PL. Valor de $PL = 25.5$.

$$(25) \quad -x_{02} - x_{12} - x_{23} - x_{24} \leq -1$$

Restrição de eliminação de subcircuito / Corte do 2 / Clique.

Equivalente a: $x_{01} + x_{03} + x_{04} + x_{13} + x_{14} + x_{34} \leq 3$.

$$(26) \quad -x_{02} - x_{13} - x_{23} - x_{34} \leq -1$$

DCE.

Equivalente a: $x_{01} + x_{03} + x_{04} + x_{12} + x_{14} + x_{24} \leq 3$.

DC: $x_{01} + x_{03} + x_{14} + x_{24} \leq 3$.

$$(27) \quad -x_{03} - x_{13} - x_{23} - x_{34} \leq -1$$

Restrição de eliminação de subcircuito / Corte do 3 / Clique.

Equivalente a: $x_{01} + x_{02} + x_{04} + x_{12} + x_{14} + x_{24} \leq 3$.

$$(28) \quad -x_{04} - x_{14} - x_{24} - x_{34} \leq -1$$

Restrição de eliminação de subcircuito / Corte do 4 / Clique.

Equivalente a: $x_{01} + x_{02} + x_{03} + x_{12} + x_{13} + x_{23} \leq 3$.

$$(29) \quad -x_{13} - x_{14} - x_{23} - x_{24} \leq -1$$

DCE.

Equivalente a: $x_{01} + x_{02} + x_{03} + x_{04} + x_{12} + x_{34} \leq 3$.

DC: $x_{01} + x_{02} + x_{04} + x_{34} \leq 3$.

Corta PL. Valor de $PL = 25.5$.

$$(30) \quad -x_{13} \quad -x_{14} \quad -x_{23} \quad -x_{34} \leq -1$$

DCE.

Equivalente a: $x_{01} + x_{02} + x_{03} + x_{04} + x_{12} + x_{24} \leq 3$.

DC: $x_{01} + x_{02} + x_{03} + x_{24} \leq 3$.

$$(31) \quad -x_{02} \quad +x_{03} \quad -x_{13} \quad -x_{14} \quad -x_{23} \quad -x_{24} \leq -1$$

Desigualdade de Cobertura Levantada.

Equivalente a: $x_{01} + 2x_{03} + x_{04} + x_{12} + x_{34} \leq 3$.

DC: $x_{01} + x_{04} + x_{12} + x_{34} \leq 3$.

Corta PL. Valor de $PL = 25.5$.

$$(32) \quad -x_{02} \quad +x_{03} \quad -x_{13} \quad -x_{14} \quad -x_{23} \quad -x_{34} \leq -1$$

DGCIL (Fixação a zero e a um).

Escolha das variáveis a fixar: $x_{14} = 1$ e $x_{02} = x_{13} = x_{23} = x_{34} = 0$.

DCI: $x_{03} \leq 0$.

$$(33) \quad -x_{02} \quad +x_{03} \quad +x_{04} \quad -2x_{13} \quad -2x_{14} \quad -x_{23} \leq -1$$

DGCIL (Fixação a zero e a um).

Escolha das variáveis a fixar: $x_{13} = x_{04} = 1$ e $x_{02} = x_{14} = x_{23} = 0$.

DCI: $x_{03} \leq 0$.

$$(34) \quad -x_{02} \leq 0$$

Equivalente a: $x_{02} \geq 0$.

$$(35) \quad -x_{03} \leq 0$$

Equivalente a: $x_{03} \geq 0$.

$$(36) \quad -x_{04} \leq 0$$

Equivalente a: $x_{04} \geq 0$.

$$(37) \quad -x_{12} \leq 0$$

Equivalente a: $x_{12} \geq 0$.

$$(38) \quad -x_{13} \leq 0$$

Equivalente a: $x_{13} \geq 0$.

$$(39) \quad -x_{14} \leq 0$$

Equivalente a: $x_{14} \geq 0$.

$$(40) \quad -x_{23} \leq 0$$

Equivalente a: $x_{23} \geq 0$.

$$(41) \quad -x_{24} \leq 0$$

Equivalente a: $x_{24} \geq 0$.

$$(42) \quad -x_{34} \leq 0$$

Equivalente a: $x_{34} \geq 0$.

$$(43) \quad +x_{03} \quad -x_{13} \quad -x_{14} \leq 0$$

DCILUL (Fixação a zero).

Escolha das variáveis a fixar: $x_{13} = x_{14} = 0$.

Estas arestas pertencem à árvore de suporte de peso mínimo.

DCI: $x_{03} \leq 0$.

$$(44) \quad +x_{04} \quad -x_{13} \quad -x_{14} \quad -x_{23} \leq 0$$

DCILUL (Fixação a zero).

Escolha das variáveis a fixar: $x_{13} = x_{14} = x_{23} = 0$.

Estas arestas pertencem à árvore de suporte de peso mínimo.

DCI: $x_{04} \leq 0$.

$$(45) \quad -x_{02} \quad +x_{03} \quad +x_{04} \quad -x_{13} \quad -x_{14} \leq 0$$

DCILUL (Fixação a zero).

Escolha das variáveis a fixar: $x_{02} = x_{13} = x_{14} = 0$.

Estas arestas pertencem à árvore de suporte de peso mínimo.

DCI: $x_{04} \leq 0$.

$$(46) \quad -x_{02} \quad +x_{03} \quad +x_{12} \quad -x_{13} \quad -x_{14} \quad -x_{23} \leq 0$$

DCILUL (Fixação a zero).

Escolha das variáveis a fixar: $x_{02} = x_{13} = x_{14} = x_{23} = 0$.

Estas arestas pertencem à árvore de suporte de peso mínimo.

DCI: $x_{12} \leq 0$.

Corta PL. Valor de $PL = 25.5$.

$$(47) \quad -x_{02} + x_{03} + x_{12} - 2x_{14} - x_{24} - x_{34} \leq 0$$

DCILUL (Fixação a zero).

Escolha das variáveis a fixar: $x_{02} = x_{14} = x_{24} = x_{34} = 0$.

As arestas $\{0, 2\}$ e $\{1, 4\}$ pertencem à árvore de suporte de peso mínimo, mas as arestas $\{2, 4\}$ e $\{3, 4\}$ não pertencem.

DCI: $x_{12} \leq 0$.

$$(48) \quad +x_{34} \leq 1$$

$$(49) \quad +x_{24} \leq 1$$

$$(50) \quad +x_{23} \leq 1$$

$$(51) \quad +x_{14} \leq 1$$

$$(52) \quad +x_{13} \leq 1$$

$$(53) \quad +x_{12} \leq 1$$

$$(54) \quad +x_{04} \leq 1$$

$$(55) \quad +x_{03} \leq 1$$

$$(56) \quad +x_{02} \leq 1$$

$$(57) \quad -x_{02} + x_{03} + x_{04} \leq 1$$

DCILUL (Fixação a zero).

Escolha das variáveis a fixar: $x_{02} = 0$.

Esta aresta pertence à árvore de suporte de peso mínimo.

DCI: $x_{03} + x_{04} \leq 1$.

$$(58) \quad +x_{03} + x_{12} - x_{14} \leq 1$$

DCILUL (Fixação a zero).

Escolha das variáveis a fixar: $x_{14} = 0$.

Esta aresta pertence à árvore de suporte de peso mínimo.

DCI: $x_{03} + x_{12} \leq 1$.

$$(59) \quad +x_{03} \quad +x_{04} \quad -x_{13} \quad -x_{14} \quad +x_{34} \leq -3$$

DCILUL (Fixação a zero).

Escolha das variáveis a fixar: $x_{13} = x_{14} = x_{34} = 0$.

As arestas $\{1, 3\}$ e $\{2, 4\}$ pertencem à árvore de suporte de peso mínimo.

DCI: $x_{04} + x_{24} \leq 1$.

$$(60) \quad +x_{03} \quad +x_{04} \quad -x_{13} \quad -x_{14} \quad +x_{24} \leq 1$$

DCILUL (Fixação a zero).

