Universidade Portucalense Infante D. Henrique
Departamento de Matemática

Algoritmos de Feixe para Programação Linear Inteira Mista

Ana Maria Ramires Príncipe dos Santos

Doutoramento em Matemática
(Especialidade Optimização Discreta)

2004

Universidade Portucalense Infante D. Henrique
Departamento de Matemática

Algoritmos de Feixe para Programação Linear Inteira Mista

Ana Maria Ramires Príncipe dos Santos

Tese submetida à Universidade Portucalense Infante D. Henrique para obtenção do grau de Doutor em Matemática (especialidade Optimização Discreta), realizada sob a orientação do Professor Doutor João Luís Cardoso Soares, Professor Auxiliar do Departamento de Matemática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

2004

Ao Pedro, Ricardo e Miguel.

Agradecimentos

Agradeço ao meu supervisor, Professor Doutor João Luís Cardoso Soares, toda a orientação e apoio científicos. Nunca puderei esquecer a sua disponibilidade, persistência e encorajamento durante o desenvolvimento deste trabalho, sem o que não teria conseguido torná-lo realidade.

Ao Departamento de Matemática da Universidade Portucalense e, em particular, ao Professor Doutor F. Costa Durão, por todo o apoio e incentivo; aos meus colegas que sempre me encorajaram, em particular, ao José Matos, Helena Dulce, M. João Polidoro, Natércia Durão, Raquel Oliveira, Stella Abreu e, em especial, ao Professor Doutor F. Maciel Barbosa; ao Departamento de Informática pelo apoio técnico; e à Professora Doutora Cristina Abreu pelo apoio na aquisição de bibliografia e *software*.

À minha família e aos meus amigos, pelo apoio e encorajamento. Em particular, quero agradecer à minha cunhada Isabel a disponibilidade sempre mostrada.

À minha Mãe, por toda a dedicação, altruísmo, força e amor.

Ao Ricardo pela sua energia, ao Miguel pela sua ternura e ao Pedro pela sua paciência; aos três, pelo seu amor e pela forma como aceitaram as restrições impostas à nossa vida familiar nos últimos anos.

Finalmente, agradeço o apoio financeiro suportado pelo PRODEP, formação avançada de docentes do ensino superior, no âmbito dos concursos 3/PRODEP/98 (Medida 5-Acção 5.2) e 2/PRODEP/2000 (Medida 5-Acção 5.3).

Resumo

Modelos de optimização são muito comuns na literatura das áreas da Investigação Operacional e das Ciências de Gestão. Neste trabalho, abordamos a resolução de uma classe de modelos de optimização conhecidos por modelos de *programação linear disjuntiva*, que se caracterizam por serem problemas de programação linear cuja região de admissibilidade é união de poliedros. Quase todos os problemas de optimização combinatória (como o problema do caixeiro viajante, por exemplo) podem ser modelados como problemas de programação linear disjuntiva.

Problemas de programação linear disjuntiva (incluindo muitos dos problemas de optimização combinatória) são melhor resolvidos por métodos do tipo enumerativo (denominados *branch-and-bound*), eventualmente combinados com métodos de planos cortantes (denominados *branch-and-cut*). No caso de problemas de optimização combinatória, os cortes mais eficazes são os que constituem facetas do poliedro associado. No caso de problemas sem estrutura identificável, os *cortes mais profundos* propostos em [17] são vistos, em geral, como uma boa opção. Contudo, a sua caracterização obriga à resolução de um adequado problema de optimização cuja resolução algébrica obriga à duplicação de estruturas de dados. Em geral, não são conhecidas simplificações se o problema possuir estrutura.

Neste trabalho, mostramos como é possível gerar cortes mais profundos para problemas com estrutura como é o caso, por exemplo, dos problemas de optimização combinatória. Para a resolução do problema de optimização subjacente à caracterização do corte mais profundo, usaremos algoritmos de feixe (do inglês, *bundle methods*). O desenvolvimento que efectuámos encontra-se fundamentalmente relacionado com o algoritmo proposto por P. Wolfe na década de 70 [128], que determina o ponto pertencente ao invólucro convexo de um conjunto finito de pontos que está mais próximo da origem, no sentido da norma Euclidiana.

Assim, propomos uma generalização do algoritmo de P. Wolfe [128] para qualquer métrica, diferenciável ou não. Tal como aquele, o algoritmo que propomos é composto por um ciclo externo e um ciclo interno, termina ao fim de um número finito de iterações e gera a mesma sucessão de pontos quando a métrica é a Euclidiana. O

algoritmo foi implementado como gerador de cortes disjuntivos mais separadores em alguns problemas de programação inteira, com e sem estrutura, para uma selecção de problemas teste da MIPLIB e da ORLIB.

Efectuámos um trabalho de síntese, inspirado em [91] e [43], no qual se demonstram que alguns cortes combinatórios bem conhecidos são válidos para determinadas relaxações disjuntivas do problema de optimização combinatória subjacente. A identificação dessas relaxações permitirá obter cortes mais profundos de acordo com a metodologia mencionada no parágrafo anterior. Os cortes combinatórios analisados dizem respeito aos problemas de partição por cliques, corte máximo, subdigrafo acíclico, ordenamento linear, cobertura de conjuntos e caixeiro viajante assimétrico. As provas apresentadas nesta dissertação usam argumentos primais e constituem uma alternativa às provas apresentadas em [91], cujo autor usou argumentos duais.

Finalmente, procurou-se pôr em prática algumas das metodologias sugeridas num problema específico de optimização combinatória. A versão assimétrica do problema do caixeiro viajante é um problema de optimização combinatória que tem vindo a servir de plataforma de teste para diversos métodos de resolução em optimização combinatória. Este facto e a importância de descobrir bons limites inferiores para este problema motivou a proposta de um algoritmo do tipo Lagrangeano para obter um limite inferior para o valor óptimo do problema do caixeiro viajante assimétrico, melhor do que o que advém do problema de afectação, através da resolução sucessiva de problemas de afectação. O algoritmo é um método de primeira ordem baseado na função de penalidade exponencial cujas direcções de deslocamento são definidas com base numa relaxação disjuntiva que se propõe ser de dois tipos: uma baseada em ciclos e a outra baseada em cliques.

Palavras-chave: Optimização, Optimização Combinatória, Programação Disjuntiva, Projecção, Separação, Ponto mais próximo, Geração de colunas, Limites Inferiores, Problema do Caixeiro Viajante

Abstract

Optimization models are very common in the Operations Research and Management Science literature. In this work, it is proposed the study of the solution of a class of optimization models known by disjunctive convex programs, witch are characterized as linear programs whose feasible region is the union of polyhedra. Almost any combinatorial optimization problems (like the traveling salesman problem, for example) can be modeled as a disjunctive linear problem.

Disjunctive linear programs (including many of the combinatorial optimization problems) are best solved by methods of the enumerative type (called branch-and-bound) or combined with cutting planes methods (called branch-and-cut). In the case of combinatorial optimization problems, the most efficient cuts are the ones that induce facets of the associated polyhedron. In the case of problems without identifiable structure, deepest cuts, considered in [17], are seen, in general, as a good option. However, its characterization compels to the resolution of one adequate optimization problem whose algebraic resolution involves the duplication of data structures. In general, simplifications are not known if the problem possesses structure.

In this work, we show how to to generate deepest cuts for problems with structure as it is the case, for example, of the combinatorial optimization problems. We will use bundle methods to solving the optimization problem underlying the characterization of the deepest cut. Our development is related to P. Wolfe's algorithm, developed in the 70's [128], for the problem of finding the nearest point of smallest Euclidean norm in the convex hull of a given finite point set. We consider a generalization of P. Wolfe's algorithm [128] for any metric, differentiable or not. The algorithm we propose is composed of an external cycle and an internal cycle, stops after a finite number of iterations and generates the same iterates as Wolfe's algorithm for the Euclidean metric. The algorithm was implemented as a deepest disjunctive cut generator for some integer programming problems, with and without structure, and numerical results were obtained for a selection of test problems from MIPLIB and ORLIB.

We made a synthesis work, inspired by [91] and [43], in which we demonstrate that some well known combinatorial cuts are valid for specific disjunctive relaxations of the

underlying combinatorial optimization problem. The identification of these relaxations will allow to get deeper cuts according to the methodology mentioned in the previous paragraph. Combinatorial problems that were analyze are clique partitioning, max-cut, acyclic subdigraph, linear ordering, covering partitioning and asymmetric traveling salesman problems. The proofs presented in this dissertation use primal arguments and are alternatives to the proofs in [91], whose author used dual arguments.

Finally, it was looked to put in practice some of the methodologies suggested, in a specific combinatorial optimization problem. The asymmetric version of the traveling salesman problem is a combinatorial problem frequently used as test platform for solutions methods in combinatorial optimization. This fact and the importance as finding good lower bounds for this problem motivated the proposal of an algorithm of the Lagrangean type, to get lower bounds for the optimal value of the asymmetric traveling salesman problem, that dominates the bound that comes from the assignment relaxation, through the solving of a sequence of assignment problems. The algorithm that we propose is a first-order method based on the exponential penalty function. Directions of movement are derived from a disjunctive relaxation that we propose being one of two possible classes, one based on cycles, the other based on cliques.

Keywords: Optimization, Combinatorial Optimization, Disjunctive Programming, Projection, Separation, Nearest point, Column generation, Lower Bounds, Traveling Salesman Problem

Conteúdo

Índice de Tabelas **x**

Índice de Figuras **xii**

1 Introdução **13**

1.1 Introdução 13

1.2 Histórico 15

1.3 Estrutura da tese 17

1.4 Notação e alguns resultados básicos 19

1.4.1 Teoria poliedral e optimização combinatória 20

1.4.2 Teoria de grafos 23

2 Projectar: dual de Separar **27**

2.1 Introdução 27

2.2 Uma generalização do algoritmo de P. Wolfe 29

2.3 Álgebra linear numérica 34

2.3.1 Norma Euclidiana 35

2.3.2 Normas l_1 e l_∞ 38

2.4 Experiência computacional 38

2.4.1 Experiência computacional com diferentes normas 42

2.4.2 Experiência computacional com problemas lineares 0-1 43

2.4.3 Experiência computacional com problemas de partição 49
2.5 Conclusões 49

3 Dedução de cortes combinatórios como cortes disjuntivos 55
3.1 Introdução 55
3.2 Característica inteira de um poliedro 56
3.2.1 Programação inteira pura 56
3.2.2 Programação inteira mista 59
3.3 Dedução de alguns cortes combinatórios como cortes disjuntivos 62
3.3.1 O problema de partição por cliques 63
3.3.2 O problema de corte máximo 65
3.3.3 O problema do subdigrafo acíclico 69
3.3.4 O problema de ordenamento linear 71
3.3.5 O problema da cobertura de conjuntos 72
3.3.6 O problema do caixeiro viajante assimétrico 73

4 Um melhor limite inferior para o Problema do Caixeiro Viajante Assimétrico baseado na Afectação 77
4.1 Introdução 77
4.2 O Problema do Caixeiro Viajante Assimétrico 78
4.2.1 A formulação clássica 79
4.2.2 Uma relaxação linear disjuntiva baseada em ciclos 81
4.2.3 Uma relaxação linear disjuntiva baseada em cliques 85
4.2.4 Determinação de um limite inferior melhorado 90
4.2.4.1 Ilustração com um pequeno exemplo 93
4.3 Conclusões 98
4.4 Corte global de capacidade mínima num grafo dirigido 98
4.4.1 Algoritmo de Golberg e Tarjan (corte $s-t$) 100

4.4.2 Algoritmo de Hao e Orlin (corte global) 110
4.4.3 Experiência computacional 123
4.5 O problema de afectação 125
4.5.1 Emparelhamentos Perfeitos 128
4.5.2 O Método Húngaro 131
4.5.3 Histórico 134
4.5.3.1 Emparelhamento de máxima cardinalidade 134
4.5.3.2 Afectação 136

5 Conclusões e perspectivas de trabalho futuro **138**

A Fortalecimento do corte $\alpha x \geq \beta$ **143**

B Regra de Ryan-Foster **149**

C Fluxos e prefluxos **151**

Referências **166**

Lista de Tabelas

2.1 Organização de um ficheiro em formato MPS. 39

2.2 Problemas do Tipo 0: número médio de Ciclos maiores e menores, número de pontos e tempo de CPU. 44

2.3 Problemas do Tipo 1: número médio de Ciclos maiores e menores, número de pontos e tempo de CPU. 45

2.4 Problemas do Tipo 2: número médio de Ciclos maiores e menores, número de pontos e tempo de CPU. 46

2.5 Problemas do Tipo 3: número médio de Ciclos maiores e menores, número de pontos e tempo de CPU. 47

2.6 Descrição dos problemas teste lineares 0-1. 48

2.7 Resultados computacionais do *Bundle_sc*. 51

2.8 Resultados computacionais do *Bundle_1d*. 52

2.9 Resultados computacionais do *Bundle_2d*. 52

2.10 Resultados computacionais do *Bundle_c*. 52

2.11 Descrição dos problemas teste de partição. 53

2.12 Resultados computacionais do *Bundle_RF*- 2 e 3 restrições. 54

4.1 Matriz de custos de uma instância do ATSP com 8 vértices. 94

4.2 Resultados computacionais da implementação do algoritmo de Hao e Orlin. 124

Lista de Figuras

2.1 Algoritmo modificado de P. Wolfe. 31

3.1 Exemplo de polítopo com característica de Chvátal 2. 58
3.2 Exemplo de "2-chorded odd cycle" com 7 ciclos. 64
3.3 Exemplo de "odd wheel" com 7 ciclos. 66
3.4 Exemplo de "odd bicycle wheel" com 5 ciclos. 68
3.5 Exemplo de circulante-$(5, 2)$. 69
3.6 Exemplo de escada de Möbius com 5 diciclos. 71
3.7 Exemplos de D_k^+ e D_k^-. 76

4.1 Árvore de enumeração dos caminhos simples em $S = \{i_1, i_2, i_3\}$. 86
4.2 $P_{(j_1,j_2,\ldots,j_k)}$. 87
4.3 Algoritmo para determinar um limite inferior melhorado para o ATSP. 93
4.4 Soluções óptimas inteiras sobre P^{ATSP} e P^{ASS}, respectivamente. 94
4.5 Iteração 0. 95
4.6 Iteração 1. 95
4.7 Iteração 2. 96
4.8 Iteração 3. 96
4.9 Iteração 4. 97
4.10 Iteração 5: $G[x_6]$. 97
4.11 Subrotina *Inicializar*. 102

4.12 Subrotina *Empurrar (i, j)*. 103

4.13 Subrotina *Rotular (i)*. 103

4.14 Algoritmo *empurrar-prefluxo* de Golberg e Tarjan. 105

4.15 Algoritmo *DetCorteMin(s)* de Padberg e Rinaldi. 111

4.16 Subrotinas *InicializarMod* e *RotularMod (i)*. 113

4.17 Algoritmo *DetCorteMin(s)* de Hao e Orlin. 114

4.18 Algoritmo de determinação do emparelhamento de máxima cardinalidade num grafo bipartido. 129

4.19 Método Húngaro. 131

4.20 Redefinição das variáveis duais no método Húngaro. 134

Capítulo 1

Introdução

Um rio caudaloso pode-nos deixar indiferentes à sua passagem... Uma gota de água pode, pelo contrário, submergir-nos completamente.

Fogo Sobre Fogo, Jorge de Sousa Braga, 1998

1.1 Introdução

Os modelos de optimização são muito comuns na literatura das áreas da Investigação Operacional e das Ciências de Gestão. Neste trabalho, abordamos a resolução de uma classe de modelos de optimização conhecidos por modelos de *programação linear disjuntiva*, que se caracterizam por ser problemas de programação linear cuja região de admissibilidade é união de poliedros. Quase todos os problemas de optimização combinatória (como o problema do caixeiro viajante, por exemplo) podem ser modelados como problemas de programação linear disjuntiva. Exemplos de aplicações incluem, entre outros, os problemas do caixeiro viajante, de localização de serviços/equipamentos, de rotas de veículos, de desenho de redes de telecomunicações, da selecção/composição da carteira, planeamento de máquinas, calendarização de tripulações, partição de conjuntos, problemas com restrições de limites superiores nas variáveis e o problema linear complementar. Estes problemas também têm aplicações em outras áreas das ciências como a biologia, ciências dos computadores e matemáticas discretas. As aplicações incluem sequenciamento de DNA, desenho de VLSI e compiladores e a solução de diversos problemas em matemáticas discretas como, por exemplo, o problema das quatro cores.

Um problema de *programação linear* é um problema de optimização, em que a função

a optimizar é linear e a região admissível é a solução de um número finito de restrições (desigualdades e equações) lineares. Geometricamente, a região admissível é um poliedro convexo.

Se o programa linear contém uma restrição adicional que indica que só algumas das variáveis de decisão podem assumir valores inteiros, designa-se por *programa linear inteiro misto*. Geometricamente, a região admissível deste problema é a união de poliedros individuais, em que cada poliedro corresponde a uma afectação particular de valores inteiros às variáveis inteiras. Se a restrição adicional indica que todas as variáveis de decisão só podem assumir valores inteiros, o problema designa-se por *programa linear inteiro (puro)*. Neste problema, a região admissível é constituída por todos os pontos inteiros que pertencem à região admissível do programa linear e, geometricamente, é um reticulado. Se a restrição adicional indica que todas as variáveis de decisão estão restritas aos valores 0 e 1, o problema designa-se por *programa inteiro binário ou 0-1*.

Se o programa linear possui condições lógicas, isto é, declarações conjuntivas e disjuntivas sobre restrições lineares, designa-se por *programa linear disjuntivo* (extensão da programação linear introduzida por Balas [9]). Geometricamente, a região admissível deste problema é a união de poliedros individuais porque qualquer sistema lógico de restrições pode ser equivalentemente escrito como uma cadeia de declarações disjuntivas, em que cada declaração individual só contém restrições conjuntivas. Na prática, qualquer programa inteiro ou inteiro misto pode ser formulado como um programa disjuntivo. Reciprocamente, qualquer programa disjuntivo limitado pode ser reformulado como um programa linear inteiro misto, usando variáveis binárias para representar a disjunção.

Existe muita literatura dedicada à resolução de classes específicas de programas convexos disjuntivos. Por exemplo, o problema linear complementar monótono é resolvido em tempo polinomial, existindo algoritmos muito eficientes para problemas de pequena e grande dimensão. O problema do caixeiro viajante e outros problemas de optimização combinatória são programas disjuntivos, em que os conjuntos individuais são pontos isolados. A resolução eficiente deste problema baseia-se no estudo da sua estrutura poliedral, o que implica a identificação de desigualdades válidas, e no desenvolvimento de métodos para a sua determinação. Porém, existem muitos problemas que não possuem qualquer estrutura particular.

O principal objectivo deste trabalho é a determinação de melhores cortes disjuntivos para problemas de programação linear inteira mista, utilizando algoritmos de feixe, no sentido de serem mais profundos e proporcionarem, na prática, uma convergência mais rápida a um algoritmo de planos cortantes. Este trabalho está fundamentalmente relacionado com a teoria e experiência prática desenvolvidas com a abordagem *lift-and-*

project de Balas, Ceria e Cornuéjols [16, 17], o algoritmo proposto por P. Wolfe [128] que determina o ponto pertencente ao invólucro convexo de um conjunto finito de pontos que está mais próximo da origem, no sentido da norma Euclidiana, e os desenvolvimentos teóricos descritos por Soares em [120] sobre a solução de problemas disjuntivos convexos.

A abordagem de Balas et al. [16, 17] constitui um dos contributos práticos mais importantes para a solução de problemas de programação inteira binária, através de algoritmos de planos cortantes, desde o trabalho de Gomory nos anos 60. Os cortes são seleccionados entre as desigualdades válidas de uma determinada relaxação linear disjuntiva do problema original e a sua determinação consiste em resolver um problema linear (ou quadrático). No caso linear, a geração dos cortes disjuntivos baseia-se no conhecimento da representação algébrica do polar da relaxação linear disjuntiva (conjunto de todas as desigualdades válidas). No caso não linear, em que essa representação algébrica não é explícita, o problema da geração dos cortes disjuntivos pode ser resolvido, conforme sugerido por Owen e Mehrotra [104], usando o seu dual. Esta plataforma primal-dual não só permite a determinação de cortes no caso não linear como também fornece a interpretação geométrica do processo *lift-and-project* de geração de cortes.

1.2 Histórico

A maioria da teoria fundamental e definições usadas na programação disjuntiva foram desenvolvidas e apresentadas por Balas, nos anos 70, em dois relatórios técnicos [8, 9] posteriormente publicados em [10, 11, 12]. Os dois resultados fundamentais da programação disjuntiva, o *Princípio Disjuntivo* e o *Procedimento de Convexificação Sequencial* foram introduzidos em [8] e [9], respectivamente.

O Procedimento de Convexificação Sequencial permite obter a caracterização de todas as desigualdades válidas de um poliedro disjuntivo, visto como o fecho do invólucro convexo da união de poliedros individuais explícitos, e estabelece um procedimento para a determinação dessas desigualdades quando são conhecidas desigualdades válidas para cada um desses poliedros individuais. Quando obtidas através deste procedimento, as desigualdades válidas tomam a designação de *disjuntivas* ou simplesmente *cortes disjuntivos*. Sob determinados pressupostos de facialidade, este procedimento permite obter a representação explícita do poliedro disjuntivo, considerando sequencialmente os invólucros convexos parciais. Este resultado foi usado por Jeroslow [83] e por Balas et al. [16] para provar a convergência finita dos algoritmos propostos para programação disjuntiva (limitada) e para programação linear inteira binária, respectivamente.

Anteriormente aos cortes disjuntivos introduzidos por Balas, os cortes que eram apli-

cados na resolução de programas lineares disjuntivos eram os cortes de Gomory (para programas lineares inteiros mistos [68, 69]), os cortes de intersecção [6], os cortes de convexidade [62], os cortes diamante (polares ou polaroide) [27] e os cortes de anexação poliedral [64]. Em [82], Jeroslow sumaria os pressupostos básicos da maioria destes cortes e relaciona-os com os cortes disjuntivos. Balas, Bowman, Glover e Sommer em [15], Balas em [7] e Glover em [63] apresentam novos desenvolvimentos e casos especiais.

Na referência [117], Sherali e Shetty sumariam a programação linear disjuntiva. O princípio disjuntivo é usado para formular o problema de determinar o *corte mais profundo*, no sentido de uma certa medida da profundidade entre a origem e uma dada relaxação de programação disjuntiva. É proposto um procedimento para determinar o corte mais profundo, embora os resultados da experiência computacional sejam limitados.

Boyd em [25, 26] usa o conceito de *corte mais profundo* na resolução de problemas de programação linear inteira e inteira binária. A selecção do corte é feita entre todas as desigualdades válidas para um poliedro mochila particular, que contém a região de admissibilidade original. A primeira referência apresenta bons resultados da experiência computacional, embora o sucesso desta abordagem dependa da estrutura particular do problema, que é um factor determinante na eficiência da resolução do problema de geração do corte.

Com a dissertação de doutoramento de Ceria [34], a programação disjuntiva tornou-se uma plataforma eficiente para a resolução de problemas práticos. Ceria desenvolveu um algoritmo puro de planos cortantes para programas lineares mistos binários, que usa o conceiro de *corte mais profundo* no sentido de uma certa medida de profundidade entre a solução óptima do programa linear relaxado e uma dada relaxação de programação disjuntiva. Os cortes resultantes da aplicação deste método passam a designar-se por *cortes lift-and-project*. É estabelecida a relação do trabalho de Balas em programação disjuntiva com as matrizes de cortes de Lovász e Schrijver [94], a hierarquia de relaxações para programação inteira mista de Sherali e Adams [116] e os cortes de intersecção de Balas [6]. No trabalho desenvolvido com Balas e Cornuéjols [16, 17] é provada a equivalência destes cortes com os cortes disjuntivos. São apresentados resultados teóricos sobre cortes obtidos a partir da disjunção 0-1 e um algoritmo de planos cortantes finito para problemas lineares mistos binários. É demonstrada a eficiência da implementação do método de *Lift-and-project* numa plataforma de *branch-and-cut* com o desenvolvimento do código MIPO para programas lineares inteiros mistos sem estrutura combinatória particular.

1.3 Estrutura da tese

A tese, além desta introdução, inclui três capítulos e três apêndices, cujo conteúdo se passa a descrever.

Nesta tese, mostra-se como é possível gerar cortes mais profundos para problemas com estrutura como é o caso, por exemplo, dos problemas de optimização combinatória. Para a resolução do problema de optimização subjacente à caracterização do corte mais profundo, serão usados algoritmos de feixe (do inglês, *bundle methods*). O desenvolvimento efectuado encontra-se fundamentalmente relacionado com o algoritmo proposto por P. Wolfe [128] que determina o ponto pertencente ao invólucro convexo de um conjunto finito de pontos que está mais próximo da origem, no sentido da norma Euclidiana. Assim, no Capítulo 2, baseados em argumentos disjuntivos através de algoritmos, propõe-se uma generalização do algoritmo de P. Wolfe [128] para qualquer métrica, diferenciável ou não. O algoritmo proposto é composto por um ciclo externo e um ciclo interno, termina ao fim de um número finito de iterações e gera a mesma sucessão de pontos que o algoritmo de P. Wolfe quando a métrica é a Euclidiana. Analisa-se a álgebra linear numérica inerente à sua implementação, distintamente para as normas l_2, l_1 e l_∞. O algoritmo foi implementado com o objectivo de resolver o problema de determinar o ponto pertencente ao invólucro convexo de um conjunto finito de pontos, com diferentes normas - l_1, l_2 e l_∞, e, como gerador de cortes disjuntivos mais separadores em problemas de programação inteira, tendo sido obtidos resultados numéricos para uma selecção de problemas teste da MIPLIB e da ORLIB.

No Capítulo 3, é elaborado um trabalho de síntese desenvolvido com base em dois trabalhos, nomeadamente, o de Cornuéjols e Li [43] sobre fechos elementares para programação inteira e o de Letchford [91] sobre cortes disjuntivos para optimização combinatória.

A obtenção da solução óptima de um problema linear inteiro ou inteiro misto através de métodos de planos cortantes obriga à geração de desigualdades que, embora não sejam válidas para a região admissível da relaxação linear subjacente, são válidas para o invólucro convexo da região de interesse. Na Secção 3.2, são recordados os procedimentos que permitem obter uma caracterização algébrica explícita desse invólucro, com base no trabalho de Cornuéjols e Li [43].

Uma das abordagens mais eficientes para resolver problemas de optimização combinatória com estrutura particular é a abordagem *branch-and-cut* (Padberg e G. Rinaldi [106], Caprara e Fishetti [30]), em que os problemas são formulados como programas lineares inteiros e resolvidos através de uma combinação de planos cortantes e *branch-and-bound*. Para um conjunto de problemas importantes, Letchford em [91] demonstrou que alguns cortes combinatórios bem conhecidos são válidos para deter-

minadas relaxações disjuntivas do problema de optimização combinatória subjacente. A identificação dessas relaxações permitirá obter cortes mais profundos de acordo com a metodologia apresentada no Capítulo 2. As provas apresentadas na Secção 3.3 usam argumentos primais e constituem uma alternativa às provas apresentadas em [91] cujo autor usa argumentos duais. Os cortes combinatórios analisados dizem respeito aos problemas de partição por cliques, corte máximo, subdigrafo acíclico, ordenamento linear, cobertura de conjuntos e caixeiro viajante assimétrico.

No Capítulo 4, procura-se pôr em prática algumas das metodologias sugeridas num problema específico de optimização combinatória. O problema do caixeiro viajante assimétrico (do inglês, *Asymmetric Traveling Salesman Problem* e, por isso, denotado ATSP) é um problema de optimização combinatória que tem vindo a servir de plataforma de inúmeros métodos de resolução. Este facto e a importância de descobrir bons limites inferiores para este problema motivaram a sua escolha para testar a implementação do algoritmo de planos cortantes proposto. Assim, é apresentada uma estratégia para obter um limite inferior para o valor óptimo do ATSP, melhor do que o que advém do problema de afectação. O algoritmo proposto é um método de primeira ordem baseado na função de penalidade exponencial que requer apenas a resolução sucessiva de problemas de afectação como subproblemas. Apresentam-se duas relaxações lineares disjuntivas do poliedro definido pelo invólucro convexo das soluções admissíveis do ATSP, que se baseiam no conhecimento de um circuito não Hamiltoniano ou de um clique que permite a identificação de um hiperplano separador entre esse poliedro e um ponto extremo do poliedro definido pelo invólucro convexo das soluções admissíveis do problema de afectação. Explica-se como determinar esse circuito ou clique e como reoptimizar a resolução paramétrica dos problemas de afectação inerentes à identificação desse hiperplano.

O problema de identificar restrições de eliminação de subpercursos para o ATSP é, conforme sugerido por Padberg e Rinaldi [105], uma aplicação do problema do corte mínimo global num grafo dirigido, que se descreve na Secção 4.4. É recordado o algoritmo de *empurrar-prefluxo* de Goldberg e Tarjan [66] para o problema de determinar o fluxo máximo entre dois vértices designados e o algoritmo de Hao e Orlin [74], que determina o corte global de capacidade mínima num grafo dirigido. O algoritmo de Hao e Orlin baseia-se no algoritmo *DetCorteMin(s)* proposto por Padberg e Rinaldi [105], que resolve a variante do problema de corte mínimo onde um vértice está fixo, e é, essencialmente, uma versão adaptada do algoritmo de *empurrar-prefluxo* de Goldberg e Tarjan. A descrição do algoritmo de *empurrar-prefluxo* de Goldberg e Tarjan seguiu o desenvolvimento de Hao e Orlin [74], pois as descrições no artigo original [66] ou no livro de Ahuja, Magnanti e Orlin [1, Capítulo 7] não permitem identificar, tão claramente, as modificações introduzidas por Hao e Orlin.

Na Secção 4.5, é descrito o problema de afectação e o método Húngaro, proposto

por Kuhn [86]. O método usa o algoritmo para determinação do emparelhamento de máxima cardinalidade num grafo bipartido, recordado na Secção 4.5.1, e cujo desenvolvimento algorítmico é, essencialmente, baseado no livro de Wolsey [129, Capítulo 4].

O capítulo final desta tese é dedicado à apresentação das conclusões principais e descrição de um conjunto de possíveis desenvolvimentos a serem trabalhados num futuro próximo.

No Apêndice A, apresentam-se os procedimentos para o fortalecimento de cortes disjuntivos propostos por Balas e Jeroslow [18] e utilizados nas aplicações descritas no Capítulo 2. No Apêndice B, descreve-se a regra de ramificação sugerida por Ryan e Foster [112] para a resolução de problemas de partição através do método *branch-and-bound*, e também aplicada por Ceria e Pataki [35] na geração de cortes disjuntivos. A regra de Ryan-Foster é aplicada na geração de cortes disjuntivos em problemas de partição, na implementação do algoritmo de P. Wolfe descrita no Capítulo 2. No Apêndice C, apresentam-se alguns resultados sobre fluxos e prefluxos que complementam o desenvolvimento da Secção 4.4.

1.4 Notação e alguns resultados básicos

O objectivo desta secção é fornecer o suporte matemático necessário à compreensão dos restantes capítulos. É apresentada a notação usual, seguida da revisão de alguns conceitos e resultados básicos da teoria poliedral, da programação inteira e da teoria dos grafos. Todos os termos são usados no sentido usual. A maioria da notação e terminologia que se apresentam foi retirada do livro de Schrijver [114]. Os conceitos topológicos que se usam são os presentes na literatura da análise convexa, por exemplo, nos livros de Urruty e Lemaréchal [77, 78].

O símbolo $\mathbb{R}^n$ denota o espaço real $n-$dimensional, e $\mathbb{R}^n_+$, o conjunto de vectores não negativos de $\mathbb{R}^n$. Os elementos de $\mathbb{R}^n$ são vectores linha ou coluna, devendo a distinção ser inferida pelo contexto ou pela variável que é utilizada (em geral, os vectores coluna são representados pelas letras do alfabeto latino, e os vectores linha e escalares, pelas letras do alfabeto grego).

Dada uma matriz $A \in \mathbb{R}^{m\times n}$, a_{ij} denota o elemento na linha i e coluna j, $A_{i\cdot}$ denota a sua $i-$ésima linha, e $A_{\cdot j}$ denota a sua $j-$ésima coluna. Do mesmo modo, $A_{I\cdot}$ denota a submatriz de A composta pelas linhas indexadas pelo conjunto de índices I, e $A_{\cdot J}$ denota a submatriz de A composta pelas colunas indexadas pelo conjunto de índices J. Dado um vector $x \in \mathbb{R}^n$, x_S denota o subvector de x indexado pelo conjunto de índices S; o número de elementos de S é denotado por $|S|$. As matrizes e os vectores

são indexados superiormente, como em A^i e x^i, devendo ser inferido pelo contexto se i representa a $i-$ésima potência ou o $i-$ésimo elemento de alguma sequência. Os escalares são indexados inferiormente, como em α_i. A sequência $x_1, x_2, \ldots, x_n, \ldots$ é denotada por $\{x_i\}_{i=1}^{\infty}$. A transposta de uma matriz A denota-se por A^T, e o determinante de uma matriz quadrada A por $det(A)$.

Dada uma matriz A, os escalares $\lambda_{\min}(A)$ e $\lambda_{\max}(A)$ denotam o menor e o maior valor próprio de A, respectivamente. A matriz A diz-se *definida positiva* se $\lambda_{\min}(A) > 0$ e *semidefinida positiva* se $\lambda_{\min}(A) \geqslant 0$.

Se α é um número real, então $\lfloor \alpha \rfloor$ denota o arredondamento para o inteiro imediatamente inferior de α, e $\lceil \alpha \rceil$ para o inteiro imediatamente superior de α. Se α e β são números reais, então $[\alpha, \beta]$ denota o conjunto $\{x \in \mathbb{R}\colon \alpha \leqslant x \leqslant \beta\}$, e (α, β), o conjunto $\{x \in \mathbb{R}\colon \alpha < x < \beta\}$.

Se o conjunto $S \subseteq \mathbb{R}^n$ é limitado superiormente, então existe o menor número real y tal que $x \leqslant y$ para todo o $x \in S$; y diz-se o *supremo* de S e denota-se por $\sup(P)$. Se S é não vazio mas limitado superiormente, então $\sup(S) = +\infty$. Similarmente, o ínfimo de S denota-se por $\inf(S)$ e se S é não vazio, mas limitado inferiormente, então $\inf(S) = -\infty$.

1.4.1 Teoria poliedral e optimização combinatória

Dado um conjunto $P \subseteq \mathbb{R}^n$, o *interior* de P denota-se por int (P), o *fecho* de P por cl (P) e a *fronteira* de P por fr (P). O *interior relativo* de P é o interior de P relativo ao invólucro afim de P e denota-se por ri (P). O *invólucro cónico* de P é o conjunto de todas as combinações finitas não-negativas (cónicas) de elementos do conjunto P e denota-se por cone (P). O *invólucro afim* de P é o conjunto de todas as combinações finitas afins de P e denota-se por aff (P). O *invólucro convexo* de P é o conjunto de todas as combinações finitas convexas de P e denota-se por conv (P). A dimensão de P é a dimensão do seu invólucro afim.

Seja $P \subseteq \mathbb{R}^n$ um conjunto convexo fechado não vazio. O *cone de recessão* de P é o conjunto dos vectores $d \in \mathbb{R}^n$ tal que, para qualquer $x \in P$ e para qualquer escalar $t \in \mathbb{R}_+$, o vector $x + td \in P$ e denota-se por P_∞.

A representação algébrica do conjunto $P \subseteq \mathbb{R}^n$ é da forma

$$P = \{x \in \mathbb{R}^n\colon G(x) \leqslant 0\},$$

em que $G\colon \mathbb{R}^n \to \mathbb{R}^m$ é o vector mapeamento, cujos componentes são funções convexas fechadas. O conjunto de *pontos extremos* ou *vértices* de P é o conjunto dos pontos

em P que não podem ser escritos como combinação convexa de dois quaisquer outros pontos também em P; denota-se por vert(P).

O conjunto $P \subseteq \mathbb{R}^n$ é um conjunto poliédrico ou um *poliedro* se e só se P é a intersecção de um número finito de semi-espaços, isto é,

$$P = \{x \in \mathbb{R}^n \colon Ax \leqslant b, Dx = d\}, \tag{1.1}$$

em que A, D são matrizes, e b, d são vectores de dimensões finitas adequadas. Um vector $d \in \mathbb{R}^n$ é uma *direcção* do poliedro P se, para qualquer $x \in P$ e para qualquer escalar $t \in \mathbb{R}_+$, o vector $x + td \in P$. O conjunto de todas as direcções de um poliedro P é um *cone poliedral* e denota-se por P_∞. Se P é um poliedro limitado e não vazio cuja única direcção é o vector nulo, então P diz-se um *polítopo*. Todo o poliedro P admite uma caracterização do tipo de Carathéodory da forma

$$P = \text{conv } (S) + \text{cone } (C), \tag{1.2}$$

em que S e C são conjuntos finitos. Reciprocamente, se P admite a caracterização (1.2), então P é um poliedro.

Um *hiperplano* de $\mathbb{R}^n$ é um conjunto $H = \{x \in \mathbb{R}^n \colon \alpha x = \beta\}$ para algum $(\alpha, \beta) \in \mathbb{R}^{n+1}$. O vector α diz-se o vector normal do hiperplano H, e o escalar β, o termo independente. Os semi-espaços fechados positivo e negativo de $\mathbb{R}^n$ são definidos por $H_+ = \{x \in \mathbb{R}^n \colon \alpha x \geqslant \beta\}$ e $H_- = \{x \in \mathbb{R}^n \colon \alpha x \leqslant \beta\}$, respectivamente. A desigualdade "$\alpha x \leqslant \beta$" diz-se uma *desigualdade válida* para o poliedro P se $\alpha x \leqslant \beta$ para todo o $x \in P$. O conjunto *polar*(P) é o conjunto de pontos $(\alpha, \beta) \in \mathbb{R}^{n+1}$ que definem desigualdades válidas para o poliedro P.

Uma *face* F de P é um conjunto poliédrico contido em P, para o qual existe alguma desigualdade "$\alpha x \leqslant \beta$" tal que $F = \{x \in P \colon \alpha x \leqslant \beta\} = \{x \in P \colon \alpha x = \beta\}$. Então, F diz-se uma face definida por "$\alpha x = \beta$". Uma face diz-se *própria* se $F \neq \emptyset$ e $F \subset P$. Uma *faceta* é uma face própria maximal ou, equivalentemente, uma face F é uma faceta se dim $(F) =$ dim $(P) - 1$. Existe uma correspondência um para um entre as desigualdades em (1.1) que não são redundantes e as facetas de P.

Dado um poliedro $P = \{x \in \mathbb{R}^n \colon Ax \leqslant b\}$, define-se o *invólucro convexo inteiro* de P como $P_I =$ conv $\{x \in \mathbb{Z}^n \colon x \in P\}$. O invólucro convexo inteiro de um poliedro é um poliedro se A e b forem racionais ou P for limitado. O poliedro $\overline{P}$ tal que $\overline{P} \supseteq P$ define uma *relaxação linear* de P.

Seja Q um conjunto finito de pontos $\{q_1, q_2, \ldots, q_m\}$ ou uma matriz $[q_1 q_2 \cdots q_m]$, com a distinção inferida pelo contexto. A *cardinalidade* de Q é o número de elementos do conjunto Q e denota-se por $|Q|$.

O *invólucro afim* de Q é o conjunto $\text{aff}\,(Q) = \{x \in \mathbb{R}^n \colon x = Qw, ew = 1\}$, em que e é um vector linha com todos os componentes iguais a 1; geometricamente, é uma variedade linear que passa por todos os pontos de Q. O *invólucro convexo* de Q é o conjunto $\text{conv}\,(Q) = \{x \in \mathbb{R}^n \colon x = Qw, ew = 1, w \geqslant 0\}$; geometricamente, é um polítopo e o menor conjunto convexo que contém Q. O *invólucro cónico* de Q é o conjunto $\text{cone}\,(Q) = \{x \in \mathbb{R}^n \colon x = Qw, ew = 1, w > 0\}$; geometricamente, é o menor cone poliedral que contém Q. O *interior relativo do invólucro convexo* de Q é o conjunto $\text{ri}\,(\text{conv}\,(Q)) = \{x \in \mathbb{R}^n \colon x = Qw, w > 0\}$; geometricamente, é o conjunto de pontos de Q que não estão contidos em nenhuma face própria de Q.

O conjunto Q diz-se *independente afim* se, para todo $q \in Q$, $q \notin \text{aff}\,(Q \setminus \{q\})$. Se Q é um conjunto independente afim, então todo o $x \in \text{aff}\,(Q)$ possui uma representação única da forma $x = Qw$ tal que $ew = \sum_{i=1}^{m} w_i = 1$, em que w é designado por vector das coordenadas baricêntricas de x relativamente a Q. A dimensão de Q é o número máximo de vectores independentes afim de Q menos um.

O poliedro K é um *polítopo* se e só se existe um conjunto finito Q tal que $K = \text{conv}\,(Q)$. Se $K = \text{conv}\,(Q)$, então Q é minimal se e só se Q é o conjunto dos pontos extremos de K. Por isso, a dimensão do poliedro K é a cardinalidade do maior conjunto independente afim contido em K menos um.

O conjunto polar (K) é o conjunto de pontos $(\alpha, \beta) \in \mathbb{R}^{n+1}$ que definem desigualdades válidas para o poliedro K. Uma face F do poliedro K é um conjunto poliédrico contido em K tal que existe $(\alpha, \beta) \in \mathit{polar}(K)$ com $F = \{x \in K \colon \alpha x = \beta\}$. Uma faceta de K é uma face de dimensão $\dim\,(K) - 1$. Um ponto extremo ou vértice $q \in K$ é uma face de dimensão zero . Para qualquer subconjunto $S \subseteq Q$, o conjunto $\text{conv}\,(S)$ é uma face de K tal que $\dim\,(\text{conv}\,(S)) = |S| - 1$.

Seja $\overline{x} \notin K = \text{conv}\,(Q)$. Então, existe um ponto único $\widetilde{x} \in K$ que está à distância Euclidiana mínima de $\overline{x}$ e que se caracteriza por uma das seguintes expressões equivalentes:

$$(\widetilde{x} - \overline{x})^T x \geqslant \|\widetilde{x} - \overline{x}\|^2 \quad (\forall x \in K),$$
$$(\widetilde{x} - \overline{x})^T q \geqslant \|\widetilde{x} - \overline{x}\|^2 \quad (\forall q \in Q).$$

Neste caso, a desigualdade válida para K, violada por $\overline{x}$ e que está à sua distância mínima é caracterizada por $\alpha = (\widetilde{x} - \overline{x})^T / \|\widetilde{x} - \overline{x}\|$ e $\beta = \|\widetilde{x} - \overline{x}\|$.

Seja f uma função convexa de $\mathbb{R}^n$ em $\mathbb{R}$. O subdiferencial de f em $x \in \mathbb{R}^n$, denota-se $\partial f(x)$, é o conjunto de todos os subgradientes de f em x (são vectores linha). Se, adicionalmente, f é contínua em x, então o conjunto $\partial f(x)$ é não vazio e compacto; se f é diferenciável em x então $\partial f(x)$ possui um único elemento denominado o gradiente de f em x e denota-se por $\nabla f(x)$.

Seja $\|\cdot\|$ uma qualquer norma em $\mathbb{R}^n$ e $\overline{x} \notin K = \text{conv}\,(Q)$, com Q finito; então existe pelo menos um ponto $\widehat{x} \in \text{conv}\,(Q)$ que está à distância mínima de $\overline{x}$. A unicidade só é garantida se a norma for continuamente diferenciável. Qualquer um daqueles pontos $\widehat{x}$ é caracterizado por: para $f(x) = \|x - \overline{x}\|$, existe um subgradiente $\alpha \in \partial f(\widehat{x})$ e um escalar $\beta = f(\widehat{x})$ tal que $\alpha x \leqslant \beta$ para todo o $x \in K$. Recorde-se que $\partial f(x) = \{\xi \in \mathbb{R}^n \colon \xi(x - \overline{x}) = \|x - \overline{x}\|, \|\xi\|_* \leqslant 1\}$, é um conjunto que, na maioria do casos de interesse (l_1, l_∞, l_p, Frobenius, etc.), é facilmente caracterizável em termos algébricos. A norma $\|\cdot\|_*$ é denotada por norma conjugada (ou dual) da norma $\|\cdot\|$ e é definida por $\|\alpha\|_* \equiv \max\{\alpha x \colon \|x\| \leqslant 1\}$. Exemplos de normas vectoriais em $\mathbb{R}^n$ são: a norma l_2 ou Euclidiana, $\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$, a norma l_1, $\|x\|_1 = \sum_{i=1}^n |x_i|$, e a norma l_∞, $\|x\|_\infty = \max_{i=1,\cdots,n} |x_i|$. A norma l_2 é dual de si própria, a norma l_1 é dual da norma l_∞ e *vice-versa*.

1.4.2 Teoria de grafos

Seguidamente, apresentam-se as definições dos termos de teoria de grafos que serão usados. Para consulta de definições adicionais e resultados teóricos, refira-se o livro de West [127].

Grafos não dirigidos

Seja $G = (V, E)$ um *grafo não dirigido* definido por um conjunto de vértices V e um conjunto de arestas E. Para identificar uma aresta usa-se a letra e ou o conjunto $\{i, j\}$ ou o par (i, j). Arestas são subconjuntos de V com cardinalidade dois; isto é, se $i, j \in V$ com $i \neq j$, então $e = \{i, j\}$ é uma aresta e i, j são as extremidades da aresta e. Denota-se por K_n um grafo *completo* com n vértices (isto é, $K_n = (V, E)$), em que $V = \{1, \ldots, n\}$ e $E = V \times V$.

Para qualquer vértice $v \in V$, o conjunto de arestas incidentes a v designa-se por *estrela de* v e denota-se por $\delta(v)$. O grau de um vértice v é igual ao número de arestas incidentes a v, isto é, $|\delta(v)|$.

Se $E' \subset E$, então $G(E')$ diz-se o *subgrafo* de G induzido pelo conjunto de arestas E' e, similarmente, se $V' \subset V$, então $G(V')$ diz-se o *subgrafo* de G induzido pelo conjunto de vértices V'. Um subconjunto de vértices $V' \subset V$ cujo subgrafo $G(V')$ é completo diz-se um *clique*.

Uma sequência $v_1 e_1 v_2 e_2 \cdots v_n e_n v_{n+1}$, em que $v_1, v_2, \ldots, v_n, v_{n+1}$ são vértices e $e_1, e_2, \ldots, e_n$ são arestas tais que $e_i = (v_i, v_{i+1})$ para $i = 1, \ldots, n$, designa-se por *caminho* de comprimento n. Um caminho diz-se *fechado* se $v_1 = v_{n+1}$. Um *circuito* de comprimento n $v_1 e_1 v_2 e_2 \cdots v_n e_n v_1$ é um caminho fechado.

Se o conjunto de vértices V puder ser particionado em dois conjuntos V_1 e V_2 tais que cada aresta de E tem um vértice em V_1 e outro em V_2, então G diz-se *bipartido* e denota-se por $G = (V_1, V_2, E)$. O grafo G é bipartido se só se não contém circuitos de tamanho ímpar.

Para qualquer subconjunto de vértices $S \subset V$, define-se

$$\begin{aligned} E(S) &= \{(i,j) \in E \colon i,j \in S\}, \\ \delta(S) &= \{(i,j) \in E \colon i \in S, j \in V \setminus S\}. \end{aligned}$$

O conjunto $\delta(S)$ diz-se um *corte* em G determinado por S. Se S é um conjunto singular, isto é, $S = \{v\}$, então usa-se a notação $\delta(v)$ como abreviatura de $\delta(\{v\})$. O conjunto de arestas $\delta(S)$ é um *corte-*(i,j) em G. Os vértices i, j dizem-se *separados* pelo corte $\delta(S)$ ou $\delta(S)$ é um corte-(i,j) se exactamente um dos vértices i ou j estiver em S.

Um subconjunto $\mathcal{M} \subseteq E$ diz-se um *emparelhamento* se não existirem duas arestas em $\mathcal{M}$ que possuam uma extremidade em comum. Um vértice $i \in V$ diz-se *exposto* (relativamente a um emparelhamento $\mathcal{M}$) se i não é extremidade de qualquer aresta de $\mathcal{M}$. Quando não existem vértices expostos, diz-se que o emparelhamento $\mathcal{M}$ é *perfeito*. Emparelhamentos perfeitos possuem cardinalidade $|V|/2$.

Diz-se que P é um *caminho alternado* relativamente a um emparelhamento $\mathcal{M}$ se P é um caminho simples definido por

$$P = \Big\{ \underbrace{\{v_0, v_1\}}_{e_1}, \underbrace{\{v_1, v_2\}}_{e_2}, \ldots, \underbrace{\{v_{p-1}, v_p\}}_{e_p} \Big\} \tag{1.3}$$

tal que $\{e_i, e_{i+1}\}$ intersecta $\mathcal{M}$ e $E \setminus \mathcal{M}$ para todo $i = 0, 1, \ldots, p-1$. Recorde-se que, sendo P um caminho simples, $\{v_0, v_1, \ldots, v_p\}$ é um conjunto de vértices distintos.

Diz-se que P é um *caminho aumentado* relativamente a um emparelhamento $\mathcal{M}$ se P é um caminho alternado definido por (1.3) e v_0, v_p são dois vértices expostos. Consequentemente, p é ímpar, $e_1, e_3, \ldots, e_p \in E \setminus \mathcal{M}$, e $e_2, e_4, \ldots, e_{p-1} \in \mathcal{M}$.

A *matriz de adjacência* de um grafo não dirigido $G = (V, E)$ é uma matriz com ambas linhas e colunas indexadas por V, em que a entrada na posição (v, e) é igual ao número de arestas que conectam os vértices v e e. A *matriz de incidência* de G é uma matriz de zeros e uns com linhas e colunas indexadas por V e E, respectivamente, em que a entrada na posição (v, e) é um se e só se o vértice v é incidente com a aresta e.

Grafos dirigidos

Seja $G = (V, E)$ um *grafo dirigido* ou *digrafo* definido por um conjunto de vértices V e um conjunto de arcos $E \subset (V \times V)$. Para qualquer subconjunto de vértices $S \subseteq V$, define-se

$$\begin{aligned} E(S) &= \{(i,j) \in E \colon i,j \in S\}, \\ \delta^+(S) &= \{(i,j) \in E \colon i \in S, j \in V \setminus S\} \quad = \delta^-(V \setminus S), \\ \delta^-(S) &= \{(i,j) \in E \colon i \in V \setminus S, j \in S\} \quad = \delta^+(V \setminus S). \end{aligned}$$

Se S for um conjunto singular, isto é, $S = \{j\}$, então utiliza-se a notação $\delta^+(j)$ ou $\delta^-(j)$, respectivamente. Se $S = \{1,2\}$, então $E(S) = \{(i,j) \in E \colon i,j \in \{1,2\}\}$, $\delta^+(S) = \{(i,j) \in E \colon i \in \{1,2\}, j \in V \setminus \{1,2\}\}$ e $\delta^-(S) = \{(i,j) \in E \colon i \in V \setminus \{1,2\}, j \in \{1,2\}\}$.

Uma sequência $v_1\ e_1\ v_2\ e_2 \cdots v_n\ e_n\ v_{n+1}$, em que $v_1, v_2, \ldots, v_n, v_{n+1} \in V$ e $e_1, e_2, \ldots, e_n \in E$ são tais que $e_i = (v_i, v_{i+1})$ para $i = 1, \ldots, n$, designa-se por *caminho* (dirigido) de comprimento n. Um caminho diz-se *fechado* se $v_1 = v_{n+1}$. Um *ciclo* de comprimento n, $v_1\ e_1\ v_2\ e_2 \cdots v_n\ e_n\ v_1$, é um caminho fechado.

Seja $G' = (V', E')$ um grafo dirigido, em que $V' = \{v_1, \ldots, v_p\} \subseteq V$ e $E' = \{(v_1, v_2), (v_2, v_3), \ldots, (v_p, v_1)\} \subseteq E$. Então, G' diz-se um *ciclo Hamiltoniano* se $p = n$ ou diz-se um *subpercurso* se $p < n$.

Seja $S = \{v_1, v_2, \ldots, v_k\} \subseteq V$ um conjunto de vértices distintos. Se

$$E(S) = \{(v_1, v_2), (v_2, v_3), \ldots, (v_{k-1}, v_k)\} \subseteq E,$$

então $E(S)$ designa-se por *caminho dirigido* em G. Se

$$E(S) = \{(v_1, v_2), (v_2, v_3), \ldots, (v_{k-1}, v_k), (v_k, v_1)\} \subset E,$$

então $E(S)$ designa-se por *ciclo dirigido* ou *circuito* em G. Se o grafo G não possui ciclos dirigidos, então diz-se *acíclico*. Se cada par de vértices em S é adjacente em G, então S diz-se um *clique* de G.

A matriz de incidência de um digrafo $G = (V, E)$ é uma matriz com linhas e colunas indexadas por V e E, respectivamente, em que a entrada na posição (v, e) é -1, +1 ou 0 se o vértice v é a cabeça de e, a cauda de e ou nenhum dos casos, respectivamente.

Seja T um conjunto de arcos $(i,j) \in E$. Se, para todo o par de vértices $i, j \in V$ distintos, T contém ou o arco (i,j) ou (j,i), mas não ambos, então T diz-se um *torneio*. Um torneio sem ciclos dirigidos diz-se acíclico.

Seja $G = (V, E)$ um grafo dirigido constituído por uma sequência ordenada de m ciclos dirigidos distintos denotados por $C_1, \ldots, C_m$. O grafo G diz-se um grafo *escada de Möbius* se forem verificadas as seguintes condições:

1. o grau de cada vértice em V é pelo menos 3;
2. m é um inteiro ímpar tal que $m \geqslant 3$;
3. $|C_i| \in \{3, 4\}$ para $i = 1, \ldots, m$.
4. Seja (C_i, C_{i+1}), para $i = 1, \ldots, m$, um par de ciclos dirigidos adjacentes, em que $C_{m+1} = C_1$. Então, $C_i \cap C_{i+1} = e_i \in E$, para $i = 1, \ldots, m$, e $C_i \cap C_j = \emptyset$ para $j \in \{1, \ldots, i-1, i+1, \ldots, m\}$.
5. Seja (C_i, C_j) com $i < j$ um qualquer par de ciclos dirigidos que possuem um vértice comum $v_i \in V$. Então, v_i pertence a todos os ciclos dirigidos $C_i, \ldots, C_j$ ou $C_j, \ldots, C_m, C_1, \ldots, C_i$.
6. Seja C_j, para $j = 1, \ldots, m$ um qualquer ciclo dirigido e $J = \{1, \ldots, m\} \cap (\{j-2, j-4, \ldots\} \cup \{j+1, j+3, \ldots\})$ um conjunto de índices. Então, $E \setminus \{e_i \colon i \in J\}$ contém um só ciclo dirigido.

Capítulo 2

Projectar: dual de Separar

2.1 Introdução

Conhecido um ponto $\overline{x}$ de $\mathbb{R}^n$ e um conjunto finito de polítopos $K^i \subset \mathbb{R}^n$, $i \in I$, considera-se o problema de obter o hiperplano *mais separador* entre $\overline{x}$ e o polítopo K definido implicitamente através de

$$K \equiv \text{conv} \left(\bigcup_{i \in I} K^i \right), \tag{2.1}$$

onde conv $(\cdot)$ denota invólucro convexo. O hiperplano *mais separador* entre $\overline{x}$ e K é caracterizado pela solução dual óptima no seguinte par de problemas

$$\left.\begin{array}{ll} \min & \|x - \overline{x}\| \\ \text{s.a} & x \in K \end{array}\right\} = \left\{\begin{array}{ll} \max & \beta - \alpha\overline{x} \\ \text{s.a} & (\alpha, \beta) \in \text{polar } (K),\ \|\alpha\|_* \leqslant 1, \end{array}\right. \tag{2.2}$$

onde $\|\cdot\|$ e $\|\cdot\|_*$ designam normas vectoriais mutuamente conjugadas, e o conjunto polar $(K) \subseteq \mathbb{R}^{n+1}$ designa o conjunto dos $(n+1)$-uplos (α, β) tais que "$\alpha x \geqslant \beta$" é uma desigualdade válida para K. A igualdade (2.2) traduz a dualidade entre determinar a projecção de $\overline{x}$ sobre K (designado por problema primal) e determinar o hiperplano mais separador entre $\overline{x}$ e K (problema dual).

O interesse na resolução desta classe de problemas de optimização é motivado pela geração de hiperplanos mais separadores em programação inteira. Por exemplo, quando na resolução de problemas de programação inteira (ou inteira mista), $\overline{x}$ é minimizante de uma função linear cx sobre uma relaxação $\overline{K}$ de um conjunto K, e interessa obter uma desigualdade válida para K e inválida em $\overline{x}$. Em geral, o problema de minimização (2.2) é tão difícil quanto o próprio problema de programação inteira que se pretende

resolver. Por isso, na prática, a separação deve ocorrer entre o ponto $\overline{x}$ e uma outra relaxação de K definida com base em argumentos disjuntivos (por exemplo, $x_j = 0$ ou $x_j = 1$) que não contenha $\overline{x}$.

Neste contexto, duas questões científicas decorrem naturalmente. Que tipo e número de disjunções se devem considerar em (2.2)? Um maior número de disjunções pode favorecer a obtenção de cortes mais profundos. Contudo, a caracterização algébrica explícita de K pode ser impeditiva em termos de espaço de memória e/ou de velocidade de convergência. Que norma vectorial se deve considerar em (2.2)? Existe alguma evidência teórica, ainda não observável na prática, de que normas como a Euclidiana, por exemplo, poderão proporcionar convergência mais rápida ao algoritmo de planos cortantes, veja-se Soares [120], por exemplo. Mais, a resolução de (2.2) pode obrigar à resolução de problemas não lineares e/ou não diferenciáveis, quadráticos no caso da norma Euclidiana, e para os quais não é óbvia a escolha dos melhores algoritmos que aproveitem a estrutura específica do problema em causa. Por outro lado, tem-se observado experimentalmente que a norma escolhida em (2.2) influencia a esparsidade do corte assim caracterizado, o que também parece ser um aspecto relevante na eficácia do algoritmo de planos cortantes. Nesse aspecto, os cortes mais separadores baseados na norma Euclidiana tendem ser menos esparsos do que cortes baseados em normas poliedrais como as normas l_1 e l_∞.

A dualidade (2.2) sugere a utilização de algoritmos que, com o propósito de determinar um hiperplano separador, possam utilizar de forma eficiente a estrutura combinatória de K. Por exemplo, quando a norma é a Euclidiana, o algoritmo proposto por P. Wolfe [128] resolve (2.2) através da resolução sucessiva de problemas lineares sobre K. Sendo K o invólucro convexo de conjuntos poliédricos, a optimização de uma função linear sobre K pode ser massivamente paralelizável ou eficientemente resolvida por algoritmos do tipo paramétrico. Neste capítulo, é proposta uma generalização do algoritmo de P. Wolfe para a resolução do problema (2.2) com qualquer norma, diferenciável ou não, atendendo ao interesse dessa abordagem na geração de planos cortantes baseados em argumentos disjuntivos. São também apresentados resultados computacionais obtidos com uma implementação combinada de MATLAB e CPLEX.

Na próxima secção, propõe-se uma generalização do algoritmo de P. Wolfe para a resolução do problema de obter o hiperplano mais separador entre um ponto conhecido e o invólucro convexo de conjuntos poliedrais definido com base em argumentos disjuntivos com qualquer norma, diferenciável ou não. Apresenta-se uma prova da sua convergência finita e analisa-se a álgebra linear numérica inerente à sua implementação, distintamente para as normas l_2, l_1 e l_∞. Na Secção 2.4, apresentam-se alguns resultados computacionais da implementação do algoritmo proposto para a resolução do problema de determinar o ponto pertencente ao invólucro convexo de um conjunto finito de pontos, com diferentes normas - l_1, l_2 e l_∞, e, como gerador de cortes, para a

resolução de problemas de optimização discreta com diversos tipos de disjunções e a norma Euclideana.

2.2 Uma generalização do algoritmo de P. Wolfe

P. Wolfe propôs, em [128], um algoritmo para determinar o ponto pertencente ao invólucro convexo de um conjunto finito de pontos $K^0 \equiv \{p_1, p_2, \ldots, p_m\} \subset \mathbb{R}^n$, aqui denotado por $K \equiv \text{conv}\ (K^0)$, que está mais próximo da origem no sentido da norma Euclidiana. Algebricamente, o algoritmo resolve o problema de optimização

$$\begin{array}{ll} \min & \|x\| \\ \text{s.a} & x = \displaystyle\sum_{k=1}^{m} w_k p_k, \quad \sum_{k=1}^{m} w_k = 1, \quad w_k \geqslant 0, \quad k = 1, 2, \ldots, m, \end{array} \tag{2.3}$$

onde $\|\cdot\|$ denota norma Euclidiana. O algoritmo de P. Wolfe não requer o conhecimento explícito e antecipado de K^0, antes manipulando subconjuntos de K^0 gerados iterativamente - neste aspecto, o algoritmo pode ser considerado um algoritmo de geração de colunas. A cardinalidade desses subconjuntos nunca ultrapassa $\dim\ (K) + 1$, o que torna a abordagem atractiva quando $|K^0|$ é elevado.

Propõe-se, na Figura 2.1, uma generalização do algoritmo de P. Wolfe para a resolução do problema (2.3) com qualquer norma. Tal como aquele, o algoritmo que se propõe é composto por um ciclo externo (Ciclo maior) e um ciclo interno (Ciclo menor) com invariantes semelhantes, terminando ao fim de um número finito de passos e gerando a mesma sucessão de pontos quando a norma é a Euclidiana.

Cada iteração do Ciclo maior inicia-se com o conhecimento de um par de subconjuntos $P_1, P_2 \subseteq K^0$, com P_1 independente afim, e um ponto $X \in \text{ri}\ \text{conv}\ (P_1)$ tal que $X \in \arg\min\{\|x\| : x \in \text{aff}\ (P_1)\}$. Como se mostra no Lema 2.1, o subconjunto P_2 é necessário para lidar com a averiguação de optimalidade. Se X verifica a condição necessária e suficiente de optimalidade sobre K (conforme Secção 1.4.1) com um dado subgradiente, então o algoritmo termina. Caso contrário, é identificado um novo ponto $\widehat{p} \in K^0$ que deve ser acrescentado a $P \equiv P_1 \cup P_2$. Quando a norma é não diferenciável, existe a dificuldade adicional da identificação do subgradiente que valida a optimalidade de X.

Se a inserção de $\widehat{p}$ em P permite identificar uma direcção de descida, então $\widehat{p}$ é inserido em P_1 (possivelmente com outros pontos de P_2) e P_2 fica vazio. Prossegue a execução do Ciclo menor que consiste na eliminação sucessiva de pontos de P_1 até que seja alcançado $X \in \text{ri}\ \text{conv}\ (P_1)$ tal que $X \in \arg\min\{\|x\| : x \in \text{aff}\ (P_1)\}$. Se a inserção de $\widehat{p}$ não favorece a identificação de uma direcção de descida, então $\widehat{p}$ é

inserido em P_2, e é imediatamente iniciada uma nova iteração do Ciclo maior sem qualquer redefinição de X, mas com um novo subgradiente. Esta última situação, que favorecerá o aparecimento do tal subgradiente que assegura a optimalidade de X ao fim de um número finito de passos, só acontece quando a norma não é continuamente diferenciável.

Uma dada iteração do Ciclo maior é designada por *séria* se $\|X\|$ decresce nessa iteração, caso contrário essa iteração é designada por *nula*. Esta notação também é adoptada na descrição de alguns algoritmos para optimização convexa não diferenciável, descritos por Urruty e Lemaréchal [78], por exemplo. As iterações nulas de um Ciclo maior são constituídas por nenhuma execução do Ciclo menor, enquanto que as iterações sérias são constituídas por pelo menos uma execução do Ciclo menor.

A próxima sequência de lemas destina-se a demonstrar a convergência finita do algoritmo proposto. É consequência imediata de resultados clássicos que $X \in \arg\min \{\|x\| \colon x \in \operatorname{aff}(P_1)\}$ se e só se existe $\alpha \in \partial\|\cdot\|(X)$ tal que $\alpha X = \alpha p$ para todo $p \in P_1$. Em particular, se $P_1 = \{X\}$, então isso é verdade com qualquer $\alpha \in \partial\|\cdot\|(X)$. O Lema 2.1 assegura a veracidade, para algum α, da afirmação entre parêntesis no início de cada uma das iterações do Ciclo maior.

Lema 2.1 *Seja $\widehat{x} = \sum_{p^i \in P} \widehat{w}_i p^i$ admissível para (2.4). Então, $\widehat{x}$ é solução óptima para (2.4) se e só se existe $\widehat{\alpha} \in \partial\|\cdot\|(\widehat{x})$ tal que*

$$\begin{aligned} &\widehat{\alpha}(p - \widehat{x}) = 0 \quad \text{para todo } p \in P_1 \cup \{p^i \in P_2 \colon \widehat{w}_i > 0\}, \\ &\widehat{\alpha}(p - \widehat{x}) \geqslant 0 \quad \text{para todo } p \in \{p^i \in P_2 \colon \widehat{w}_i = 0\}. \end{aligned}$$

Mais,

1. *se $\|\widehat{x}\| < \|X\|$, então $\widehat{w}_0 > 0$ para $p^0 = \widehat{p} \in P_2$;*
2. *se $\|\widehat{x}\| = \|X\|$, então $\widehat{\alpha} \in \partial\|\cdot\|(X)$.*

Demonstração: Seja $C \subseteq \mathbb{R}^n$ o conjunto dos pontos x admissíveis para (2.4). Sendo C um conjunto convexo fechado, $\widehat{x}$ é óptimo para $\min\{\|x\| \colon x \in C\}$ se e só se $0 \in \partial\|\cdot\|(\widehat{x}) + N_C(\widehat{x})$, com $N_C(\widehat{x})$ a denotar o cone normal de C em $\widehat{x}$ (veja-se, por exemplo, Urruty e Lemaréchal [77]). Como

$$N_C(\widehat{x}) = \left\{ s \in \mathbb{R}^n \colon \begin{array}{ll} s(p - \widehat{x}) = 0, & p \in P_1 \cup \{p^i \in P_2 \colon \widehat{w}_i > 0\}, \\ s(p - \widehat{x}) \leqslant 0, & p \in \{p^i \in P_2 \colon \widehat{w}_i = 0\} \end{array} \right\}$$

conclui-se a primeira parte do lema. Agora, suponha-se que $\|\widehat{x}\| < \|X\|$. Como $\alpha \in \partial\|\cdot\|(X)$ e X é admissível para (2.4),

$$\|\widehat{x}\| \geqslant \|X\| + \alpha(\widehat{x} - X) > \|\widehat{x}\| + \alpha(\widehat{x} - X), \tag{2.6}$$

Inicialização:

Escolher $P_2 = \emptyset$, $P_1 \subseteq K^0$ independente afim e um vector $X \in \text{ri conv } (P_1)$ tal que $X \in \arg\min\{\|x\| : x \in \text{aff } (P_1)\}$.

Ciclo maior:

(Seja $P = P_1 \cup P_2 \subseteq K^0$ e $X \equiv P_1\overline{w} \in ri$ conv (P_1), com $\overline{w} > 0$ e $e\overline{w} = 1$, tal que existe $\alpha \in \partial\|\cdot\|(X)$ satisfazendo $\alpha(p - X) = 0$ para todo $p \in P_1$, e $\alpha(p - X) \geqslant 0$ para todo $p \in P_2$)

Determinar $\widehat{p} \in \arg\ \min\{\alpha x : x \in K^0\}$.

Se $\alpha\widehat{p} = \alpha X$
então STOP: $X \in \arg\ \min\{\|x\| : x \in K\}$.
senão $P_2 := P_2 \cup \{\widehat{p}\}$.

Determinar uma solução óptima $\widehat{x} \equiv \sum_{p^i \in P} \widehat{w}_i p^i$ do problema

$$\begin{array}{ll} \min & \|x\| \\ \text{s.a} & x = \sum_{p^i \in P} w_i p^i, \quad \sum_{p^i \in P} w_i = 1, \quad w_i \geqslant 0, \quad i : p^i \in P_2. \end{array} \tag{2.4}$$

Se $\|\widehat{x}\| = \|X\|$
então Repetir Ciclo maior.
senão $\theta := \min\left(1,\ \max\{t : \overline{w}_i + t(\widehat{w}_i - \overline{w}_i) \geqslant 0, \quad i : p^i \in P_1\}\right)$
$X := X + \theta(\widehat{x} - X)$.
$P_1 := P_1 \cup \{p^i \in P_2 : \widehat{w}_i > 0\}$ e $P_2 := \emptyset$.
Redefinir P_1 de modo a ser independente afim e $X \in \text{ri } (\text{conv } (P_1))$.

Ciclo menor:

Determinar uma solução óptima $\widehat{x} \equiv \sum_{p^i \in P_1} \widehat{w}_i p^i$ do problema

$$\begin{array}{ll} \min & \|x\| \\ \text{s.a} & x \in \text{aff } (P_1) \end{array} \tag{2.5}$$

Se $\|\widehat{x}\| = \|X\|$
então Repetir o Ciclo maior.
senão $\theta := \min\left(1,\ \max\{t : \overline{w}_i + t(\widehat{w}_i - \overline{w}_i) \geqslant 0, \quad i : p^i \in P_1\}\right)$
$X := X + \theta(\widehat{x} - X)$.
Redefinir P_1 de modo que $X \in \text{ri conv } (P_1)$.
Se $\widehat{x} \in \text{conv } (P_1)$
então Repetir o Ciclo maior.
senão Repetir o Ciclo menor.

Figura 2.1: *Algoritmo modificado de P. Wolfe.*

pelo que $\alpha(\widehat{x} - X) < 0$. Por outro lado,

$$\alpha(\widehat{x} - X) = \sum_{p^i \in P_1} \widehat{w}_i \alpha(p^i - X) + \sum_{p^i \in P_2 \setminus \{\widehat{p}\}} \widehat{w}_i \alpha(p^i - X) + \widehat{w}_0 \alpha(\widehat{p} - X) \geqslant \widehat{w}_0 \alpha(\widehat{p} - X),$$

devido à propriedade de α no início do Ciclo maior. Por isso, também $\widehat{w}_0 \alpha(\widehat{p} - X) < 0$. Como $\alpha\widehat{p} < \alpha X$, por definição de $\widehat{p}$, então $\widehat{w}_0 > 0$. Finalmente, se $\|\widehat{x}\| = \|X\|$, então $\widehat{\alpha} \in \partial\|\cdot\|(X)$ é consequência de [29, Lema 2(a)], o que conclui a demonstração. □

No caso de $f(x) \equiv \|x\|$ ser uma norma continuamente diferenciável, $\alpha = \nabla f(x)$ e, portanto, $\alpha\,(\widehat{p} - X) < 0$ implica $\|\widehat{x}\| < \|X\|$, pelo que não existirão iterações nulas. Em particular, no problema (2.4), $P_2 = \{\widehat{p}\}$ sempre. Como, pelo Lema 2.1, $\widehat{w}_0 > 0$, então o problema (2.4) é essencialmente um problema sem restrições e, portanto, equivalente a um adequado sistema de equações lineares que, como é possível demonstrar, tem solução única.

No caso de $f(x) \equiv \|x\|$ ser uma norma l_1 ou l_∞, o problema (2.4) é equivalente a um adequado problema linear. Na Secção 2.3.2, explica-se como obter $\widehat{\alpha}$ a partir de informação dual básica. Por 2. do Lema 2.1, caso $\|\widehat{x}\| = \|X\|$, este vector $\widehat{\alpha}$ servirá para iniciar uma nova iteração do Ciclo maior.

Lema 2.2 *Se, após a resolução de (2.4),* $\|\widehat{x} \equiv \sum_{p^i \in P} \widehat{w}_i p^i\| < \|X \equiv \sum_{p^i \in P_1} \overline{w}_i p^i\|$, *então:*

1. $\theta \equiv \min\,(1,\ \max\{t\colon \overline{w}_i + t(\widehat{w}_i - \overline{w}_i) \geqslant 0$ *para todo* $i\colon p^i \in P_1\}) \in (0, 1]$;
2. $X + \theta(\widehat{x} - X) \in$ *conv* $\ (P_1' \equiv P_1 \cup \{p^i \in P_2\colon \widehat{w}_i > 0\})$, *e*
3. $\|X + \theta(\widehat{x} - X)\| < \|X\|$.

Demonstração: O facto de $\theta \in (0, 1]$ decorre de $\overline{w}_i > 0$ para todo $i\colon p^i \in P_1$. Portanto, para todo o escalar $t \geqslant 0$,

$$X + t(\widehat{x} - X) = \sum_{p^i \in P_1} \left(\overline{w}_i + t(\widehat{w}_i - \overline{w}_i)\right) p^i + \sum_{p^i \in P_2} (t\widehat{w}_i)\, p^i.$$

Como $\widehat{w}_i \geqslant 0$ para $i\colon p^i \in P_2$, e $\overline{w}_i > 0$ para $i\colon p^i \in P_1$, tem-se $X + t(\widehat{x} - X) \in$ conv (P_1') para todo $t \in [0, \theta]$. Finalmente, $\|X + \theta(\widehat{x} - X)\| = \|(1 - \theta)X + \theta\widehat{x}\| \leqslant (1 - \theta)\|X\| + \theta\|\widehat{x}\| < (1 - \theta)\|X\| + \theta\|X\| = \|X\|$. □

Após a resolução de (2.4) e no caso de $\|\widehat{x}\| < \|X\|$, ocorre uma redefinição de X e de P_1, de modo a que a definição de θ assegura $X \in$ conv (P_1), e, atendendo a 1. do Lema 2.1, pelo menos um novo vector é inserido em P_1. O próximo resultado diz respeito à independência afim de P_1.

Lema 2.3 *Suponha-se que* $\widehat{x} = \sum_{p^i \in P} \widehat{w}_i p^i$ *para algum* $\widehat{w}$ *tal que* $\widehat{w}_i \geqslant 0$ *para todo* $i\colon p^i \in P_2$, *e que* P_1 *é independente afim. Se* $\widehat{w}$ *é ponto extremo do poliedro*

$$W = \left\{ w\colon \widehat{x} = Pw, w_i \geqslant 0, \quad i\colon p^i \in P_2 \right\},$$

então $P_1 \cup \{p^i \in P_2\colon \widehat{w}_i > 0\}$ *é independente afim.*

Demonstração: Suponha-se que $P_1 \cup \{p^i \in P_2\colon \widehat{w}_i > 0\}$ não é independente afim. Então, existe $\overline{w} \neq 0$ tal que $\overline{w}_i = 0$ para todo $i\colon p^i \in P_2$ e $\widehat{w}_i = 0$, e

$$\sum_{p^i \in P} \overline{w}_i p^i = 0, \quad \sum_{p^i \in P} \overline{w}_i = 0.$$

Para $t > 0$ e suficientemente pequeno, $\widehat{w} + t\overline{w}$ e $\widehat{w} - t\overline{w}$ pertencem a W. Chega-se a um absurdo, pois $\widehat{w}$ é combinação convexa desses dois pontos de W. □

Portanto, se a solução óptima de (2.4) estiver caracterizada adequadamente, então o conjunto P_1 mantém-se independente afim durante todo o algoritmo, sem que isso seja imposto artificialmente. Note-se que, em geral, se $X \in \text{conv}\ (P_1)$, então um procedimento análogo ao que está implícito na demonstração do Lema 2.3 permite encontrar $P_1' \subseteq P_1$ independente afim e tal que $X \in \text{ri conv}\ (P_1')$.

Na primeira iteração de uma passagem pelo Ciclo menor, a solução óptima $\widehat{x}$ do problema (2.4) permanece óptima em (2.5) nessa iteração. A demonstração do Lema 2.4 é análoga à do Lema 2.2.

Lema 2.4 *Se, após a resolução de (2.5),* $\|\widehat{x}\| < \|X\|$ *e* $\widehat{x} \notin$ *conv* (P_1), *então*

$$\theta \equiv \min\left(1,\ \max\{t\colon X + t(\widehat{x} - X) \in \textit{conv}\ (P_1)\}\right) \in (0,1),$$
$$\|X + \theta(\widehat{x} - X)\| < \|X\|,$$

e P_1 *fica com menos um elemento, pelo menos.*

Seguidamente, é demonstrada a convergência finita do algoritmo proposto.

Teorema 2.1 *O algoritmo da Figura 2.1 termina num número finito de iterações dos Ciclos maior e menor.*

Demonstração: Pelo Lema 2.4, em cada iteração do Ciclo maior, ocorre um número finito de iterações do Ciclo menor. Em cada iteração (com a possível excepção da primeira e da última), o conjunto P fica com menos um elemento. Por isso, basta provar que o algoritmo termina num número finito de iterações do Ciclo maior.

Primeiro, prova-se que existe um número finito de iterações nulas consecutivas. Entre duas iterações nulas consecutivas, P_1 mantém-se constante, e P_2 tem mais um elemento. Como K^0 é finito, esta sequência não pode decorrer indefinidamente. Segundo, na iteração seguinte a uma iteração séria, $\|X\| = \min\{\|x\| \colon x \in \text{aff}\,(P_1)\}$ e $X \in \text{ri conv}\,(P_1)$. Entre duas iterações sérias consecutivas, $\|X\|$ decresce estritamente. Portanto, de iteração séria em iteração séria, os conjuntos P_1 não podem repetir-se. Como existem em número finito, conclui-se que existe um número finito de iterações sérias e também de todo o algoritmo. □

2.3 Álgebra linear numérica

Nesta secção, explica-se como se devem organizar os cálculos algébricos envolvidos na implementação do algoritmo descrito na Figura 2.1 na resolução do problema de minimização em (2.2). Duas características desse problema não são capturadas em (2.3), nomeadamente, a função objectivo é $\|x - \overline{x}\|$ e K^0, o conjunto dos pontos extremos de K, não é conhecido explicitamente. Mas, como

$$\begin{array}{ll} \min & \|x - \overline{x}\| \\ \text{s.a} & x \in K \end{array} = \begin{array}{ll} \min & \|y\| \\ \text{s.a} & y \in K - \overline{x} \end{array} \tag{2.7}$$

e $(K - \overline{x})^0 = K^0 - \overline{x}$, então o algoritmo proposto também resolve o problema de minimização em (2.2) com devidos ajustamentos.