Escolha das variáveis a fixar: $x_{13} = x_{14} = 0$.

Estas duas arestas pertencem à árvore de suporte de peso mínimo.

DCI: $x_{04} + x_{24} \leq 1$.

$$(61) \quad +x_{23} \quad +x_{24} \quad +x_{34} \leq 2$$

Restrição de eliminação de subcircuito / Clique.

$$(62) \quad +x_{13} \quad +x_{14} \quad +x_{34} \leq 2$$

Restrição de eliminação de subcircuito / Clique.

$$(63) \quad +x_{12} \quad +x_{14} \quad +x_{24} \leq 2$$

Restrição de eliminação de subcircuito / Clique.

$$(64) \quad +x_{12} \quad +x_{13} \quad +x_{23} \leq 2$$

Restrição de eliminação de subcircuito / Clique.

$$(65) \quad +x_{03} \quad +x_{12} \quad +x_{34} \leq 2$$

DCI.

$$(66) \quad +x_{03} \quad +x_{12} \quad +x_{24} \leq 2$$

DCI.

$$(67) \quad +x_{03} \quad +x_{04} \quad +x_{24} \leq 2$$

DCI.

$$(68) \quad +x_{03} \quad +x_{04} \quad +x_{23} \leq 2$$

DCI.

$$(69) \quad +x_{02} \quad +x_{04} \quad +x_{24} \leq 2$$

Restrição de eliminação de subcircuito / Clique.

$$(70) \quad +x_{02} +x_{03} \quad \quad +x_{23} \quad \leq 2$$

Restrição de eliminação de subcircuito / Clique.

$$(71) \quad +x_{03} +x_{04} +x_{12} +x_{13} \quad +x_{23} \quad \leq 3$$

DCE.

$$\text{DC: } x_{03} + x_{04} + x_{13} + x_{23} \leq 3.$$

$$(72) \quad +x_{12} +x_{13} +x_{14} +x_{23} +x_{24} +x_{34} \leq 3$$

Restrição de eliminação de subcircuito / Corte do nodo 0 / Clique.

$$(73) \quad +x_{02} +x_{03} +x_{04} \quad \quad +x_{23} +x_{24} +x_{34} \leq 3$$

Restrição de eliminação de subcircuito / Corte do nodo 1 / Clique.

$$(74) \quad +x_{02} +x_{03} +x_{04} +x_{12} +x_{13} +x_{14} +x_{23} +x_{24} +x_{34} \leq 4$$

Esta desigualdade é equivalente à seguinte: $x_{01} \geq 0$.

Notação Utilizada

$G = (V, E) \rightsquigarrow$ Grafo completo não orientado.

$V \rightsquigarrow$ Conjunto de nodos do grafo não orientado G , $V = \{0, 1, \dots, n-1\}$.

$E \rightsquigarrow$ Conjunto de arestas do grafo não orientado G , $E = \{e = \{i, j\}, i, j \in V, i \neq j\}$.

$c_e \rightsquigarrow$ Custo associado à aresta $e \in E$.

$w_e \rightsquigarrow$ Peso associado à aresta $e \in E$.

$E_T \rightsquigarrow$ Conjunto de arestas de $T = (V, E_T)$ árvore de suporte no grafo G .

$C(T) \rightsquigarrow$ Custo da árvore de suporte T .

$W(T) \rightsquigarrow$ Peso da árvore de suporte T .

$W \rightsquigarrow$ Valor estabelecido para o limite de peso da restrição saco-mochila.

$\vartheta(WMST) \rightsquigarrow$ Valor ótimo do Problema WMST.

$H \rightsquigarrow$ Matriz de pesos.

$C \rightsquigarrow$ Matriz de custos.

$T_c \rightsquigarrow$ Árvore de suporte de custo mínimo.

$T_w \rightsquigarrow$ Árvore de suporte de peso mínimo.

$T_p \rightsquigarrow$ Árvore de suporte ponderada mínima.

$G = (V, A) \rightsquigarrow$ Grafo completo orientado.

$A \rightsquigarrow$ Conjunto de arcos do grafo orientado G , $A = \{(i, j), i \in V, j \in V \setminus \{0\}, i \neq j\}$.

$c_{ij} \rightsquigarrow$ Custo associado ao arco $(i, j) \in A$.

$w_{ij} \rightsquigarrow$ Peso associado ao arco $(i, j) \in A$.

$p_{ij} \rightsquigarrow$ Valor ponderado associado ao arco $(i, j) \in A$, sendo combinação linear de c_{ij} e w_{ij} .

$P(T) \rightsquigarrow$ Valor ponderado da árvore de suporte T .

$A(S_1, S_2) \rightsquigarrow$ Conjunto de arcos com um extremo em S_1 e outro extremo em S_2 .

$X_{WMST} \rightsquigarrow$ Conjunto das soluções admissíveis do Problema WMST.

$X_L \rightsquigarrow$ Conjunto das soluções admissíveis da relaxação linear do Problema WMST.

$X_T \rightsquigarrow$ Conjunto das soluções admissíveis do Problema MST.

$X_K \rightsquigarrow$ Conjunto das soluções admissíveis do problema saco-mochila binário.

$d_k \rightsquigarrow$ Direção na iteração k .

$s_k \rightsquigarrow$ Tamanho do passo na iteração k .

λ e $\nu \rightsquigarrow$ Multiplicadores de Lagrange.

$tol \rightsquigarrow$ Tolerância usada no critério de paragem dos algoritmos Lagrangeanos.

LI \rightsquigarrow Limite inferior.

LS \rightsquigarrow Limite superior.

$P_{WMST} \rightsquigarrow$ Invólucro convexo do conjunto X_{WMST} .

$E_0 \rightsquigarrow$ Conjunto de variáveis fixas a zero.

$E_1 \rightsquigarrow$ Conjunto de variáveis fixas a um.

$\tilde{c}_e \rightsquigarrow$ Custo modificado associado à aresta $e \in E$.

TL \rightsquigarrow Tempo limite imposto.

$maxIter \rightsquigarrow$ Número máximo de iterações a realizar na Heurística *Feasibility Pump*.

$\tilde{x} \rightsquigarrow$ Solução obtida por arredondamento.

$\Delta(x, \tilde{x}) \rightsquigarrow$ Distância entre os pontos x e \tilde{x} .

$\bar{x} \rightsquigarrow$ Solução de referência.

$\bar{S} \rightsquigarrow$ Suporte binário de \bar{x} .

$\mathcal{N}(\bar{x}, k) \rightsquigarrow$ Vizinhança k -OPT de \bar{x} , onde k é o número de variáveis binárias que podem trocar de valor.

$\Delta(x, \bar{x}) \leq k \rightsquigarrow$ Restrição *Local Branching*, onde k é o número de variáveis binárias que podem trocar de valor.

$\lfloor p \rfloor \rightsquigarrow$ Converte o número p no maior número inteiro menor ou igual a p .

$\lceil p \rceil \rightsquigarrow$ Converte o número p no menor número inteiro maior ou igual a p .

$\lfloor p \rfloor \rightsquigarrow$ Converte o número p no valor inteiro mais próximo de p .

$\sigma_{gap}, \sigma_{tempo} \rightsquigarrow$ Desvio padrão do gap e do tempo, respetivamente.

$OPT \rightsquigarrow$ Valor ótimo ($\vartheta(WMST)$) ou um valor do limite superior para o valor do custo.