Comece-se por analisar a primeira escolha para o conjunto P_1 que dá início ao algoritmo. Em geral, uma escolha possível é $P_1 := \{p - \overline{x}\}$, com p um qualquer ponto extremo de K. No entanto, quando já é conhecido um conjunto $Q_1 = \{q^i\} \subseteq K^0$ proveniente de, por exemplo, uma prévia resolução de (2.7) com um $\overline{x}$ diferente, pode-se inicializar P_1 de modo a acelerar o algoritmo. Primeiro, obtém-se o vector $X = \sum_i w_i q^i \in \arg\min\{\|x - \overline{x}\| \colon x \in \text{conv}\,(Q_1)\}$, usando um algoritmo adequado, após o que se define $P_1 = \{q^i \in Q_1 \colon w_i > 0\} - \overline{x}$ de modo a que P_1 seja independente afim e $X - \overline{x} \in \text{ri conv}\,(P_1)$.

Outro aspecto a ter em conta na implementação do algoritmo proposto é a determinação do novo ponto $\widehat{p} \in K^0 - \overline{x}$. Se K é da forma (2.1), então

$$\begin{array}{ll} \min & \alpha y \\ \text{s.a} & y \in K^0 - \overline{x} \end{array} = \min_{i \in I} \left(\begin{array}{ll} \min & \alpha x \\ \text{s.a} & x \in K^i \end{array} \right) - \alpha \overline{x}, \tag{2.8}$$

pelo que $\widehat{p} = \widehat{q}^k - \overline{x}$, com $\widehat{q}^k$ ponto extremo de um dos políticos $K^k, k \in I$. Nas próximas subsecções, analisa-se a álgebra linear numérica inerente à implementação do algoritmo proposto, distintamente para as normas l_2, l_1 e l_∞.

2.3.1 Norma Euclidiana

Dado um conjunto $Q_1 \subseteq K^0$ tal que $|Q_1|$ não é elevada, o problema $\min\{\|x - \overline{x}\| : x \in \text{conv } (Q_1)\}$ pode ser resolvido pelo algoritmo NNLS proposto por Lawson e Hanson [90] para a resolução de problemas de mínimos quadrados com restrições de não negatividade. O algoritmo NNLS está disponível como função interna do MATLAB e também está disponível em FORTRAN em `http://www.netlib.org/lawson-hanson/all`. O algoritmo NNLS é um método de restrições activas baseado na decomposição QR.

O Lema 2.5 mostra que o problema (2.9), abaixo, está relacionado com o problema de mínimos quadrados com restrições de não negatividade (2.10) e justifica a utilização da rotina NNLS. Considerem-se então os problemas

$$\begin{array}{lll} \min & \|x-\overline{x}\|^2 & = \min \quad \|Q_1 v - \overline{x}\|^2 \quad = \min \quad \|(Q_1 - \overline{x}e)v\|^2 \\ \text{s.a} & x \in \text{conv } (Q_1) & \text{s.a} \quad ev = 1, v \geqslant 0 \qquad \text{s.a} \quad ev = 1, v \geqslant 0, \end{array} \tag{2.9}$$

$$\begin{array}{lll} \min & \left\| \left[\begin{array}{c} Q_1 - \overline{x}e \\ e \end{array} \right] u - \left[\begin{array}{c} 0 \\ 1 \end{array} \right] \right\|^2 & = \quad \min \quad \|(Q_1 - \overline{x}e)u\|^2 + (eu-1)^2 \\ \text{s.a} & u \geqslant 0 & \text{s.a} \quad u \geqslant 0, \end{array} \tag{2.10}$$

em que a matriz $Q_1 - \overline{x}e$ é obtida subtraindo $\overline{x}$ a cada coluna da matriz Q_1. Resultados similares podem ser deduzidos na ausência, total ou parcial, de restrições de não negatividade. Esse lema é mencionado por P. Wolfe em [128] sem demonstração. Inclui-se aqui uma possível demonstração.

Lema 2.5 *Se $\overline{u}$ é solução óptima de (2.10), então o valor óptimo é $1 - e\overline{u}$, $\overline{v} = \overline{u}/(e\overline{u})$ é solução óptima de (2.9), e $(1 - e\overline{u})/(e\overline{u})$ o respectivo valor óptimo.*

Demonstração: Das condições necessárias e suficientes de optimalidade de (2.10) e usando $Q \equiv Q_1 - \overline{x}e$, tem-se

$$\overline{u} \geqslant 0, \quad (Q^T Q + e^T e)\,\overline{u} - e^T \geqslant 0, \quad \overline{u}^T \left[(Q^T Q + e^T e)\,\overline{u} - e^T\right] = 0.$$

Então, $\overline{u}^T Q^T Q \overline{u} + (e\overline{u})^2 = e\overline{u}$ ou, de modo equivalente, $\|Q\overline{u}\|^2 + (1 - e\overline{u})^2 = 1 - e\overline{u}$. Note-se que $e\overline{u} > 0$ porque se $e\overline{u} = 0$, então $\overline{u} = 0$, o que é absurdo. Observe-se que $\overline{v} \equiv \overline{u}/(e\overline{u}) \geqslant 0$ e $e\overline{v} = 1$. Se $\lambda = (1 - e\overline{u})/(e\overline{u})$, então

$$\begin{array}{lll} & Q^T Q \overline{v} + \lambda e^T & = \left[\left(Q^T Q + e^T e\right)\overline{u} - e^T\right]/(e\overline{u}) \quad \geqslant \quad 0 \\ \text{e} & & \\ & \overline{v}^T \left(Q^T Q \overline{v} + \lambda e^T\right) & = \frac{1}{(e\overline{u})^2}\overline{u}^T \left[\left(Q^T Q + e^T e\right)\overline{u} - e^T\right] \quad = \\ & & = \frac{1}{(e\overline{u})^2}\left[\overline{u}^T \left(Q^T Q + e^T e\right)\overline{u} - e\overline{u}\right] \quad = \quad 0. \end{array}$$

Por isso, $\overline{v}$ é óptimo para (2.9), e o valor óptimo é

$$\|Q\overline{v}\|^2 = \frac{1}{(e\overline{u})^2}\|Q\overline{u}\|^2 = \frac{e\overline{u}(1-e\overline{u})}{(e\overline{u})^2} = \frac{1-e\overline{u}}{e\overline{u}}.$$

□

No final do algoritmo NNLS, as componentes positivas de $\overline{u}$ estão associadas a um subconjunto $P_1 - \overline{x}e$ de $Q_1 - \overline{x}e$ que é independente afim. Então, como P_1 também é independente afim e $X = Q_1\overline{v} \in \text{ri conv}(P_1)$, X é única solução óptima de problema de optimização

$$\begin{array}{lll} \min & \|x-\overline{x}\|^2 & = \quad \min \quad \|P_1 v - \overline{x}\|^2 \\ \text{s.a} & x \in \text{aff}(P_1) & \quad \text{s.a} \quad ev = 1 \end{array} \tag{2.11}$$

que, por sua vez, está relacionado com o problema de mínimos quadrados sem restrições

$$\min \quad \left\| \begin{bmatrix} P_1 - \overline{x}e \\ e \end{bmatrix} u - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\|^2 \quad = \quad \min \quad \|(P_1 - \overline{x}e)u\|_2 + (eu-1)^2, \tag{2.12}$$

conforme está implícito no Lema 2.5.

Seguidamente, explica-se a álgebra linear numérica inerente à introdução de novas colunas e remoção de colunas de P_1. Se P_1 é um conjunto independente afim, então existe uma matriz $(n+1)\times(n+1)$ ortogonal Q, uma matriz $m \times m$ triangular superior e não singular R e um escalar não nulo $\overline{b}_2$ tais que

$$\widetilde{P}_1 \equiv \begin{bmatrix} P_1 - \overline{x}e & 0 \\ e & 1 \end{bmatrix} = Q \begin{bmatrix} R & \overline{b}_1 \\ 0 & \overline{b}_2 \\ 0 & 0 \end{bmatrix}. \tag{2.13}$$

Como é simples de verificar, a solução óptima do problema (2.12) é a solução $\overline{u}$ do sistema de equações lineares $Ru = \overline{b}_1$, e o respectivo valor óptimo é $\overline{b}_2^2 = 1 - e\overline{u}$. Consequentemente, por um resultado análogo ao do Lema 2.5, $\overline{v} = \overline{u}/(e\overline{u})$ é solução óptima de (2.11). Em seguida, explica-se como actualizar a matriz triangular superior em (2.13) após a inserção e remoção de um elemento em P_1. Considere-se a inserção de um elemento no conjunto P_1.

Lema 2.6 *Seja $P_1 \cup \{\widehat{p}\}$ um conjunto independente afim, e suponha-se conhecida a decomposição (2.13) de $\widetilde{P}_1$. Então, existe uma matriz $(n+1)\times(n+1)$ ortogonal $\overline{Q}$ tal que*

$$\overline{P}_1 \equiv \begin{bmatrix} P_1 - \overline{x}e & \widehat{p} - \overline{x} & 0 \\ e & 1 & 1 \end{bmatrix} = \overline{Q} \begin{bmatrix} R & a & \overline{b}_1 \\ 0 & \rho & f \\ 0 & 0 & g \\ 0 & 0 & 0 \end{bmatrix}, \tag{2.14}$$

onde, para $s = (P_1 - \overline{x}e)^T(\widehat{p} - \overline{x}) + e^T$,

$$a = R^{-T}s, \qquad \rho = \sqrt{(\widehat{p} - \overline{x})^T(\widehat{p} - \overline{x}) + 1 - a^T a},$$
$$f = (1 - a^T\overline{b}_1)/\rho, \quad g = \sqrt{\overline{b}_2^2 - f^2}.$$

Demonstração: Primeiro, observe-se que

$$\widetilde{P}_1^T\widetilde{P}_1 = \begin{bmatrix} (P_1 - \overline{x}e)^T(P_1 - \overline{x}e) + e^Te & e^T \\ e & 1 \end{bmatrix} = \begin{bmatrix} R & \overline{b}_1 \\ 0 & \overline{b}_2 \end{bmatrix}^T \begin{bmatrix} R & \overline{b}_1 \\ 0 & \overline{b}_2 \end{bmatrix} =$$
$$= \begin{bmatrix} R^TR & R^T\overline{b}_1 \\ \overline{b}_1^TR & \overline{b}_1^T\overline{b}_1 + \overline{b}_2^2 \end{bmatrix}.$$

Seja $\overline{R}$ a matriz $(m+1) \times (m+1)$ triangular superior em (2.14). Então,

$$\overline{R}^T\overline{R} = \begin{bmatrix} R & a & \overline{b}_1 \\ 0 & \rho & f \\ 0 & 0 & g \end{bmatrix}^T \begin{bmatrix} R & a & \overline{b}_1 \\ 0 & \rho & f \\ 0 & 0 & g \end{bmatrix} = \begin{bmatrix} R^TR & R^Ta & R^T\overline{b}_1 \\ a^TR & a^Ta + \rho^2 & a^T\overline{b}_1 + \rho f \\ \overline{b}_1^TR & \overline{b}_1^Ta + \rho f & \overline{b}_1^T\overline{b}_1 + f^2 + g^2 \end{bmatrix}$$
$$= \begin{bmatrix} (P_1 - \overline{x}e)^T(P_1 - \overline{x}e) + e^Te & (P_1 - \overline{x}e)^T(\widehat{p} - \overline{x}) + e^T & e^T \\ (\widehat{p} - \overline{x})^T(P - \overline{x}e) + e & (\widehat{p} - \overline{x})^T(\widehat{p} - \overline{x}) + 1 & 1 \\ e & 1 & 1 \end{bmatrix} = \overline{P}_1^T\overline{P}_1.$$

□

Este lema permite caracterizar a solução óptima $(\widehat{u}, \widehat{u}_0)$ de

$$\min \left\| \begin{bmatrix} P_1 - \overline{x}e & \widehat{p} - \overline{x} \\ e & 1 \end{bmatrix} \begin{bmatrix} u \\ u_0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\|^2 \tag{2.15}$$

como sendo a solução do sistema de equações lineares

$$\begin{bmatrix} R & a \\ 0 & \rho \end{bmatrix} \begin{bmatrix} u \\ u_0 \end{bmatrix} = \begin{bmatrix} \overline{b}_1 \\ f \end{bmatrix} \iff \begin{bmatrix} \widehat{u} \\ \widehat{u}_0 \end{bmatrix} = \begin{bmatrix} \overline{u} - (f/\rho)a \\ f/\rho \end{bmatrix}.$$

Agora, considere-se a remoção de uma das primeiras m colunas de P_1, denotada por p_j. Considere-se a matriz

$$\overline{P}_1 \equiv \begin{bmatrix} p_1 - \overline{x} & \ldots & p_{j-1} - \overline{x} & p_{j+1} - \overline{x} & \ldots & p_m - \overline{x} & 0 \\ 1 & \ldots & 1 & 1 & \ldots & 1 & 1 \end{bmatrix} = Q \begin{bmatrix} \widetilde{R} & \overline{b}_1 \\ 0 & \overline{b}_2 \\ 0 & 0 \end{bmatrix},$$

onde a matriz $\widetilde{R}$ só não é triangular superior porque os elementos nas posições $(j, j+1)$, $\ldots, (m, m+1)$ são não nulos. Esta matriz pode ser conduzida à forma triangular superior, efectuando uma multiplicação à esquerda por $m-j+1$ matrizes de Givens, que são matrizes ortogonais. Deste modo, obtém-se a decomposição QR da matriz $\overline{P}_1$.

2.3.2 Normas l_1 e l_∞

No caso da norma l_1, o problema (2.4) pode ser formulado como um problema linear cujo par primal-dual se apresenta de seguida:

$$\left.\begin{array}{lll} \min & e\sigma & \\ \text{s.a} & Pw+\sigma & \geqslant \overline{x} \\ & -Pw+\sigma & \geqslant -\overline{x} \\ & ew & = 1 \\ & w_i \geqslant 0 \ (i : p^i \in P_2) & \end{array}\right\} = \left\{\begin{array}{lll} \max & (\gamma_1-\gamma_2)\overline{x}+\rho & \\ \text{s.a} & (\gamma_1-\gamma_2)P_1+\rho e & = 0 \\ & (\gamma_1-\gamma_2)P_2+\rho e & \leqslant 0 \\ & \gamma_1+\gamma_2 & = e \\ & \gamma_1,\gamma_2 \geqslant 0 & \end{array}\right. \tag{2.16}$$

No caso da norma l_∞, o problema (2.4) pode ser formulado como um problema linear cujo par primal-dual se apresenta de seguida:

$$\left.\begin{array}{lll} \min & \sigma & \\ \text{s.a} & Pw+e\sigma & \geqslant \overline{x} \\ & -Pw+e\sigma & \geqslant -\overline{x} \\ & ew & = 1 \\ & w_i \geqslant 0 \ (i : p^i \in P_2) & \end{array}\right\} = \left\{\begin{array}{lll} \max & (\gamma_1-\gamma_2)\overline{x}+\rho & \\ \text{s.a} & (\gamma_1-\gamma_2)P_1+\rho e & = 0 \\ & (\gamma_1-\gamma_2)P_2+\rho e & \leqslant 0 \\ & (\gamma_1+\gamma_2)e_n & = 1 \\ & \gamma_1,\gamma_2 \geqslant 0 & \end{array}\right. \tag{2.17}$$

Em qualquer dos casos, se $(\widehat{\rho}, \widehat{\gamma}_1, \widehat{\gamma}_2)$ denota uma qualquer solução dual óptima, então o vector $\widehat{\alpha}$ a que se refere o Lema 2.1 é caracterizado por $\widehat{\alpha} = \widehat{\gamma}_2 - \widehat{\gamma}_1$. Mais, se $(\widehat{\sigma}, \widehat{w})$ denota uma qualquer solução primal básica admissível que seja óptima, então, pelo Lema 2.3, o novo conjunto P_1 é independente afim sem que isso seja imposto à parte.

2.4 Experiência computacional

Implementou-se o algoritmo da Figura 2.1 e efectuaram-se diversos testes computacionais. Começou-se por testar a implementação com base nos problemas analisados por P. Wolfe em [128], que são essencialmente pontos uniformemente distribuídos sobre o cubo n-dimensional. Os resultados são apresentados na Subsecção 2.4.1.

Testou-se também o algoritmo da Figura 2.1 como gerador de cortes para problemas de optimização discreta, definidos apenas pela norma Euclidiana. Os resultados obtidos com uma colecção de problemas teste proveniente da MIPLIB [23, 24], disponível electronicamente em `http://miplib.zib.de/miplib3/miplib.html`, são apresentados na Subsecção 2.4.2. Os resultados obtidos com uma colecção de problemas teste proveniente da OR LIB, disponível electronicamente em `http://mscmga.ms.ic.ac.uk/jeb/orlib/sppinfo.html`, são apresentados na Subsecção 2.4.3. O primeiro conjunto contém essencialmente problemas lineares com variáveis binárias, enquanto que o segundo conjunto contém problemas de partição (do inglês, *set partitioning problems*).

A implementação é constituída por um conjunto de rotinas escritas em MATLAB, versão 6.0 [81]. Todos os testes computacionais foram obtidos num PC com processador Intel Pentium II a 350MHz, com 128Mb de RAM e 256K de cache.

Os problemas teste são lidos em formato MPS [101, 102] (abreviatura de *Mathematical Programming System*), que consiste num ficheiro ASCII que contém os dados de um problema de programação matemática em formato fixo por coluna (ver Tabela 2.1). Originalmente, os problemas de partição encontravam-se em formato *airline* (porque correspondem a problemas de calendarização de tripulação de linhas aéreas) e foram convertidos para o formato MPS, utilizando o programa em linguagem Perl de David Levine disponível em `http://www.ms.ic.ac.uk/jeb/pub/`.

Field:	1	2	3	4	5	6
Columns:	2-3	5-12	15-22	25-36	40-47	50-61
	NAME	problem name				
	ROWS					
	type	name				
	COLUMNS					
		column name	row name	value	row name	value
	RHS					
		rhs name	row name	value name	row	value
	RANGES					
		range name	row name	value name	row	value
	BOUNDS					
	type	bound name	column name	value		
	ENDATA					

Tabela 2.1: *Organização de um ficheiro em formato MPS.*

De acordo com objectivos específicos, alteraram-se os valores por defeito dos seguintes parâmetros do CPLEX:

AGGIND: Por defeito, o CPLEX efectua o pré-processamento sobre os dados do problema inteiro pré-resolvendo e agregando (em termos de linhas e colunas) de modo a reduzir o tamanho do programa inteiro e fortalecer a relaxação linear inicial. Quando activado, este parâmetro invoca o *Cplex Aggregator* que é aplicado o número de vezes especificado ou necessário até que não sejam possíveis mais reduções de linhas e colunas. O parâmetro foi desactivado em todas as chamadas do CPLEX.

PREIND: Por defeito, o CPLEX invoca o *Cplex Presolve* e as rotinas de pré--processamento determinam se o problema é ilimitado ou impossível. Quando este parâmetro é desactivado, o método de optimização utilizado devolve a informação referente à solução, isto é, se o problema é imposível ou possível e, neste último caso, se é ilimitado ou não. O parâmetro foi desactivado em todas as chamadas do CPLEX.

CPX_PARAM_CUTUP: Para um dado problema inteiro, o CPLEX declara a sua optimalidade quando determina a solução inteira e todos os nodos tenham sido processados, sendo, neste caso, a optimalidade relativa à tolerância e aos critérios de optimalidade estabelecidos. É neste contexto que o parâmetro de *cut-off* actua, terminando a optimização do problema inteiro quando encontra um limite estabelecido, pois restringe a pesquisa da optimalidade da solução inteira, eliminando os nodos que correspondem a valores da função objectivo superiores ou inferiores ao valor de *cut-off* dado. De forma a reduzir o número de nodos do procedimento de *branch-and--bound* da rotina *CPXmipopt* ao resolver problemas inteiros de minimização, usou--se o parâmetro *CPX_PARAM_CUTUP* de modo a eliminar a pesquisa de soluções admissíveis inteiras com valor superior ao indicado pelo parâmetro. Como a utilização de um valor muito restritivo de *cut-off* pode eliminar a determinação de soluções inteiras, considerou-se a tolerância de 10^{-5}.

Seguidamente, é apresentada uma breve descrição das rotinas que estabelecem a ligação entre o MATLAB e o CPLEX. As rotinas *read_problem.c*, *lp_cplex.c* e *mip_cplex.c* estão presentes em todas as aplicações, dado o seu carácter genérico, tendo as restantes rotinas características concordantes com o tipo específico de disjunção utilizada.

read_problem.c: Os dados que definem o problema inteiro são lidos de um ficheiro em formato MPS, o problema é optimizado e a solução inteira é obtida através da chamada à rotina *CPXmipopt*. A rotina devolve os dados originais do problema na forma matricial, o resultado da optimização e o número de nodos do procedimento de *branch-and-bound* usados na resolução.

lp_cplex.c: Efectua a leitura dos dados de um problema linear na forma matricial (correspondente à relaxação linear do problema lido pela rotina *read_problem.c*),

optimiza-o e obtém a solução e base óptimas, através da chamada à rotina *CPXlpopt*. Por defeito, é utilizado o optimizador dual do Simplex, mas nesta rotina optou-se por utilizar o método primal do Simplex. O número máximo de iterações a serem executadas pelo método Simplex é limitado a 10.000 por atribuição do parâmetro *CPX_PARAM_ITLIM*.

mip_cplex.c: Esta rotina lê os dados de um problema inteiro na forma matricial, optimiza-o e obtém a solução inteira, através da chamada à rotina *CPXmipopt*, e o número de nodos do procedimento de *branch-and-bound* usados na resolução. Esta rotina permite a utilização do parâmetro *CPX_PARAM_CUTUP* quando existe o conhecimento de um limite superior do valor da solução inteira.

solve_base.c: (em Bundle_1d) Nesta rotina, após a leitura dos dados de um problema linear na forma matricial, é executado um de dois procedimentos, dependendo do valor de *mode*.

Se $mode = 0$, previamente à chamada desta rotina foi definida uma disjunção simples do tipo $x_j = 0 \vee x_j = 1$, pelo que, neste caso, se pretende determinar a solução e bases óptimas de dois problemas lineares que diferem entre si na afectação distinta do valor da variável binária x_j. Isto é, para o primeiro problema, o valor de x_j é fixado a zero e, para o segundo, a um através da alteração adequada dos seus limites superiores e inferiores. Os problemas são optimizados através da chamada à rotina *CPXlpopt*, utilizando o conhecimento de uma base inicial.

Se $mode = 1$, o objectivo da chamada desta rotina é fortalecer o valor de β num dado corte $\alpha x \geqslant \beta$ através da aplicação do procedimento de *strong branching* (ver Cplex [80, página 252]). Neste caso, associada a este corte, está a definição de uma disjunção simples do tipo $x_j = 0 \vee x_j = 1$ e o conhecimento do conjunto de variáveis que são fraccionárias na solução óptima da relaxação do problema inteiro. Seja k a cardinalidade deste conjunto. No primeiro passo a x_j é afectado o valor um, sendo resolvidos $2 * k$ problemas, nos quais cada uma das variáveis fraccionárias é fixada sucessivamente a um e a zero. Da resolução destes problemas é extraído o valor da maior solução obtida. No segundo passo a x_j é afectado o valor zero, sendo resolvidos $2 * k$ problemas, nos quais cada uma das variáveis fraccionárias é fixada sucessivamente a um e a zero. Da resolução destes problemas é extraído o valor da maior solução obtida. Dos dois valores obtidos é seleccionado o menor, que será usado para o fortalecimento de β. A afectação dos valores das variáveis a zero ou a um é implementada através da alteração adequada dos seus limites superiores e inferiores. A resolução de todos os problemas lineares é feita de forma eficiente, usando em cada reoptimização o conhecimento de bases iniciais e a rotina *CPXlpopt*.

solve_base_2d.c: (em Bundle_2d) Esta rotina, após efectuar a leitura dos dados de um problema linear na forma matricial, determina a solução e base óptimas de quatro

problemas lineares. Antes da chamada da rotina foi definida uma disjunção, que envolve as duas variáveis mais fraccionárias na solução óptima da relaxação linear de um problema inteiro, do tipo $x_j = 0 \vee x_j = 1 \vee x_k = 0 \vee x_k = 1$. Neste caso, os problemas lineares a optimizar diferem entre si na afectação distinta do valor das variáveis binárias x_j e x_k. A afectação dos valores das variáveis a zero ou a um é implementada através da alteração adequada dos seus limites superiores e inferiores. Os problemas lineares são optimizados através da chamada à rotina *CPXlpopt*, utilizando o conhecimento de uma base inicial.

solve_base_rf.c: (em Bundle_RF) Previamente à chamada desta rotina, foi definida a disjunção de Ryan-Foster [112] (ver Apêndice B), na qual a afectação do valor zero às variáveis binárias pertencentes a dois conjuntos distintos determina a diferença entre os problemas lineares a resolver. Neste caso, a rotina determina a solução óptima de dois problemas lineares: para o primeiro, cada variável do primeiro conjunto é afectada ao valor zero; para o segundo, cada variável do segundo conjunto é afectada, também, ao valor zero. A afectação dos valores das variáveis a zero é implementada através da afectação do valor zero aos valores do limites superiores e inferiores de cada variável binária. Os problemas são optimizados através da chamada à rotina *CPXlpopt*, utilizando o conhecimento de uma base inicial, e, para cada problema, são determinadas a solução e bases óptimas.

2.4.1 Experiência computacional com diferentes normas

Testou-se a implementação com a resolução do problema (2.3) para as normas l_1, l_2 e l_∞. Conforme sugerido por P. Wolfe [128], o conjunto de base é um conjunto K^0 de m pontos uniformemente distribuídos sobre o hipercubo n-dimensional $[0,2]^n$. Cada problema é caracterizado por:

Tipo 0: K^0;

Tipo 1: $X^0 + K^0$, sendo X^0 um ponto escolhido aleatoriamente no conjunto $2K^0$;

Tipo 2: $\{(1 + 10^{-3}x_1, x_2, \ldots, x_n) \colon (x_1, x_2, \ldots, x_n) \in K^0\}$, e

Tipo 3: $\{10^{-2} + 10^{-3}x_1, x_2, \ldots, x_n) \colon (x_1, x_2, \ldots, x_n) \in K^0\}$.

As Tabelas 2.2, 2.3, 2.4 e 2.5 apresentam o comportamento médio do algoritmo para problemas do Tipo 0, Tipo 1, Tipo 2 e Tipo 3, respectivamente, em termos do número de Ciclos maiores e menores, número de pontos de K necessários para determinar a solução e tempo de CPU, utilizando as normas l_1, l_2 e l_∞. Os resultados referem-se a valores médios de 10 amostras de várias dimensões. Quando $m > n$, o número de

pontos de K necessários para determinar a solução em problemas do Tipo 0 é $n+1$, do Tipo 1 é, em geral, pequeno (entre 2 e 8), do Tipo 2 e 3 é aproximadamente n, com a excepção no Tipo 3 e norma l_∞ que é 1.

2.4.2 Experiência computacional com problemas lineares 0-1

Testou-se o algoritmo da Figura 2.1 como gerador de cortes em problemas lineares 0-1. Considerou-se apenas a norma Euclideana, pelo que o Figura 2.1 se torna o algoritmo original de P. Wolfe, e diversos tipos de disjunções. A Tabela 2.6 descreve os problemas teste seleccionados de acordo com a seguinte estrutura: as primeiras três colunas contêm o nome, o número de restrições e o número de variáveis; a quarta e quinta colunas contêm o valor óptimo e o número de nodos da árvore do *branch-and-bound* requeridos para optimizar o problema através do CPLEX 8.0 [80], com definições *default* dos parâmetros; e, as duas últimas colunas contêm, relativamente à relaxação linear, o valor óptimo e o número de componentes fraccionárias na solução óptima encontrada.

Seguidamente, descrevem-se as implementações efectuadas do algoritmo de P. Wolfe, como gerador de cortes, em cada uma das aplicações *Bundle_sc*, *Bundle_1d*, *Bundle_2d* e *Bundle_c*. Na aplicação *Bundle_sc*, o objectivo foi testar a qualidade de cortes mais profundos que separam o invólucro convexo inteiro da solução óptima da relaxação linear. Nas aplicações *Bundle_1d* e *Bundle_2d*, o objectivo foi testar a qualidade de cortes disjuntivos baseados nas restrições de integralidade das variáveis. Na aplicação *Bundle_c*, o objectivo foi testar a qualidade de cortes mais profundos com base no invólucro inteiro de cada uma das restrições do problema, separadamente.

As Tabelas 2.7, 2.8, 2.9 e 2.10 resumem os resultados numéricos obtidos. Para o número de cortes gerados pelo algoritmo de P. Wolfe, que se apresenta na coluna CUTS, a coluna W-IT indica o número acumulado de iterações do Ciclo maior do algoritmo efectuadas, a coluna W-LP, o número total de iterações feitas pelo CPLEX ao resolver os problemas lineares envolvidos na geração desses cortes e a coluna W-ND, o número total de nodos utilizados pelo CPLEX ao resolver os problemas lineares 0-1 envolvidos na geração desses cortes. A coluna ND indica o número de nodos utilizados pelo CPLEX para optimizar o problema original após a adição do número de cortes indicado na coluna CUTS, ou A-CUTS. O valor apresentado na coluna IMP corresponde à melhoria obtida, relativa ao número de nodos da árvore do *branch-and-bound* requerido para optimizar o problema através do CPLEX com definições *default* dos parâmetros. Seguidamente, descrevem-se os aspectos mais específicos de cada aplicação.

Bundle_sc: A aplicação inicia-se com a leitura e optimização da relaxação linear $\overline{P}$. O seguinte procedimento é repetido até que o número de nodos na resolução do problema original seja zero ou o número de cortes gerados seja igual a dez. Gerar um

Tipo 0								
Norma l_1	m=20	m=40	m=60	m=80	m=100	m=120	m=140	m=160
n=10	12,9	11,7	11,7	11,1	11,1	11,2	11,0	11,2
	14,0	11,4	11,4	10,2	10,2	10,4	10,0	10,4
	10,8	11,0	11,0	11,0	11,0	11,0	11,0	11,0
	0,290	0,234	0,257	0,289	0,276	0,303	0,320	0,355
n=20	16,5	26,7	22,4	21,8				
	17,9	31,9	22,8	21,6				
	14,1	20,5	21,0	21,0				
	0,382	0,816	0,638	0,648				
n=30	17,7	30,3						
	19,1	35,7						
	15,3	23,9						
	0,506	1,187						
Norma l_2	m=20	m=40	m=60	m=80	m=100	m=120	m=140	m=160
n=10	11,9	11,3	11,0	11,0	11,0	11,0	11,0	11,0
	12,1	10,6	10,0	10,0	10,0	10,0	10,0	10,0
	11,0	11,0	11,0	11,0	11,0	11,0	11,0	11,0
	0,141	0,165	0,210	0,336	0,310	0,344	0,394	0,438
n=20	15,8	23,6	21,2	21,9				
	15,8	25,7	20,4	21,8				
	14,8	20,5	21,0	21,0				
	0,334	0,513	0,522	0,651				
n=30	17,4	27,3						
	17,0	29,6						
	16,8	24,0						
	0,547	0,901						
Norma l_∞	m=20	m=40	m=60	m=80	m=100	m=120	m=140	m=160
n=10	13,4	11,7	11,5	11,3	11,6	11,3	11,7	11,3
	15,0	11,4	11,0	10,6	11,2	10,6	11,4	10,6
	10,8	11,0	11,0	11,0	11,0	11,0	11,0	11,0
	0,252	0,205	0,233	0,263	0,290	0,292	0,333	0,340
n=20	16,0	25,0	22,5	22,8				
	17,4	28,9	23,0	23,6				
	13,6	20,1	21,0	21,0				
	0,305	0,628	0,545	0,617				
n=30	17,6	31,5						
	19,6	38,9						
	14,6	23,1						
	0,388	0,986						

Tabela 2.2: *Problemas do Tipo 0: número médio de Ciclos maiores e menores, número de pontos e tempo de CPU.*

Tipo 1								
Norma l_1	m=20	m=40	m=60	m=80	m=100	m=120	m=140	m=160
n=10	3,1	4,6	4,4	4,0	5,4	5,2	5,3	5,1
	2,4	4,8	4,3	3,6	6,1	5,1	6,0	6,1
	2,8	3,4	3,5	3,4	3,7	4,3	3,6	3,1
	0,041	0,085	0,086	0,084	0,143	0,140	0,164	0,176
n=20	3,4	4,4	5,6	5,4				
	2,7	4,1	5,8	5,2				
	3,1	3,7	4,4	4,6				
	0,050	0,089	0,131	0,131				
n=30	4,5	4,9						
	3,7	4,1						
	4,3	4,7						
	0,079	0,100						
Norma l_2	m=20	m=40	m=60	m=80	m=100	m=120	m=140	m=160
n=10	3,9	4,6	4,6	3,9	4,7	4,4	3,9	4,8
	2,9	3,8	3,8	2,9	3,9	3,5	2,9	4,1
	3,9	4,4	4,4	3,9	4,5	4,3	3,9	4,5
	0,047	0,101	0,096	0,096	0,141	0,149	0,149	0,201
n=20	4,2	5,1	5,2	5,5				
	3,2	4,1	4,2	4,6				
	4,2	5,1	5,2	5,4				
	0,098	0,140	0,163	0,197				
n=30	5,2	7,0						
	4,2	6,0						
	5,2	7,0						
	0,173	0,265						
Norma l_∞	m=20	m=40	m=60	m=80	m=100	m=120	m=140	m=160
n=10	5,7	6,6	7,0	7,2	8,1	7,6	9,5	7,7
	6,1	8,0	8,7	9,1	10,0	9,6	12,6	9,9
	4,3	4,2	4,3	4,3	5,1	4,6	5,4	4,5
	0,091	0,126	0,150	0,177	0,217	0,218	0,302	0,259
n=20	7,7	10,1	10,2	11,6				
	8,9	13,3	12,6	15,0				
	5,5	5,9	6,8	7,2				
	0,136	0,228	0,247	0,312				
n=30	9,4	9,6						
	11,0	11,0						
	6,8	7,2						
	0,182	0,210						

Tabela 2.3: *Problemas do Tipo 1: número médio de Ciclos maiores e menores, número de pontos e tempo de CPU.*

Tipo 2								
Norma l_1	m=20	m=40	m=60	m=80	m=100	m=120	m=140	m=160
n=10	13,3	20,8	23,5	30,7	29,5	30,9	31,3	36,6
	16,5	30,6	36,0	47,1	48,0	47,5	49,8	56,8
	9,1	10,0	10,0	13,3	10,0	10,0	11,8	15,4
	0,277	0,550	0,693	0,981	1,055	1,129	1,230	1,505
n=20	15,6	34,2	48,5	52,4				
	16,8	44,1	72,0	80,0				
	13,4	23,3	24,0	23,8				
	0,359	1,118	1,953	2,282				
n=30	16,6	31,6						
	17,3	38,7						
	14,9	23,5						
	0,441	1,283						
Norma l_2	m=20	m=40	m=60	m=80	m=100	m=120	m=140	m=160
n=10	14,2	19,7	23,3	26,1	25,2	28,3	27,9	28,7
	18,0	28,4	35,6	41,2	39,5	45,6	44,8	46,4
	9,4	10,0	10,0	10,0	9,9	10,0	10,0	10,0
	0,175	0,327	0,500	0,664	0,745	0,960	1,069	1,212
n=20	15,5	29,2	42,1	45,1				
	15,4	38,0	63,2	69,2				
	14,6	19,4	20,0	20,0				
	0,319	0,668	1,125	1,396				
n=30	15,7	28,6						
	15,2	31,4						
	15,2	24,8						
	0,507	0,938						
Norma l_∞	m=20	m=40	m=60	m=80	m=100	m=120	m=140	m=160
n=10	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
	0,004	0,006	0,009	0,011	0,013	0,016	0,018	0,021
n=20	1,0	1,1	1,0	1,0				
	0,0	0,1	0,0	0,0				
	1,0	1,1	1,0	1,0				
	0,004	0,006	0,009	0,011				
n=30	1,0	1,0						
	0,0	0,0						
	1,0	1,0						
	0,004	0,007						

Tabela 2.4: *Problemas do Tipo 2: número médio de Ciclos maiores e menores, número de pontos e tempo de CPU.*

Tipo 3								
Norma l_1	m=20	m=40	m=60	m=80	m=100	m=120	m=140	m=160
n=10	13,4	22,2	23,2	25,6	29,2	34,8	28,8	30,6
	16,3	33,4	35,4	40,2	47,4	51,3	46,6	50,2
	9,5	10,0	10,0	10,0	10,0	17,3	10,0	10,0
	0,272	0,754	0,895	0,853	1,043	1,241	1,140	1,297
n=20	16,4	31,1	44,8	50,0				
	18,9	41,7	68,6	79,0				
	12,9	19,5	20,0	20,0				
	0,403	1,054	1,852	2,245				
n=30	16,5	30,6						
	17,2	38,0						
	14,8	22,2						
	0,449	1,236						
Norma l_2	m=20	m=40	m=60	m=80	m=100	m=120	m=140	m=160
n=10	13,4	20,6	23,8	26,5	27,6	28,7	29,8	28,2
	16,4	30,2	36,6	42,0	44,2	46,4	48,6	45,4
	9,4	10,0	10,0	10,0	10,0	10,0	10,0	10,0
	0,160	0,495	0,621	0,670	0,822	0,977	1,136	1,192
n=20	15,0	29,1	40,0	47,1				
	15,3	37,7	59,0	73,2				
	13,7	19,5	20,0	20,0				
	0,319	0,655	1,058	1,459				
n=30	15,5	25,3						
	15,0	26,9						
	15,0	22,7						
	0,509	0,853						
Norma l_∞	m=20	m=40	m=60	m=80	m=100	m=120	m=140	m=160
n=10	14,6	19,2	21,4	24,1	25,2	25,4	26,2	25,9
	18,8	27,7	32,1	37,6	39,8	40,2	41,9	41,1
	9,4	9,7	9,7	9,6	9,6	9,6	9,5	9,7
	0,284	0,552	0,613	0,738	0,835	0,901	0,993	1,036
n=20	16,1	29,3	37,8	41,3				
	17,9	38,7	55,3	62,4				
	13,3	18,9	19,3	19,2				
	0,309	0,846	1,345	1,595				
n=30	16,9	31,4						
	18,5	38,5						
	14,3	23,3						
	0,366	0,970						

Tabela 2.5: *Problemas do Tipo 3: número médio de Ciclos maiores e menores, número de pontos e tempo de CPU.*

Problema	Linhas	Colunas	Solução inteira	Número de nodos	Solução linear	Número de var. fracc.
bm23	20	27	34	306	20.57	6
enigma	21	100	0	1068	0	6
lseu	28	89	1120	30037	834.68	10
p0033	16	33	3089	225	2520.57	7
p0040	13	40	62027	17	61830	2
p0201	133	201	7615	571	6875	40
pipex	25	48	788.263	517	773.8	6
sentoy	30	60	-7772	319	-7839	8
stein9	13	9	5	15	4	6
stein27	118	27	18	3862	13	21
stein45	331	45	30	54595	22	35

Tabela 2.6: *Descrição dos problemas teste lineares 0-1.*

corte que separa $\overline{x}$, uma solução óptima da relaxação linear $\overline{P}$, do invólucro inteiro, acrescentar esse corte a $\overline{P}$ e optimizar o problema linear 0-1 resultante. Após cada corte introduzido, registar também o número de nodos (em ND) que seriam necessários para resolver o problema linear 0-1 resultante. A Tabela 2.7 resume os resultados obtidos.