Siglas Utilizadas

Siglas em Geral

- E** \rightsquigarrow Grupo de instâncias Euclidianas. Pag. 15.
- MIP** \rightsquigarrow Programação Linear Inteira Mista. Pag. 241.
- MST** \rightsquigarrow Árvore de Suporte de Custo Mínimo. Pag. 7.
- MST_P** \rightsquigarrow Árvore de Suporte Ponderada Mínima. Pag. 13.
- MST_W** \rightsquigarrow Árvore de Suporte de Peso Mínimo. Pag. 11.
- NFA** \rightsquigarrow Limite superior nunca foi atualizado. Pag. 308.
- PLI** \rightsquigarrow Programação Linear Inteira. Pag. 19.
- QC** \rightsquigarrow Grupo de instâncias Quase Caminhos. Pag. 16.
- R** \rightsquigarrow Grupo de instâncias Aleatórias. Pag. 16.
- TSP** \rightsquigarrow Problema do Caixeiro Viajante. Pag. 27.
- WMST** \rightsquigarrow Problema da Árvore de Suporte de Custo Mínimo com Restrições de Peso. Pag. 1.

Formulações para o Problema WMST

- CS** \rightsquigarrow Formulação de Cortes. Pag. 23.
- CS_L** \rightsquigarrow Relaxação linear da Formulação de Cortes. Pag. 23.
- ES** \rightsquigarrow Formulação de Eliminação de subcircuitos. Pag. 22.

ES_L \rightsquigarrow Relaxação linear da Formulação de Eliminação de subcircuitos. Pag. 23.

MF \rightsquigarrow Formulação de Fluxos. Pag. 24.

MF_L \rightsquigarrow Relaxação linear da Formulação de Fluxos. Pag. 26.

MTZ \rightsquigarrow Formulação Miller-Tucker-Zemlin. Pag. 27.

MTZ_L \rightsquigarrow Relaxação linear da Formulação Miller-Tucker-Zemlin. Pag. 28.

MTZli \rightsquigarrow Formulação derivada da MTZ com a restrição (3.22) substituída pela Desigualdade Levantada i , $i = 1, 2, 3, 4$. Pag. 29.

MTZli_L \rightsquigarrow Relaxação linear da Formulação derivada da MTZ com a restrição (3.22) substituída pela Desigualdade Levantada i , $i = 1, 2, 3, 4$. Pag. 31.

P-CS \rightsquigarrow Procedimento de Introdução de Cortes usando a Formulação CS. Pag. 42.

P-CS_L \rightsquigarrow Relaxação linear do Procedimento de Introdução de Cortes usando a Formulação CS. Pag. 42.

P-MTZ+C \rightsquigarrow Procedimento de Introdução de Cortes usando a Formulação MTZ. Pag. 42.

P-MTZ+C_L \rightsquigarrow Relaxação linear do Procedimento de Introdução de Cortes usando a Formulação MTZ. Pag. 42.

P-WMTZ+C \rightsquigarrow Procedimento de Introdução de Cortes usando a Formulação WMTZ. Pag. 42.

P-WMTZ+C_L \rightsquigarrow Relaxação linear do Procedimento de Introdução de Cortes usando a Formulação WMTZ. Pag. 42.

WE \rightsquigarrow Formulação de Fluxo com Especificação de Pesos. Pag. 36.

WE_L \rightsquigarrow Relaxação linear da Formulação de Fluxo com Especificação de Pesos. Pag. 38.

WMST _{λ} \rightsquigarrow Formulação genérica do Problema WMST com a restrição de peso relaxada. Pag. 75.

WMST-C _{$\lambda\nu$} \rightsquigarrow Formulação genérica do Problema WMST com a restrição de peso e uma desigualdade válida relaxadas. Pag. 98.

WMTZ \rightsquigarrow Formulação de Peso Miller-Tucker-Zemlin. Pag. 31.

WMTZ_L \rightsquigarrow Relaxação linear da Formulação de Peso Miller-Tucker-Zemlin. Pag. 33.

WMTZli \rightsquigarrow Formulação derivada da WMTZ com a restrição (3.33) substituída pela Desigualdade Levantada i , $i = 1, 2, 3, 4$. Pag. 34.

WMTZli_L \rightsquigarrow Relaxação linear da Formulação derivada da WMTZ com a restrição (3.33) substituída pela Desigualdade Levantada i , $i = 1, 2, 3, 4$. Pag. 35.

Desigualdades para o Problema WMST

CI \rightsquigarrow Cobertura Implícita. Pag. 138.

DC \rightsquigarrow Desigualdade de Cobertura. Pag. 135.

DCE \rightsquigarrow Desigualdade de Cobertura Estendida. Pag. 141.

DCI \rightsquigarrow Desigualdade de Cobertura Implícita. Pag. 139.

DCIE \rightsquigarrow Desigualdade de Cobertura Implícita Estendida. Pag. 143.

DCIL \rightsquigarrow Desigualdade de Cobertura Implícita Levantada. Pag. 145.

DCILDL \rightsquigarrow Desigualdade de Cobertura Implícita Levantada por *Down-Lifting*. Pag. 153.

DCILUL \rightsquigarrow Desigualdade de Cobertura Implícita Levantada por *Up-Lifting*. Pag. 158.

DGCIL \rightsquigarrow Desigualdade Generalizada de Cobertura Implícita Levantada. Pag. 164.

Algoritmos e Heurísticas para o Problema WMST

ABB-P-WMTZ+C \rightsquigarrow Algoritmo *Branch and Bound* do procedimento de introdução de cortes P-WMTZ+C. Pag. 308.

ALagB \rightsquigarrow Algoritmo Lagrangeano Base para o Problema WMST. Pag. 78.

- ALBC** \rightsquigarrow Algoritmo *Local Branching* Clássico para o Problema WMST. Pag. 284.
- ALB1** \rightsquigarrow Algoritmo *Local Branching* 1 (ALBC com limite de tempo). Pag. 292.
- ALB2** \rightsquigarrow Algoritmo *Local Branching* 2 (ALB1 com mecanismos de Intensificação e de Diversificação e interrupção do *solver* assim que se obtenha uma solução inteira admissível no Algoritmo *Branch and Bound* usando a Formulação WMTZ). Pag. 294.
- ALB3** \rightsquigarrow Algoritmo *Local Branching* 3 (ALB2 com limite de tempo na exploração das vizinhanças). Pag. 295.
- ALB4** \rightsquigarrow Algoritmo *Local Branching* 4 (solução de referência inicial é obtida pela Heurística FPB2). Pag. 297.
- ALB5** \rightsquigarrow Algoritmo *Local Branching* 5 (solução de referência inicial é a solução da correspondente árvore T_w). Pag. 298.
- Alg1** \rightsquigarrow Algoritmo Lagrangeano 1 (baseado no cálculo dos valores ponderados da forma $p_{ij} = (1 - \gamma_k)w_{ij} + \gamma_k c_{ij}$). Pag. 85.
- Alg2** \rightsquigarrow Algoritmo Lagrangeano 2 (baseado no cálculo dos valores ponderados da forma $p_{ij} = \lambda_k w_{ij} + c_{ij}$). Pag. 87.
- Alg3** \rightsquigarrow Algoritmo Lagrangeano 3 (baseado no Método do subgradiente). Pag. 88.
- Alg4** \rightsquigarrow Algoritmo Lagrangeano 4 (baseado na técnica de pesquisa binária). Pag. 90.
- Alg5** \rightsquigarrow Algoritmo Lagrangeano 5 (baseado na comparação de valores da função Lagrangeana). Pag. 92.
- Alg6** \rightsquigarrow Algoritmo Lagrangeano 6 (baseado na convexidade da função Lagrangeana). Pag. 96.
- Alg7** \rightsquigarrow Algoritmo Lagrangeano 7 (baseado na introdução de desigualdades válidas). Pag. 98.
- DCIL-E1** \rightsquigarrow Estratégia que a partir de uma DCI obtém uma DCIL. Pag. 188.
- DCIL-E2** \rightsquigarrow Estratégia que só obtém a DCIL se a DCI cortar a solução da relaxação linear. Pag. 188.
- DL-Ei** \rightsquigarrow Estratégia *Down-Lifting* i , $i = 1, 2, 3, 4$ (usada nos algoritmos de separação para o caso das DCILDLs). Pag. 226.