Bundle_1d: A aplicação inicia-se com a leitura e optimização da relaxação linear $\overline{P}$. O seguinte procedimento é repetido para cada componente fraccionária $\overline{x}_j$ na solução óptima $\overline{x}$ da relaxação linear $\overline{P}$. Gerar o corte mais profundo entre $\overline{x}$ e o invólucro convexo de $P \cap \{x\colon x_j \in \{0,1\}\}$. Os coeficientes das variáveis nesse corte são fortalecidos, usando os procedimentos descritos por Balas e Jeroslow [18]. Do conjunto de todos os cortes gerados, selecciona-se apenas o mais profundo no sentido da distância Euclidiana[1]. O termo independente desse corte é fortalecido, por aplicação do procedimento de *strong branching* (ver Cplex [80, página 252]), o corte assim obtido é acrescentado à relaxação linear $\overline{P}$ e o problema linear resultante é optimizado. Seja $\overline{x}_k$ a componente que definiu o corte adicionado. Se esta componente permanece fraccionária, então ocorre o seguinte procedimento: é gerado apenas mais um corte com base na disjunção $x_k = 0 \vee x_k = 1$. Este corte é fortalecido e acrescentado à relaxação linear $\overline{P}$. O problema linear 0-1 resultante é optimizado. A coluna A-CUTS indica se este procedimento ocorreu (2 cortes acrescentados) ou não (1 só corte acrescentado). A Tabela 2.8 resume os resultados obtidos.

Bundle_2d: A aplicação inicia-se com a leitura e optimização da relaxação linear $\overline{P}$. Obter uma solução óptima $\overline{x}$ da relaxação linear e efectuar o seguinte procedimento. Identificar o par $(\overline{x}_j, \overline{x}_k)$ de componentes mais fraccionárias e gerar o corte mais profundo entre $\overline{x}$ e o invólucro convexo de $\overline{P} \cap \{x\colon x_j, x_k \in \{0,1\}\}$. Este corte é acrescentado à relaxação linear. Se ambas as componentes se mantêm fraccionárias,

[1] a distância Euclideana de $\overline{x}$ ao hiperplano $\alpha x = \beta$ é de valor igual $(\beta - \alpha\overline{x})/\|\alpha\|_2$.

então é gerado um novo corte mais profundo usando a mesma disjunção, à semelhança da aplicação anterior. A Tabela 2.9 resume os resultados obtidos.

Bundle_c: A aplicação inicia-se com a leitura e optimização da relaxação linear $\overline{P}$. O seguinte procedimento é repetido para cada restrição do problema $a_i x \leqslant b_i$. Gerar o corte mais profundo entre $\overline{x}$, uma solução óptima da relaxação linear $\overline{P}$, e o invólucro convexo de $\{x\colon a_i x \leqslant b_i, x \in \{0,1\}^n\}$. Do conjunto de todos os cortes gerados, selecciona-se apenas o mais profundo no sentido da distância Euclidiana, que é acrescentado à relaxação linear $\overline{P}$. O problema linear 0-1 resultante é optimizado. A Tabela 2.10 resume os resultados obtidos.

2.4.3 Experiência computacional com problemas de partição

Testou-se o algoritmo de P. Wolfe como gerador de cortes em problemas de partição (veja-se o Apêndice B). A Tabela 2.11 descreve as características dos problemas teste seleccionados. Seguidamente descreve-se a aplicação.

Bundle_RF: A aplicação inicia-se com a leitura e optimização da relaxação linear $\overline{P}$. Gerar o corte disjuntivo mais profundo que separa $\overline{x}$, uma solução óptima da relaxação linear $\overline{P}$, do invólucro inteiro, baseado na disjunção de Ryan-Foster. Para a selecção dos dois conjuntos de variáveis que definem a disjunção, as restrições do problema são analisadas de modo a identificar o melhor conjunto de restrições, de cardinalidade dois ou três, que verifica a regra de Ryan-Foster. O corte gerado é acrescentado a $\overline{P}$ e o problema linear 0-1 resultante é optimizado. A Tabela 2.12 resume os resultados obtidos. As colunas T e CT indicam a cardinalidade dos dois conjuntos de variáveis que definem a disjunção de Ryan-Foster, a coluna W-IT indica o número de iterações do Ciclo maior do algoritmo de P. Wolfe efectuadas, e a coluna W-NC, o número total de nodos utilizados pelo CPLEX ao resolver os problemas lineares envolvidos na geração desse corte. A coluna ND indica o número de nodos utilizados pelo CPLEX para optimizar o problema linear 0-1 resultante.

2.5 Conclusões

Para testar a implementação efectuada do algoritmo de P. Wolfe como gerador de cortes definidos apenas pela norma Euclidiana para a resolução de problemas de optimização discreta, os cortes foram gerados utilizando metodologias diferentes. O objectivo principal foi comparar os procedimentos propostos entre si e com o procedimento de optimização dos problemas através de um programa *standard*, o CPLEX, com definições *default* dos parâmetros, em termos do número de nodos da árvore do

branch-and-bound.

Nas aplicações *Bundle_sc*, *Bundle_1d*, *Bundle_2d* e *Bundle_c*, testou-se o algoritmo na resolução de problemas lineares com variáveis binárias e na aplicação *Bundle_RF*, na resolução de problemas de partição.

Na aplicação *Bundle_sc*, os cortes gerados separam o invólucro convexo inteiro da solução óptima da relaxação linear. Os resultados da experiência computacional apresentam, até determinada altura, uma redução gradual significativa do número de nodos da árvore do *branch-and-bound* na resolução do problema linear 0-1, o que pode ser justificado pelo facto dos cortes baseados na norma Euclideana introduzirem maior imprecisão numérica.

As aplicações *Bundle_1d* e *Bundle_2d* baseiam-se nas restrições de integralidade das variáveis. A primeira usa disjunções simples numa componente e a segunda, em duas componentes. Observa-se que a *Bundle_2d* gera cortes disjuntivos mais profundos, o que só beneficia do facto de os cortes serem fortalecidos em *Bundle_1d* e em *Bundle_2d* não. Isto só não se observou nos problemas *enigma* e *lseu*.

Na aplicação *Bundle_c*, os cortes gerados separam o invólucro convexo inteiro de cada uma das restrições do problema. Comparando os resultados obtidos, observam-se melhorias positivas na aplicação da metodologia da *Bundle_c*, em relação aos resultados das aplicações *Bundle_1d* ou *Bundle_2d*, com excepção dos problemas *p0033* e *p0201*.

Na aplicação *Bundle_RF* os cortes gerados separam o invólucro convexo inteiro da solução óptima da relaxação linear e baseiam-se na disjunção de Ryan-Foster. Observou-se que a identificação do melhor conjunto de restrições, de cardinalidade dois ou três, que verifica a regra de Ryan-Foster demora algum tempo. Observam-se melhorias em alguns problemas, o que sugere a possibilidade de obtenção de melhores resultados se forem consideradas regras de fortalecimento específicas para cortes Ryan-Foster.

Problema	CUTS	W-ND	W-LP	ND	CUTS	W-IT	W-ND	ND
bm23	3	41	38074	53	7	88	70078	15
	4	52	42955	34	8	103	76684	11
	5	66	50163	25	9	115	82133	13
	6	80	55778	22	10	126	89871	12
enigma	1	4	29745	82	2	6	30329	0
lseu	3	261	15311007	23326	7	621	33445775	27354
	4	343	20316100	55137	8	727	37199421	18098
	5	421	29749384	40474	9	817	41784287	41951
	6	511	30898128	37764	10	902	47864066	28797
p0033	3	36	23483	446	7	94	60923	122
	4	54	45053	185	8	108	62702	115
	5	65	46308	128	9	123	64347	172
	6	81	52788	153	10	137	67029	149
p0040	1	20	127	13	2	32	193	0
p0201	3	338	4656973	432	7	806	10032626	470
	4	453	5279997	586	8	938	16546412	513
	5	563	6342858	582	9	1060	17441090	479
	6	677	7708475	350	10	1193	25037954	528
pipex	3	53	253682	404	7	131	581305	89
	4	72	382924	222	8	153	958915	36
	5	89	441291	211	9	175	1166006	29
	6	112	546802	85	10	199	1237271	28
sentoy	3	137	70953	309	7	345	223194	214
	4	196	128616	365	8	399	290285	223
	5	249	157361	255	9	454	349572	244
	6	296	180552	276	10	506	511560	183
stein9	3	24	88	2	7	52	101	2
	4	31	90	2	8	60	110	2
	5	39	95	2	9	66	112	2
	6	46	100	2	10	72	115	2
stein27	3	74	135002	527	7	168	137112	210
	4	91	135290	1098	8	193	138485	147
	5	117	135748	178	9	217	138964	20
	6	145	136514	543	10	243	140552	308
stein45	3	137	929718	42318	7	322	1220791	49135
	4	180	972963	50486	8	365	1272358	45304
	5	233	1111637	43929	9	411	1323384	45663
	6	280	1156658	43479	10	467	1492910	44585

Tabela 2.7: *Resultados computacionais do* Bundle_sc.

Problema	CUTS	W-IT	W-LP	A-CUTS	ND	IMP
bm23	6	160	2535	1	309	-0.980%
enigma	7	128	1727	1	41	+96.161%
lseu	11	1238	15498	2	34926	-16.277%
p0033	8	335	2476	2	216	+4.000%
p0040	2	25	88	1	21	-23.529%
p0201	40	6220	168816	1	301	+47.285%
pipex	6	132	1213	1	752	-45.455%
sentoy	8	586	9712	1	591	-85.266%
stein9	7	46	314	2	11	+26.667%
stein27	22	398	8821	2	3315	+14.164%
stein45	36	1570	61239	2	56548	-3.577%

Tabela 2.8: *Resultados computacionais do* Bundle_1d.

Problema	A-CUTS	W-IT	W-LP	ND	IMP
bm23	1	17	406	303	+0.980%
enigma	2	32	691	1271	-19.007%
lseu	1	61	1657	36562	-21.723%
p0033	1	21	291	182	+19.111%
p0040	1	12	51	21	-23.529%
p0201	2	282	11105	268	+53.065%
pipex	1	15	269	609	-17.795%
sentoy	2	116	4029	277	+13.166%
stein9	-	-	-	-	-
stein27	1	5	379	3819	+1.113%
stein45	2	46	3143	53306	+2.36%

Tabela 2.9: *Resultados computacionais do* Bundle_2d.

Problema	CUTS	W-IT	W-ND	ND	IMP
bm23	20	354	1724	244	+20.261%
enigma	21	244	5966124	584	+45.318%
lseu	28	622	18829	36097	-20.175%
p0033	16	235	1236	289	-28.444%
p0040	11	60	0	17	0%
p0201	133	2119	9203062	646	-107.717%
pipex	25	356	1149	691	-33.656%
sentoy	30	738	12115	361	-13.166%
stein9	13	60	0	16	-6.667%
stein27	118	985	0	3511	+9.089%
stein45	331	2626	0	49590	+9.168%

Tabela 2.10: *Resultados computacionais do* Bundle_c.

Problema	Linhas	Colunas	Solução inteira	Número de nodos	Solução linear	Número de var. fracc.
nw21	25	577	7408	1	7380.00	7
nw22	23	619	6984	1	6942.00	7
nw23	19	711	12534	21	12317.00	6
nw24	19	1366	6314	3	5843.00	6
nw25	20	1217	5960	2	5852.00	8
nw26	23	771	6796	1	6743.00	5
nw27	22	1355	9933	1	9877.50	3
nw28	18	1210	8298	2	8169.00	3
nw29	18	2540	4274	32	4185.33	12
nw30	26	2653	3942	7	3726.80	8
nw31	26	2662	8038	9	7980.00	5
nw32	19	294	14877	10	14570.00	4
nw33	23	3068	6678	2	6484.00	6
nw34	20	899	10488	1	10453.50	4
nw35	23	1709	7216	1	7206.00	4
nw36	20	1783	7314	11	7260.00	6
nw37	19	770	10068	2	9961.50	4
nw38	23	1220	5558	1	5552.00	6
nw39	25	677	10080	3	9868.50	3
nw40	19	404	10809	1	10658.25	9
nw41	17	197	11307	2	10972.50	3
nw42	23	1079	7656	3	7485.00	7
nw43	18	1072	8904	1	8897.00	7

Tabela 2.11: *Descrição dos problemas teste de partição.*

Problema	2 restrições					3 restrições				
	T	CT	W-IT	W-NC	ND	T	CT	W-IT	W-NC	ND
nw21	180	234	52	835	1	120	253	46	693	1
nw22	119	239	42	664	1	57	381	38	579	1
nw23	276	252	47	510	19	82	456	24	310	24
nw24	409	548	40	572	3	268	877	56	683	3
nw25	356	428	77	1112	2	141	732	59	839	2
nw26	269	292	45	554	2	158	470	56	761	2
nw27	509	637	91	1355	1	388	749	83	1271	1
nw28	537	521	93	1131	3	311	843	133	1639	3
nw29	661	1167	122	3407	21	349	1850	180	5199	21
nw30	524	1516	136	3137	7	409	1657	137	2964	7
nw31	762	988	55	1418	6	374	1358	58	1399	18
nw32	17	117	27	307	19	20	206	22	284	13
nw33	888	1320	77	1925	2	517	1876	100	2167	2
nw34	208	229	37	734	1	161	522	45	759	1
nw35	591	723	37	1017	0	289	981	46	850	0
nw36	369	995	77	981	33	293	1113	46	1051	33
nw37	191	224	49	991	0	94	476	68	1217	0
nw38	507	475	48	813	4	396	634	54	1029	6
nw39	109	308	56	856	3	74	425	48	818	3
nw40	112	210	42	794	2	71	271	45	715	2
nw41	35	91	20	187	2	21	111	23	195	3
nw42	227	523	59	1185	2	189	460	44	1008	2
nw43	229	397	58	1438	1	99	721	61	1353	1

Tabela 2.12: *Resultados computacionais do* Bundle_RF- *2 e 3 restrições.*

Capítulo 3

Dedução de cortes combinatórios como cortes disjuntivos

3.1 Introdução

A obtenção da solução óptima de um problema linear inteiro ou inteiro misto através de métodos de planos cortantes obriga à geração de desigualdades que, embora não sejam válidas para a região admissível P da relaxação linear subjacente, são válidas para o conjunto $P_I \equiv \text{conv}\ (P \cap \mathbb{Z}^n)$ - o invólucro convexo da região de interesse. A literatura da especialidade é rica em trabalhos científicos onde se demonstra que determinadas classes de desigualdades definem facetas de um determinado poliedro P_I. Isso é sinónimo de demonstrar que se tratam de desigualdades essenciais na definição do poliedro P_I. Tais desigualdades são reconhecidas como "boas".

Em alternativa, podem-se caracterizar classes de desigualdades válidas para P_I como sendo "boas" quando, para k elevado, elas são válidas para P^k, um dado fecho elementar de ordem k, mas não são válidas para P^{k-1}. Nesse sentido, a comunidade científica internacional tem, recentemente, vindo a colocar várias questões científicas. Para um dado problema específico, qual é o menor inteiro k tal que $P^k = P_I$? Quão "profunda" é uma dada desigualdade relativamente ao número de operações de fecho elementar?

Por isso, em jeito meramente introdutório, recordam-se, na Secção 3.2, os procedimentos que permitem obter uma caracterização algébrica explícita de P_I. O desenvolvimento realizado teve como ponto de partida o trabalho de Cornuéjols e Li [43]. Na Secção 3.3, apresentam-se provas alternativas da existência de desigualdades indutoras de facetas que podem ser deduzidas a partir de disjunções simples para alguns cortes combinatórios bem conhecidos. O desenvolvimento proposto teve por base o trabalho de A. Letchford [91]. São apresentadas demonstrações por diferirem das daquele autor.

Os cortes combinatórios analisados dizem respeito aos problemas de partição por cliques, corte máximo, subdigrafo acíclico, ordenamento linear, cobertura de conjuntos e caixeiro viajante assimétrico.

3.2 Característica inteira de um poliedro

Em programação inteira, o conjunto convexo definido pela intersecção de todos os cortes de P pertencentes a uma dada família diz-se o *fecho elementar* de P. Um *corte* é uma desigualdade válida para $P \cap \mathbb{Z}^n$, que se obtém com base na caracterização algébrica de P. O número de vezes que é necessário, recursivamente, aplicar a operação de fecho elementar até obter uma caracterização algébrica de $P_I \equiv \text{conv}\ (P \cap \mathbb{Z}^n)$ diz-se a *característica inteira* de P (tradução do inglês, *rank*).

Nesta secção, recordam-se os procedimentos que permitem obter os fechos elementares de várias das famílias de cortes específicas mais conhecidas, distintamente para problemas inteiros puros e inteiros mistos gerais. Naturalmente, os mesmos procedimentos podem ser aplicados ao caso particular de problemas 0-1 puros e 0-1 mistos.

3.2.1 Programação inteira pura

Seja $P \equiv \{x \in \mathbb{R}^n_+ \colon Ax \leqslant b\}$ um poliedro com A uma matriz $m \times n$ e b um vector $m \times 1$, contendo apenas números racionais. Seja $P_I \equiv \text{conv}\ (P \cap \mathbb{Z}^n)$, que também é um poliedro e, portanto, também se pode descrever explicitamente, para uma adequada matriz A' e vector b', da forma

$$P_I = \{x \in \mathbb{R}^n_+ \colon A'x \leqslant b'\}, \tag{3.1}$$

embora esta caracterização seja, normalmente, desconhecida *a priori*.

O *Procedimento de Chvátal-Gomory*, assim designado porque se baseia no método de planos cortantes proposto por Gomory [67] e porque foi formalizado por Chvátal [37], consiste na introdução sucessiva de cortes Chvátal-Gomory. Um corte de Chvátal-Gomory ou corte C-G ou simplesmente corte de Chvátal é uma desigualdade definida por algum vector $u \in \mathbb{R}^m_+$ através de

$$\lfloor uA \rfloor x \leqslant \lfloor ub \rfloor, \tag{3.2}$$

em que $\lfloor \cdot \rfloor$ representa o arredondamento para o inteiro imediatamente inferior ou igual, componente a componente. Como é simples de provar, toda a desigualdade (3.2) é válida para P_I.

O Teorema 3.1 estabelece que, para todo o poliedro racional, o sistema constituído por todas as desigualdades de Chvátal-Gomory define ainda um poliedro racional. A sua prova pode ser consultada em Cook et al. [41, Teorema 6.34], por exemplo.

Teorema 3.1 (Schrijver [113];Chvátal [37] no caso limitado) *Se* $P = \{x \in \mathbb{R}^n_+ \colon Ax \leqslant b\}$ *é um poliedro definido por quantidades racionais* A *e* b*, então*

$$P^{(1)} \equiv \left\{x \in \mathbb{R}^n_+ \colon \lfloor uA \rfloor x \leqslant \lfloor ub \rfloor,\ u \in \mathbb{R}^m_+\right\} = \left\{x \in \mathbb{R}^n_+ \colon A^{(1)}x \leqslant b^{(1)}\right\}$$

com $A^{(1)}$ *e* $b^{(1)}$ *definidos por quantidades racionais. O poliedro* $P^{(1)}$ *é denominado o fecho elementar de Chvátal-Gomory de* P*.*

Fica então bem definido o seguinte procedimento, denominado Procedimento de Chvátal-Gomory, que consiste na geração da seguinte sucessão de poliedros:

$$\left.\begin{array}{lcll} P^{(0)} & \equiv & \{x \in \mathbb{R}^n_+ \colon A^{(0)}x \leqslant b^{(0)}\} & \\ & = & \{x \in \mathbb{R}^n_+ \colon Ax \leqslant b\} & \\ P^{(k+1)} & \equiv & \{x \in \mathbb{R}^n_+ \colon A^{(k+1)}x \leqslant b^{(k+1)}\} & \\ & = & \{x \in \mathbb{R}^n_+ \colon \lfloor uA^{(k)} \rfloor x \leqslant \lfloor ub^{(k)} \rfloor,\ u \geqslant 0\}, & k = 0, 1, \ldots \end{array}\right\} \tag{3.3}$$

O Teorema 3.2 mostra que, após um número finito de operações de fecho elementar, se obtém o poliedro (3.1). A sua prova pode ser consultada em, por exemplo, Cook et al. [41, Teorema 6.35].

Teorema 3.2 (Schrijver [113];Chvátal [37] no caso limitado) *Se* $P^{(0)} \equiv P = \{x \in \mathbb{R}^n_+ \colon Ax \leqslant b\}$ *é um poliedro definido por quantidades racionais, então, após um número finito* k *de iterações do Procedimento de Chvátal-Gomory, obtém-se* $P^{(k)} = P_I \equiv$ *conv* $(P \cup \mathbb{Z}^n)$*. O menor inteiro* k *tal que* $P^{(k)} = P_I$ *designa-se por característica de Chvátal de* P *(veja-se exemplo na Figura 3.1).*

O *Procedimento Disjuntivo* consiste na introdução sucessiva de cortes disjuntivos, cortes introduzidos por Balas [11]. Um corte disjuntivo é uma desigualdade que, para algum vector $(\pi, \pi_0) \in \mathbb{Z}^{n+1}$, é válida para o seguinte conjunto

$$P(\pi, \pi_0) \equiv \text{cl conv}\left(\left(P \cap \{x \colon \pi x \leqslant \pi_0\}\right) \cup \left(P \cap \{x \colon \pi x \geqslant \pi_0 + 1\}\right)\right). \tag{3.4}$$

Como é simples de provar, para todo $(\pi, \pi_0) \in \mathbb{Z}^{n+1}$, o conjunto $P(\pi, \pi_0)$ é um poliedro que contém P_I. Mais, todo o corte de Chvátal-Gomory é um corte disjuntivo, em que pelo menos um dos poliedros $P \cap \{x \colon \pi x \leqslant \pi_0\}$ ou $P \cap \{x \colon \pi x \geqslant \pi_0 + 1\}$ é vazio.

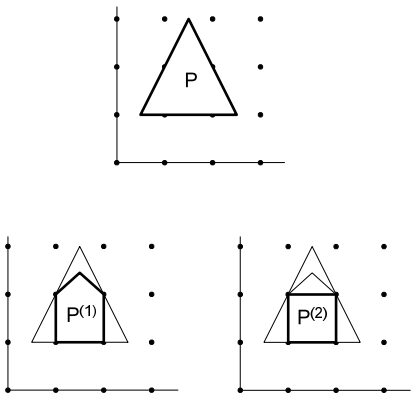


Figura 3.1: *Exemplo de polítopo com característica de Chvátal* 2.

O Teorema 3.3 estabelece que, para todo o poliedro racional, o sistema constituído por todas as desigualdades disjuntivas define ainda um poliedro racional. A sua prova pode ser consultada em, por exemplo, Cook, Kannan e Schrijver [42].

Teorema 3.3 (Cook et al. [42], Nemhauser e Wolsey [103]) *Se* $P = \{x \in \mathbb{R}^n_+ \colon Ax \leqslant b\}$ *é um poliedro definido por quantidades racionais* A *e* b*, então*

$$P^{(1)} \equiv \bigcap_{(\pi,\pi_0)\in\mathbb{Z}^{n+1}} P(\pi,\pi_0) = \left\{x \in \mathbb{R}^n_+ \colon A^{(1)}x \leqslant b^{(1)}\right\}$$

com $A^{(1)}$ *e* $b^{(1)}$ *definidos por quantidades racionais. O poliedro* $P^{(1)}$ *é denominado fecho elementar disjuntivo de* P*.*

Fica então bem definido o seguinte procedimento, denominado Procedimento Disjuntivo, que consiste na geração da seguinte sucessão de poliedros:

$$\left.\begin{array}{lcll} P^{(0)} & \equiv & \{x \colon A^{(0)}x \leqslant b^{(0)}\} & = \{x \colon Ax \leqslant b\} = P, \\ P^{(k+1)} & \equiv & \{x \colon A^{(k+1)}x \leqslant b^{(k+1)}\} & = \displaystyle\bigcap_{(\pi,\pi_0)\in\mathbb{Z}^{n+1}} P^{(k)}(\pi,\pi_0), \qquad k = 0,1,\ldots. \end{array}\right\} \tag{3.5}$$

Nem sempre é possível obter P_I ao fim de um número finito de operações, vejam-se o Exemplo 6.1 de Balas [11] e o exemplo da Secção 4.1 de Owen e Mehrotra [104], em que $P^{(k)} \supseteq P_I$ para todo k, sendo, no entanto, verdade que $\lim_{k\to\infty} P^{(k)} = P_I$. No entanto, se $P \subset [0,1]^n$, então o Teorema 3.6 permite afirmar que existe um inteiro k tal que $P^{(k)} = P_I$.

3.2.2 Programação inteira mista

Seja $P \equiv \{(x,y) \in \mathbb{R}_+^{n+p}\colon Ax + Gy \leqslant b\} \subseteq \mathbb{R}^{n+p}$ um poliedro, em que A, G são matrizes $m \times n$ e $m \times p$, respectivamente, e b um vector $m \times 1$, todos definidos por números racionais. Seja ainda $P_I \equiv \text{conv}\,(P \cap \mathbb{Z}^n \times \mathbb{R}^p)$, que também é um poliedro e, por isso, se pode descrever, para adequadas matrizes A', G' e vector b', na forma

$$P_I \quad = \quad \{(x,y) \in \mathbb{R}_+^{n+p}\colon A'x + G'y \leqslant b'\}, \tag{3.6}$$

embora esta caracterização seja, normalmente, desconhecida *a priori*.

O *Procedimento Inteiro Misto de Chvátal-Gomory* consiste na introdução sucessiva de cortes inteiros mistos de Chvátal-Gomory [68]. Um corte inteiro misto de Chvátal-Gomory ou corte fraccionário de Gomory ou simplesmente corte inteiro misto de Gomory é uma desigualdade definida, para algum vector $u \in \mathbb{R}^m$, através da seguinte sequência de operações. Primeiro, introduzem-se variáveis de folga s definindo-se o poliedro

$$\begin{aligned} Q &= \left\{(x,y,s) \in \mathbb{R}_+^{n+p+m}\colon Ax + Gy + Is = b\right\} \\ &= \left\{(x,z) \in \mathbb{R}_+^{n+p+m}\colon Ax + Hz = b\right\}, \end{aligned}$$

em que $z = (y,s)$ e $H = [G\ I]$. Depois, define-se a seguinte desigualdade

$$\sum_{i\colon f_i \leqslant f_0} f_i x_i + \frac{f_0}{1-f_0} \sum_{i\colon f_i > f_0} (1-f_i)x_i + \sum_{j\colon \overline{g}_j \geqslant 0} \overline{g}_j z_j - \frac{f_0}{1-f_0} \sum_{j\colon \overline{g}_j < 0} \overline{g}_j z_j \geqslant f_0, \tag{3.7}$$

em que

$$\begin{aligned} f_i &= \overline{a}_i - \lfloor \overline{a}_i \rfloor, \quad i = 0, 1, \ldots, n, \\ \overline{a} &= uA, \\ \overline{a}_0 &= ub, \\ \overline{g} &= uH. \end{aligned}$$

Após efectuar a substituição $s = b - Ax - Gy$ na desigualdade (3.7), obtém-se uma desigualdade definida apenas em termos das variáveis (x,y). Gomory [68] demonstrou que a desigualdade (3.7) escrita em termos das variáveis (x,y) é válida para P_I, e que, quando $p = 0$, este procedimento pode originar desigualdades que dominem estritamente qualquer desigualdade de Gomory.

O Teorema 3.4 estabelece que, para todo o poliedro, o sistema constituído por todas as desigualdades inteiras mistas de Gomory define ainda um poliedro. A sua prova pode ser consultada em, por exemplo, Cook et al. [42].

Teorema 3.4 (Cook et al. [42], Nemhauser e Wolsey [103]) *Se* $P = \{(x,y) \in$

$\mathbb{R}_+^{n+p}$: $Ax + Gy \leqslant b\}$ *é um poliedro definido por quantidades racionais* A, G *e* b*, então o conjunto dos pontos* (x, y) *que satisfazem todas as desigualdades (3.7) (projectadas no espaço das variáveis* (x, y)*) é um poliedro*

$$P^{(1)} = \left\{(x, y) \in \mathbb{R}_+^{n+p}\colon A^{(1)}x + G^{(1)}y \leqslant b^{(1)}\right\}$$

definido por quantidades racionais $A^{(1)}, G^{(1)}$ *e* $b^{(1)}$*. O poliedro* $P^{(1)}$ *é denominado fecho elementar inteiro misto de* P*.*

Fica então bem definido o seguinte procedimento, denominado Procedimento Inteiro Misto de Chvátal-Gomory, que consiste na geração da seguinte sucessão de poliedros:

$$\left.\begin{array}{rcll} P^{(0)} & \equiv & \{(x, y) \in \mathbb{R}_+^{n+p}\colon A^{(0)}x + G^{(0)}y \leqslant b^{(0)}\} & \\ & = & \{(x, y) \in \mathbb{R}_+^{n+p}\colon Ax + Gy \leqslant b\}, & \\ P^{(k)} & \equiv & \{(x, y) \in \mathbb{R}_+^{n+p}\colon A^{(k+1)}x + G^{(k+1)}y \leqslant b^{(k+1)}\} & \\ & = & \{(x, y) \in \mathbb{R}_+^{n+p}\colon \text{ verificam (3.7) para todo } u\}, & k = 1, 2, \ldots. \end{array}\right\} \quad (3.8)$$

O Teorema 3.5 mostra que, após um número finito de operações de fecho elementar inteiro misto, se obtém o poliedro (3.6). Nem sempre é possível obter P_I ao fim de um número finito de operações, vejam-se os exemplos da prova do Teorema 3 de Cornuéjols e Li [44] e o exemplo 2 de Cook et al. [42], em que $P^{(k)} \supseteq P_I$ para todo k, sendo, no entanto, verdade que $\lim_{k\to\infty} P^{(k)} = P_I$.

Teorema 3.5 ([44]) *Se* $P^{(0)} = P = \{x \in \mathbb{R}_+^n\colon Ax + Gy \leqslant b\}$ *é um poliedro definido por quantidades racionais, então, para todo o inteiro* k*, os poliedros* $P^{(k)}$ *satisfazem* $P_I \subseteq P^{(k)} \subseteq P^{(k-1)} \ldots \subseteq P^{(1)} \subseteq P$*.*

O menor inteiro k tal que $P^{(k)} = P_I$ designa-se por *característica inteira mista* de P. Em particular, para problemas 0-1 mistos e 0-1 puros com n variáveis binárias, Cornuéjols e Li provaram que a característica inteira mista de P é, no máximo, n [44] .

Teorema 3.6 ([44]) *Para um problema 0-1 misto com* n *variáveis binárias,* $P^{(n)} = P_I$*.*

O *Procedimento MIR* (abreviatura do inglês *Mixed Integer Rounding*) consiste na introdução sucessiva de cortes MIR, propostos por Nemhauser e Wolsey [103].

Um corte MIR ou corte arredondado inteiro misto é uma desigualdade definida com base em dois quaisquer vectores $u^1, u^2 \in \mathbb{R}_+^m$ através da seguinte sequência de opera-

ções. Primeiro, definem-se as seguintes desigualdades válidas para P

$$\sum_{j=1}^{n} \pi_j^1 x_j + \sum_{j=1}^{p} \mu_j^1 y_j \leqslant \pi_0^1,$$
$$\sum_{j=1}^{n} \pi_j^2 x_j + \sum_{j=1}^{p} \mu_j^2 y_j \leqslant \pi_0^2,$$

em que $\pi^i = u^i A$, $\mu^i = u^i G$ e $\pi_0^i = u^i b$ para $i = 1, 2$. Depois, define-se a seguinte desigualdade

$$\sum_{j=1}^{n} \left\lfloor \pi_j^2 - \pi_j^1 \right\rfloor x_j + \frac{1}{1 - f_0} \left(\sum_{j=1}^{n} \pi_j^1 x_j + \sum_{j=1}^{p} \min\{\mu_j^1, \mu_j^2\} y_j - \pi_0^1 \right) \leqslant \left\lfloor \pi_0^2 - \pi_0^1 \right\rfloor, \quad (3.9)$$

em que $f_0 = \pi_0^2 - \pi_0^1 - \left\lfloor \pi_j^2 - \pi_j^1 \right\rfloor$. Mostra-se que as desigualdades (3.9) são válidas para P_I.

Teorema 3.7 (Nemhauser e Wolsey [103]) *Se $P = \{(x, y) \in \mathbb{R}_+^{n+p}\colon Ax + Gy \leqslant b\}$ é um poliedro definido por quantidades racionais A, G e b, então $P^{(1)}$, o conjunto dos pontos (x, y) que satisfazem a desigualdade (3.9), também é um poliedro definido por quantidades racionais $A^{(1)}, G^{(1)}$ e $b^{(1)}$, denominado fecho elementar MIR de P.*

Fica então bem definido o seguinte procedimento, denominado Procedimento MIR, que consiste na geração da seguinte sucessão de poliedros:

$$\left.\begin{array}{lcl} P^{(0)} & \equiv & \{(x, y) \in \mathbb{R}_+^{n+p}\colon A^{(0)}x + G^{(0)}y \leqslant b^{(0)}\} \\ & = & \{(x, y) \in \mathbb{R}_+^{n+p}\colon Ax + Gy \leqslant b\} \\ P^{(k+1)} & \equiv & \{(x, y) \in \mathbb{R}_+^{n+p}\colon A^{(k+1)}x + G^{(k+1)}y \leqslant b^{(k+1)}\} \\ & = & \{(x, y) \in \mathbb{R}_+^{n+p}\colon \text{ verificam (3.9) para todo } u_1, u_2 \geqslant 0\},\ k = 0, 1, \ldots . \end{array}\right\} \quad (3.10)$$

Teorema 3.8 (Nemhauser e Wolsey [103]) *Seja $P^{(0)} = P = \{(x, y) \in \mathbb{R}_+^{n+p}\colon Ax + Gy \leqslant b\}$ um poliedro definido por quantidades racionais. O fecho elementar MIR de P coincide com o fecho elementar inteiro-misto de Gomory.*

O *Procedimento Disjuntivo* consiste na aplicação do Procedimento MIR, usando uma recursão baseada nos cortes disjuntivos propostos por Cook et al. [42]. Na verdade, os cortes disjuntivos são simultaneamente uma relaxação dos cortes de Chvátal-Gomory e uma classe simples de cortes disjuntivos, que podem ser definidos mesmo na ausência de condições de não negatividades das variáveis, ao contrário dos cortes inteiros mistos

e MIR. A desigualdade $cx + hy \leqslant c_0$ diz-se uma desigualdade disjuntiva para P se é válida para o seguinte poliedro

$$P(\pi, \pi_0) \equiv \text{cl conv}\left(\left(P \cap \{(x,y)\colon \pi x \leqslant \pi_0\}\right) \cup \left(P \cap \{(x,y)\colon \pi x \geqslant \pi_0 + 1\}\right)\right) \quad (3.11)$$

para algum vector $(\pi, \pi_0) \in \mathbb{Z}^{n+1}$.

O Teorema 3.9 mostra que, após um número finito de operações de fecho elementar disjuntivo, se obtém o poliedro (3.6). A sua prova pode ser consultada em Andersen, Cornuéjols e Li [4] e Cook et al. [42].

Teorema 3.9 (Cook et al. [42]) *Se $P = \{(x,y) \in \mathbb{R}^{n+p}\colon Ax + Gy \leqslant b\}$ é um poliedro definido por quantidades racionais A, G e b, então*

$$P^{(1)} \equiv \bigcap_{(\pi,\pi_0)\in\mathbb{Z}^{n+1}} P(\pi, \pi_0)$$

também é um poliedro definido por quantidades racionais $A^{(1)}, G^{(1)}$ e $b^{(1)}$, denominado fecho elementar disjuntivo de P.

Fica então bem definido o seguinte procedimento, denominado Procedimento Disjuntivo, que consiste na geração da seguinte sucessão de poliedros:

$$\left.\begin{array}{rcl} P^{(0)} & \equiv & \{(x,y) \in \mathbb{R}^{n+p}_+\colon A^{(0)}x + G^{(0)}y \leqslant b^{(0)}\} \\ & = & \{(x,y) \in \mathbb{R}^{n+p}_+\colon Ax + Gy \leqslant b\} = P \\ P^{(k+1)} & \equiv & \{(x,y) \in \mathbb{R}^{n+p}_+\colon A^{(k+1)}x + G^{(k+1)}y \leqslant b^{(k+1)}\} \\ & = & \displaystyle\bigcap_{(\pi,\pi_0)\in\mathbb{Z}^{n+1}} P^{(k)}(\pi, \pi_0), \qquad k = 0, 1, \ldots. \end{array}\right\} \quad (3.12)$$

Teorema 3.10 (Nemhauser e Wolsey [103]) *Seja $P^{(0)} = P = \{(x,y) \in \mathbb{R}^{n+p}_+\colon Ax + Gy \leqslant b\}$ um poliedro definido por quantidades racionais. O fecho elementar disjuntivo de P coincide com o fecho elementar inteiro-misto de Chvátal-Gomory.*

3.3 Dedução de alguns cortes combinatórios como cortes disjuntivos

O desenvolvimento desta secção teve por base o trabalho de A. Letchford [91], onde prova, para alguns problemas de optimização combinatória importantes, que existem

desigualdades indutoras de facetas que podem ser deduzidas a partir de disjunções simples, usando, nas suas demonstrações, argumentos duais. Os cortes combinatórios analisados dizem respeito aos problemas de partição por cliques, corte máximo, subdigrafo acíclico, ordenamento linear, cobertura de conjuntos e caixeiro viajante assimétrico. Para estes problemas, apresentam-se provas alternativas da sua dedução como cortes disjuntivos, usando argumentos primais.

3.3.1 O problema de partição por cliques

Seja $G = (V, E)$ um grafo não dirigido completo, em que a cada aresta $e \in E$ está associado um escalar real $w_e \geqslant 0$ designado por *peso* da aresta. O problema de partição por cliques consiste em determinar uma partição $S_1 \cup S_2 \cup \cdots \cup S_q$ (em que q não está especificado) do conjunto V de forma a maximizar a soma dos pesos das arestas em $E(S_1) \cup E(S_2) \cup \cdots \cup E(S_q)$.

Uma possível formulação deste problema como um problema de programação linear inteira 0-1 utiliza uma variável binária x_{ij} ($\equiv x_{ji}$) por cada aresta (i, j) ($\equiv (j, i)$) $\in E$, que indica se a aresta (i, j) pertence ou não à partição por cliques, e é a seguinte:

$$\begin{aligned}
&\max \sum_{(i,j)\in E} w_{ij}x_{ij} \\
&\text{s.a } x_{ij} + x_{ik} - x_{jk} \leqslant 1, && (i,j),(i,k),(j,k) \in E \text{ distintas}, && (3.13)\\
&\quad x_{ij} \geqslant 0, && (i,j) \in E, && (3.14)\\
&\quad x_{ij} \in \mathbb{Z}, && (i,j) \in E. && (3.15)
\end{aligned}$$

Os Teoremas 3.11 e 3.12 mostram que as desigualdades (3.16), propostas por Grötschel e Wakabayashi [72], e (3.19), propostas por Chopra e Rao [36], são cortes disjuntivos. Nos enunciados dos teoremas, P denota a região admissível da relaxação linear de (3.13)-(3.15).