- FP** \rightsquigarrow Heurística *Feasibility Pump*. Pag. 241.
- FPB1** \rightsquigarrow Heurística *Feasibility Pump* Básica 1. Pag. 258.
- FPB2** \rightsquigarrow Heurística *Feasibility Pump* Básica 2. Pag. 259.
- FPO1** \rightsquigarrow Heurística *Feasibility Pump* com Alteração da Função Objetivo 1. Pag. 259.
- FPO2** \rightsquigarrow Heurística *Feasibility Pump* com Alteração da Função Objetivo 2. Pag. 259.
- GL-Ei** \rightsquigarrow Estratégia Generalizada i , $i = 1, 2, 3$ (usada nos algoritmos de separação para o caso das DGCIL). Pag. 231.
- Hcoef** \rightsquigarrow Algoritmo heurístico baseado em relaxação Lagrangeana para o cálculo dos coeficientes das variáveis a efetuar levantamento. Pag. 191.
- HXNR** \rightsquigarrow Heurística do *Xpress* Aplicada ao nodo raiz utilizando a Formulação WMTZ. Pag. 263.
- LB** \rightsquigarrow Heurística *Local Branching*. Pag. 290.
- OrdFixW** \rightsquigarrow Ordenação de arestas (ordem decrescente de peso, w_e , primeiro as arestas com $x_e = 1$ e depois as restantes arestas ($0 < x_e < 1$)). Pag. 169.
- OrdFixWX** \rightsquigarrow Ordenação de arestas (ordem decrescente de valor de $w_e \times x_e$, primeiro as arestas com $x_e = 1$ e depois as restantes arestas ($0 < x_e < 1$)). Pag. 169.
- OrdKnap** \rightsquigarrow Ordenação de arestas (ordem decrescente de valor de $(1 - x_e)/w_e$). Pag. 169.
- OrdW** \rightsquigarrow Ordenação de arestas (ordem decrescente de peso, w_e). Pag. 169.
- OrdWX** \rightsquigarrow Ordenação de arestas (ordem decrescente de valor de $w_e \times x_e$). Pag. 169.
- OrdX** \rightsquigarrow Ordenação de arestas (ordem decrescente de valor de x_e). Pag. 169.
- PSIABB** \rightsquigarrow Primeira solução inteira admissível obtida pelo *solver* do *Xpress* através da Formulação WMTZ, quando usa o procedimento *Branch and Bound*. Pag. 263.
- Sep1** \rightsquigarrow Algoritmo de Separação Genérico 1 (Inserção de uma desigualdade válida

no modelo por iteração). Pag. 168.

Sep2 \rightsquigarrow Algoritmo de Separação Genérico 2 (Inserção de várias desigualdades válidas no modelo por iteração). Pag. 170.

UL-Ei \rightsquigarrow Estratégia *Up-Lifting* i , $i = 1, 2, 3, 4$ (usada nos algoritmos de separação para o caso das DCILULs). Pag. 228.

Outras Siglas

Ai \rightsquigarrow Critério de Arredondamento i , $i = 1, 2, 3$ (Heurística FP). Pag. 251.

FVDi \rightsquigarrow Fixação de Variáveis Di , $i = 1, 2, 3$ (nas DCILDLs). Pag. 202.

FVUi \rightsquigarrow Fixação de Variáveis Ui , $i = 1, 2, 3, 4$ (nas DCILULs). Pag. 208.

LVi \rightsquigarrow Levantamento das Variáveis i , $i = 1, 2, 3$ (nas DCILs). Pag. 186.

Pi \rightsquigarrow Mecanismo de Perturbação i , $i = 1, 2$ (Heurística FP). Pag. 252.

PL-DL \rightsquigarrow Procedimento de Levantamento DL (primeiro efetua-se o *down-lifting* e depois o levantamento usual). Pag. 203.

PL-DLU \rightsquigarrow Procedimento de Levantamento DLU (primeiro efetua-se o *down-lifting*, depois o levantamento usual e por fim o *up-lifting*). Pag. 216.

PL-DUL \rightsquigarrow Procedimento de Levantamento DUL (primeiro efetua-se o *down-lifting*, depois o *up-lifting* e por fim o levantamento usual). Pag. 216.

PL-LD \rightsquigarrow Procedimento de Levantamento LD (primeiro efetua-se o levantamento usual e depois o *down-lifting*). Pag. 203.

PL-LDU \rightsquigarrow Procedimento de Levantamento LDU (primeiro efetua-se o levantamento usual, depois o *down-lifting* e por fim o *up-lifting*). Pag. 216.

PL-LU \rightsquigarrow Procedimento de Levantamento LU (primeiro efetua-se o levantamento usual e depois o *up-lifting*). Pag. 209.

PL-LUD \rightsquigarrow Procedimento de Levantamento LUD (primeiro efetua-se o

levantamento usual, depois o *up-lifting* e por fim o *down-lifting*). Pag. 217.

PL-UDL \rightsquigarrow Procedimento de Levantamento UDL (primeiro efetua-se o *up-lifting*, depois o *down-lifting* e por fim o levantamento usual). Pag. 216.

PL-UL \rightsquigarrow Procedimento de Levantamento LU (primeiro efetua-se o *up-lifting* e depois o levantamento usual). Pag. 209.

PL-ULD \rightsquigarrow Procedimento de Levantamento ULD (primeiro efetua-se o *up-lifting*, depois o levantamento usual e por fim o *down-lifting*). Pag. 216.

Lista de Figuras

2.1	Árvore de suporte de custo mínimo.	10
2.2	Árvore de suporte com custo 27 e peso 19.	10
2.3	Árvore de suporte de peso mínimo.	13
2.4	Árvore de suporte ponderada mínima ($p_{ij} = 0,5w_{ij} + 0,5c_{ij}$ para todos os $(i, j) \in A$).	14
2.5	Árvore de suporte ponderada mínima ($p_{ij} = w_{ij} + 0,2c_{ij}$ para todos os $(i, j) \in A$).	15
3.1	Exemplo com o valor das variáveis de fluxo na Formulação MF.	26
3.2	Exemplo do cálculo das posições dos nodos na árvore.	28
3.3	Exemplo do cálculo dos estados dos pesos dos nodos na árvore.	32
3.4	Exemplo com o valor das variáveis de fluxo na Formulação WE.	37
3.5	Solução ótima da relaxação linear obtida pelas Formulações ES, MF, WE, MTZl1 e WMTZl2.	39
3.6	Representação da solução da relaxação linear da Formulação WMTZ.	43
3.7	Representação da solução da relaxação linear da Formulação WMTZ após a introdução do corte 1.	44
3.8	Representação da solução da relaxação linear da Formulação WMTZ após a introdução do corte 2.	45
3.9	Tempos médios, em segundos, necessários para a obtenção do valor ótimo da Formulação MTZ com as várias desigualdades levantadas em instâncias de 10, 20 e 40 nodos.	51
3.10	Tempos médios, em segundos, necessários para a obtenção do valor ótimo da Formulação WMTZ com as várias desigualdades levantadas em instâncias de 10, 20 e 40 nodos.	54
3.11	Comparação das percentagens de tempo médio de execução (em segundos) da relaxação linear e do Algoritmo <i>Branch and Bound</i> do Procedimento P-WMTZ+C em instâncias R.	69