Teorema 3.11 (Letchford [91]) *Seja $C = \{v_1, v_2, \ldots, v_{2p+1}\} \subseteq V$ um conjunto de vértices de cardinalidade $2p + 1 \geqslant 5$. A desigualdade "2-chorded odd cycle" associada a C*

$$\sum_{i=1}^{2p+1} x_{v_i v_{i+1}} - \sum_{i=1}^{2p+1} x_{v_i v_{i+2}} \leqslant p, \tag{3.16}$$

onde $v_{2p+2} \equiv v_1$ e $v_{2p+3} \equiv v_2$ (veja-se a Figura 3.2), é válida para o poliedro conv $(P^0 \cup P^1)$, em que

$$P^0 = \{x \in P \colon x_{v_1 v_2} = 0\} \quad e \quad P^1 = \{x \in P \colon x_{v_1 v_2} = 1\}.$$

Demonstração: Seja $x \in P^0 \cup P^1$ qualquer. Se $x \in P^0$, então a expressão do lado esquerdo da desigualdade (3.16) avaliada em x é igual a

$$x_{v_1v_2} + \sum_{i=1}^{p}(x_{v_{2i}v_{2i+1}} + x_{v_{2i+1}v_{2i+2}} - x_{v_{2i}v_{2i+2}}) - \left[\sum_{i=1}^{p}(x_{v_{2i-1}v_{2i+1}}) + x_{v_1v_3} + x_{v_{2p+1}v_2}\right]. \tag{3.17}$$

Por hipótese, $x_{v_1v_2} = 0$; atendendo às desigualdades (3.13), o primeiro somatório da expressão (3.17) é menor ou igual a p; e, atendendo às desigualdades (3.14), o segundo somatório é não negativo. Conclui-se que a expressão (3.17) é menor ou igual a p. Se $x \in P^1$, então a expressão do lado esquerdo da desigualdade (3.16) avaliada em x é igual a

$$\sum_{i=1}^{p}(x_{v_{2i-1}v_{2i}} + x_{v_{2i}v_{2i+1}} - x_{v_{2i-1}v_{2i+1}}) - \sum_{i=1}^{p} x_{v_{2i}v_{2i+2}} + (x_{v_{2p+1}v_1} - x_{v_{2p+1}v_2}). \tag{3.18}$$

Atendendo às desigualdades (3.13), o primeiro somatório da expressão (3.18) é menor ou igual a p; atendendo às desigualdades (3.14), o segundo somatório é não negativo; e, neste caso, $x_{v_{2p+1}v_1} = x_{v_{2p+1}v_2}$ porque, atendendo às desigualdades (3.13) para $(i,j,k) = (v_{2p+1}, v_1, v_2)$ e $(i,j,k) = (v_1, v_2, v_{2p+1})$, tem-se

$$\left\{\begin{array}{l} x_{v_1v_{2p+1}} + x_{v_1v_2} - x_{v_{2p+1}v_2} \leqslant 1 \\ x_{v_2v_1} + x_{v_2v_{2p+1}} - x_{v_1v_{2p+1}} \leqslant 1 \end{array}\right. \Leftrightarrow \left\{\begin{array}{l} x_{v_1v_{2p+1}} \leqslant x_{v_{2p+1}v_2} \\ x_{v_1v_{2p+1}} \geqslant x_{v_2v_{2p+1}} \end{array}\right.$$

que implica $x_{v_1v_{2p+1}} = x_{v_2v_{2p+1}}$. Conclui-se que a expressão (3.18) é menor ou igual a p. □

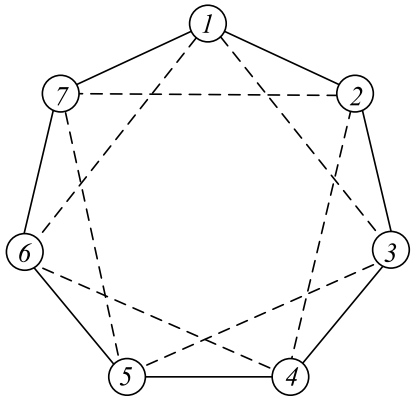


Figura 3.2: *Exemplo de "2-chorded odd cycle" com 7 ciclos.*

Teorema 3.12 (Letchford [91]) *Seja $C = \{v_1, v_2, \ldots, v_{2p+1}, u\} \subseteq V$ um conjunto de vértices de cardinalidade $2p+2$ tal que $2p+1 \geqslant 5$. A desigualdade "odd wheel"*

associada a C

$$\sum_{i=1}^{2p+1} x_{uv_i} - \sum_{i=1}^{2p+1} x_{v_iv_{i+1}} \leqslant p, \tag{3.19}$$

onde $v_{2p+2} \equiv v_1$ *(veja-se a Figura 3.3), é válida para o poliedro conv* $(P^0 \cup P^1)$*, em que*

$$P^0 = \{x \in P\colon x_{uv_1} = 0\} \quad e \quad P^1 = \{x \in P\colon x_{uv_1} = 1\}.$$

Demonstração: Seja $x \in P^0 \cup P^1$ qualquer. Se $x \in P^0$, então a expressão do lado esquerdo da desigualdade (3.19) avaliada em x é igual a

$$\sum_{i=1}^{p}(x_{uv_{2i}} + x_{uv_{2i+1}} - x_{v_{2i}v_{2i+1}}) + x_{uv_1} - \left(\sum_{i=1}^{p} x_{v_{2i-1}v_{2i}} + x_{v_{2p+1}v_1}\right). \tag{3.20}$$

Atendendo às desigualdades (3.13), o primeiro somatório da expressão (3.20) é menor ou igual a p; por hipótese, $x_{uv_1} = 0$; e, atendendo às desigualdades (3.14), o segundo somatório é não negativo. Conclui-se então que a expressão (3.20) é menor ou igual a p. Se $x \in P^1$, então a expressão do lado esquerdo da desigualdade (3.19) avaliada em x é igual a

$$\sum_{i=1}^{p}(x_{uv_{2i-1}} + x_{uv_{2i}} - x_{v_{2i-1}v_{2i}}) - \sum_{i=1}^{p} x_{v_{2i}v_{2i+1}} + x_{uv_{2p+1}} - x_{v_{2p+1}v_1}. \tag{3.21}$$

Atendendo às desigualdades (3.13), o primeiro somatório da expressão (3.21) é menor ou igual a p; atendendo às desigualdades (3.14), o segundo somatório é não negativo; e, neste caso, $x_{uv_{2p+1}} = x_{v_{2p+1}v_1}$ porque, atendendo à desigualdade (3.13) para $(i,j,k) = (v_{2p+1}, u, v_1)$ e $(i,j,k) = (v_{2p+1}, v_1, u)$, tem-se

$$\begin{cases} x_{uv_{2p+1}} + x_{uv_1} - x_{v_{2p+1}v_1} \leqslant 1 \\ x_{v_1v_{2p+1}} + x_{v_1u} - x_{v_{2p+1}u} \leqslant 1 \end{cases} \Leftrightarrow \begin{cases} x_{uv_{2p+1}} \leqslant x_{v_{2p+1}v_1} \\ x_{v_{2p+1}u} \geqslant x_{v_1v_{2p+1}} \end{cases}$$

que implica $x_{uv_{2p+1}} = x_{v_1v_{2p+1}}$. Conclui-se que a expressão (3.21) é menor ou igual a p. □

3.3.2 O problema de corte máximo

Seja $G = (V, E)$ um grafo não dirigido completo, em que a cada aresta $(i,j) \in E$ se associa um escalar real $w_{ij} \geqslant 0$ designado por *peso* da aresta (i,j). O problema do corte máximo consiste em determinar um conjunto $S \subset V$ que maximiza a soma dos pesos das arestas em $\delta(S)$.

Uma possível formulação deste problema como um problema de programação linear

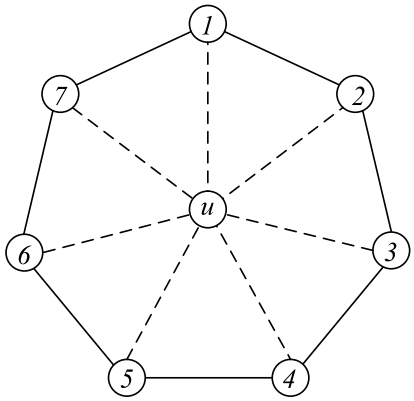


Figura 3.3: *Exemplo de "odd wheel" com 7 ciclos.*

inteira 0-1 utiliza uma variável binária $x_{ij}(\equiv x_{ji})$ por cada aresta $(i,j)(\equiv (j,i)) \in E$, que indica se a aresta (i,j) pertence ou não ao corte $\delta(S)$, e é a seguinte:

$$\begin{aligned}
&\max \sum_{(i,j)\in E} w_{ij}x_{ij} \\
&\text{s.a } x_{ij} + x_{ik} + x_{jk} \leqslant 2, && (i,j),(i,k),(j,k) \in E \text{ distintas}, && (3.22)\\
&\quad x_{ij} - x_{ik} - x_{jk} \leqslant 0, && (i,j),(i,k),(j,k) \in E \text{ distintas}, && (3.23)\\
&\quad 0 \leqslant x_{ij} \leqslant 1, && ((i,j) \in E, && (3.24)\\
&\quad x_{ij} \in \mathbb{Z}, && (i,j) \in E. && (3.25)
\end{aligned}$$

Os Teoremas 3.13 e 3.14 mostram que as desigualdades (3.26), propostas por Barahona e Mahjoub [20], e (3.29), propostas por Poljak e Turzik [108], são cortes disjuntivos. Nos enunciados dos teoremas, P denota a região admissível da relaxação linear de (3.22)-(3.25).

Teorema 3.13 (Letchford [91]) *Seja $C = \{v_1, v_2, \ldots, v_{2p+1}, s, t\} \subseteq V$ um conjunto de vértices de cardinalidade $2p+3$ tal que $2p+1 \geqslant 3$. A desigualdade "odd bicycle wheel" associada a C*

$$\sum_{i=1}^{2p+1} x_{v_iv_{i+1}} + \sum_{i=1}^{2p+1} x_{sv_i} + \sum_{i=1}^{2p+1} x_{tv_i} + x_{st} \leqslant 4p+2, \tag{3.26}$$

onde $v_{2p+2} \equiv v_1$ (veja-se a Figura 3.4), é válida para o poliedro conv $(P^0 \cup P^1)$, em que

$$P^0 = \{x \in P\colon x_{sv_1} = 0\} \quad e \quad P^1 = \{x \in P\colon x_{sv_1} = 1\}.$$

Demonstração: Seja $x \in P^0 \cup P^1$ qualquer. Se $x \in P^0$, então a expressão do lado

esquerdo da desigualdade (3.26) avaliada em x é igual a

$$x_{sv_1}+\sum_{i=1}^{p}(x_{tv_{2i-1}}+x_{tv_{2i}}+x_{v_{2i-1}v_{2i}})+\sum_{i=1}^{p}(x_{sv_{2i}}+x_{sv_{2i+1}}+x_{v_{2i}v_{2i+1}})+(x_{v_{2p+1}v_1}+x_{tv_{2p+1}}+x_{st}). \tag{3.27}$$

Por hipótese, $x_{sv_1}=0$; atendendo às desigualdades (3.22), o primeiro e o segundo somatórios da expressão (3.27) são ambos menores ou iguais a $2p$; e, neste caso, $x_{v_{2p+1}v_1}+x_{tv_{2p+1}}+x_{st}\leqslant 2$ porque, atendendo às desigualdadee (3.22) para $(i,j,k)=(v_{2p+1},v_1,t)$ e (3.23) para $(i,j,k)=(s,t,v_1)$, tem-se

$$\left\{\begin{array}{l} x_{v_{2p+1}v_1}+x_{v_{2p+1}t}+x_{v_1t}\leqslant 2\\ x_{st}-x_{sv_1}-x_{tv_1}\leqslant 0\end{array}\right. \Leftrightarrow \left\{\begin{array}{l} x_{v_{2p+1}v_1}+x_{v_{2p+1}t}\leqslant 2-x_{v_1t}\\ x_{st}\leqslant x_{tv_1}\end{array}\right.$$

o que implica $x_{v_{2p+1}v_1}+x_{tv_{2p+1}}+x_{st}\leqslant 2$. Conclui-se que a expressão (3.27) é menor ou igual a $4p+2$. Se $x\in P^1$, então a expressão do lado esquerdo da desigualdade (3.26) avaliada em x é igual a

$$\sum_{i=1}^{p}(x_{sv_{2i-1}}+x_{sv_{2i}}+x_{v_{2i-1}v_{2i}})+\sum_{i=1}^{p}(x_{tv_{2i}}+x_{tv_{2i+1}}+x_{v_{2i}v_{2i+1}})+x_{v_{2p+1}v_1}+$$
$$+x_{sv_{2p+1}}+x_{tv_1}+x_{st}. \tag{3.28}$$

Neste caso, $x_{v_{2p+1}v_1}+x_{sv_{2p+1}}+x_{tv_1}+x_{st}\leqslant 2$ porque, por hipótese, $x_{sv_1}=1$, atendendo à desigualdade (3.22) para $(i,j,k)=(v_{2p+1},v_1,s)$ e $(i,j,k)=(s,t,v_1)$, tem-se

$$\left\{\begin{array}{l} x_{v_{2p+1}v_1}+x_{v_{2p+1}s}+x_{v_1s}\leqslant 2\\ x_{st}+x_{tv_1}+x_{sv_1}\leqslant 2\end{array}\right. \Leftrightarrow \left\{\begin{array}{l} x_{v_{2p+1}v_1}+x_{v_{2p+1}s}\leqslant 1\\ x_{st}+x_{tv_1}\leqslant 1\end{array}\right.$$

o que implica $x_{v_{2p+1}v_1}+x_{v_{2p+1}s}+x_{st}+x_{tv_1}\leqslant 2$; e, atendendo às desigualdades (3.22), o primeiro e o segundo somatórios da expressão (3.28) são ambos menores ou iguais a $2p$. Conclui-se que a expressão (3.28) é menor ou igual a $4p+2$. □

Teorema 3.14 (Letchford [91]) *Seja $C=\{v_1,v_2,\ldots,v_{2p+1}\}\subseteq V$ um conjunto de vértices de cardinalidade $2p+1\geqslant 5$. A desigualdade circulante-$(2p+1,2)$ associada a C*

$$\sum_{i=1}^{2p+1}x_{v_iv_{i+1}}+\sum_{i=1}^{2p+1}x_{v_iv_{i+2}}\leqslant 3p, \tag{3.29}$$

onde $v_{2p+2}\equiv v_1$ e $v_{2p+3}\equiv v_2$ (veja-se a Figura 3.5), é válida para o poliedro conv $(P^0\cup P^1)$, em que

$$P^0=\{x\in P\colon x_{v_1v_2}=0\}\quad e\quad P^1=\{x\in P\colon x_{v_1v_2}=1\}.$$

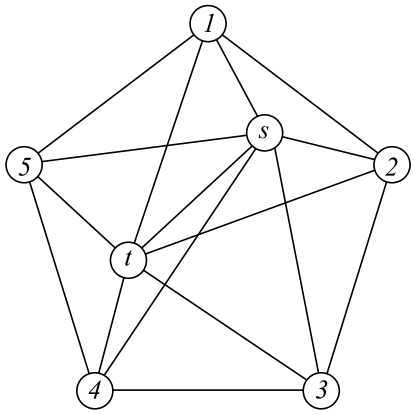


Figura 3.4: *Exemplo de "odd bicycle wheel" com* 5 *ciclos.*

Demonstração: Seja $x \in P^0 \cup P^1$ qualquer. Prove-se, por indução em p, que a desigualdade (3.29) é válida para os dois conjuntos poliédricos P^0 e P^1.

Comece-se por supôr que $p = 2$. Se $x \in P^0$, então a expressão do lado esquerdo da desigualdade (3.29) avaliada em x é igual a

$$(x_{v_2v_3} + x_{v_3v_4} + x_{v_2v_4}) + (x_{v_4v_5} + x_{v_5v_1} + x_{v_4v_1}) + x_{v_1v_2} + x_{v_1v_3} + (x_{v_3v_5} + x_{v_5v_2}). \tag{3.30}$$

Atendendo às desigualdades (3.22), a primeira e segunda parcelas são ambas menores ou iguais a 2; por hipótese, $x_{v_1v_2} = 0$; atendendo às desigualdades (3.23), $x_{v_1v_3} \leqslant x_{v_2v_3}$; e, atendendo às desigualdades (3.22), $x_{v_3v_5} + x_{v_5v_2} \leqslant 2 - x_{v_2v_3}$. Conclui-se então que a expressão (3.30) é menor ou igual a $6(= 3p)$. Se $x \in P^1$, então a expressão do lado esquerdo da desigualdade (3.29) avaliada em x é igual a

$$(x_{v_1v_2} + x_{v_2v_3} + x_{v_1v_3}) + (x_{v_3v_4} + x_{v_4v_5} + x_{v_3v_5}) + (x_{v_5v_1} + x_{v_5v_2}) + (x_{v_2v_4} + x_{v_4v_1}). \tag{3.31}$$

Por hipótese, $x_{v_1v_2} = 1$; atendendo às desigualdades (3.22), a primeira e segunda parcelas são ambas menores ou iguais a 2; e atendendo às desigualdades (3.22), $x_{v_5v_1} + x_{v_5v_2} = x_{v_2v_4} + x_{v_4v_1} \leqslant 2$. Conclui-se que a expressão (3.31) é menor ou igual a $6(= 3p)$ e que a desigualdade (3.29) é válida para o poliedro $P^0 \cup P_1$ quando $p = 2$.

Suponha-se que a desigualdade (3.29) é válida para algum p e prove-se que também é válida para $p+1$. Seja $x \in P^0 \cup P^1$ qualquer. Neste caso, a expressão do lado esquerdo da desigualdade (3.29) avaliada em x é igual a

$$\sum_{i=1}^{2p+1} (x_{v_iv_{i+1}} + x_{v_iv_{i+2}}) + (x_{v_{2p+2}v_{2p+3}} + x_{v_{2p+3}v_1} + x_{v_{2p+2}v_1}) + x_{v_{2p+3}v_2}. \tag{3.32}$$

Sendo, por hipótese, a desigualdade (3.29) válida para p, a primeira parcela é menor ou igual a $3p$; e, atendendo às desigualdades (3.22), a segunda parcela é menor ou igual a 2. Conclui-se que a expressão (3.32) é menor ou igual a $3p+2-x_{v_{2p+3}v_2} \leqslant 3(p+1)$, o que conclui a demonstração. □

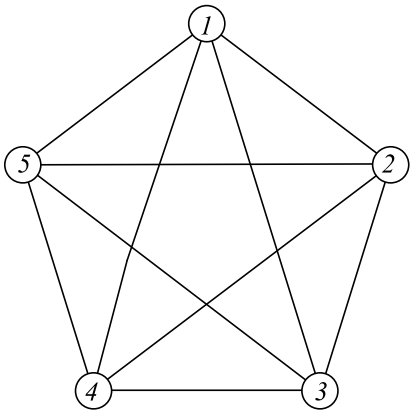


Figura 3.5: *Exemplo de circulante*-$(5,2)$.

3.3.3 O problema do subdigrafo acíclico

Seja $G = (V, E)$ um grafo dirigido completo com n vértices e m arcos (que, em particular, verifica a propriedade "$(i,j) \in E \Rightarrow (j,i) \in E$"), em que a cada arco $(i,j) \in E$ se associa um escalar real $w_{ij} \geqslant 0$ designado por *peso* do arco (i,j). O problema do subdigrafo acíclico consiste em determinar em G um subdigrafo acíclico de peso máximo. Um subdigrafo acíclico é um subgrafo do grafo dirigido G que não contém ciclos dirigidos.

Uma possível formulação deste problema como um problema de programação linear inteira 0-1 utiliza uma variável binária x_{ij} por cada arco $(i,j) \in E$, que indica se o arco (i,j) pertence ou não ao subdigrafo acíclico, e é a seguinte:

$$\max \sum_{(i,j)\in E} w_{ij}x_{ij}$$

$$\text{s.a } x(C) \leqslant |C| - 1, \qquad C = \{(i_1,i_2),(i_2,i_3),\ldots,(i_k,i_1)\} \subseteq E, \tag{3.33}$$

$$0 \leqslant x_{ij} \leqslant 1, \qquad (i,j) \in E, \tag{3.34}$$

$$x_{ij} \in \mathbb{Z}, \qquad (i,j) \in E, \tag{3.35}$$

onde C identifica um qualquer ciclo dirigido em G. A classe de desigualdades (3.33)

deve ler-se

$$\sum_{j=1}^{k} x_{i_j i_{j+1}} \leqslant k-1,$$

onde $i_{k+1} \equiv i_1$.

O Teorema 3.15 mostra que as desigualdades (3.36), propostas por Grötschel, Jünger e Reinelt [70], são cortes disjuntivos. No seu enunciado, P denota a região admissível da relaxação linear de (3.33)-(3.35).

Teorema 3.15 (Letchford [91]) *Seja $\{C_1, \ldots, C_m\}$ um conjunto distinto de ciclos dirigidos em G e $m \geqslant 3$ um inteiro ímpar. Denote-se por D_i o caminho dirigido comum a cada par de ciclos dirigidos adjacentes (C_i, C_{i+1}) para $i = 1, \ldots, m$ (veja-se a Figura 3.6). Note-se que nenhum par de caminhos dirigidos $C_i \setminus D_i$, para $i = 1, \ldots, m$, possui arcos em comum. A desigualdade escada de Möbius associada ao conjunto $\{C_1, \ldots, C_m\}$*

$$x(C_1 \cup \cdots \cup C_m) \leqslant \sum_{i=1}^{m} |C_i \setminus D_i| - (m+1)/2 \tag{3.36}$$

é válida para o poliedro conv $(P^0 \cup P^1)$, em que

$$P^0 = \{x \in P \colon x(D_m) = |D_m| - 1\} \quad e \quad P^1 = \{x \in P \colon x(D_m) = |D_m|\},$$

onde $D_m = C_m \cap C_1$.

Demonstração: Seja $x \in P^0 \cup P^1$. Se $x \in P^0$, então a expressão do lado esquerdo da desigualdade (3.36) avaliada em x é igual a

$$\sum_{i=2,4}^{m-1} x(C_i) + \sum_{i=1,3}^{m} x(C_i \setminus (D_i \cup D_{i-1})) + x(D_m). \tag{3.37}$$

Por hipótese, $x(D_m) = |D_m| - 1$; atendendo às desigualdades (3.33),

$$\begin{aligned}(3.37) &\leqslant \sum_{i=2,4}^{m-1} |C_i| - (m-1)/2 + \sum_{i=1,3}^{m} |C_i \setminus D_i| - \sum_{i=2,4}^{m} |D_i| - (m+1)/2 + |D_m| - 1 \\ &\leqslant \sum_{i=1}^{m} |C_i \setminus D_i| + |D_m| - m - 1;\end{aligned}$$

atendendo a que m é um inteiro ímpar maior ou igual a 3; e como $|C_i| \in \{3, 4\}$, então $|D_m| - m - 1 \leqslant -(m+1)/2$, conclui-se que a desigualdade (3.36) é válida para P^0.

Se $x \in P^1$, então a expressão do lado esquerdo da desigualdade (3.36) avaliada em x é igual a

$$\sum_{i=1,3}^{m} x(C_i) - x(D_m) + \sum_{i=2,4}^{m-1} x(C_i \setminus (D_i \cup D_{i-1})). \tag{3.38}$$

Por hipótese, $x(D_m) = |D_m|$; atendendo a (3.33),

$$\begin{aligned}(3.38) &\leqslant \sum_{i=1,3}^{m} |C_i| - (m+1)/2 - |D_m| + \sum_{i=2,4}^{m-1} |C_i \setminus D_i| - \sum_{i=1,3}^{m-2} |D_i| - (m-1)/2 \\ &\leqslant \sum_{i=1}^{m} |C_i \setminus D_i| - m;\end{aligned}$$

atendendo a que m é um inteiro ímpar maior ou igual a 3 e $-m \leqslant -(m+1)/2$, conclui-se que a desigualdade (3.36) é válida para P^1. □

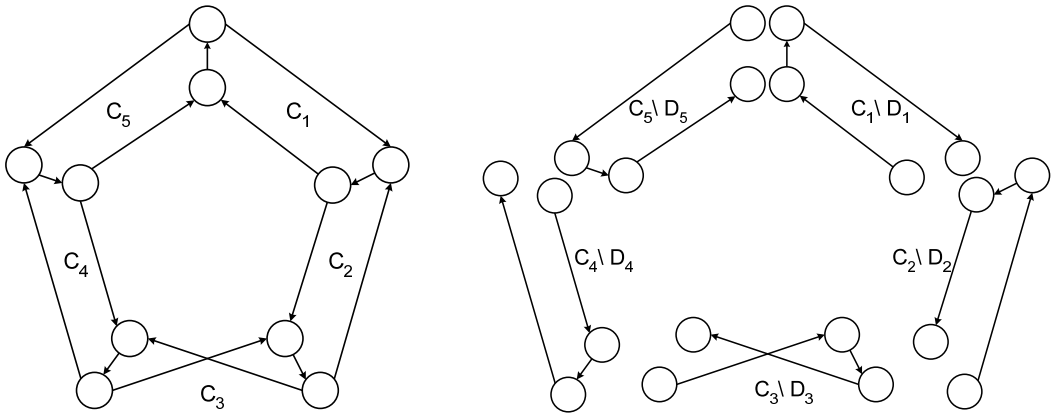


Figura 3.6: *Exemplo de escada de Möbius com* 5 *diciclos.*

3.3.4 O problema de ordenamento linear

Seja $G = (V, E)$ um grafo dirigido completo com n vértices e m arcos, em que a cada arco $(i, j) \in E$ se associa um escalar real $w_{ij} \geqslant 0$ designado por *peso* do arco (i, j). O problema de ordenamento linear consiste em determinar em G um *torneio acíclico* de peso máximo. Um torneio acíclico é um torneio (subconjunto de E tal que, para todo o par de vértices $i, j \in V$ distintos, contém o arco (i, j) ou (j, i), mas não ambos) sem ciclos dirigidos.

Uma possível formulação deste problema como um problema de programação inteira 0-1 utiliza uma variável binária x_{ij} por cada arco $(i, j) \in E$, que indica se o vértice i

está posicionado ou não antes de j no ordenamento linear, e é a seguinte:

$$\begin{aligned} \max & \sum_{(i,j)\in E} w_{ij}x_{ij} \\ \text{s.a } & (3.33), \quad (3.34), \quad (3.35), \\ & x_{ij} + x_{ji} = 1, \quad (i,j) \in E. \end{aligned} \tag{3.39}$$

As desigualdades *escada de Möbius* apresentadas para o problema do subdigrafo acíclico são válidas para o problema de ordenamento linear. No enunciado do Teorema 3.16, P denota a região admissível da relaxação linear de (3.33), (3.34), (3.35) e (3.39).

Teorema 3.16 (Letchford [91]) *A desigualdade (3.36) é válida para o poliedro conv* $(P^0 \cup P^1)$*, em que*

$$P^0 = \{x \in P\colon x(D_m) = 0\}, \quad P^1 = \{x \in P\colon x(D_m) = 1\}$$

e $D_m = C_m \cap C_1$.

Demonstração: Consequência imediata do Teorema 3.15 porque $|D_m| = 1$. □

3.3.5 O problema da cobertura de conjuntos

Seja A uma matriz $m \times n$ de zeros e uns, $M = \{1, \ldots, m\}$ e $N = \{1, \ldots, n\}$ conjuntos de índices de linhas e colunas de A, respectivamente. Seja c um vector $1 \times n$, em que $c_j \geqslant 0$, para $j \in N$, representa o *custo de selecção* da coluna j. Diz-se que a coluna $j \in N$ *cobre* a linha $i \in M$ se $a_{ij} = 1$. O problema da cobertura de conjuntos consiste em determinar o conjunto $S \subseteq N$ de custo mínimo tal que cada linha $i \in M$ é coberta pelo menos por uma coluna $j \in S$.

Uma possível formulação deste problema utiliza uma variável binária x_j por cada coluna $j \in N$, que indica se $j \in S$ ou não, e é a seguinte:

$$\begin{aligned} \min & \sum_{j\in N} c_j x_j \\ \text{s.a } & \sum_{j\in N} x_j \geqslant 1, && i \in M, && (3.40) \\ & x_j \in \{0,1\}, && j \in N. && (3.41) \end{aligned}$$

O Teorema 3.17 mostra que as desigualdades (3.43), propostas por Euler, Jünger e Reinelt [51] e Müller e Schulz [100], são cortes disjuntivos. No seu enunciado, P

é a região admissível da relaxação linear de (3.40)-(3.41), e, para a linha $i \in M$, $S(i) = \{j \in N\colon a_{ij} = 1\}$ denota o conjunto das colunas que cobrem a linha i.

Teorema 3.17 (Letchford [91]) *Sejam p e q inteiros tais que p não é múltiplo de q e $2 \leqslant q < p$. Sejam $N_0, N_1, \ldots, N_{p-1} \subseteq N$ subconjuntos não vazios e não necessáriamente disjuntos. Supor que existem índices das linhas de A, sejam $r_k \in N$ $(k = 0, \ldots, p-1)$, tais que*

$$S(r_k) \subseteq N_k \cup N_{k+1} \cup \cdots \cup N_{k+q-1}. \tag{3.42}$$

A desigualdade de ciclo generalizada associada ao conjunto $\{N_0, N_1, \ldots, N_{p-1}\}$

$$\sum_{k=0}^{p-1} x(N_k) \geqslant \lceil p/q \rceil, \tag{3.43}$$

em que $\lceil \cdot \rceil$ denota o arredondamento para o inteiro imediatamente superior, é válida para o poliedro

$$conv\left(\bigcup_{j \in S(r_0)} (P \cap \{x\colon x_j = 1\})\right),$$

onde $S(r_0) = N_0 \cup N_1 \cup \cdots \cup N_{q-1}$.

Demonstração: Seja $x \in P$ tal que $x_j = 1$ para $j \in S(r_0)$. Se $j \in S(r_0)$, então existe um índice $j \in N_t$ tal que $t \in \{0, 1, \ldots, q-1\}$ e $x(N_t) \geqslant 1$; a expressão do lado esquerdo da desigualdade (3.43) avaliada em x é então igual a

$$x(N_t) + \sum_{k=0, k\neq t}^{p-1} x(N_k) \geqslant x(N_t) + \sum_{k=0}^{\lfloor p/q \rfloor - 1} x(N_{t+kq+1} \cup N_{t+kq+2} \cup \cdots \cup N_{t+kq+q}). \tag{3.44}$$

Atendendo à relação (3.42), conclui-se que

$$(3.44) \geqslant x(N_t) + \sum_{k=0}^{\lfloor p/q \rfloor - 1} x(S(r_{t+kq+1})) \geqslant 1 + \lfloor p/q \rfloor - 1 + 1 = \lceil p/q \rceil.$$

□

3.3.6 O problema do caixeiro viajante assimétrico

Seja $G = (V, E)$ um grafo dirigido tal que a cada arco $(i, j) \in E$ está associado um escalar real $c_{ij} \geqslant 0$ designado por *custo* do arco (i, j). O problema do caixeiro viajante

assimétrico consiste em determinar em G um circuito dirigido simples com n arcos e de custo mínimo.

Uma possível formulação do ATSP como um problema de programação inteira 0-1 utiliza uma variável binária x_{ij} por cada arco $(i,j) \in E$, que indica se o arco (i,j) pertence ou não ao circuito procurado, e é a seguinte:

$$\begin{aligned}
&\min \sum_{(i,j)\in E} c_{ij}x_{ij} \\
&\text{s.a} \sum_{(i,j)\in\delta^+(i)} x_{ij} = 1, && i \in V, && (3.45)\\
&\sum_{(i,j)\in\delta^-(j)} x_{ij} = 1, && j \in V, && (3.46)\\
&\sum_{(i,j)\in E(S)} x_{ij} \leqslant |S| - 1, && S \subseteq V, S \neq \emptyset, V, && (3.47)\\
&x_{ij} \in \{0,1\}, && (i,j) \in E. && (3.48)
\end{aligned}$$

O Teorema 3.18 mostra que as desigualdades (3.49) e (3.50), propostas por Grötschel e Padberg [19] e Fischetti [54], são cortes disjuntivos. No seu enunciado, P denota a região admissível da relaxação linear de (3.45)-(3.48).

Teorema 3.18 (Letchford [91]) *Seja* $C = \{(v_1,v_2),(v_2,v_3),\ldots,(v_k,v_1)\} \subseteq E$ *um circuito dirigido simples tal que* $3 \leqslant k \leqslant n-1$.

1. *A desigualdade* D_k^+ *associada a* C

$$\sum_{i=1}^{k-1} x_{v_iv_{i+1}} + x_{v_kv_1} + 2\sum_{i=3}^{k} x_{v_1v_i} + \sum_{i=4}^{k}\sum_{j=3}^{i-1} x_{v_iv_j} \leqslant k-1 \qquad (3.49)$$

(veja-se a Figura 3.7(a)) é válida para o poliedro conv $\left(\cup_{i=1}^{k} P^{v_t}\right)$ *com*

$$P^{v_t} = \{x \in P\colon x_{v_1v_t} = 1\}, \quad t \in \{2,3,\ldots,k\}$$

e

$$P^{v_1} = \{x \in P\colon x_{v_1v_2} = x_{v_1v_3} = \cdots = x_{v_1v_k} = 0\}.$$

2. *A desigualdade* D_k^- *associada a* C

$$\sum_{i=1}^{k-1} x_{v_iv_{i+1}} + x_{v_kv_1} + 2\sum_{i=2}^{k-1} x_{v_iv_1} + \sum_{i=3}^{k-1}\sum_{j=2}^{i-1} x_{v_iv_j} \leqslant k-1 \qquad (3.50)$$

(veja-se a Figura 3.7(b)) é válida para o poliedro conv $\left(\cup_{i=1}^{k} P^{v_t}\right)$ *com*

$$P^{v_t} = \{x \in P\colon x_{v_t v_1} = 1\}, \quad t \in \{2, 3, \ldots, k\}$$

e

$$P^{v_1} = \{x \in P\colon x_{v_2 v_1} = x_{v_3 v_1} = \cdots = x_{v_k v_1} = 0\}.$$

Demonstração: Prova-se apenas a veracidade deste teorema para as desigualdades D_k^+. A demonstração, para as desigualdades D_k^-, é análoga. Seja $x \in \cup_{i=1}^{k} P^{v_t}$ qualquer. Se $x \in P^{v_t}$ para algum $t \in \{3, 4, \ldots, k\}$, então atendendo às desigualdades (3.46) para o vértice v_1 e (3.45) para o vértice v_t,

$$x_{v_1 v_i} = 0, \text{ para } i \in \{2, 3, \ldots, k\} \setminus \{t\},$$
$$x_{v_i v_t} = 0, \text{ para } i \in \{2, 3, \ldots, k\}.$$

Por isso, a expressão do lado esquerdo da desigualdade (3.49) avaliada em x é igual a

$$\sum_{j=3}^{t-1} x_{v_{j-1} v_j} + \sum_{j=t}^{k-1} x_{v_j v_{j+1}} + x_{v_k v_1} + (1 + x_{v_1 v_t}) + \sum_{j=3}^{t-1} \sum_{i=j+1}^{k} x_{v_i v_j} + \sum_{j=t+1}^{k-1} \sum_{i=j+1}^{k} x_{v_i v_j}.$$

Reorganizando os termos desta expressão obtém-se

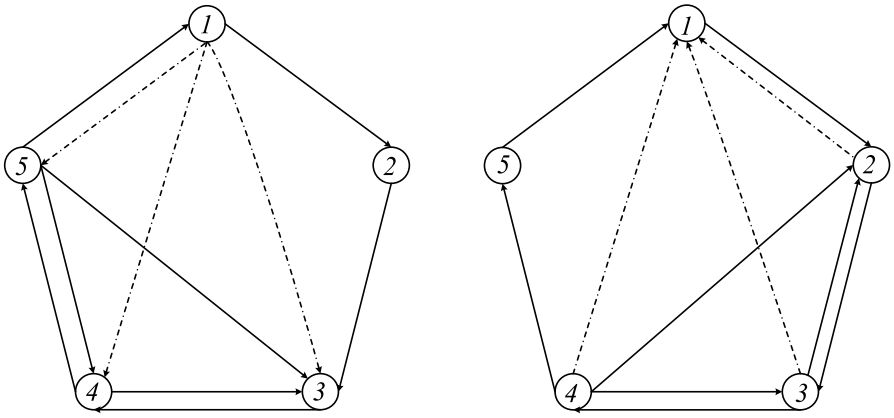
$$1 + \left[\sum_{j=3}^{t-1} \left(x_{v_{j-1} v_j} + \sum_{i=j+1}^{k} x_{v_i v_j}\right)\right] + \left(x_{v_k v_1} + x_{v_1 v_t} + \sum_{j=t}^{k-1} x_{v_j v_{j+1}} + \sum_{j=t+1}^{k-1} \sum_{i=j+1}^{k} x_{v_i v_j}\right). \tag{3.51}$$

Atendendo às desigualdades (3.46) para os vértices $v_3, \ldots, v_{t-1}$, a segunda parcela da expressão (3.51) é menor ou igual a $t - 3$. Atendendo às desigualdades (3.47) para o conjunto $S = \{v_1, v_t, v_{t+1}, \ldots, v_k\}$ e à não negatividade das variáveis, a terceira parcela é menor ou igual a $x(E(S)) \leqslant |S| - 1 = k - t + 1$. Conclui-se que a expressão (3.51) é menor ou igual a $1 + (t - 3) + (k - t + 1) = k - 1$.

Se $x \in P^{v_t}$ para algum $t \in \{1, 2\}$, então $x_{v_1 v_i} = 0$ para todo $i = 3, 4, \ldots, k$. Por isso, a expressão do lado esquerdo da desigualdade (3.49) avaliada em x é igual a

$$\sum_{i=1}^{k-1} x_{v_i v_{i+1}} + x_{v_k v_1} + \sum_{i=4}^{k} \sum_{j=3}^{i-1} x_{v_i v_j}. \tag{3.52}$$

Atendendo às desigualdades (3.47) para o conjunto $S = \{v_1, \ldots, v_k\}$ e às restrições de não negatividade, conclui-se que a expressão (3.52) é menor ou igual a $x(E(S)) \leqslant |S| - 1 = k - 1$. Conclui-se, então, que a expressão (3.51) é menor ou igual a $1 + (t - 3) + (k - t + 1) = k - 1$. □



(a) D_5^+ (b) D_5^-

Figura 3.7: *Exemplos de* D_k^+ *e* D_k^-.

Capítulo 4

Um melhor limite inferior para o Problema do Caixeiro Viajante Assimétrico baseado na Afectação

4.1 Introdução

Neste capítulo, propõe-se uma nova estratégia para obter um limite inferior para o valor óptimo do problema do caixeiro viajante assimétrico melhor do que o que advém do problema de afectação. O algoritmo proposto é um método de primeira ordem baseado na função de penalidade exponencial que requer apenas a resolução sucessiva de problemas de afectação como subproblemas. A definição das direcções de deslocamento usa uma relaxação disjuntiva que se propõe ser de dois tipos, uma baseada em ciclos e a outra baseada em cliques.

O problema de identificar restrições de eliminação de subpercursos para o problema do caixeiro viajante assimétrico é, conforme sugerido por Padberg e Rinaldi em [105], uma aplicação do problema do corte mínimo global num grafo dirigido, descrito na Secção 4.4. Recorda-se o algoritmo de *empurrar-prefluxo* de Goldberg e Tarjan [66] para o problema de determinar o fluxo máximo entre dois vértices designados e o algoritmo de Hao e Orlin [74] que determina o corte global de capacidade mínima num grafo dirigido. Este último baseia-se no algoritmo *DetCorteMin(s)* de Padberg e Rinaldi [105], que resolve a variante do problema de corte mínimo onde um vértice está fixo, e é, essencialmente, uma versão adaptada do algoritmo de *empurrar-prefluxo* de Goldberg e Tarjan. No Apêndice C, são apresentados alguns resultados sobre fluxos e prefluxos que complementam o desenvolvimento desta secção.

O problema de afectação é descrito na Secção 4.5, conjuntamente com o método

clássico utilizado para a sua resolução, o método Húngaro, proposto por Kuhn em [86]. O método usa o algoritmo para determinação do emparelhamento de máxima cardinalidade num grafo bipartido, que se recorda na Secção 4.5.1, e cujo desenvolvimento algorítmico é, essencialmente, baseado em Wolsey [129, Capítulo 4].

4.2 O Problema do Caixeiro Viajante Assimétrico

Seja $G[c] = (V, E)$ um grafo dirigido simples (i.e., sem laços e arcos múltiplos) com n vértices e m arcos tal que a cada arco $(i, j) \in E$ está associado um escalar real $c_{ij} \geqslant 0$ designado por *custo* do arco (i, j). O problema do caixeiro viajante assimétrico consiste em determinar em $G[c]$ um ciclo (dirigido) Hamiltoniano de custo mínimo. Um ciclo Hamiltoniano é um ciclo que passa por todos os vértices - num ciclo não há vértices repetidos. Para este e outros conceitos da teoria dos grafos, acompanhou-se a notação do livro de West [127].

O Problema do Caixeiro Viajante Assimétrico é um problema de optimização combinatória que tem vindo a servir de plataforma de teste para diversos métodos de resolução em Optimização Combinatória. Por isso, também foi a classe de problemas escolhida para testar a metodologia proposta.

A referência [88] sumaria a investigação no ATSP até 1985, enquanto que a referência [73] é mais actual. De entre os métodos mais eficazes, aqueles que têm tido maior sucesso pertencem à classe de métodos *branch-and-bound* (Carpaneto, Dell'Amico e Toth [31] e Fischetti, Lodi e Toth [55] para a versão assimétrica) ou *branch-and-cut* (Applegate et al. [5] e Padberg e Rinaldi [106] para a versão simétrica). Esta classe de algoritmos é essencialmente uma pesquisa em árvore na qual, em cada nó, é resolvido um problema mais simples; se a solução óptima define um ciclo Hamiltoniano, então fica encontrado um limite superior para o valor óptimo, caso contrário, ocorrem ramificações em dois nós ou introdução de planos cortantes, e o algoritmo prossegue recursivamente. É bem sabido que a eficácia desta abordagem é não só dependente da qualidade dos limites superiores que forem sendo obtidos, mas também dos limites inferiores.

Os limites inferiores também podem servir para avaliar o desempenho de heurísticas (em problemas que não possuem o valor óptimo conhecido ou é moroso obtê-lo). Nesse sentido, o limite inferior mais utilizado tem sido o valor óptimo do problema de afectação. Mais recentemente, foi proposto por Cirasella et al. em [38] um limite inferior melhorado, que é essencialmente uma generalização do limite Held-Karp para o caso simétrico [76, 75]. Este limite inferior é o valor óptimo da relaxação linear da formulação clássica, que é recordada na Subsecção 4.2.1.

Motivada pela importância de descobrir bons limites inferiores para o ATSP, é apresentada uma estratégia de obter um limite inferior para o valor óptimo do ATSP melhor do que o que advém do problema de afectação. O algoritmo que se propõe é do tipo Lagrangeano, mas, ao contrário da abordagem clássica, não requer a actualização de variáveis duais e obriga apenas à resolução sucessiva de problemas de afectação como subproblemas.

Na próxima secção, recorda-se a formulação clássica para o Problema do Caixeiro Viajante Assimétrico. Na Secções 4.2.2 e 4.2.3 explica-se como se pode fortalecer uma relaxação linear através relaxações lineares disjuntivas, que usam o facto de que um dado ponto extremo não satisfaz uma restrição de ciclo ou de clique. Explica-se, posteriormente, como optimizar adequadamente uma função linear nessas relaxações lineares disjuntivas. Na Secção 4.2.4, propõe-se um algoritmo que permite obter um limite inferior melhorado para o valor óptimo do ATSP, ilustrando-se algumas iterações desse algoritmo com um exemplo.

4.2.1 A formulação clássica

A formulação clássica do ATSP utiliza uma variável $x_{ij} \in \{0,1\}$ por cada arco $(i,j) \in E$, que indica se o arco (i,j) pertence ou não ao ciclo Hamiltoniano procurado, e é a seguinte:

$$
\begin{aligned}
z = \min \quad & \sum_{(i,j)\in E} c_{ij}x_{ij} & & \\
\text{s.a} \quad & \sum_{(i,j)\in\delta^+(i)} x_{ij} = 1, & i \in V, & \quad (4.1)\\
& \sum_{(i,j)\in\delta^-(j)} x_{ij} = 1, & j \in V, & \quad (4.2)\\
& x_{ij} \geqslant 0, & (i,j) \in E, & \quad (4.3)\\
& \sum_{(i,j)\in E(S)} x_{ij} \leqslant |S| - 1, & S \subseteq V, S \neq \emptyset, V, & \quad (4.4)\\
& x_{ij} \in \mathbb{Z}, & (i,j) \in E, & \quad (4.5)
\end{aligned}
$$

onde $\delta^+(i)$ denota o conjunto dos arcos de G que *saem* do vértice i, e $\delta^-(i)$ denota o conjunto dos arcos de G que *entram* no vértice i.

As restrições (4.1)-(4.3) definem um poliedro que se denota por P^{ASS} (ASS abreviatura de *Assignment*). Os pontos extremos de P^{ASS} satisfazem as restrições (4.5) e são os vectores característicos de ciclos Hamiltonianos e *subpercursos* de G. O valor de $\min\{cx\colon x \in P^{\text{ASS}}\}$ é um limite inferior para z que pode ser obtido em $\mathcal{O}(n^3)$ operações

aritméticas e comparações através do método Húngaro (veja-se Lawler et al. [88], por exemplo), mesmo que a função objectivo seja definida por números reais. Os vectores característicos de ciclos Hamiltonianos em G são os pontos extremos de P^{ASS} que satisfazem as restrições (4.4) e que, por isso, se designam restrições de eliminação de subpercursos. Denota-se por P^{ATSP} o invólucro convexo das soluções admissíveis de (4.1)-(4.5).

O problema $\min\{cx\colon x \in P^{\text{ATSP}}\}$ é $\mathcal{NP}$-difícil (veja-se Lawler et al. [88], por exemplo). No entanto, a sua relaxação linear pode ser resolvida em tempo polinomial através do método elipsóide (Khachiyan [84]), conforme explicado por Grötschel, Lovász e Shrijver em [71] - aqui já se torna necessário que c seja racional. Essencialmente, isso deve-se ao facto de que dado $\overline{x}$ satisfazendo as restrições (4.1)-(4.3), averiguar se $\overline{x}$ satisfaz as restrições (4.4) consiste na resolução de um problema de corte mínimo global em $G[\overline{x}]$ porque, para todo o x que satisfaça as restrições (4.1), tem-se

$$\sum_{(i,j)\in E(S)} x_{ij} = \sum_{i\in S}\left(\sum_{(i,j)\in\delta^+(i)} x_{ij} - \sum_{(i,j)\in E\colon\, j\notin S} x_{ij}\right) = |S| - \sum_{(i,j)\in\delta^+(S)} x_{ij} \tag{4.6}$$

e, portanto, cada uma das restrições (4.4) pode ser substituída por

$$\sum_{(i,j)\in\delta^+(S)} x_{ij} \geqslant 1, \qquad S\subset V, S\neq\emptyset \tag{4.7}$$

sem que a relaxação linear de (4.1)-(4.5) fique diferente. Uma formulação alternativa a (4.1)-(4.5) consiste em substituir as restrições (4.4) por

$$\sum_{(i,j)\in E(C)} x_{ij} \leqslant |E(C)| - 1, \qquad C \text{ ciclo não Hamiltoniano de } G. \tag{4.8}$$

Não é difícil mostrar que a relaxação linear deste novo problema é de qualidade inferior à relaxação linear de (4.1)-(4.5). A sua resolução é também polinomial porque cada uma das restrições (4.8) pode ser reescrita como

$$\sum_{(i,j)\in E(C)} (1 - x_{ij}) \geqslant 1, \qquad C \text{ ciclo não Hamiltoniano de } G.$$

Por isso, averiguar se $\overline{x}$ que satisfaz (4.1)-(4.3) também satisfaz as restrições (4.8) consiste em determinar em $G[1-\overline{x}]$ o ciclo não Hamiltoniano de custo mínimo. Este problema pode ser resolvido em $\mathcal{O}(n^4)$ operações aritméticas e comparações através de uma adaptação do algoritmo de Floyd-Warshall [56, 125].

4.2.2 Uma relaxação linear disjuntiva baseada em ciclos

Seja C um ciclo não Hamiltoniano de G tal que $E(C) = \{(i_1, i_2), (i_2, i_3), \ldots, (i_K, i_1)\} \subseteq E$ e seja P um conjunto poliedral satisfazendo $P^{\text{ATSP}} \subseteq P \subseteq P^{\text{ASS}}$. Defina-se a seguinte relaxação linear disjuntiva de P^{ATSP}, inspirada em Smith, Srinivasan e Thompson [119] e também usada por Carpaneto e Toth em [32],

$$P_{C,i_1} \equiv \text{conv} \left(\bigcup_{k=1}^{K} P_k \right), \tag{4.9}$$

onde

$$\left.\begin{array}{lcl} P_1 & \equiv & P \cap \{x\colon x_{i_1 i_2} = 0\}, \\ P_2 & \equiv & P \cap \{x\colon x_{i_1 i_2} = 1, x_{i_2 i_3} = 0\}, \\ P_3 & \equiv & P \cap \{x\colon x_{i_1 i_2} = x_{i_2 i_3} = 1, x_{i_3 i_4} = 0\}, \\ & \ldots & \\ P_K & \equiv & P \cap \{x\colon x_{i_1 i_2} = x_{i_2 i_3} = \ldots = x_{i_{K-1} i_K} = 1, x_{i_K i_1} = 0\}. \end{array}\right\} \tag{4.10}$$

Não é difícil verificar que P_{C,i_1} é um poliedro que satisfaz

$$P^{\text{ATSP}} \subseteq P_{C,i_1} \subseteq P \cap \left\{ x\colon \sum_{(i,j)\in E(C)} x_{ij} \leqslant |E(C)| - 1 \right\} \subseteq P \subseteq P^{\text{ASS}}.$$

Mais, os poliedros $P_1, P_2, \ldots, P_K$ são faces de P (e de P^{ATSP}). Repare-se que, para cada $k \in \{1, 2, \ldots, K\}$, $P_k = P \cap \{x\colon \alpha^k x = \beta_k\}$, com $\beta_k = -(k-1)$ e $\alpha^k \in \mathbb{R}^m$ um vector linha definido, componente a componente, por

$$\alpha^k_{ij} = \left\{ \begin{array}{ll} 1 & \text{se } (i,j) = (i_k, i_{k+1}), \\ -1 & \text{se } (i,j) \in \{(i_1, i_2), (i_2, i_3), \ldots, (i_{k-1}, i_k)\}, \\ 0 & \text{noutros casos,} \end{array} \right\}, (i,j) \in E.$$

e que $\alpha^k x \geqslant \beta_k$ para todo o $x \in P^{\text{ASS}} \supseteq P$. Não é verdade que, em geral, P_{C,i_1} seja uma face de P.

Lema 4.1 *Seja C um ciclo não Hamiltoniano de G e P um conjunto poliedral satisfazendo $P^{ATSP} \subseteq P \subseteq P^{ASS}$. Se os pontos extremos de P são pontos extremos de P^{ASS}, então $P_{C,v} \equiv P_C$ é independente de $v \in V(C)$, e os pontos extremos de P_C são os pontos extremos de P que satisfazem*

$$\sum_{(i,j)\in E(C)} x_{ij} \leqslant |E(C)| - 1. \tag{4.11}$$

Demonstração: Suponha-se que $E(C) = \{(i_1, i_2), (i_2, i_3), \ldots, (i_K, i_1)\}$. É suficiente provar que $P_{C,i_1} \equiv \text{conv}\,(\cup_{k=1}^{K} P_k)$, definido em (4.9), coincide com $P_{C,i_2} \equiv \text{conv}\,(\cup_{k=1}^{K} P'_k)$, definido por

$$P'_k \equiv P \cap \{x\colon x_{i_2 i_3} = x_{i_3 i_4} = \ldots = x_{i_k i_{k+1}} = 1, x_{i_{k+1} i_{k+2}} = 0\},$$

admitindo que $i_{K+1} \equiv i_1$ e $i_{K+2} \equiv i_2$. Seja $\overline{x}$ um ponto extremo de P_{C,i_1}; então $\overline{x} \in P_k$ para algum $k \in \{1, 2, \ldots, K\}$. Se $k \geqslant 2$, então $\overline{x} \in P'_{k-1}$. Se $k = 1$, então existe $l \in \{1, 2, \ldots, K\}$ tal que

$$\overline{x}_{i_2 i_3} = \overline{x}_{i_3 i_4} = \ldots = \overline{x}_{i_l i_{l+1}} = 1, \overline{x}_{i_{l+1} i_{l+2}} = 0,$$

e, neste caso, $\overline{x} \in P'_l$. Em qualquer dos casos, $\overline{x} \in P_{C,i_2}$. Reciprocamente, seja $\overline{x}$ um ponto extremo de P_{C,i_2} tal que $\overline{x} \in P'_k$. Se $\overline{x}_{i_1 i_2} = 1$, então $k \leqslant K - 1$ e $\overline{x} \in P_{k+1}$; caso contrário, $\overline{x} \in P_1$. Em qualquer dos casos, $\overline{x} \in P_{C,i_1}$.

Os pontos extremos de P_C são os pontos extremos de cada um dos conjuntos $P_k, k = 1, 2, \ldots, K$ que, como é fácil verificar, satisfazem (4.11). Reciprocamente, para todo o ponto extremo x de P que satisfaça (4.11), existe um conjunto P_k tal que $x \in P_k$. Então, x terá que ser ponto extremo de P_C porque $P_k \subset P$. □

Em geral, $P_{C,v} \subseteq P \cap \{x\colon \sum_{(i,j)\in E(C)} x_{ij} \leqslant |E(C)| - 1\}$ para todo $v \in V(C)$. No caso particular do lema acima,

$$P_C = \text{conv}\left(P \cap \left\{x\colon \sum_{(i,j)\in E(C)} x_{ij} \leqslant |E(C)| - 1\right\} \cap \mathbb{Z}^m\right).$$

Uma consequência imediata do Lema 4.1 é a seguinte generalização do procedimento de convexificação sequencial de Balas [9].

Proposição 4.1 *Seja $\mathcal{C} = \{C_1, C_2, \ldots, C_l\}$ o conjunto de todos os ciclos não Hamiltonianos do grafo $G = (V, E)$. Então,*

$$P^{ATSP} = \left(\ldots\left(\left(P^{ASS}_{C_1}\right)_{C_2}\right)_{\ldots}\right)_{C_l}. \tag{4.12}$$

Demonstração: Pelo Lema 4.1, os pontos extremos do conjunto do lado direito de (4.12) são os pontos extremos de P^{ASS} que satisfazem as desigualdades (4.8). □

Aborda-se agora a questão da existência de um hiperplano separador entre P_{C,i_1}, para algum ciclo não Hamiltoniano C, e um ponto extremo de P.

Lema 4.2 *Seja $\overline{x}$ um ponto extremo de um poliedro P tal que $P^{ATSP} \subseteq P \subseteq P^{ASS}$.*

Então,

1. *se existe um ciclo não Hamiltoniano C tal que $\sum_{(i,j)\in E(C)}(1-\overline{x}_{ij})<1$, então $\overline{x}\notin P_{C,v}$ para todo $v\in V(C)$;*
2. *se $\overline{x}_{ij}\in(0,1)$ para algum $(i,j)\in E$, então $\overline{x}\notin P_{C,i}$ para todo o ciclo não Hamiltoniano C que contenha o arco (i,j).*

Demonstração: Para a primeira parte, por hipótese, $\sum_{(i,j)\in E(C)}\overline{x}_{ij}>|E(C)|-1$, mas, para todo $v\in V(C)$ e para todo $x\in P_{C,v}$, $\sum_{(i,j)\in E(C)}x_{ij}\leqslant|E(C)|-1$. Conclui-se que $\overline{x}\notin P_{C,v}$. Para a segunda parte, se $\overline{x}$ pertencesse a $P_{C,i}$, então também seria um ponto extremo de $P_{C,i}$, o que é absurdo pois todos os pontos extremos de $P_{C,i}$ possuem zero ou um na componente da posição (i,j). □

Pelo Lema 4.2, qualquer ciclo não Hamiltoniano que contenha o arco (i,j) associado a uma componente fraccionária de $\overline{x}$ serve o propósito de identificar um hiperplano separador. Pode-se, por exemplo, escolher o caminho mais curto de j para i no conjunto dos grafos $G[1-\overline{x}]-\{k\}$ $(k\in V\setminus\{i,j\})$ que, conjuntamente com o arco (i,j), define um ciclo pretendido. Em alternativa, pode-se resolver o problema do ciclo não Hamiltoniano mais curto em $G[1-\overline{x}]$ da maneira que se explica a seguir.

Seja $G[d]=(V,E)$ um grafo dirigido tal que a cada arco $(i,j)\in E$ está associado um escalar $d_{ij}\geqslant 0$ que se designa por *distância* do arco (i,j). O mais curto ciclo não Hamiltoniano em $G[d]$ possui distância total igual

$$\min_{k\in V}\left\{\min\left\{\pi^k_{ji}+d_{ij}\colon (i,j)\in E, i,j\in V\setminus\{k\}\right\}\right\}, \tag{4.13}$$

onde π^k_{ji} denota o comprimento do caminho mais curto em $G[d]$ do vértice j para o vértice i sem passar pelo vértice k. Para cada $k\in K$ fixo, todos aqueles valores podem ser calculados através do algoritmo de Floyd-Warshall [56, 125], que pode funcionar em tempo proporcional a $\mathcal{O}(n^3)$ [1, Secção 5.6], aplicado ao grafo $G[d]-\{k\}$. Portanto, o mais curto ciclo não Hamiltoniano em $G[d]$ pode ser identificado em $\mathcal{O}(n^4)$ operações (este número pode ser reduzido para $\mathcal{O}(n^3)$). Se (4.13) for inferior a um, então a respectiva solução óptima C é tal que $\overline{x}\notin P_{C,v}$ para todo $v\in V(C)$. Se (4.13) for superior ou igual a um e $\overline{x}_{ij}\in(0,1)$, então o ciclo não Hamiltoniano C de comprimento

$$\min\left\{\pi^k_{ji}+d_{ij}\colon k\in V\setminus\{i,j\}\right\}, \tag{4.14}$$

é tal que $\overline{x}\notin P_{C,i}$. Em ambos os casos, é possível identificar um hiperplano separador, óptimo nalgum aspecto.

Para $P=P^{\text{ASS}}$, o problema de optimização linear sobre P_C pode ser resolvido em $\mathcal{O}(n^3)$ operações aritméticas e comparações, portanto, comparável ao tempo de resolução de

apenas um problema de afectação. De facto, a resolução do problema

$$\begin{array}{ll}\min & cx \\ \text{s.a} & x \in P_C\end{array} \quad = \quad \min_{k=1,2,\ldots,K} \left(\begin{array}{ll}\min & cx \\ \text{s.a} & x \in P_k\end{array} \right) \tag{4.15}$$

pode fazer-se parametricamente através de uma optimização (envolvendo $\mathcal{O}(n^3)$ operações) e $K-1$ reoptimizações (cada uma envolvendo $\mathcal{O}(n)$ operações) da maneira explicada a seguir.

Suponha-se que se pretende obter um par de soluções primal-dual óptimas para $z_{k+1} = \min\{cx\colon x \in P_{k+1}\}$, sendo conhecida essa informação para $z_k = \min\{cx\colon x \in P_k\}$. Seja $\overline{x} \in \{0,1\}^m$ uma solução primal óptima de

$$z_k = \left\{ \begin{array}{lll}\min & cx & \\ \text{s.a} & x \in P & \\ & x_{ij} = 1, & (i,j) \in E(C_k) \equiv \{(i_1,i_2),(i_2,i_3),\ldots,(i_{k-1},i_k)\}, \\ & x_{ij} = 0, & (i,j) = (i_k,i_{k+1}),\end{array} \right\} \tag{4.16}$$

que é vector característico de um emparelhamento perfeito M, e seja $(\overline{u},\overline{v}) \in \mathbb{R}^{2n}$ uma solução óptima para o correspondente problema dual

$$z_k \quad = \quad \sum_{(i,j)\in E(C_k)} c_{ij} + \left\{ \begin{array}{lll}\max & \displaystyle\sum_{i\notin\{i_1,i_2,\ldots,i_{k-1}\}} u_i + \sum_{j\notin\{i_2,i_3,\ldots,i_k\}} v_j & \\ \text{s.a} & u_i + v_j \leqslant c_{ij}, & (i,j) \in E \setminus E(C_{k+1}).\end{array} \right\}$$

Relativamente a (4.16), o novo problema $z_{k+1} = \min\{cx\colon x \in P_{k+1}\}$ possui a restrição "$x_{i_k i_{k+1}} = 1$" no lugar da restrição "$x_{i_k i_{k+1}} = 0$" e uma nova restrição "$x_{i_{k+1} i_{k+2}} = 0$". Consequentemente, no novo problema dual aparece mais uma parcela constante "$c_{i_k i_{k+1}}$" e desaparecem as variáveis u_{i_k} e $v_{i_{k+1}}$ da função objectivo, e desaparece a restrição "$u_{i_{k+1}} + v_{i_{k+2}} \leqslant c_{i_{k+1} i_{k+2}}$". Por isso, o vector $\overline{x}$ já não é primal admissível, mas o vector $(\overline{u},\overline{v})$ permanece dual admissível no novo problema, e "$z_k + c_{i_k i_{k+1}} - \overline{u}_{i_k} - \overline{v}_{i_{k+1}}$" surge como primeiro limite inferior ao valor de z_{k+1}. Pode-se iniciar o método Húngaro com a solução dual admissível $(\overline{u},\overline{v})$ e o vector característico do emparelhamento

$$M' \equiv M \setminus \{(i,i_{k+1}),(i_{k+1},j),(i_{k+1},i_{k+2})\} \cup \{(i_k,i_{k+1})\}$$

para algum $i \notin \{i_1,i_2,\ldots,i_{k-1},i_k\}$ e $j \in \{i_1,i_2,\ldots,i_{k-1},i_k\}$. O novo emparelhamento M' possui menos dois arcos se (i_{k+1},i_{k+2}) pertencia ao emparelhamento anterior M ou menos um arco se não pertencia. A solução óptima do novo problema pode ser então obtida em $\mathcal{O}(n)$ operações aritméticas e comparações. Como existem, no máximo, $K-1$ reoptimizações a fazer e $K < n$, conclui-se que o esforço computacional global na resolução do problema (4.15) é $\mathcal{O}(n^3)$.

4.2.3 Uma relaxação linear disjuntiva baseada em cliques

Nesta secção, é construída uma relaxação disjuntiva de P^{ATSP} que usa cliques do mesmo modo que a relaxação anterior usava ciclos. A nova relaxação requer um número muito mais elevado de disjunções. Também neste caso, a optimização nesse poliedro pode ser feita de modo paramétrico.

Seja $S \subseteq V$ um conjunto de cardinalidade $K < n$. O número de caminhos simples em $G[S]$, o subgrafo de G induzido por S, a partir de um determinado vértice de S é, no máximo, igual a

$$p_K = \sum_{k=1}^{K} \frac{(K-1)!}{(K-k)!} \quad \big(= 1 + (K-1)p_{K-1} \big). \tag{4.17}$$

Por exemplo, se $S = \{i_1, i_2, i_3\}$ e i_1 é o vértice designado, cada um desses caminhos corresponde a um arco da subárvore da esquerda da árvore enumerativa ilustrada na Figura 4.1. Por exemplo, o arco 1 está associado ao caminho constituído pelo vértice i_1 e nenhum arco, o arco 2 está associado ao caminho $i_1(i_1, i_2)i_2$, etc.

Para um conjunto poliedral P satisfazendo $P^{\text{ATSP}} \subseteq P \subseteq P^{\text{ASS}}$, defina-se a seguinte relaxação linear disjuntiva de P^{ATSP},

$$P_S \equiv \text{conv} \left(\bigcup_{(j_1,j_2,\ldots,j_k):\ \{j_1,j_2,\ldots,j_k\} \subseteq S, k \geqslant 1} P_{(j_1,j_2,\ldots,j_k)} \right), \tag{4.18}$$

onde cada conjunto $P_{(j_1,j_2,\ldots,j_k)}$ é definido pelos pontos $x \in P$ tais que

$$\begin{aligned} x_{ij} &= 1, \quad (i,j) \in E(C_k), \\ x_{ij} &= 0, \quad (i,j) \in E(S) \setminus E(C_k), j \in V(C_k), \end{aligned}$$

sendo $C_k \equiv \{j_1, (j_1, j_2), j_2, (j_2, j_3), j_3, \ldots, j_{k-1}, (j_{k-1}, j_k), j_k\}$. Portanto, o conjunto $P_{(j_1,j_2,\ldots,j_k)} \cap \mathbb{Z}^m$ é o subconjunto de $P^{\text{ASS}} \cap \mathbb{Z}^m$ dos vectores que pertencem a P e são característicos de conjuntos de arcos que usam o caminho

$$j_0(j_0, j_1)j_1(j_1, j_2)j_2(j_2, j_3)j_3 \ldots j_k(j_k, j_{k+1})j_{k+1}$$

para alguns $j_0, j_{k+1} \in V \setminus S$ - veja-se a Figura 4.2. No caso do exemplo da Figura 4.1, existem 15 poliedros na definição (4.18), que se apresentam pela ordem de numeração dos arcos:

$$
\begin{aligned}
P_{(i_1)} &\equiv P \cap \{x\colon x_{i_1i_2} = x_{i_2i_1} = x_{i_1i_3} = x_{i_3i_1} = 0\},\\
P_{(i_2)} &\equiv P \cap \{x\colon x_{i_2i_1} = x_{i_1i_2} = x_{i_2i_3} = x_{i_3i_2} = 0\},\\
P_{(i_3)} &\equiv P \cap \{x\colon x_{i_3i_1} = x_{i_1i_3} = x_{i_3i_2} = x_{i_2i_3} = 0\},\\
P_{(i_1,i_2)} &\equiv P \cap \{x\colon x_{i_1i_2} = 1, x_{i_2i_1} = x_{i_2i_3} = x_{i_3i_1} = 0\},\\
P_{(i_1,i_2,i_3)} &\equiv P \cap \{x\colon x_{i_1i_2} = x_{i_2i_3} = 1, x_{i_3i_1} = 0\},\\
P_{(i_1,i_3)} &\equiv P \cap \{x\colon x_{i_1i_3} = 1, x_{i_3i_1} = x_{i_3i_2} = x_{i_2i_1} = 0\},\\
P_{(i_1,i_3,i_2)} &\equiv P \cap \{x\colon x_{i_1i_3} = x_{i_3i_2} = 1, x_{i_2i_1} = 0\},\\
P_{(i_2,i_1)} &\equiv P \cap \{x\colon x_{i_2i_1} = 1, x_{i_1i_2} = x_{i_1i_3} = x_{i_3i_2} = 0\},\\
P_{(i_2,i_1,i_3)} &\equiv P \cap \{x\colon x_{i_2i_1} = x_{i_1i_3} = 1, x_{i_3i_2} = 0\},\\
P_{(i_2,i_3)} &\equiv P \cap \{x\colon x_{i_2i_3} = 1, x_{i_3i_1} = x_{i_3i_2} = x_{i_1i_2} = 0\},\\
P_{(i_2,i_3,i_1)} &\equiv P \cap \{x\colon x_{i_2i_3} = x_{i_3i_1} = 1, x_{i_1i_2} = 0\},\\
P_{(i_3,i_1)} &\equiv P \cap \{x\colon x_{i_3i_1} = 1, x_{i_1i_2} = x_{i_1i_3} = x_{i_2i_3} = 0\},\\
P_{(i_3,i_1,i_2)} &\equiv P \cap \{x\colon x_{i_3i_1} = x_{i_1i_2} = 1, x_{i_2i_3} = 0\},\\
P_{(i_3,i_2)} &\equiv P \cap \{x\colon x_{i_3i_2} = 1, x_{i_2i_1} = x_{i_2i_3} = x_{i_1i_3} = 0\},\\
P_{(i_3,i_2,i_1)} &\equiv P \cap \{x\colon x_{i_3i_2} = x_{i_2i_1} = 1, x_{i_1i_3} = 0\},
\end{aligned}
$$

Não é difícil verificar que P_S é um poliedro que satisfaz

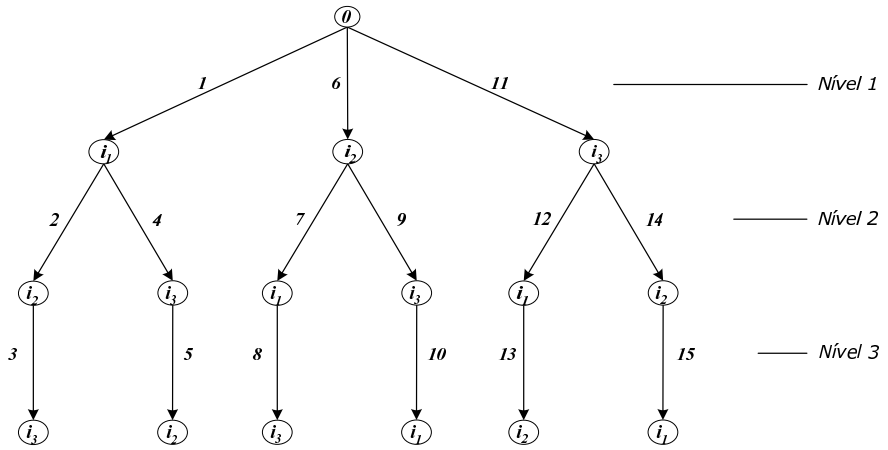


Figura 4.1: *Árvore de enumeração dos caminhos simples em* $S = \{i_1, i_2, i_3\}$.

$$
P^{\mathrm{ATSP}} \subseteq P_S \subseteq P \cap \left\{ x\colon \sum_{(i,j)\in E(S)} x_{ij} \leqslant |S| - 1 \right\} \subseteq P \subseteq P^{\mathrm{ASS}}.
$$

O poliedro P_S é o invólucro convexo da união de, no máximo, $p = Kp_K$ poliedros. Cada um desses poliedros $P_l \equiv P_{(j_1,j_2,\ldots,j_k)}$, com $l = 1, 2, \ldots, p$, é uma face de P. Mais, se os pontos extremos de P são pontos extremos de P^{ASS}, então os pontos extremos

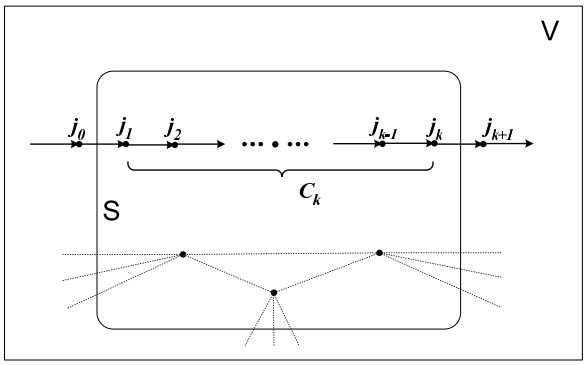


Figura 4.2: $P_{(j_1,j_2,\ldots,j_k)}$.

de P_S são os pontos extremos de P que satisfazem

$$\sum_{(i,j)\in E(S)} x_{ij} \leqslant |S| - 1. \tag{4.19}$$

De facto, os pontos extremos de P_S são os pontos extremos de cada um dos conjuntos P_l, para $l = 1, 2, \ldots, p$, que satisfazem (4.19). Além disso, se $x \in P_l$ então, para algum $k \in S$,

$$\sum_{(i,j)\in E(S)} x_{ij} = \sum_{i\in S\setminus\{k\}} \left(\sum_{j\in S:\ (i,j)\in E} x_{ij} \right) \leqslant |S| - 1.$$

Reciprocamente, para todo o ponto extremo x de P que seja vector característico e satisfaça (4.19), existe um conjunto P_l tal que $x \in P_l$. Então, x terá que ser ponto extremo de P_S porque $P_l \subseteq P$. Uma consequência imediata é a seguinte generalização do procedimento de convexificação sequencial de Balas [9]. A sua demonstração é idêntica à da Proposição 4.1.

Proposição 4.2 *Seja $\mathcal{S} = \{S_1, S_2, \ldots, S_l\}$ a família de todos os subconjuntos próprios de V. Então,*

$$P^{ATSP} = \left(\ldots \left(\left(P^{ASS}_{S_1} \right)_{S_2} \right)_{\ldots} \right)_{S_l}. \tag{4.20}$$

Aborda-se agora a questão da existência de um hiperplano separador entre P^{ATSP} e um ponto extremo de uma relaxação linear de P^{ATSP} do tipo descrito.

Lema 4.3 *Seja P tal que $P^{ATSP} \subseteq P \subseteq P^{ASS}$, e seja $\overline{x}$ um ponto extremo de P tal que $\overline{x}_{uv_1}, \overline{x}_{uv_2} \in (0,1)$ para alguns $(u,v_1),(u,v_2) \in E$. Então,*

1. *se existe um conjunto $S \subset V$ tal que $\sum_{(i,j)\in\delta^+(S)} \overline{x}_{ij} < 1$, então $\overline{x} \notin P_S$;*

2. *para todo o conjunto* S *tal que* $\sum_{(i,j)\in\delta^+(S)} \overline{x}_{ij} = 1$, $u, v_1 \in S$ *e* $v_2 \notin S$, *tem-se* $\overline{x} \notin P_S$.

Demonstração: Para a primeira parte, atendendo a (4.6), por hipótese, $\sum_{(i,j)\in E(S)} \overline{x}_{ij} > |S| - 1$. Mas, para todo $x \in P_S$, tem-se $\sum_{(i,j)\in E(S)} x_{ij} \leqslant |S| - 1$. Conclui-se que $\overline{x} \notin P_S$. Para a segunda parte, se $\overline{x}$ pertencesse a P_S, então também seria um ponto extremo de um dos conjuntos $P_l, l = 1, 2, \ldots, p$. Por isso, existe um vértice $k \in S$ tal que

$$1 = \sum_{w\notin S:\ (k,w)\in E} \overline{x}_{kw} \leqslant \sum_{(i,j)\in\delta^+(S)} \overline{x}_{ij} = 1. \tag{4.21}$$

Se $k = u$, chega-se a um absurdo porque $\overline{x}_{uv_1} > 0$ e $v_1 \in S$, o que implica $\sum_{w\notin S:\ (u,w)\in E} \overline{x}_{uw} < 1$. Se $k \neq u$, também se chega a um absurdo porque, de (4.21),

$$0 = \sum_{w\notin S:\ (u,w)\in E} \overline{x}_{uw} \geqslant \overline{x}_{uv_2} > 0.$$

□

Pelo Lema 4.3, conhecido um ponto extremo $\overline{x}$ de uma relaxação linear P de P^{ATSP} com uma componente $\overline{x}_{ij}$ fraccionária, pode-se identificar um conjunto $S \subset V$ tal que $\overline{x} \notin P_S$ através da resolução do problema do corte global de capacidade mínima no grafo $G[\overline{x}]$ da maneira que se explica a seguir.

Seja $G[u] = (V, E)$ um grafo dirigido simples com n vértices e m arcos tal que a cada arco $(i, j) \in E$ está associado um escalar real $u_{ij} \geqslant 0$ designado por *capacidade* do arco (i, j). O problema de determinar o corte global de capacidade mínima em G pode escrever-se como

$$\min \left\{ \sum_{(i,j)\in\delta^+(S)} u_{ij} \colon S \subset V, S \neq \emptyset \right\}. \tag{4.22}$$

Se se englobassem as restrições $s \in S$ e $t \in V \setminus S$ para dois vértices designados, então o problema (4.22) seria dual do problema de determinar o fluxo máximo de s para t (veja-se Ford e Fulkerson [57]). Não estando especificados os vértices origem s e terminal t, o problema (4.22) é dual do problema de determinar o maior valor de fluxo que é possível enviar entre qualquer par de vértices de G. Consequentemente, o problema (4.22) pode ser resolvido após a resolução de $n(n-1)/2$ problemas de fluxo máximo com vértices origem e terminal designados. Hao e Orlin em [74] propuseram um algoritmo específico que usa apenas uma sequência de $2n - 2$ problemas de fluxo máximo. A abordagem, que é inspirada no algoritmo *preflow-push* de Golberg e Tarjan [66] (para o problema do fluxo máximo entre dois vértices designados) e na abordagem de Padberg e Rinaldi [105], resulta num algoritmo que requer $\mathcal{O}(nm \log(n^2/m))$ operações aritméticas e comparações se u for um vector de números racionais. Portanto, o esforço

computacional é comparável à resolução de n problemas de fluxo máximo entre dois vértices designados.

Se o valor óptimo de (4.22), para $u = \overline{x}$, for inferior a um, então a solução óptima S é tal que $\overline{x} \notin P_S$ pelo Lema 4.3. Se o valor óptimo de (4.22) for igual a um, então um conjunto S, nas condições do Lema 4.3, pode ser encontrado do seguinte modo. Se $f_2 = \overline{x}_{uv_2} \in (0,1)$, então determine-se o fluxo máximo de u para v_2 no grafo $G[\overline{x}] - \{(u, v_2)\}$, que se sabe saber ser igual a $1 - f_2$. Pelo Teorema do Fluxo Máximo-Corte Mínimo, existe um conjunto de vértices S, contendo u e todos os vértices $v \neq v_2$ tais que $\overline{x}_{uv} \in (0,1)$, tal que, relativamente ao grafo $G - \{(u, v_2)\}$,

$$\sum_{(i,j)\in\delta^+(S)} \overline{x}_{ij} = 1 - f_2.$$

Por isso, relativamente ao grafo G original, $\sum_{(i,j)\in\delta^+(S)} \overline{x}_{ij} = 1 - f_2 + f_2 = 1$. Pelo Lema 4.3, $\overline{x} \notin P_S$. Conclui-se que a determinação de S pode ser, em qualquer dos casos, efectuada em tempo polinomial. Este conjunto S, assim determinado, é óptimo em algum aspecto.