3.12	Comparação das percentagens de tempo médio de execução (em segundos) da relaxação linear e do Algoritmo <i>Branch and Bound</i> do Procedimento P-WMTZ+C em instâncias E.	70
3.13	Comparação das percentagens de tempo médio de execução (em segundos) da relaxação linear e do Algoritmo <i>Branch and Bound</i> do Procedimento P-WMTZ+C em instâncias QC.	70
3.14	Relação entre os valores obtidos pela relaxação linear das diferentes formulações e procedimentos de introdução de cortes.	71
4.1	Representação gráfica de $\vartheta(WMST_\lambda)$ em função do valor de λ	76
4.2	Representação geométrica dos pesos e dos custos das árvores T_c e T_w	81
4.3	Representação geométrica dos pesos e dos custos das árvores T_c e T_w e da reta $w = W$ com $W(T_w) > W$	82
4.4	Representação geométrica dos pesos e dos custos das árvores T_c e T_w e da reta $w = W$ com $W(T_c) \leq W$	83
4.5	Representação geométrica dos pesos e dos custos das árvores T_c e T_w e da reta $w = W$, onde $W(T_w) < W < W(T_c)$	83
4.6	Comparação dos tempos de execução em instâncias QC para diferentes valores de ρ	107
4.7	Comparação dos Gap_R médios, em percentagem, entre os Algoritmos Alg6 e Alg7 nas instâncias QC.	113
4.8	Tempos médios, em segundos, obtidos pelos algoritmos Lagrangeanos (instâncias QC).	116
4.9	Número médio de árvores de suporte obtidas pelos Algoritmos Alg5 e Alg6 nas instâncias QC.	118
4.10	Comparação dos Gap_R médios, em percentagem, entre os Algoritmos Alg6 e Alg7 nas instâncias R.	119
4.11	Tempos médios, em segundos, obtidos pelos algoritmos Lagrangeanos (instâncias R).	122
4.12	Número médio de árvores de suporte obtidas pelos Algoritmos Alg5 e Alg6 nas instâncias R.	124
4.13	Comparação dos Gap_R médios, em percentagem, entre os Algoritmos Alg6 e Alg7 nas instâncias E.	125
4.14	Tempos médios, em segundos, obtidos pelos algoritmos Lagrangeanos (instâncias E).	128
4.15	Número médio de árvores de suporte obtidas pelos Algoritmos Alg5 e Alg6 nas instâncias E.	130
5.1	Representação das arestas da Cobertura S_4 no grafo.	137
5.2	Árvore de suporte com custo 24 e peso 21.	140

5.3	Desigualdade de Cobertura e Desigualdade de Cobertura Implícita.	140
5.4	Ciclos formados com as arestas da Cobertura Implícita S do Exemplo 5.3	144
5.5	Árvore de suporte com custo 24 e peso 21.	152
5.6	Árvore de suporte com custo 28 e peso 22.	157
5.7	Árvore de suporte com custo 24 e peso 21.	162
6.1	Árvore de suporte de peso mínimo incluindo a aresta $\{0, 4\}$ com custo 47 e peso 14.	179
6.2	Árvore de suporte de peso mínimo incluindo as arestas $\{0, 1\}$ e $\{0, 4\}$ com custo 32 e peso 18.	179
6.3	Árvore de suporte de peso mínimo incluindo as arestas $\{0, 1\}$, $\{0, 4\}$ e $\{1, 2\}$ com custo 24 e peso 21.	180
6.4	Esquemas de obtenção de uma DCIL dada uma DCI.	188
6.5	Árvores de suporte T_{c_I} e T_{w_I} obtidas no Passo 1 do Algoritmo HCoef no cálculo do coeficiente de levantamento usual β_{34}	196
6.6	Árvore de suporte T_{p_I} obtida no Passo 2 do Algoritmo HCoef no cálculo do coeficiente de levantamento usual β_{34}	196
6.7	Árvores de suporte T_{c_I} e T_{w_I} obtidas no Passo 1 do Algoritmo HCoef no cálculo do coeficiente de levantamento usual β_{03}	197
6.8	Primeira árvore de suporte T_{p_I} obtida no Passo 2 do Algoritmo HCoef no cálculo do coeficiente de levantamento usual β_{03}	198
6.9	Segunda árvore de suporte T_{p_I} obtida no Passo 2 do Algoritmo HCoef no cálculo do coeficiente de levantamento usual β_{03}	199
6.10	Esquemas de obtenção de uma DCIL dada uma DCIE.	200
6.11	Árvores de suporte T_{c_I} e T_{w_I} obtidas no Passo 1 do Algoritmo HCoef no cálculo do coeficiente <i>down-lifting</i> β_{12}	204
6.12	Árvore de suporte T_{p_I} obtida no Passo 2 do Algoritmo HCoef no cálculo do coeficiente <i>down-lifting</i> β_{12}	205
6.13	Árvore de suporte T_{w_I} obtida no Passo 1.1 do Algoritmo HCoef no cálculo do coeficiente <i>up-lifting</i> β_{23}	210
6.14	Árvores de suporte T_{c_I} e T_{w_I} obtidas no Passo 1 do Algoritmo HCoef no cálculo do coeficiente <i>up-lifting</i> β_{02}	211
6.15	Árvore de suporte T_{p_I} obtida no Passo 2 do Algoritmo HCoef no cálculo do coeficiente <i>up-lifting</i> β_{02}	211

6.16	Percentagem de instâncias, em cada uma das melhores estratégias heurísticas de separação com tempo inferior ao tempo do Procedimento P-WMTZ+ C_L em instâncias QC de 10 a 60 nodos.	233
6.17	Percentagem de soluções ótimas obtidas com as melhores estratégias heurísticas de separação em instâncias QC de 10 a 60 nodos.	234
6.18	Percentagem de instâncias, em cada uma das melhores estratégias heurísticas de separação com o número de nodos no <i>Branch and Bound</i> inferior ao número usado pelo Procedimento P-WMTZ+C em instâncias QC de 10 a 60 nodos.	234
7.1	Árvore de suporte $T_{\tilde{x}}$ com custo 27 e peso 19.	243
7.2	Árvore de suporte $T_{\tilde{x}}$ com custo 24 e peso 21.	244
7.3	Representação geométrica das duas trajetórias de custos (instância QC80-7). . . .	254
7.4	Representação geométrica das duas trajetórias de pesos (instância QC80-7). . . .	254
7.5	Evolução da distância $\Delta(x^*, \tilde{x})$ ao longo das iterações (instância QC80-7).	255
7.6	Evolução da distância $\Delta(x^*, \tilde{x})$ ao longo das iterações na instância QC80-7 nas quatro estratégias heurísticas propostas.	260
7.7	Evolução da distância $\Delta(x^*, \tilde{x})$ ao longo das iterações na instância R80-5 nas quatro estratégias heurísticas propostas.	261
7.8	Evolução da distância $\Delta(x^*, \tilde{x})$ ao longo das iterações na instância E80-3 nas quatro estratégias heurísticas propostas.	261
7.9	Percentagem de <i>gaps</i> nulos nas diferentes heurísticas propostas nas instâncias QC. .	264
7.10	Comparação dos <i>gaps</i> médios, em percentagem, entre as Heurísticas FP propostas e a Heurística HXNR nas instâncias QC.	266
7.11	Diferenças médias de <i>gaps</i> , em percentagem, entre as Heurísticas HXNR e FPB2 nas instâncias QC.	268
7.12	Comparação dos <i>gaps</i> médios, em percentagem, entre as Heurísticas FPB2 e a PSI-ABB nas instâncias QC.	269
7.13	Percentagem de <i>gaps</i> nulos nas diferentes heurísticas propostas nas instâncias R. . .	270
7.14	Comparação dos <i>gaps</i> médios, em percentagem, entre as Heurísticas FP propostas nas instâncias R.	272
7.15	Percentagem de <i>gaps</i> nulos nas diferentes heurísticas propostas nas instâncias E. . .	275
7.16	Comparação dos <i>gaps</i> médios, em percentagem, entre as Heurísticas FP propostas nas instâncias E.	277
8.1	Esquema de ramificação obtido usando o Algoritmo ALBC na instância QC10-8. . .	286
8.2	Representação da trajetória de custos da instância QC10-8.	291

8.3	Representação da trajetória de pesos da instância QC10-8.	292
8.4	Esquema de ramificação obtido usando o Algoritmo ALB3 na instância QC10-10. . .	297
8.5	Tempos médios de execução, em segundos, obtidos usando o Algoritmo ALB4 para diferentes valores de k'	300
8.6	Percentagem de soluções ótimas obtidos pelos algoritmos <i>Local Branching</i> nas instâncias QC.	302
8.7	Percentagem de soluções ótimas obtidas pelos Algoritmos <i>Local Branching</i> nas instâncias R.	304
8.8	Percentagem de soluções ótimas obtidas pelos algoritmos <i>Local Branching</i> nas instâncias E.	306