Para $P = P^{\text{ASS}}$, o problema de optimização linear sobre P_S pode ser resolvido em $\mathcal{O}(n^3 + (p-1)n)$ operações aritméticas e comparações. De facto, a resolução do problema

$$\begin{array}{ll} \min & cx \\ \text{s.a} & x \in P_S \end{array} \quad = \quad \min_{(j_1,j_2,\ldots,j_k):\ \{j_1,j_2,\ldots,j_k\}\subseteq S, k\geqslant 1} \left(\begin{array}{ll} \min & cx \\ \text{s.a} & x \in P_{(j_1,j_2,\ldots,j_k)} \end{array} \right) \tag{4.23}$$

pode fazer-se parametricamente através de uma optimização (envolvendo $\mathcal{O}(n^3)$ operações) e $K-1$ reoptimizações (cada uma envolvendo $\mathcal{O}(n)$ operações) da maneira que se explica a seguir.

Sejam P_k e P_{k+1} poliedros associados a um arco do nível k e a um arco do nível $k+1$, sucessivos, na árvore de enumeração dos caminhos, conforme ilustrado na Figura 4.1. Seguidamente, explica-se como se pode obter um par de soluções primal-dual óptimas para $z_{k+1} = \min\{cx\colon x \in P_{k+1}\}$, sendo conhecida essa informação para $z_k = \min\{cx\colon x \in P_k\}$. Seja $\overline{x} \in \{0,1\}^m$ uma solução primal óptima de

$$z_k = \left\{ \begin{array}{lll} \min & cx & \\ \text{s.a} & x \in P & \\ & x_{ij} = 1, & (i,j) \in E(C_k) \equiv \{(j_1,j_2),(j_2,j_3),\ldots,(j_{k-1},j_k)\}, \\ & x_{ij} = 0, & (i,j) \in E(S) \setminus E(C_k), j \in V(C_k), \end{array} \right\} \tag{4.24}$$

que é vector característico de um emparelhamento perfeito M, e seja $(\overline{u}, \overline{v}) \in \mathbb{R}^{2n}$ uma

solução óptima para o correspondente problema dual

$$z_k = \sum_{(i,j)\in E(C_k)} c_{ij} + \left\{ \begin{array}{lll} \max & \sum_{i\notin\{i_1,i_2,\ldots,i_{k-1}\}} u_i + \sum_{j\notin\{i_2,i_3,\ldots,i_k\}} v_j & \\ \text{s.a} & u_i + v_j \leqslant c_{ij}, & (i,j) \in E \setminus E(S), \\ & u_i + v_j \leqslant c_{ij}, & (i,j) \in E(S) \setminus E(C_k), j \notin V(C_k). \end{array} \right\}$$

Relativamente a (4.24), o novo problema $z_{k+1} = \min\{cx\colon x \in P_{k+1}\}$ possui a restrição "$x_{j_k j_{k+1}} = 1$" no lugar da restrição "$x_{j_k j_{k+1}} = 0$" e novas restrições "$x_{j_{k+1} j} = 0, j \in S\setminus V(C_{k+1})$" e "$x_{i j_{k+1}} = 0, i \in S\setminus V(C_{k+1})$". No novo problema dual, aparece mais uma parcela constante "$c_{j_k j_{k+1}}$" e desaparecem as variáveis u_{j_k} e $v_{j_{k+1}}$ da função objectivo, e desaparecem as restrições "$u_{j_{k+1}} + v_j \leqslant c_{j_{k+1} j}, j \in S \setminus V(C_k)$". Por isso, o vector $\overline{x}$ já não é primal admissível, mas o vector $(\overline{u}, \overline{v})$ permanece dual admissível no novo problema e "$z_k + c_{j_k j_{k+1}} - \overline{u}_{j_k} - \overline{v}_{j_{k+1}}$" surge como limite inferior ao valor óptimo do novo problema. Pode-se inicializar o método Húngaro com a solução dual admissível $(\overline{u}, \overline{v})$ e o vector característico do emparelhamento

$$M' \equiv M \setminus \{(s, j_{k+1}), (j_k, t), (j_{k+1}, w)\} \cup \{(j_k, j_{k+1})\}$$

para algum $t \notin S$ e $s, w \in V$ - não é necessário excluir (j_{k+1}, w) se $w \notin S$. A solução óptima do novo problema pode ser então obtida em $\mathcal{O}(n)$ operações aritméticas e comparações. Como existem, no máximo, $p-1$ reoptimizações a fazer, conclui-se que o esforço computacional global na resolução de (4.23) é $\mathcal{O}(n^3 + (p-1)n)$. Realce-se que se a árvore for percorrida por *breath-first-search*, então as soluções iniciais estarão prontamente disponíveis se forem armazenadas uma por cada nível. Portanto, não é necessário armazenar mais do que um par primal-dual óptimo por nível.

4.2.4 Determinação de um limite inferior melhorado

Nesta secção, propõe-se uma abordagem Lagrangeana para obter um limite inferior melhorado para o valor de $z = \min\{cx\colon x \in P^{\text{ATSP}}\}$ que usa apenas a resolução de problemas de afectação como subproblemas. O algoritmo proposto, que é formalmente descrito na Figura 4.3, é essencialmente um método descendente de primeira ordem para a minimização da função de penalidade

$$f(x) \equiv cx + \rho \sum_{i=1}^{K} \left(e^{(a_i x - b_i)/\rho} - 1 \right), \tag{4.25}$$

onde "$a_i x \leqslant b_i$", para $i = 1, 2, \ldots, K$, são desigualdades válidas para P^{ATSP}. Este conjunto de desigualdades vai sendo expandido à medida que o algoritmo decorre.

Em cada iteração genérica-k do algoritmo proposto, começa-se por identificar uma direcção de descida, usando informação de primeira ordem. Observe-se que

$$\nabla f(x) = c + y(x)A, \quad \nabla^2 f(x) = A^T Y(x) A/\rho,$$

onde A é uma matriz $K \times n$, contendo as colunas a_i^T para $i = 1, 2, \ldots, K$, e $Y(x)$ é uma matriz diagonal $K \times K$ cujos elementos diagonais coincidem com o vector $y(x)$ definido, componente a componente, por

$$y_i(x) = e^{(a_i x - b_i)/\rho}, \qquad i = 1, 2, \ldots, K.$$

Enquanto não é identificada uma direcção de descida, o parâmetro de penalidade é reduzido. Depois de identificada uma direcção de descida $d_k = q_k - x_k$ para f a partir de x_k e para algum $q_k \in \arg\min\{\nabla f(x_k) q \colon q \in P^{\text{ASS}}\} \cap \mathbb{Z}^m$, decide-se o tamanho do passo λ_k ao longo dessa direcção para obter a nova aproximação $x_{k+1} = x_k + \lambda_k d_k$. O escalar λ_k é solução óptima para $\min\{g(\lambda) \equiv f(x + \lambda d) \colon \lambda \in (0, 1]\}$. Se $g'(1) \leqslant 0$, então $\lambda_k = 1$ é a solução óptima porque g é convexa. Caso contrário, o escalar λ_k deve ser aproximado através do método de Newton. Algumas simplificações ocorrem na correspondente fórmula recursiva. Para x e d fixos, tem-se

$$g(\lambda) = c(x + \lambda d) + \rho \sum_{i=1}^{K} \left(e^{(a_i(x+\lambda d) - b_i)/\rho} - 1 \right) = z_0 + \lambda w_0 + \rho \sum_{i=1}^{K} \left(y_i(\lambda) - 1 \right),$$

onde $z_0 = cx, w_0 = cd, z = Ax - b, w = Ad$ e $y(\lambda) \equiv y(x + \lambda d) = e^{(z_i + \lambda w_i)/\rho}$. Por isso,

$$\begin{aligned} g'(\lambda) &= (c + y(\lambda)A)\, d = w_0 + \sum_{i=1}^{K} w_i y_i(\lambda), \\ g''(\lambda) &= d^T A^T Y(\lambda) A d/\rho = \sum_{i=1}^{K} w_i^2 y_i(\lambda)/\rho, \end{aligned}$$

pelo que o método de Newton consiste na aplicação da seguinte fórmula recursiva a partir de $\lambda^{(0)} = 0$,

$$\lambda^{(j+1)} = \lambda^{(j)} - \frac{g'(\lambda^{(j)})}{g''(\lambda^{(j)})} = \lambda^{(j)} - \rho \left[\frac{w_0 + \sum_{i=1}^{K} y_i(\lambda^{(j)}) w_i}{\sum_{i=1}^{K} y_i(\lambda^{(j)}) w_i^2} \right], \qquad j = 0, 1, \ldots .$$

O último passo da iteração genérica-k consiste na identificação do corte global $\delta^+(S)$ de capacidade mínima em $G[q_k]$. Note-se que q_k é um vector de zeros e uns e, por isso, a identificação de $\delta^+(S)$ pode ser efectuada por um método *standard* de averiguação de conexidade num grafo.

Segue-se a determinação da desigualdade válida para P_S^{ASS} *mais profunda* entre q_k e P_S^{ASS}. A determinação desse corte obriga à resolução do seguinte par de problemas para $\overline{x} = q_k$ e $S \subset V$:

$$\begin{array}{ll} \min & ||x - \overline{x}|| \\ \text{s.a} & x \in P_S^{\text{ASS}} \end{array} \quad \equiv \quad \begin{array}{ll} \max & \alpha\overline{x} - \beta \\ \text{s.a} & (\alpha, \beta) \in \text{polar } (P_S^{\text{ASS}}), \quad ||\alpha||_* \leqslant 1. \end{array} \tag{4.26}$$

Usando o algoritmo modificado de P. Wolfe, conforme descrito no Capítulo 2, obtém-se a solução do problema (4.26), resolvendo uma sequência de problemas lineares:

$$\begin{array}{ll} \min & ax \\ \text{s.a} & x \in P_S^{\text{ASS}} \end{array} \quad = \quad \min_{(j_1, j_2, \ldots, j_k) \colon \{j_1, j_2, \ldots, j_k\} \subseteq S, k \geqslant 1} \left(\begin{array}{ll} \min & ax \\ \text{s.a} & x \in P_{(j_1, j_2, \ldots, j_k)}^{\text{ASS}} \end{array} \right), \tag{4.27}$$

com a modificado ao longo do algoritmo. Tal como foi explicado na Secção 4.2.3, esta resolução pode ser feita parametricamente.

Finalmente, a inclusão do corte "$a_{K+1}x \leqslant b_{K+1}$" assim obtido na função penalidade só é efectivada se o ângulo entre a sua normal e a normal de cada corte previamente adicionado for suficientemente não nulo. Balas, Ceria e Cornuéjols em [16, 17], propuseram a seguinte regra de decisão. Escolha-se um valor para o parâmetro θ, inferior a 1 e, por exemplo, 0.9999. Se os co-senos dos ângulos formados entre a normal do corte a adicionar e a de cada corte já adicionado forem pelo menos θ, então o corte é inserido na função (4.25).

Resta agora explicar como calcular o limite inferior. A função f goza das seguintes propriedades. Como $f(x) \leqslant cx$ para todo $x \in P^{\text{ASS}}$ tal que $a_i x \leqslant b_i$ para $i = 1, 2, \ldots, K$, então

$$\begin{aligned} z \equiv \min\{cx \colon x \in P^{\text{ATSP}}\} &\geqslant \min\{cx \colon x \in P^{\text{ASS}},\ a_i x \leqslant b_i,\ i = 1, 2, \ldots, K\} \\ &\geqslant \min\{f(x) \colon x \in P^{\text{ASS}},\ a_i x \leqslant b_i,\ i = 1, 2, \ldots, K\} \\ &\geqslant \min\{f(x) \colon x \in P^{\text{ASS}}\}. \end{aligned}$$

Além disso, f é convexa e continuamente diferenciável. Por isso, $f(x_k) + \nabla f(x_k)d_k$ constitui um limite inferior ao valor de z, uma vez que

$$\min\{f(x)\colon x \in P^{\text{ASS}}\} \geqslant \min\{f(x_k) + \nabla f(x_k)(x - x_k)\colon x \in P^{\text{ASS}}\} = f(x_k) + \nabla f(x_k)d_k.$$

Inicialização:

Determinar $x_0 \in \arg\min\{cx\colon x \in P^{\text{ASS}}\} \cap \mathbb{Z}^m$.

Seja $\delta^+(S)$ o corte global de capacidade mínima em $G[x_0]$.

Obter o corte $a_1 x \leqslant b_1$ mais profundo entre x_0 e P_S^{ASS}.

Inicializar $\rho = 1$, $K = 1$, $k = 0$ e definir f através de (4.25).

Iteração genérica-k:

Enquanto $x_k \in \arg\min\{\nabla f(x_k)q\colon q \in P^{\text{ASS}}\}$

Reduzir ρ.

Se $\rho < \epsilon_{\text{TOL}}$, então STOP.

Seja $d_k = q_k - x_k$ para algum $q_k \in \arg\min\{\nabla f(x_k)q\colon q \in P^{\text{ASS}}\} \cap \mathbb{Z}^m$.

Determinar $\lambda_k \in \arg\min\{f(x_k + \lambda d_k)\colon \lambda \in (0, 1]\}$.

Afectar $x_{k+1} = x_k + \lambda_k d_k$.

Seja $\delta^+(S)$ o corte global de capacidade mínima em $G[q_k]$.

Obter o corte $a_{K+1}x \leqslant b_{K+1}$ mais profundo entre q_k e P_S^{ASS}.

Se "$a_{K+1}x \leqslant b_{K+1}$" é *suficientemente* distinto dos cortes anteriores, então $K := K + 1$.

$k := k + 1$.

Figura 4.3: *Algoritmo para determinar um limite inferior melhorado para o ATSP.*

4.2.4.1 Ilustração com um pequeno exemplo

Considere-se a instância do problema caixeiro viajante assimétrico, observada em Balas e Toth [19, Pag. 381], cujo custo do arco genérico (i, j) corresponde à entrada (i, j) da matriz da Tabela 4.1. Este problema tem valor óptimo 26, que corresponde ao ciclo Hamiltoniano identificado na Figura 4.4(a) através de $G[x^*]$.

São ilustradas seis iterações do algoritmo da Figura 4.3, utilizando a regra de redução do parâmetro de penalidade $\rho := \min\{\rho/10, \rho^{1.5}\}$, inspirada em Cominetti e Dussault [39], Dussault [48] e Mongeau e Sartenaer [98], e a norma l_∞ em (4.26). Como

-	2	11	10	8	7	6	5
6	-	1	8	8	4	6	7
5	12	-	11	8	12	3	11
11	9	10	-	1	9	8	10
11	11	9	4	-	2	10	9
12	8	5	2	11	-	11	9
10	11	12	10	9	12	-	3
7	10	10	10	6	3	1	-

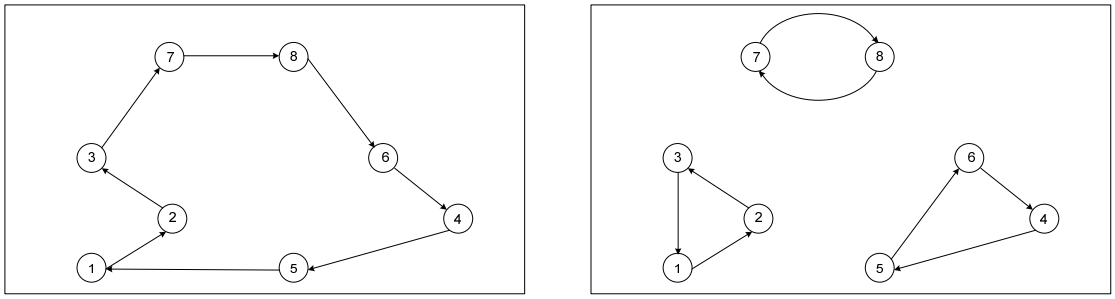
Tabela 4.1: *Matriz de custos de uma instância do ATSP com* 8 *vértices.*

se pode observar, o primeiro limite inferior é 17 e, após a introdução de três cortes, aumenta para 21.5184.

No Passo de Inicialização, o valor óptimo de $\min\{cx\colon x \in P^{\text{ASS}}\}$ é 17 e a solução óptima é x_0, ilustrada na Figura 4.4(b) através de $G[x_0]$. O corte global de capacidade mínima em $G[x_0]$ é caracterizado por $S = \{7, 8\}$, por exemplo. O corte mais profundo entre x_0 e P_S^{ASS} é o corte

$$x_{78} + x_{87} \leqslant 1,$$

obtido pelo algoritmo modificado de P. Wolfe. Termina o Passo de Inicialização com a correspondente inserção do corte na função penalidade.



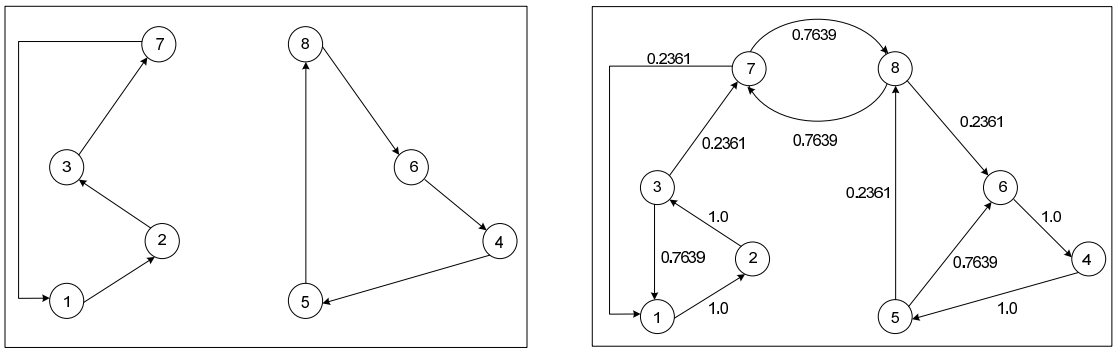
(a) $G[x^*]$ (b) $G[x_0]$

Figura 4.4: *Soluções óptimas inteiras sobre* P^{ATSP} *e* P^{ASS}*, respectivamente.*

Na iteração 0, o primeiro ciclo é interrompido com $\rho = 0.1$. A solução óptima do problema $\min\{\nabla f(x_0)q\colon q \in P^{\text{ASS}}\}$ é q_0, ilustrada na Figura 4.5(a). Não se obtém melhoria no limite inferior pois $f(x_0)+\nabla f(x_0)(q_0-x_0) = -102.6718$. Com a aplicação do método de Newton, obtém-se $\lambda_0 = 0.2361$ ao que corresponde $x_1 = x_0 + \lambda_0(q_0 - x_0)$, ilustrada na Figura 4.5(b). O corte global de capacidade mínima em $G[q_0]$ é caracterizado por $S = \{4, 5, 6, 8\}$, por exemplo. O corte mais profundo entre q_0 e P_S^{ASS} é o corte

$$x_{12} + x_{23} + x_{37} + x_{71} \leqslant 3,$$

que é acrescentado à função penalidade.



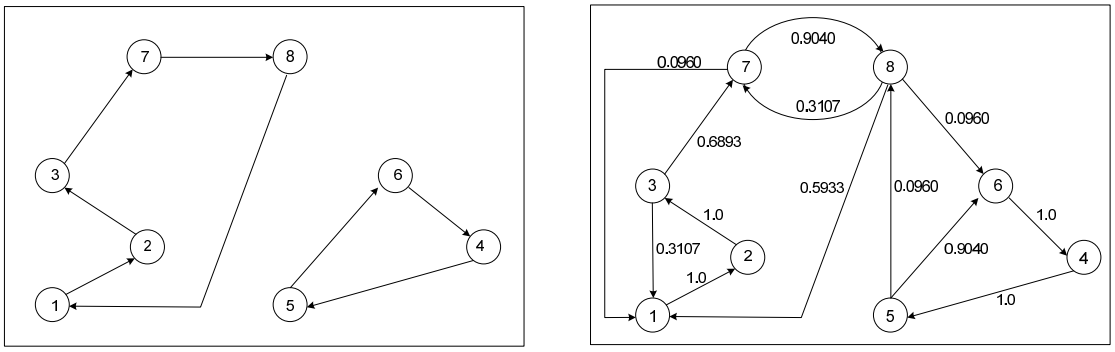
(a) $G[q_0]$ (b) $G[x_1]$

Figura 4.5: *Iteração 0.*

Na iteração 1, a solução óptima de $\min\{\nabla f(x_1)q\colon q \in P^{\text{ASS}}\}$ é q_1, ilustrada na Figura 4.6(a). Obtém-se uma melhoria no limite inferior pois $f(x_1)+\nabla f(x_1)(q_1-x_1) = 18.5673$. Com a aplicação do método de Newton, obtém-se $\lambda_1 = 0.5933$ ao que corresponde $x_2 = x_1 + \lambda_1(q_1 - x_1)$, ilustrada na Figura 4.6(b). O corte global de capacidade mínima em $G[q_1]$ é caracterizado por $S = \{1, 2, 3, 7, 8\}$, por exemplo. O corte mais profundo entre q_1 e P_S^{ASS} é o corte

$$x_{45} + x_{56} + x_{64} \leqslant 2,$$

que é acrescentado à função penalidade.

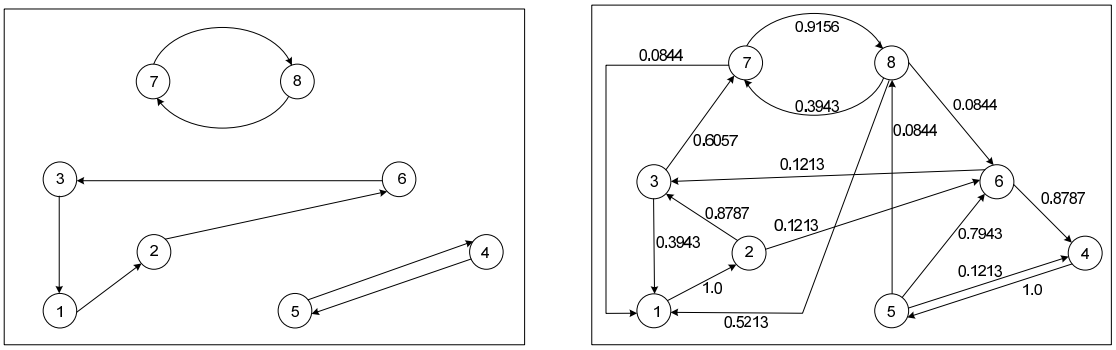


(a) $G[q_1]$ (b) $G[x_2]$

Figura 4.6: *Iteração 1.*

Na iteração 2, a solução óptima de $\min\{\nabla f(x_2)q\colon q \in P^{\text{ASS}}\}$ é q_2, ilustrada na Figura 4.7(a). Não se obtém uma melhoria no limite inferior pois $f(x_2)+\nabla f(x_2)(q_2-x_2) = 15.0568$. Com a aplicação do método de Newton, obtém-se $\lambda_2 = 0.1213$ ao

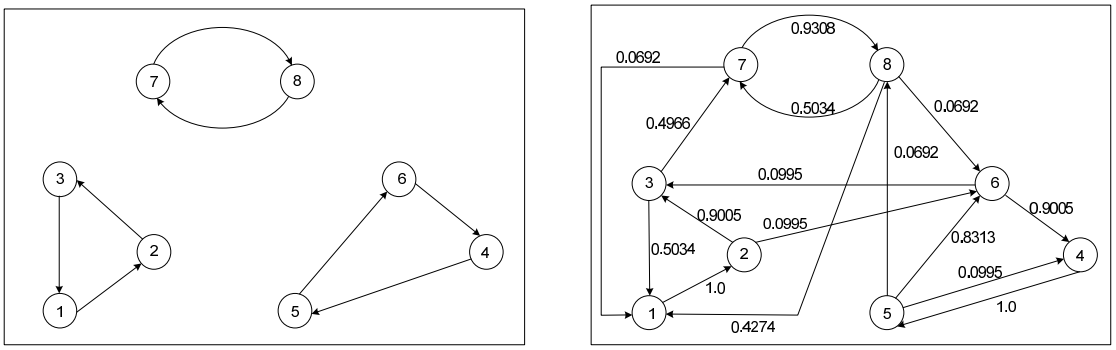
que corresponde $x_3 = x_2 + \lambda_2(q_2 - x_2)$, ilustrada na Figura 4.7(b). O corte global de capacidade mínima em $G[q_2]$ é caracterizado por $S = \{7, 8\}$, por exemplo. O corte mais profundo entre q_2 e P_S^{ASS} é o mesmo que foi obtido no Passo de Inicialização e, por isso, não é acrescentado à função penalidade.



(a) $G[q_2]$ (b) $G[x_3]$

Figura 4.7: *Iteração 2.*

Na iteração 3, a solução óptima de $\min\{\nabla f(x_3)q\colon q \in P^{\text{ASS}}\}$ é q_3, ilustrada na Figura 4.8(a). Obtém-se uma melhoria no limite inferior pois $f(x_3)+\nabla f(x_3)(q_3-x_3) = 20.7516$. Com a aplicação do método de Newton, obtém-se $\lambda_3 = 0.1801$ ao que corresponde $x_4 = x_3 + \lambda_3(q_3 - x_3)$, ilustrada na Figura 4.8(b). O corte global de capacidade mínima em $G[q_3]$ é caracterizado por $S = \{7, 8\}$. O corte mais profundo entre q_3 e P_S^{ASS} é o mesmo que foi obtido no Passo de Inicialização.



(a) $G[q_3]$ (b) $G[x_4]$

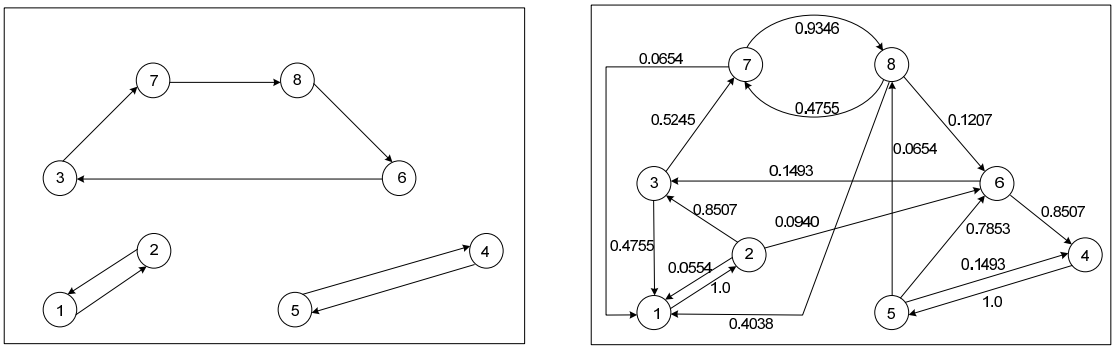
Figura 4.8: *Iteração 3.*

Na iteração 4, a solução óptima de $\min\{\nabla f(x_4)q\colon q \in P^{\text{ASS}}\}$ é q_4, ilustrada na Figura 4.9(a). Não se obtém-se uma melhoria no limite inferior pois $f(x_4)+\nabla f(x_4)(q_4-x_4) = 20.1955$. Com a aplicação do método de Newton, obtém-se $\lambda_4 = 0.0554$ ao que corresponde $x_5 = x_4 + \lambda_4(q_4 - x_4)$, ilustrada na Figura 4.9(b). O corte global

de capacidade mínima em $G[q_4]$ é caracterizado por $S = \{3, 6, 7, 8\}$. O corte mais profundo entre q_4 e P_S^{ASS} é o corte

$$x_{12} + x_{21} + x_{45} + x_{54} \leqslant 3,$$

que é acrescentado à função penalidade.



(a) $G[q_4]$ (b) $G[x_5]$

Figura 4.9: *Iteração 4.*

Na iteração 5, a solução óptima de $\min\{\nabla f(x_5)q\colon q \in P^{\text{ASS}}\}$ é q_5, que coincide com q_3. Obtém-se nova melhoria no limite inferior pois $f(x_5) + \nabla f(x_5)(q_5 - x_5) = 21.5184$. Com a aplicação do método de Newton, obtém-se $\lambda_5 = 0.0647$ ao que corresponde $x_6 = x_5 + \lambda_5(q_5 - x_5)$, ilustrada na Figura 4.10. O corte global de capacidade mínima em $G[q_5]$ é caracterizado por $S = \{7, 8\}$. Uma vez mais, o corte mais profundo entre q_2 e P_S^{ASS} é o mesmo que foi obtido no Passo de Inicialização.

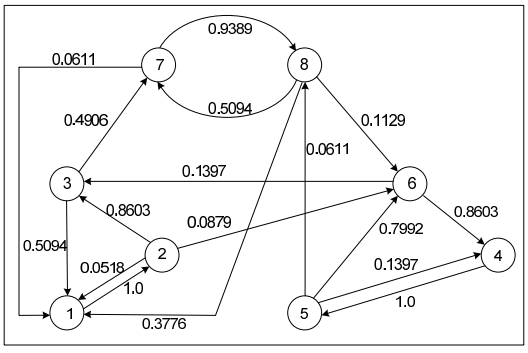


Figura 4.10: *Iteração 5:* $G[x_6]$.

4.3 Conclusões

O algoritmo proposto requer um estudo computacional mais aprofundado. O exemplo pequeno, que se acompanhou, serviu para testar uma primeira implementação feita em MATLAB com chamadas das rotinas `INDUS3` e `APPMIX` disponíveis na `Netlib` na biblioteca de algoritmos da `ACM`, em `http://www.netlib.org/toms/750`. Experiências computacionais preliminares permitiram identificar diversas limitações. Em primeiro lugar, observou-se que o algoritmo de primeira ordem pode tornar-se lento, progredindo com λ's demasiado pequenos.

Em segundo lugar e algo inesperadamente, observou-se que, em geral, o corte mais profundo entre q_k e P_S^{ASS} é uma desigualdade de ciclo - portanto, até menos profunda que a correspondente desigualdade de clique em S. Fica em aberto a questão de identificar para que normas $\|\cdot\|$ essa desigualdade será sempre o corte mais profundo em (4.26), quando $\overline{x}$ é o vector característico de um subcircuito.

Analisou-se também o comportamento do algoritmo se, no último passo da iteração genérica k do algoritmo da Figura 4.3, o vector q_k for substituído por x_k. Neste caso, foram obtidas algumas desigualdades que não são de circuito mas também ainda não são desigualdades de clique. Fica também em aberto a questão de saber que tipo de desigualdades se conseguirão gerar se x^k satisfizer todas as restrições de clique e for ponto extremo de uma relaxação linear de P^{ATSP}.

4.4 Corte global de capacidade mínima num grafo dirigido

Seja $G = (V, E)$ um grafo dirigido tal que a cada arco $(i, j) \in E$ está associado um escalar real $u_{ij} \geqslant 0$ designado por *capacidade* do arco (i, j). O problema de determinar o corte global de capacidade mínima em G consiste em resolver o seguinte problema de optimização:

$$\begin{array}{ll} \min & u(S) \equiv \displaystyle\sum_{(i,j)\in\delta^+(S)} u_{ij} \\ \text{s.a} & \emptyset \neq S \subset V, \end{array} \tag{4.28}$$

onde $\delta^+(S) \equiv \{(i, j) \in E\colon i \in S, j \notin S\}$ designa um *corte* de G.

Um corte de G é um conjunto minimal de arcos de G que se removidos fazem com que, para conjuntos de vértices complementares S e $V \setminus S$, deixe de existir um caminho orientado de qualquer vértice $i \in S$ para qualquer vértice $j \in V \setminus S$. O problema (4.28) consiste em identificar o corte de G cuja soma das capacidades dos respectivos arcos é mínima. Atendendo à não negatividade das capacidades, pode-se assumir, sem perda

de generalidade, que o corte procurado é um conjunto da forma $\delta^+(S)$ para algum $S \subset V$.

Hao e Orlin em [74] descrevem várias aplicações do problema (4.28). A aplicação que se tem em mente é a identificação de restrições de eliminação de sub-percursos para o problema do caixeiro viajante assimétrico, conforme sugerido por Padberg e Rinaldi em [105].

Se fossem englobadas as restrições $s \in S$ e $t \in V \setminus S$ para dois vértices s e t previamente escolhidos, então o problema (4.28) seria dual do (clássico) problema de determinar o fluxo máximo de s para t, conforme demonstrado por Ford e Fulkerson em [57]. Para resolver este último problema, existe o algoritmo das *cadeias incrementais* de Ford e Fulkerson [57] e o algoritmo *empurrar-prefluxo* de Golberg e Tarjan [66], entre outros (veja-se, por exemplo, o livro de Ahuja, Magnanti e Orlin [1]). O algoritmo mais rápido (em termos de complexidade computacional) é o algoritmo *empurrar-prefluxo*, que pode ser implementado para funcionar em $\mathcal{O}(nm \log(n^2/m))$ operações (aritméticas e comparações), sendo n o número de vértices e m o número de arcos do grafo G.

Assim, o problema (4.28) pode ser resolvido em $\mathcal{O}(n^3 m \log(n^2/m))$ operações inerentes à resolução de $n(n-1)/2$ problemas de fluxo máximo entre dois vértices porque o problema (4.28) é igual a

$$\min_{s,t \in V} \left(\begin{array}{lll} \min & u(S) & \\ \text{s.a} & s \in S, & t \notin S \end{array} \right).$$

Padberg e Rinaldi em [105] mostraram que também é possível resolver o problema (4.28) através da resolução de n adequados problemas de fluxo máximo de subconjuntos de V para o vértice t.

Hao e Orlin em [74] propuseram um algoritmo que resolve todos aqueles n problemas de fluxo máximo em $\mathcal{O}(nm \log(n^2/m))$ operações - isto é, um número de operações proporcional ao número de operações necessário para resolver apenas um problema de fluxo máximo entre dois vértices. Esse algoritmo, descrito na Secção 4.4.2, é, essencialmente, uma adaptação do algoritmo de *empurrar-prefluxo* de Golberg e Tarjan, que também é descrito na Secção 4.4.1. Para descrever este último, acompanhou-se o desenvolvimento de Hao e Orlin em [74], pois as descrições no artigo original ou no livro de Ahuja, Magnanti e Orlin [1, Capítulo 7] não permitem identificar, tão claramente, as modificações introduzidas.

4.4.1 Algoritmo de Golberg e Tarjan (corte $s-t$)

Seja $G=(V,E)$ um grafo dirigido com n vértices, m arestas e verificando a seguinte propriedade "$(i,j)\in E \Rightarrow (j,i)\in E$", sem perda de generalidade. Sejam $s,t\in V$ dois vértices designados por vértice *origem* e vértice *terminal*, respectivamente. A cada arco $(i,j)\in E$ está associado um escalar real $u_{ij}\geqslant 0$, eventualmente $+\infty$, designado por *capacidade* do arco (i,j). O problema do *fluxo máximo de s para t* é o seguinte problema de programação linear:

$$\begin{array}{ll} \max & v \\ \text{s.a} & \displaystyle\sum_{j\in\delta^-(i)} x_{ji} - \sum_{j\in\delta^+(i)} x_{ij} = \left\{ \begin{array}{ll} -v & \text{se } i=s \\ 0 & \text{se } i\neq s,t \\ v & \text{se } i=t \end{array} \right\}, \quad i\in V, \\ & 0\leqslant x\leqslant u. \end{array} \tag{4.29}$$

Um *fluxo em G de s para t* é uma solução admissível x para o problema linear (4.29) para algum valor de $v\geqslant 0$. Pode-se escrever o problema (4.29) sem a variável v como

$$\begin{array}{lll} \max & e_t(x) & \\ \text{s.a} & e_i(x)=0, & i\in V\setminus\{s,t\}, \\ & 0\leqslant x\leqslant u, & \end{array} \tag{4.30}$$

onde $e_i(x)$ denota o *excesso de fluxo* do vértice $i\in V$ definido por

$$e_i(x) \equiv \sum_{(j,i)\in\delta^-(i)} x_{ji} - \sum_{(i,j)\in\delta^+(i)} x_{ij} = \sum_{(j,i)\in\delta^-(i)} (x_{ji}-x_{ij}), \tag{4.31}$$

devido à condição imposta aos arcos de G. As restrições de conservação de fluxo $e_i(x)=0$, para $i\in V\setminus\{s,t\}$, impõem que, para qualquer fluxo x, "*o fluxo que entra em i é igual ao fluxo que sai de i*" seja verdade para todo o vértice $i\neq s,t$ - relação conhecida por *lei de Kirchoff* ou de *conservação de fluxo*. A restrição $e_s(x)=-e_t(x)$ não é considerada no problema (4.30) por ser redundante uma vez que, para todo $x\in\mathbb{R}^n$,

$$\sum_{i\in V} e_i(x) = 0,$$

o que, por sua vez, implica que, para todo x admissível para o problema (4.30),

$$e_s(x) = -e_t(x) + \sum_{i \in V \setminus \{s,t\}} e_i(x) = -e_t(x).$$

O dual do problema linear (4.30) é uma formulação do seguinte problema de optimização combinatória, conhecido por problema do *corte $s-t$ de capacidade mínima*,

$$\begin{array}{ll} \min & u(S) \equiv \displaystyle\sum_{(i,j) \in \delta^+(S)} u_{ij} \\ \text{s.a} & s \in S, \quad t \in V \setminus S, \end{array} \tag{4.32}$$

onde, neste caso, $\delta^+(S)$ define um corte especial de G designado por *corte $s-t$ de G*.

O conhecimento de um fluxo x^* óptimo para o problema (4.30) permite caracterizar uma solução óptima para o problema (4.32) por observação do grafo residual $G[x^*]$: o conjunto S^* constituído pelos vértices i tais que existe um caminho de s para i em $G[x^*]$ é uma solução óptima para o problema (4.32). No entanto, o conhecimento de uma solução óptima S^* para o problema (4.32) não é suficiente para caracterizar um fluxo x^* óptimo para o problema (4.30), conforme se ilustra no exemplo seguinte, referido por Hao e Orlin em [74]. Seja S^* o corte $s-t$ de um grafo dirigido $G = (V, E)$ de capacidade mínima $u(S^*)$. Considere-se o grafo $G' = (V', E')$, onde $V' = \{s'\} \cup V$, $E' = \{(s', s)\} \cup E$, e o novo arco (s', s) possui capacidade $u_{s's} = u(S^*)$. Então, $S = \{s'\}$ define um corte $s' - t$ em G' de capacidade mínima. No entanto, esse conhecimento não ajuda a caracterizar a forma como o fluxo será distribuído pelos arcos de G a partir de s.

Até esta data, todo o algoritmo conhecido, para a resolução do problema (4.32), que não use a sua formulação em termos de um problema linear é, essencialmente, um algoritmo que visa a resolução directa do problema (4.30).