Lista de Tabelas

3.1	Comparação dos valores obtidos usando as Formulações MTZ_L e $WMTZ_L$	33
3.2	Comparação das formulações usando um exemplo de 5 nodos.	39
3.3	Comparação das formulações usando um exemplo de 10 nodos.	47
3.4	Comparação dos valores da relaxação linear da Formulação MTZ com as várias desigualdades levantadas em instâncias de 10 nodos.	49
3.5	Comparação dos valores da relaxação linear da Formulação MTZ com as várias desigualdades levantadas em instâncias de 20 nodos.	50
3.6	Comparação dos valores da relaxação linear da Formulação WMTZ com as várias desigualdades levantadas em instâncias de 10 nodos.	52
3.7	Comparação dos valores da relaxação linear da Formulação WMTZ com as várias desigualdades levantadas em instâncias de 20 nodos.	53
3.8	Comparação dos tempos médios (em segundos) e dos desvios padrões necessários para a obtenção do valor da relaxação linear nas Formulações MTZ e WMTZ em instâncias QC.	55
3.9	Comparação dos tempos médios (em segundos) e dos desvios padrões necessários para a obtenção do valor da relaxação linear nas Formulações MTZ e WMTZ em instâncias R.	56
3.10	Comparação dos tempos médios (em segundos) e dos desvios padrões necessários para a obtenção do valor da relaxação linear nas Formulações MTZ e WMTZ em instâncias E.	56
3.11	Comparação das formulações em instâncias QC de 10 e 20 nodos.	58
3.12	Comparação das formulações em instâncias QC de 40, 60 e 80 nodos.	60
3.13	Comparação dos Procedimentos P-CS e P-WMTZ+C em termos de tempo médio de execução (em segundos) da relaxação linear em instâncias QC de 10, 20, 40, 60 e 80 nodos.	61
3.14	Comparação das formulações em instâncias R de 10 nodos.	62
3.15	Comparação das formulações em instâncias R de 40, 60, 80 e 100 nodos.	64

3.16	Comparação das formulações em instâncias E de 10 nodos.	65
3.17	Comparação das formulações em instâncias E de 20, 40, 60 e 80 nodos.	66
3.18	Tempos médios, em segundos, e respectivos desvios padrões obtidos com a aplicação do Procedimento P-WMTZ+C na obtenção do valor ótimo ou de um limite superior para o valor ótimo em cada grupo de instâncias.	68
4.1	Resultados da aplicação do Algoritmo Alg1 para o exemplo de 5 nodos.	86
4.2	Resultados da aplicação do Algoritmo Alg2 para o exemplo de 5 nodos.	87
4.3	Resultados da relaxação Lagrangeana aplicando o Algoritmo Alg3 para diferentes valores de ρ	89
4.4	Resultados da aplicação do Algoritmo Alg4 para o exemplo de 5 nodos.	91
4.5	Resultados da aplicação do Algoritmo Alg5 para obtenção do intervalo $[\ell_k, u_k]$. . .	95
4.6	Resultados da aplicação do Algoritmo Alg6 para obtenção do intervalo $[\ell_k, u_k]$. . .	97
4.7	Comparação dos algoritmos Lagrangeanos no Exemplo 2.1.	105
4.8	Valores de ρ utilizados no Algoritmo Alg3 nas instâncias Aleatórias e Euclidianas de 10 a 100 nodos.	107
4.9	Valores de ρ utilizados no Algoritmo Alg3 nas instâncias Aleatórias e Euclidianas de 150 a 1000 nodos.	107
4.10	Comparação das duas estratégias de inicialização do Algoritmo Alg5 em instâncias QC.	108
4.11	Percentagens de <i>gaps</i> nulos obtidas através do LS nos vários algoritmos por cada grupo de instâncias.	110
4.12	Percentagens de Gap_{LS} que são inferiores aos Gap_R nos vários algoritmos por cada grupo de instâncias.	111
4.13	<i>Gaps</i> médios, em percentagem, obtidos pelos algoritmos Lagrangeanos nas instâncias QC.	114
4.14	Desvios Padrões dos <i>gap</i> obtidos pelos algoritmos Lagrangeanos nas instâncias QC. .	115
4.15	Tempos médios, em segundos, obtidos pelos algoritmos Lagrangeanos (instâncias QC). .	117
4.16	Desvios padrões dos tempos obtidos pelos algoritmos Lagrangeanos (instâncias QC). .	117
4.17	<i>Gap</i> médios, em percentagem, obtidos pelos algoritmos Lagrangeanos nas instâncias R. .	120
4.18	Desvios Padrões dos <i>gap</i> obtidos pelos algoritmos Lagrangeanos nas instâncias R. .	121
4.19	Tempos médios, em segundos, obtidos pelos algoritmos Lagrangeanos (instâncias R). .	123
4.20	Desvios padrões dos tempos obtidos pelos algoritmos Lagrangeanos (instâncias R). .	123
4.21	<i>Gap</i> médios, em percentagem, obtidos pelos algoritmos Lagrangeanos nas instâncias E. .	126
4.22	Desvios Padrões dos <i>gap</i> obtidos pelos algoritmos Lagrangeanos nas instâncias E. .	127

4.23	Tempos médios, em segundos, obtidos pelos algoritmos Lagrangeanos (instâncias E).	129
4.24	Desvios padrões dos tempos obtidos pelos algoritmos Lagrangeanos (instâncias E).	129
6.1	Principais resultados por iteração do Algoritmo Sep1 para as DCs aplicado ao Exemplo 2.1.	174
6.2	Principais resultados por iteração do Algoritmo Sep2 para as DCs aplicado ao Exemplo 2.1.	175
6.3	Principais resultados por iteração do Algoritmo Sep1 para as DCIs aplicado ao Exemplo 2.1.	181
6.4	Principais resultados por iteração do Algoritmo Sep2 para as DCIs aplicado ao Exemplo 2.1.	182
6.5	Principais resultados por iteração do Algoritmo Sep1 para as DCIEs aplicado ao Exemplo 5.4.	184
6.6	Principais resultados por iteração do Algoritmo Sep2 para as DCIEs aplicado ao Exemplo 5.4.	185
6.7	Principais resultados por iteração do Algoritmo Sep1 para as DCILDLs com fixação de variáveis FVD3 aplicado ao Exemplo 2.1.	205
6.8	Principais resultados por iteração do Algoritmo Sep1 para as DCILDLs com fixação de variáveis FVD2 aplicado ao Exemplo 2.1.	206
6.9	Principais resultados por iteração do Algoritmo Sep1 para as DCILULs utilizando o procedimento de levantamento de variáveis PL-UL aplicado ao Exemplo 2.1.	212
6.10	Principais resultados por iteração do Algoritmo Sep1 para as DCILULs utilizando o procedimento de levantamento de variáveis PL-LU aplicado ao Exemplo 2.1.	213
6.11	Principais resultados por iteração do Algoritmo Sep1 para as DGCILs utilizando o procedimento de levantamento de variáveis PL-DLU aplicado ao Exemplo 2.1.	218
6.12	Principais resultados por iteração do Algoritmo Sep1 para as DGCILs utilizando o procedimento de levantamento de variáveis PL-DLU aplicado ao Exemplo 2.1.	219
6.13	Comparação do Procedimento P-WMTZ+ C_L e dos Algoritmos Sep1 e Sep2 nas DCs e nas DCIs nas instâncias QC de 10 a 60 nodos.	221
6.14	Comparação do Procedimento P-WMTZ+ C_L e dos Algoritmos Sep1 e Sep2 nas DCIEs nas instâncias QC de 10 a 60 nodos.	223
6.15	Comparação do Procedimento P-WMTZ+ C_L e dos Algoritmos Sep1 e Sep2 nas DCILs nas instâncias QC de 10 a 60 nodos.	225
6.16	Comparação do Procedimento P-WMTZ+ C_L e dos Algoritmos Sep1 e Sep2 nas DCILDLs nas instâncias QC de 10 a 60 nodos.	227

6.17	Comparação do Procedimento P-WMTZ+C _L e dos Algoritmos Sep1 e Sep2 nas DCI-LULs nas instâncias QC de 10 a 60 nodos.	230
6.18	Comparação do Procedimento P-WMTZ+C _L e dos Algoritmos Sep1 e Sep2 nas DG-CILs nas instâncias QC de 10 a 60 nodos.	232
6.19	Comparação do Procedimento P-WMTZ+C _L e dos melhores algoritmos heurísticos de separação nas instâncias QC de 10 a 60 nodos.	236
6.20	Comparação do Procedimento P-WMTZ+C _L e dos dois melhores algoritmos heurísticos de separação nas instâncias QC de 80 a 150 nodos.	238
7.1	Comparação dos valores obtidos usando as quatro Heurísticas FP propostas.	262
7.2	<i>Gaps</i> médios, em percentagem, obtidos pelas heurísticas propostas nas instâncias QC.	265
7.3	Desvios Padrões dos <i>gap</i> obtidos pelas heurísticas propostas nas instâncias QC.	265
7.4	Tempos médios, em segundos, obtidos pelas heurísticas propostas nas instâncias QC.	267
7.5	Desvios padrões dos tempos obtidos pelas heurísticas propostas nas instâncias QC.	267
7.6	<i>Gaps</i> médios, em percentagem, obtidos pelas heurísticas propostas nas instâncias R.	271
7.7	Desvios Padrões dos <i>gaps</i> obtidos pelas heurísticas propostas nas instâncias R.	271
7.8	Tempos médios, em segundos, obtidos pelas heurísticas propostas nas instâncias R.	273
7.9	Desvios padrões dos tempos obtidos pelas heurísticas propostas nas instâncias R.	273
7.10	<i>Gaps</i> médios, em percentagem, obtidos pelas heurísticas propostas nas instâncias E.	276
7.11	Desvios Padrões dos <i>gaps</i> obtidos pelas heurísticas propostas nas instâncias E.	276
7.12	Tempos médios, em segundos, obtidos pelas heurísticas propostas nas instâncias E.	278
7.13	Desvios padrões dos tempos obtidos pelas heurísticas propostas nas instâncias E.	278
8.1	Comparação dos algoritmos <i>Local Branching</i> na instância QC10-8.	299
8.2	Tempo, em segundos, imposto na exploração das vizinhanças.	301
8.3	Comparação dos algoritmos <i>Local Branching</i> em termos de <i>gaps</i> médios (em percentagem) e de tempo médio de execução (em segundos) nas instâncias QC.	303
8.4	Comparação dos algoritmos <i>Local Branching</i> em termos de <i>gaps</i> médios (em percentagem) e de tempo médio de execução (em segundos) nas instâncias R.	305
8.5	Comparação dos algoritmos <i>Local Branching</i> em termos de <i>gaps</i> médios (em percentagem) e de tempo médio de execução (em segundos) nas instâncias E.	307
8.6	Comparação entre os Algoritmos ALB4 e ABB-P-WMTZ+C nas instâncias QC de 10 a 40 nodos.	309
8.7	Comparação entre os Algoritmos ALB4 e ABB-P-WMTZ+C nas instâncias QC de 60 a 100 nodos.	310

8.8	Comparação entre os Algoritmos ALB4 e ABB-P-WMTZ+C nas instâncias QC de 150 a 1000 nodos.	311
8.9	Comparação entre os Algoritmos ALB4 e ABB-P-WMTZ+C nas instâncias R de 10 a 150 nodos.	312
8.10	Comparação entre os Algoritmos ALB4 e ABB-P-WMTZ+C nas instâncias R de 200 a 1000 nodos.	313
8.11	Comparação entre os Algoritmos ALB4 e ABB-P-WMTZ+C nas instâncias E de 10 a 150 nodos.	314
8.12	Comparação entre os Algoritmos ALB4 e ABB-P-WMTZ+C nas instâncias E de 200 a 1000 nodos.	315

Bibliografia

- [1] *FICO Xpress Optimization Suite*. www.fico.com/xpress.
- [2] T. Achterberg and T. Berthold. Improving the feasibility pump. *Discrete Optimization*, 4(1):77–86, 2007.
- [3] V. Aggarwal, Y.P. Aneja, and K.P. Nair. Minimal spanning tree subject to a side constraint. *Computers and Operations Research*, 9(4):287–296, 1982.
- [4] A. Agra, A. Cerveira, C. Requejo, and E. Santos. On the weight-constrained minimum spanning tree problem. In *Proceedings of the International Network Optimization Conference*, volume 6701 of *Lecture Notes in Computer Science*, pages 156–161, 2011.
- [5] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. Network flows. In G.L. Nemhauser, A.H.G. Rinnooy Kan, and M.J. Todd, editors, *Optimization, Handbooks in Operations Research and Management Science, Vol. 1*. Elsevier Science Publishers, North-Holland, Amsterdam, The Netherlands, 1989.
- [6] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, New York, 1993.
- [7] L. Amado and P. Barcia. New polynomial bounds for matroidal knapsacks. *European Journal of Operations Research*, 95(1):201–210, 1996.
- [8] K.A. Andersen, K. Jörnsten, and M. Lind. On bicriterion minimal spanning trees: An approximation. *Computers and Operations Research*, 23(12):1171–1182, 1996.
- [9] R. Andrade, A. Lucena, and N. Maculan. Using lagrangian dual information to generate degree constrained spanning trees. *Discrete Applied Mathematics*, 154(5):703–717, 2006.

- [10] E. Balas. Facets of knapsack polytope. *Mathematical Programming*, 8(1):146–164, 1975.
- [11] E. Balas and E. Zemel. Facets of knapsack polytope from minimal covers. *SIAM Journal on Applied Mathematics*, 34(1):119–148, 1978.
- [12] L. Bertacco, M. Fischetti, and A. Lodi. A feasibility pump heuristic for general mixed-integer problems. *Discrete Optimization*, 4(1):63–76, 2007.
- [13] D. Blokh and G. Gutin. An approximation algorithm for combinatorial optimization problems with two parameters. *Australasian Journal of Combinatorics*, 14:157–164, 1996.
- [14] T. Christof and G. Reinelt. *PORTA POLYhedron Representation Transformation Algorithm*. <http://typo.zib.de/opt-long-projects/Software/Porta/>.
- [15] M. Desrochers and G. Laporte. Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints. *Operations Research*, 10(1):27–36, 1991.
- [16] D. Eppstein. Finding the k smallest spanning trees. *BIT Numerical Mathematics*, 32(2):237–248, 1992.
- [17] C. E. Ferreira, A. Martin, and R. Weismantel. Solving multiple knapsack problems by cutting planes. *SIAM Journal Optimization*, 6(3):858–877, 1996.
- [18] M. Fischetti, F. Glover, and A. Lodi. The feasibility pump. *Mathematical Programming*, 104(1):91–104, 2005.
- [19] M. Fischetti and A. Lodi. Local branching. *Mathematical Programming*, 98(1-3):23–47, 2003.
- [20] L.R. Ford and D.R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.
- [21] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, New York, 1979.
- [22] R.E. Gomory. Some polyhedra related to combinatorial problems. *Linear Algebra and its Applications*, 2(4):451–558, 1969.

- [23] L. Gouveia. A $2n$ constraint formulation for the capacitated minimal spanning tree problem. *Operations Research*, 43(1):130–141, 1995.
- [24] L. Gouveia. Using the Miller-Tucker-Zemlin constraints to formulate a minimal spanning tree problem with hop constraints. *Computers and Operations Research*, 22(9):959–970, 1995.
- [25] L. Gouveia. Multicommodity flow models for spanning trees with hop constraints. *European Journal of Operational Research*, 95(1):178–190, 1996.
- [26] L. Gouveia and T. Magnanti. Network flow models for designing diameter-constrained minimum-spanning and Steiner trees. *Networks*, 41(3):159–173, 2003.
- [27] L. Gouveia and P. Martins. The capacitated minimum spanning tree problem: revisiting hop-indexed formulations. *Computers and Operations Research*, 32(9):2435–2452, 2005.
- [28] L. Gouveia and C. Requejo. A new lagrangean relaxation approach for the hop-constrained minimum spanning tree problem. *European Journal of Operational Research*, 132(3):539–552, 2001.
- [29] Z. Gu, G.L. Nemhauser, and M.W.P. Savelsbergh. Lifted cover inequalities for 0-1 integer programs: Computation. *INFORMS Journal on Computing*, 10(4):427–437, 1998.
- [30] Z. Gu, G.L. Nemhauser, and M.W.P. Savelsbergh. Lifted cover inequalities for 0-1 integer programs: Complexity. *INFORMS Journal on Computing*, 11(1):117–123, 1999.
- [31] H.W. Hamacher and G. Ruhe. On spanning tree problems with multiple objectives. *Annals of Operations Research*, 52(4):209–230, 1994.
- [32] P.L. Hammer, E.L. Johnson, and U.N. Peled. Facets of regular 0-1 polytopes. *Mathematical Programming*, 8(1):179–206, 1975.
- [33] G. Handler and I. Zang. A dual algorithm for the constrained shortest path problem. *Networks*, 10(4):293–309, 1980.
- [34] P. Hansen, N. Mladenović, and D. Urošević. Variable neighborhood search and local branching. *Computers and Operations Research*, 33(10):3034–3045, 2006.

- [35] R. Hassin and A. Levin. An efficient polynomial time approximation scheme for the constrained minimum spanning tree problem using matroid intersection. *SIAM Journal on Computing*, 33(2):261–268, 2004.
- [36] M. Held, P. Wolfe, and H.P. Crowder. Validation of subgradient optimization. *Mathematical Programming*, 6(1):62–88, 1974.
- [37] S.T. Henn. Weight-constrained minimum spanning tree problem. Master’s thesis, Department of Mathematics, University of Kaiserslautern, Kaiserslautern, Germany, 2007.
- [38] S.P. Hong, S.J. Chung, and B.H. Park. A fully polynomial bicriteria approximation scheme for the constrained spanning tree problem. *Operations Research Letters*, 32(3):233–239, 2004.
- [39] B.A. Julstrom and G.R. Raidl. Edges sets: an effective evolutionary coding of spanning trees. *IEEE Transactions on Evolutionary Computation*, 7(3):225–239, 2003.
- [40] A. Jüttner. On resource constrained optimization problems. In *4th Japanese-Hungarian Symposium on Discrete Mathematics and its Applications*, pages 3–6, 2005.
- [41] A. Jüttner, B. Szviatovszki, I. Mécs, and Z. Rajkó. Lagrange relaxation based method for the QoS routing problem. In *Proceedings of the IEEE INFOCOM - The conference on Computer Communications*, volume 2, pages 859–868. IEEE Computer and communications Societies, 2001.
- [42] K. Kaparis and A.N. Letchford. Local and global lifted cover inequalities for the 0-1 multidimensional knapsack problem. *European Journal of Operational Research*, 186(1):91–103, 2008.
- [43] K. Kaparis and A.N. Letchford. Separation algorithms for 0-1 knapsack polytopes. *Mathematical Programming*, 124(1-2):69–91, 2010.
- [44] D. Klabjan, G.L. Nemhauser, and C. Tovey. The complexity of cover inequality separation. *Operations Research Letters*, 23(1-2):35 – 40, 1998.
- [45] T. Magnanti and L. Wolsey. Optimal trees. In M.O. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser, editors, *Network Models, Handbooks in Operations*

Research and Management Science, Vol. 7. Elsevier Science Publishers, North-Holland, 1995.

- [46] C.E. Miller, A.W. Tucker, and R.A. Zemlin. Integer programming formulation and travelling salesman problems. *Journal of the Association for Computing Machinery*, 7(4):326–329, 1960.
- [47] N. Mladenović and P. Hansen. Variable neighborhood search. *Computers and Operations Research*, 24(11):1097–1100, 1997.
- [48] S.C. Narula and C.A. Ho. Degree-constrained minimum spanning tree. *Computers and Operations Research*, 7(4):239–249, 1980.
- [49] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, New York, 1988.
- [50] D. Pisinger. Where are the hard knapsack problems? *Computers and Operations Research*, 32(9):2271–2284, 2005.
- [51] R.C. Prim. Shortest connection networks and some generalizations. *Bell Systems Technical Journal*, 36(6):1389–1401, 1957.
- [52] G.R. Raidl, S. Pirkwieser, and J. Puchinger. A Lagrangean decomposition/evolutionary algorithm hybrid for the knapsack constrained maximum spanning tree problem. *Studies in Computational Intelligence*, 153:69–85, 2008.
- [53] R.M. Ramos, S. Alonso, J. Sicilia, and C. González. The problem of the optimal biobjective spanning tree. *European Journal of Operational Research*, 111(3):617–628, 1998.
- [54] R. Ravi and M.X. Goemans. The constrained minimum spanning tree problem. In *Proceedings of the Scandinavian Workshop on Algorithm Theory*, volume 1097 of *Lecture Notes in Computer Science*, pages 66–75, 1996.
- [55] C. Requejo, A. Agra, A. Cerveira, and E. Santos. Formulations for the weight-constrained minimum spanning tree problem. In *Proceedings of the International Conference on Numerical Analysis and Applied Mathematics*, volume 1281 of *AIP Conference Proceedings*, pages 2166–2169, 2010.

- [56] C. Requejo and E. Santos. Greedy heuristics for the diameter-constrained minimum spanning tree problem. *Journal of Mathematical Sciences*, 161(6):930–943, 2009.
- [57] C. Requejo and E. Santos. Lagrangean based algorithms for the weight constrained minimum spanning tree problem. In *Proceedings of the VII ALIO-EURO - Workshop on Combinatorial Optimization*, pages 38–41, 2011.
- [58] A.W. Shogan. Constructing a minimal-cost spanning tree subject to resource constraints and flow requirements. *Networks*, 13(2):169–190, 1983.
- [59] N.Z. Shor. *Minimization Methods for Non-Differentiable Functions*. Springer-Verlag, Berlin, 1985.
- [60] F. Sourd and O. Spanjaard. A multiobjective branch-and-bound framework: application to the biobjective spanning tree problem. *INFORMS Journal on Computing*, 20(3):472–484, 2008.
- [61] S. Steiner and T. Radzik. Computing all efficient solutions of the biobjective minimum spanning tree problem. *Computers and Operations Research*, 35(1):198–211, 2008.
- [62] L.A. Wolsey. Faces for linear inequalities in 0-1 variables. *Mathematical Programming*, 8(1):165–178, 1975.
- [63] L.A. Wolsey. *Integer Programming*. John Wiley & Sons, 1998.
- [64] Y. Xiao, K. Thulasiraman, G. Xue, and A. Jüttner. The constrained shortest path problem: Algorithmic approaches and an algebraic study with generalization. *AKCE International Journal of Graphs and Combinatorics*, 2(2):63–86, 2005.
- [65] G. Xue. Primal-dual algorithms for computing weight-constrained shortest paths and weight-constrained minimum spanning trees. In *Proceedings of the IEEE International Performance, Computing, and Communications Conference*, pages 271 –277. IEEE Communications Society, 2000.
- [66] T. Yamada, K. Watanabe, and S. Kataoka. Algorithms to solve the knapsack constrained maximum spanning tree problem. *International Journal of Computer Mathematics*, 82(1):23–34, 2005.