Um *prefluxo em G de s para t* é um vector $x \in \mathbb{R}^m$ satisfazendo $0 \leqslant x \leqslant u$ e tal que $e_i(x) \geqslant 0$ para todo $i \in V \setminus \{s\}$. Portanto, um fluxo de s para t é, em particular, um prefluxo de s para t. O algoritmo *empurrar-prefluxo* de Goldberg e Tarjan [66] resolve directamente o problema de determinar o prefluxo máximo de s para t - substitua-se "$e_i(x) = 0$" por "$e_i(x) \geqslant 0$" no problema (4.30). Isto não constitui obstáculo se o objectivo for o da determinação do fluxo máximo de s para t, pois todo o prefluxo óptimo de s para t pode ser facilmente transformado num fluxo óptimo de s para t, sem alterar o valor do excesso de fluxo que alcança o vértice t, $e_t(x)$ - veja-se o Apêndice C.

Dado um prefluxo x, a capacidade *residual* $r_{ij}(x)$ ou, simplesmente, r_{ij} de um arco $(i, j) \in E$ define-se por $r_{ij} = u_{ij} - x_{ij} + x_{ji} \geqslant 0$ e representa o maior incremento do fluxo que pode ser enviado do vértice i para o vértice j, usando os arcos (i, j) e (j, i), e sem violar as restrições de capacidade nesses arcos. A capacidade residual r_{ij} tem duas

componentes: $u_{ij} - x_{ij}$, que representa o incremento máximo de fluxo ao longo do arco (i, j), e x_{ji}, que representa o decremento máximo do fluxo ao longo do arco (j, i). Um arco (i, j) diz-se *residual* (relativamente a um prefluxo x) se $r_{ij} > 0$. O *grafo residual* (relativamente a um prefluxo x) é constituído por arcos residuais e representa-se por $G[x] = (V, E[x])$.

Considere-se uma partição $V = W \cup D$. Diz-se que um conjunto de rótulos $d \in \mathbb{Z}_+^n$ é *W-válido* se $d(i) \leqslant d(j) + 1$ para cada $(i, j) \in E[x]$ tal que $i, j \in W$. Diz-se que D satisfaz a propriedade de *adormecimento* se $t \notin D$ e não existe $(i, j) \in E[x]$ tal que $i \in D$ e $j \in W$. Note-se que se D satisfaz esta propriedade, então não existe nenhum caminho no grafo residual de D para W. Um vértice $i \in W \setminus \{t\}$ diz-se *activo* se $e_i(x) > 0$ ou, simplesmente, $e(i) > 0$. Um arco $(i, j) \in E[x]$ diz-se *admissível* se $i, j \in W$ e $d(i) = d(j) + 1$. Designa-se W por conjunto de vértices *acordados* e D por conjunto de vértices *adormecidos*, por razões que se tornarão naturais mais adiante neste texto.[1]

As três subrotinas fundamentais no algoritmo de *empurrar-prefluxo* de Golberg e Tarjan são apresentadas nas Figuras 4.11, 4.12 e 4.13, nomeadamente *Inicializar*, *Empurrar (i, j)* e *Rotular (i)*[2].

Procedimento Inicializar

* Seja x um prefluxo qualquer, por exemplo, tudo-zeros *\

Para cada arco (s, j) tal que $j \in V \setminus \{s\}$, fazer
    $x_{sj} := u_{sj}$,
    $x_{js} := 0$.
$D := \{s\}$,
$W := V \setminus \{s\}$,
$d(t) := 0$.
Para cada vértice $j \in V \setminus \{t\}$, fazer
    $d(j) := 1$.

Figura 4.11: *Subrotina* Inicializar.

O objectivo do procedimento *Inicializar* é: (i) definir uma partição inicial $V = D \cup W$; (ii) definir um prefluxo x de s para t que faz com que D satisfaça a propriedade de adormecimento; e, (iii) atribuir valores iniciais aos rótulos d de modo a serem W-válidos.

O procedimento *Empurrar (i, j)* é invocado quando i é um vértice activo, e (i, j) é um

[1] W e D são as abreviaturas do inglês de *Waked* e *Dormant*, respectivamente.
[2] *Empurrar* e *Rotular* são as traduções do inglês de *Push* e *Relabel*, respectivamente.

Procedimento Empurrar (i, j)

$\delta := \min\{e(i), r_{ij}\}$,
$\delta_1 := \min\{\delta, u_{ij} - x_{ij}\}$,
$\delta_2 := \delta - \delta_1$,
$x_{ij} := x_{ij} + \delta_1$,
$x_{ji} := x_{ji} - \delta_2$.

Figura 4.12: *Subrotina* Empurrar (i, j).

Procedimento Rotular (i)

Se i é o único vértice em W com rótulo $d(i)$,
então
    $R := \{j \in W : d(j) \geqslant d(i)\}$,
    $D := D \cup R$,
    $W := W \setminus R$;
senão
    Se existir um arco $(i, j) \in G[x]$ para $j \in W$,
    então
        $d(i) := \min\{d(j) + 1 : (i, j) \in G[x]$ para $j \in W\}$;
    senão
        $D := D \cup \{i\}$,
        $W := W \setminus \{i\}$.

Figura 4.13: *Subrotina* Rotular (i).

arco admissível. Neste procedimento, as quantidades x_{ij} e x_{ji} são redefinidas de modo a *empurrar*, o mais possível, a quantidade de fluxo em excesso no vértice i ao longo dos arcos (i, j) e (j, i). Depois deste procedimento, o grafo residual pode ficar com menos um arco; diz-se então que o arco (i, j) fica *saturado*.

O procedimento *Rotular (i)* só é executado quando $i \in W$ é um vértice activo (por definição, $i \in W \setminus \{t\}$ e $e(i) > 0$), e não existe nenhum arco (i, j) admissível (por definição, um arco $(i, j) \in E[x]$ para $i, j \in W$ e $d(i) = d(j) + 1$). No final deste procedimento, uma das duas situações seguintes ocorreu: (i) i é transferido para D (eventualmente em conjunto com outros vértices de W) - neste caso, não há modificação do valor dos rótulos d; ou (ii) i permanece activo em W, e o valor do rótulo $d(i)$ foi modificado de modo a criar pelo menos um arco admissível (i, j). Em ambos os casos, mantém-se a propriedade de adormecimento de D e a W-validade dos rótulos d, conforme se formaliza no lema seguinte.

Lema 4.4 (Hao e Orlin [74]) *Seja i um vértice activo tal que não existe nenhum arco admissível (i,j). Se, à entrada do procedimento Rotular (i), d é W-válido, e D satisfaz a propriedade de adormecimento, então, à saída, d continua W-válido e D continua a satisfazer a propriedade de adormecimento. Mais, quaisquer modificações nos valores dos rótulos d correspondem a incrementos positivos.*

Demonstração: Se i é o único vértice em W com rótulo $d(i)$, então: (i) não existe nenhum arco $(i,j) \in E[x]$ tal que $j \in W \setminus R$, pois, da definição de R, ter-se-ia $d(i) \geqslant d(j)+1$ e, como $d(i) \leqslant d(j)+1$ para qualquer arco $(i,j) \in E[x]$ tal que $i,j \in W$, então ter-se-ia $d(i) = d(j)+1$, o que implicaria que (i,j) fosse um arco admissível - o que é absurdo; (ii) não existe nenhum arco $(k,j) \in E[x]$ com $k \in R \setminus \{i\}$ e $j \in W \setminus R$, pois atendendo a que $d(k) > d(i)$, então $d(i) \leqslant d(k) - 1 \leqslant (d(j)+1) - 1 \leqslant d(j)$, o que é absurdo porque $j \notin R$. Portanto, não existindo, por hipótese, nenhum arco $(k,j) \in G[x]$ tal que $k \in R$ e $j \in W \setminus R$, então $D := D \cup R$ satisfaz a propriedade de adormecimento relativamente a $W \setminus R$. Além disso, como d e $G[x]$ permanecem inalterados e $W := W \setminus R$ (portanto, W ficou mais pequeno), também se conclui que d permanece W-válido.

Se i não é o único vértice em W com rótulo $d(i)$, então consideram-se duas situações.

Primeira, se não existe nenhum arco $(i,j) \in E[x]$ para $j \in W$, então $D := D \cup \{i\}$ satisfaz a propriedade de adormecimento relativamente a $W \setminus R$. Além disso, como d e $G[x]$ permanecem inalterados, e $W := W \setminus \{i\}$ (portanto, W ficou mais pequeno), também se conclui que d permanece W-válido.

Segunda, se existe pelo menos um arco $(i,j) \in E[x]$ para $j \in W$, então, como d é W-válido, e não existe nenhum arco admissível, $d(i) < d(j)+1$ para todo o arco $(i,j) \in E[x]$ para $j \in W$. Portanto, a redefinição $d(i) := \min\{d(j)+1 \colon (i,j) \in E[x], j \in W\}$ aumenta estritamente o valor de $d(i)$. Após esta redefinição, permanece $d(i) \leqslant d(j)+1$ para todo $(i,j) \in E[x]$ para $j \in W$, e, por isso, d continua W-válido. Além disso, D continua a satisfazer a propriedade de adormecimento porque D não foi alterado.

Note-se que só há lugar à modificação dos rótulos d quando i não é o único vértice em W com rótulo $d(i)$, e existe pelo menos um arco $(i,j) \in E[x]$ para $j \in W$ - situação trabalhada no parágrafo anterior. Como foi visto, nesse caso o valor de $d(i)$, a componente de d alterada, aumenta estritamente. □

O algoritmo *empurrar-prefluxo* de Golberg e Tarjan, descrito na Figura 4.14, é constituído pelo procedimento de inicialização, seguido da execução repetida do procedimento de incremento de valor do prefluxo ao longo de um arco admissível ou do procedimento de redefinição dos rótulos.

Em cada passagem pelo ciclo do algoritmo, procura-se identificar um vértice activo

Algoritmo empurrar-prefluxo de Golberg e Tarjan

Inicializar;
Enquanto existir um vértice i activo em $G[x]$
    Se existir um arco admissível (i,j) em $G[x]$,
    então *Empurrar* (i,j);
    senão *Rotular* (i).

Figura 4.14: *Algoritmo* empurrar-prefluxo *de Golberg e Tarjan.*

i (i.e., $i \in W \setminus \{t\}$, e $e(i) > 0$) e um arco admissível (i,j) (i.e., $(i,j) \in E[x]$ para $j \in W$ tal que $d(i) = d(j) + 1$) ao longo do qual se procura *empurrar* o excesso de fluxo no vértice i através do procedimento *Empurrar* (i,j). Quando não existe um tal arco admissível, então o procedimento *Rotular* (i) permite excluir i de ser activo ou redefinir os rótulos d de modo a criar um arco admissível.

Lema 4.5 (Goldberg e Tarjan [66]) *Durante a execução do algoritmo empurrar-prefluxo de Golberg e Tarjan, x é um prefluxo de s para t.*

Demonstração: No final do procedimento *Inicializar*, x é um prefluxo de s para t. De facto, se x^0 é um qualquer prefluxo inicial de s para t (um vector de tudo-zeros, por exemplo), então, para todo $i \in V \setminus \{s\}$ tal que $(s,i) \in E[x^0]$, tem-se

$$\begin{aligned} e_i(x) &= \sum_{j \in \delta^-(i) \setminus \{s\}} (x^0_{ji} - x^0_{ij}) + (x_{si} - x_{is}) \\ &= e_i(x^0) - (x^0_{si} - x^0_{is}) + (x_{si} - x_{is}) \\ &= e_i(x^0) + (u_{si} - x^0_{si}) + x^0_{is} \\ &= e_i(x^0) + r_{si}(x^0) \geqslant e_i(x^0) \geqslant 0. \end{aligned}$$

Para todo $i \in V \setminus \{s\}$ tal que $(s,i) \notin E[x^0]$, $e_i(x) = e_i(x^0) \geqslant 0$ decorre trivialmente. Também é trivial que $0 \leqslant x \leqslant u$. Portanto, x é um prefluxo de s para t.

O procedimento *Empurrar* (i,j) só é executado quando $i \in W$ é um vértice activo, e $(i,j) \in E[x]$ é um arco admissível. A redefinição de x mantém x um prefluxo de s para t, pois, se, à entrada do procedimento, x^0 é um prefluxo de s para t, e x decorre da execução de *Empurrar* (i,j), então

$$\begin{aligned} e_i(x) &= e_i(x^0) - (x^0_{ji} - x^0_{ij}) + (x_{ji} - x_{ij}) \\ &= e_i(x^0) - (x^0_{ji} - x^0_{ij}) + [(x^0_{ji} - \delta_2) - (x^0_{ij} + \delta_1)] \\ &= e_i(x^0) - \delta \\ &= \left\{ \begin{array}{ll} 0, & \text{se } e_i(x^0) \leqslant r_{ij}(x^0) \\ e_i(x^0) - r_{ij}(x^0), & \text{se } e_i(x^0) > r_{ij}(x^0) \end{array} \right\} \geqslant 0. \end{aligned}$$

De forma análoga, se conclui que $e_j(x) = e_j(x^0) + \delta > e_j(x^0) \geqslant 0$ porque $\delta > 0$. Os excessos de fluxo dos restantes vértices mantêm-se inalterados. Para concluir que x é um prefluxo de s para t, resta mostrar que $0 \leqslant x \leqslant u$ ou, simplesmente, nas componentes x_{ij} e x_{ji}. Ora, trivialmente se verifica que $0 \leqslant x_{ij} \leqslant u_{ij}$. Por outro lado, $x_{ji} = x^0_{ji} - \delta_2$ com

$$\delta_2 = \delta - \delta_1 = \left\{ \begin{array}{ll} e(i) - e(i) = 0, & \text{se } e(i) \leqslant u_{ij} - x^0_{ij} \leqslant r_{ij}(x^0), \\ \in [0, r_{ij} - (u_{ij} - x^0_{ij})] = [0, x^0_{ji}], & \text{se } u_{ij} - x^0_{ij} \leqslant e(i) \leqslant r_{ij}(x^0), \\ r_{ij} - (u_{ij} - x^0_{ij}) = x^0_{ji}, & \text{se } u_{ij} - x^0_{ij} \leqslant r_{ij}(x^0) \leqslant e(i), \end{array} \right.$$

e, portanto, como $\delta_2 \in [0, x^0_{ji}]$, tem-se $0 \leqslant x_{ji} \leqslant u_{ji}$. □

Lema 4.6 (Hao e Orlin [74]) *Ao longo de todo o algoritmo empurrar-prefluxo de Golberg e Tarjan, verificam-se as seguintes propriedades:*

1. *os rótulos d mantêm a W-validade;*
2. *D satisfaz a propriedade de adormecimento.*

Demonstração: Começa-se por demonstrar que, no final do procedimento *Inicializar*, as propriedades 1. e 2. são válidas. A redefinição do prefluxo x *satura*[3] os arcos $(s, j) \in E$ para $j \in V \setminus \{s\}$, pelo que deixam de existir arcos $(s, j) \in G[x]$ para $j \in V \setminus \{s\}$. Pode-se então afirmar que a propriedade 2. se mantém válida dado que, no final, $D = \{s\}$. A propriedade 1. é válida porque $d(i) \leqslant 1 \leqslant d(j) + 1$ verifica-se trivialmente em qualquer arco $(i, j) \in E$.

Assuma-se agora que as propriedades 1. e 2. são válidas no início de um ciclo genérico do algoritmo *empurrar-prefluxo*. Seja $i \in W$ o vértice activo identificado à entrada desse ciclo.

Se existir um arco (i, j) admissível, então é invocado o procedimento *Empurrar* (i, j). Neste procedimento, as quantidades x_{ij} e x_{ji} são redefinidas de modo a que o grafo

[3] Um arco $(i, j) \in G[x]$ fica *saturado* quando r_{ij} passa de positivo a nulo; isto é, quando (i, j) deixa de ser um arco residual.

residual fica com menos um arco (se o arco (i,j) fica *saturado*) ou permanece o mesmo (se o arco (i,j) não fica *saturado*). Para além disso, nem W nem d são modificados pelo que, após essa operação, a propriedade 1. permanece válida. A propriedade 2. permanece válida porque não há introdução de arcos no grafo residual.

Se não existir qualquer arco (i,j) admissível, então é invocado o procedimento *Rotular (i)*. Pelo Lema 4.4, à saída do procedimento *Rotular (i)*, as propriedades 1. e 2. permanecem válidas.

Conclui-se então que, também no final do ciclo genérico (e por isso ao longo de todo o algoritmo *empurrar-prefluxo*), se verificam as propriedades 1. e 2. □

O Lema 4.7 mostra que, no final do algoritmo *empurrar-prefluxo* de Goldberg e Tarjan, fica identificado um corte $s-t$ de capacidade mínima.

Lema 4.7 (Goldberg e Tarjan [66]) *Durante o algoritmo empurrar-prefluxo de Goldberg e Tarjan, quando não existirem vértices activos em $G[x]$, $\delta^+(D)$ é um corte $s-t$ de capacidade mínima.*

Demonstração: Quando não existirem vértices activos em $G[x]$, então

$$e_t(x) = \sum_{j\in W} e_j(x) = \sum_{j\in W}\left(\sum_{i\in\delta^-(j)} x_{ij} - \sum_{i\in\delta^+(j)} x_{ji}\right) = \sum_{j\in W}\left(\sum_{i\in\delta^-(j)\cap D} x_{ij} - \sum_{i\in\delta^+(j)\cap D} x_{ji}\right),$$

o que decorre do cancelamento entre parcelas simétricas. Além disso, atendendo a que não há arcos em $G[x]$ direccionados de D para W, e, por isso, $r_{ij} = (u_{ij}-x_{ij})+x_{ji} = 0$ para $(i,j)\in\delta^+(D)$, tem-se

$$e_t(x) = \sum_{(i,j)\in\delta^+(D)} (x_{ij} - x_{ji}) = \sum_{(i,j)\in\delta^+(D)} u_{ij}.$$

Como x é um prefluxo de s para t, então verificam-se as condições do Corolário C.1 do Apêndice C, pelo que, $\delta^+(D)$ é um corte $s-t$ de capacidade mínima. □

Lema 4.8 (Goldberg e Tarjan [66]) *Ao longo de todo o algoritmo empurrar-prefluxo de Golberg e Tarjan, $d(i) \leqslant n-1$ para todo $i\in V$.*

Demonstração: Começa-se por mostrar que, ao longo de todo o algoritmo *empurrar-prefluxo* de Golberg e Tarjan, o conjunto $d(V) \equiv \{d(i)\colon i\in V\}$ é um conjunto de inteiros consecutivos (possivelmente com repetições).

No final da execução da subrotina *Inicializar*, $d(V)$ é um conjunto de inteiros consecutivos porque $d(t)=0$, e $d(j)=1$ para todo $j\in V\setminus\{t\}$.

Suponha-se que, no início de um ciclo genérico do algoritmo *empurrar-prefluxo* de Golberg e Tarjan, $d(V)$ é um conjunto de inteiros consecutivos. Seja $i \in W$ o vértice activo identificado. Se existe um arco residual (i,j) admissível, então a subrotina *Empurrar* (i,j) é invocada, o que não afecta o conjunto $d(V)$. Se não existe qualquer arco residual (i,j) admissível, então a subrotina *Rotular* (i) é invocada. Neste caso, distinguem-se três situações.

Primeiro, i é o único vértice em W com rótulo $d(i)$. A consecutividade de $d(V)$ decorre de d não ser alterado.

Segundo, i não é o único vértice em W com rótulo $d(i)$, e existe um arco residual (i,j) para $j \in W$. Então, existe um vértice $k \in W$ com $k \neq i$ tal que $d(k) = d(i)$. Assim, a redefinição $d(i) := d(j) + 1$, para algum $j \in W$, faz com que $d(V)$ permaneça um conjunto de inteiros consecutivos.

Terceiro, i não é o único vértice em W com rótulo $d(i)$, e não existe qualquer arco residual (i,j) para $j \in W$. Também aqui, a consecutividade de $d(V)$ decorre de d não ser alterado.

Conclui-se que, no final desse ciclo genérico e, portanto, ao longo de todo o algoritmo, $d(V)$ é um conjunto de inteiros consecutivos. Assim, como $d(t) = 0 \in d(V)$, pode-se afirmar que $d(i) \leqslant n-1$ para todo $i \in V$. □

Uma interessante consequência do Lema 4.8 é a de que, para cada $i \in V$, o valor da componente $d(i)$ é incrementado, no máximo, $n-1$ vezes e, por isso, o número máximo de modificações efectuadas em componentes de d, durante todo o algoritmo *empurrar-prefluxo* de Golberg e Tarjan, é inferior a n^2.

Seguidamente, mostra-se que o número máximo de execuções do procedimento *Empurrar* (i,j) que *saturam* componentes do prefluxo é inferior a nm.

Suponha-se que, em determinado instante do algoritmo, o procedimento *Empurrar* (i,j) é invocado com i um vértice activo e (i,j) um arco admissível, e a modificação da componente x_{ij} do prefluxo x provoca a *saturação* do arco (i,j). Denote-se por $d'_i \equiv d(i) = d(j) + 1$ o rótulo do vértice i nesse instante. Se o mesmo arco voltar a ser *saturado*, então é porque, entretanto, o procedimento *Empurrar* (i,j) foi executado usando o arco (j,i) como argumento. Mas, então ter-se-ia $d'_j \equiv d(j) = d(i) + 1 \geqslant d'_i + 1$ e, no instante da nova *saturação* do arco (i,j), ter-se-ia $d(i) = d(j) + 1 \geqslant d'_j + 1 \geqslant d'_i + 2$. Como $d(i) \leqslant n-1$, então não pode haver mais do que n *saturações* do arco genérico (i,j). O resultado desejado decorre então de haver m arcos.

Seguidamente, mostra-se que o número máximo de execuções do procedimento *Empurrar* (i,j) que *não saturam* componentes do prefluxo é $\mathcal{O}(n^2 m)$.

Considere-se um escalar ϕ definido no início de uma passagem genérica do ciclo do algoritmo *empurrar-prefluxo* através de

$$\phi := \sum_{i \in I} d(i),$$

onde $I := \{i \in W \setminus \{t\} : e_i(x) > 0\}$ denota o conjunto dos vértices activos. Na primeira passagem, o valor de ϕ decorre do procedimento *Inicializar* e é inferior a n. A seguinte definição do escalar ϕ é útil na análise que se segue. Defina-se $\phi := \phi^+ - \phi^-$ com ϕ^+ um acumulador de incrementos ao valor de ϕ (inicializado ao valor do ϕ inicial) e ϕ^- um acumulador de decrementos ao valor de ϕ (inicializado a zero). Analise-se o que pode acontecer a estes valores durante uma passagem genérica pelo ciclo do algoritmo. Seja $i \in W \setminus \{t\}$ o vértice activo identificado. Considerem-se duas situações:

1. Se existe um arco admissível (i, j), então é invocado o procedimento *Empurrar (i, j)*. Neste procedimento há que considerar duas situações distintas:

   (a) Se $e(i) = \min\{e(i), r_{ij}\}$, então, após a modificação da componente x_{ij} do prefluxo x, tem-se $e(i) = 0$. Se o vértice j também era activo, então ϕ é decrementado $d(i)$ unidades, caso contrário, ϕ é decrementado $d(i) - d(j) = 1$ unidades. Em ambos os casos, o valor de ϕ é reduzido pelo menos uma unidade.

   (b) Se $e(i) > \min\{e(i), r_{ij}\}$, então, após a modificação da componente x_{ij} do prefluxo x, o arco (i, j) fica saturado e o valor de ϕ pode permanecer o mesmo (se j também era vértice activo) ou aumentar $d(j)$ unidades (se j não era vértice activo). Portanto, ϕ^+ aumenta, no máximo, n unidades.

2. Se não existe qualquer arco admissível (i, j), então é invocado o procedimento *Rotular (i)*. Neste procedimento há que considerar três situações distintas:

   (a) Se i é o único vértice em W com rótulo $d(i)$, então o valor de ϕ é decrementado em pelo menos $d(i) \geqslant 1$ unidades. Portanto, o valor de ϕ é reduzido pelo menos uma unidade.

   (b) Se i não é o único vértice em W com rótulo $d(i)$ e existe um arco residual $(i, j)j \in$ para W, então a consequente redefinição de $d(i)$ implica um incremento do valor de ϕ. Portanto, o valor de ϕ é aumentado e ϕ^+ aumenta, no máximo, n unidades.

   (c) Se i não é o único vértice em W com rótulo $d(i)$ e não existe qualquer arco residual (i, j) para $j \in W$, então o valor de ϕ é decrementado em pelo menos $d(i) \geqslant 1$ unidades. Portanto, o valor de ϕ é reduzido pelo menos uma unidade.

Conforme observado num parágrafo anterior, os passos correspondentes às situações 1b e 2b ocorrem menos de nm e n^2 vezes, respectivamente, durante todo o algoritmo. O incremento no valor de ϕ^+ devido à passagem pelo passo 1b é, no máximo, $n(nm)$, enquanto que o incremento no valor de ϕ^+ devido à passagem pelo passo 2b é, no máximo, n^2. Por isso, $\phi^+ < n + n^2m + n^2$ sempre.

Por outro lado, no final do algoritmo, observa-se que $\phi = 0$ dada a inexistência de vértices activos. Por isso, os passos nos quais ϕ é decrementado pelo menos uma unidade, isto é, os passos 1a, 2a e 2c ocorrem, no máximo, $n + n^2 + n^2m = \mathcal{O}(n^2m)$ vezes.

Como foi observado num parágrafo anterior, os passos correspondentes às situações 1a e 2b ocorrem, no máximo, $nm + n^2$ vezes. Assim, juntando todo o recente desenvolvimento, conclui-se que as rotinas *Rotular (i)* e *Empurrar (i, j)* são chamadas, no máximo, $\mathcal{O}(n^2m)$. Se cada um daqueles passos for executado em $\mathcal{O}(1)$ operações (aritméticas e comparações), então obtém-se o seguinte resultado sobre a complexidade do algoritmo *empurrar-prefluxo* de Goldberg e Tarjan.

Teorema 4.1 (Goldberg e Tarjan [66]) *Seja $G = (V, E)$ um grafo dirigido com n vértices, m arcos e capacidades u_{ij}, para $(i, j) \in E$, definidas por números racionais não negativos. Sejam s e t dois vértices de G. O algoritmo empurrar-prefluxo de Golberg e Tarjan determina o fluxo máximo de s para t em $\mathcal{O}(n^2m)$ operações.*

A regra de selecção de vértices activos pode ser implementada de forma a reduzir essa ordem de complexidade. Usando árvores dinâmicas, Golberg e Tarjan obtiveram o seguinte resultado.

Teorema 4.2 (Goldberg e Tarjan [66]) *Nas mesmas condições do Teorema 4.1, o algoritmo empurrar-prefluxo de Golberg e Tarjan determina o fluxo máximo de s para t em $\mathcal{O}(nm \log(n^2/m))$ operações.*

Na secção seguinte, analisa-se o algoritmo proposto por Hao e Orlin em [74] para determinar o corte global de capacidade mínima, que é motivado pelo algoritmo que se descreveu.

4.4.2 Algoritmo de Hao e Orlin (corte global)

Nesta secção, apresenta-se o algoritmo de Hao e Orlin [74] que determina o corte global de capacidade mínima num grafo dirigido. Este algoritmo baseia-se no algoritmo *DetCorteMin(s)* de Padberg e Rinaldi [105], que se recorda abaixo, e numa versão

adaptada do algoritmo de *empurrar-prefluxo* de Goldberg e Tarjan, descrito na secção anterior.

Padberg e Rinaldi efectuaram a seguinte observação. Se $s \in V$ é um vértice qualquer e S^* é a solução óptima do problema (4.28), então, como $s \in S^*$ ou $s \in V \setminus S^*$, tem-se

$$u(S^*, V \setminus S^*) = \min \left(\begin{array}{ll} \min & u(T, V \setminus T) \\ \text{s.a} & s \in T \end{array} , \begin{array}{ll} \min & u(T, V \setminus T) \\ \text{s.a} & s \in V \setminus T \end{array} \right). \tag{4.33}$$

Por este facto, tem interesse a resolução da seguinte variante do problema de corte mínimo, onde o vértice $s \in V$ está fixo

$$\begin{array}{ll} \min & u(T, V \setminus T) \\ \text{s.a} & s \in T. \end{array} \tag{4.34}$$

Note-se que se se dispuser de um algoritmo para resolver o problema (4.34), então, após atribuir a orientação contrária a cada um dos arcos de G, o mesmo algoritmo serve para resolver o segundo problema de optimização em (4.33) - portanto, sem aumento da ordem de complexidade. Com base na observação (4.33), Padberg e Rinaldi propuseram o algoritmo *DetCorteMin(s)*, descrito na Figura 4.15, que resolve o problema (4.34), resolvendo uma sequência de n problemas do tipo seguinte, onde S e t estão fixos,

$$\begin{array}{ll} \min & u(T) \\ \text{s.a} & S \subseteq T, \\ & t \notin T, \end{array} \tag{4.35}$$

cuja solução óptima será designada por *corte* $S-t$ de G. Vejam-se mais detalhes sobre esta classe de problemas no Apêndice C.

Algoritmo DetCorteMin(s) de Padberg e Rinaldi

$S := \{s\}$,
MelhorValor $:= \infty$.
Enquanto $S \neq V$,
    seleccionar $t' \in V \setminus S$.
    Seja $\delta^+(S^*)$ o corte $S - t'$ mínimo e $z := u(S^*)$.
    Se $z <$ MelhorValor,
    então MelhorValor $:= z$ e MelhorCorte $:= \delta^+(S^*)$.
    $S := S \cup \{t'\}$.

Figura 4.15: *Algoritmo* DetCorteMin(s) *de Padberg e Rinaldi.*

Lema 4.9 (Hao e Orlin [74]) *O algoritmo DetCorteMin(s) de Padberg e Rinaldi re-*

solve o problema (4.34).

Demonstração: Suponha-se, sem perda de generalidade, que os vértices entram em S pela ordem da sua selecção como terminais o que, em particular, implica $s = 1$. Seja S' a solução óptima do problema (4.34) e $z' := u(S', V \setminus S')$ o respectivo valor óptimo. Seja $z := u(S^*, V \setminus S^*)$ o valor encontrado no fim do algoritmo *DetCorteMin(s)*. Por definição, $z \geqslant z'$. Por outro lado, se t' é o menor dos índices de vértices em $V \setminus S'$, então, na iteração t', $S := \{1, 2, \ldots, t' - 1\} \subseteq S'$, logo

$$z = u(S^*, V \setminus S^*) \leqslant \begin{array}{ll} \min & u(T, V \setminus T) \\ \text{s.a} & S \subseteq T,\ t' \in V \setminus T \end{array} \leqslant u(S', V \setminus S') = z'.$$

□

Hao e Orlin em [74] propuseram uma versão do algoritmo *DetCorteMin(s)* de Padberg e Rinaldi, que utiliza as subrotinas *InicializarMod* e *RotularMod (i)* descritas na Figura 4.16, e a subrotina *Empurrar (i, j)* descrita na Figura 4.12. O objectivo destas subrotinas é semelhante ao das subrotinas descritas nas Figuras 4.11 e 4.13.

Ao longo do algoritmo *DetCorteMin(s) de Hao e Orlin*, descrito abaixo, o conjunto D será sempre definido implicitamente através de

$$D = \bigcup_{k=0}^{D_{\max}} \mathit{Adormec}\ (k). \tag{4.36}$$

Cada um destes conjuntos *Adormec* (k) corresponde a uma transferência de vértices de W para D quando da execução da subrotina *RotularMod (i)*.

Seja $V = W \cup D$ uma partição. Para além das definições de W-validade dos rótulos d e adormecimento de D, introduzidas na secção anterior, diz-se que o par $(D, D_{\max})$ satisfaz a propriedade de *adormecimento global* se possuir as duas propriedades seguintes:

1. O conjunto D é definido por (4.36) e satisfaz a propriedade de adormecimento (isto é, não existe nenhum arco residual de um vértice de D para vértices de W).
2. Não existem arcos residuais de *Adormec* (l) para *Adormec* (k) para $0 \leqslant l < k \leqslant D_{\max}$.

No ínicio da primeira passagem pela subrotina *InicializarMod*, $t' = s$ e $W = V$. No final desta primeira passagem: (i) definiu-se uma partição $V = D \cup W$ com $D :=$ *Adormec* $(0) := \{s\}$ (portanto, $D_{\max} := 0$) e $W := W \setminus \{s\}$; (ii) escolheu-se um vértice terminal $t' \in W$; (iii) definiu-se um prefluxo x de $S :=$ *Adormec* (0) para t', que

Procedimento InicializarMod

$W := W \setminus \{t'\}$,
$Adormec\ (0) := Adormec\ (0) \cup \{t'\}$.
Para cada arco (t', j) para $j \in V \setminus Adormec\ (0)$, fazer
    $x_{t'j} := u_{t'j}$,
    $x_{jt'} := 0$.
Se $t' = s$,
então
    escolha-se $t' \in W$ qualquer,
    $d(t') := 0$.
    Para cada vértice $j \in V \setminus \{t'\}$, fazer
        $d(j) := 1$;
senão
    Se $W = \emptyset$,
    então
        $W := Adormec\ (D_{\max})$,
        $D_{\max} := D_{\max} - 1$.
    Escolha-se $t' \in W$ tal que $d(t') := \min\{d(j) \colon j \in W\}$.

Procedimento RotularMod (i)

Se i for o único vértice em W com rótulo $d(i)$,
então
    $R := \{j \in W \colon d(j) \geqslant d(i)\}$,
    $D_{\max} := D_{\max} + 1$,
    $Adormec\ (D_{\max}) := R$,
    $W := W \setminus R$;
senão
    Se existir um arco $(i, j) \in G[x]$ para $j \in W$,
    então
        $d(i) := \min\{d(j) + 1 \colon (i, j) \in G[x]$ para $j \in W\}$;
    senão
        $D_{\max} := D_{\max} + 1$,
        $Adormec\ (D_{\max}) := \{i\}$,
        $W := W \setminus \{i\}$.

Figura 4.16: *Subrotinas* InicializarMod *e* RotularMod (i).

faz com que o par $(D, D_{\max})$ satisfaça a propriedade de adormecimento global; e (iv) definiram-se os rótulos d de modo a serem W-válidos.

No início da segunda e seguintes passagens pela subrotina *InicializarMod*, conhece-se uma partição $V = D \cup W$ para a qual o par $(D, D_{\max})$ satisfaz a propriedade de

adormecimento, $t' \in W$, o conjunto de rótulos d é W-válido, e x é um prefluxo de *Adormec* (0) para t'. No final deste procedimento: (i) transferiu-se t' de W para *Adormec* (0); (ii) se essa transferência fez com que W ficasse vazio, então procedeu-se à transferência do conjunto *Adormec* ($D_{\max}$) de D para W (e consequente redefinição de $D_{\max}$); (iii) seleccionou-se um novo $t' := \arg\min\{d(j) : j \in W\}$; e (iv) redefiniu-se o prefluxo x de modo a tornar-se um prefluxo de $S :=$ *Adormec* (0) para t' e de modo a que o par $(D, D_{\max})$ satisfaça a propriedade de adormecimento global.

O procedimento *RotularMod (i)* só é executado quando $i \in W$ é um vértice activo e não existe nenhum arco admissível (i, j). No final deste procedimento, uma das duas situações seguintes ocorre: (i) i é transferido para D, eventualmente em conjunto com outros vértices de W, formando um novo conjunto *Adormec* ($D_{\max}$) (com o consequente incremento de $D_{\max}$) sem modificar os rótulos d; ou (ii) i permanece activo em W, e o valor do rótulo $d(i)$ é incrementado de modo a criar pelo menos um arco admissível (i, j).

O algoritmo *DetCorteMin(s)* de Hao e Orlin, descrito na Figura 4.17, é constituído pela execução repetida de um ciclo externo que inclui o procedimento de inicialização descrito antes, e um ciclo interno que é, essencialmente, o algoritmo *empurrar-prefluxo* de Golberg e Tarjan.

Algoritmo DetCorteMin(s) de Hao e Orlin

Adormec (0) := {},
$D_{\max} := 0$,
$W := V$,
$t' := s$,
$MelhorValor := \infty$.
* Seja x um qualquer prefluxo de s (para qualquer outro vértice), $x = 0$, por exemplo. *\
Enquanto *Adormec* (0) $\cup \{t'\} \neq V$
    InicializarMod.
    \ * $S :=$ *Adormec* (0) * \
    Enquanto existir um vértice activo i em $G[x]$
        Se existir arco admissível (i, j) em $G[x]$,
        então *Empurrar (i, j)*;
        senão *RotularMod (i)*.
    \ * $\delta^+(D)$ é o corte $S - t'$ mínimo e $z := u(D)$ o seu valor *\
    Se $z < MelhorValor$,
    então $MelhorValor := z$.

Figura 4.17: *Algoritmo* DetCorteMin(s) *de Hao e Orlin.*

Em cada passagem pelo ciclo interno, procura-se identificar um vértice activo i (i.e., $i \in W \setminus \{t'\}$ e $e(i) > 0$) e um arco admissível (i.e., $(i,j) \in E[x]$ para $j \in W$ tal que $d(i) = d(j) + 1$) ao longo do qual se procura *empurrar* o excesso de fluxo no vértice i. Quando não existe um tal arco admissível, então o procedimento de rotulação modificado permite excluir i de ser activo ou redefinir o rótulo $d(i)$ de modo a criar um arco admissível (i,j).

Lema 4.10 (Hao e Orlin [74]) *Durante toda a execução do ciclo interno algoritmo DetCorteMin(s) de Hao e Orlin, x é um prefluxo de $S(= Adormec\ (0))$ para t'.*

Demonstração: Recorde-se que um prefluxo de S para t' é um vector $x \in \mathbb{R}^m$ tal que $0 \leqslant x \leqslant u$ e $e_i(x) \geqslant 0$ para todo $i \in V \setminus S$ - ver o Apêndice C.

A modificação do prefluxo x é observada no procedimento *InicializarMod*, anterior ao ciclo interno, e no procedimento *Empurrar* (i,j), durante o ciclo interno.

Suponha-se que, à entrada do procedimento *Empurrar* (i,j) com $i \in W$ um vértice activo e $(i,j) \in G[x]$ um arco admissível, x é um prefluxo de S para t'. Uma ligeira adaptação da demonstração do Lema 4.5 mostra que, no final do procedimento *Empurrar* (i,j), x permanece um prefluxo de S para t'.

Suponha-se que, no início do procedimento *InicializarMod*, x é um prefluxo de *Adormec* (0) para t'. Uma ligeira adaptação da demonstração do Lema 4.5 mostra que a redefinição de x (saturando todos os arcos (t',i) para $i \in V \setminus Adormec\ (0)$) faz com que x seja prefluxo de $S \equiv Adormec\ (0) := Adormec\ (0) \cup \{t'\}$ para qualquer outro vértice. Consequentemente, x é um prefluxo de S para t' ao longo de todo o algoritmo. □

Lema 4.11 (Hao e Orlin [74]) *Ao longo de todo o algoritmo DetCorteMin(s) de Hao e Orlin, verificam-se as seguintes propriedades:*

1. *os rótulos d mantêm a W-validade;*
2. *o par $(D, D_{\max})$ satisfaz a propriedade de adormecimento global.*

Demonstração: No final da primeira passagem pela subrotina *InicializarMod*, $D = Adormec\ (0) = \{s\}$, $D_{\max} = 0$ e $W = V \setminus \{s\}$. A modificação do valor do prefluxo *satura* as componentes $(s,j) \in E$ para $j \in W$, pelo que não existe $(s,j) \in G[x]$ tal que $j \in W$. Pode-se então afirmar que a propriedade 2. é válida. A propriedade 1. é válida porque $d(i) \leqslant 1 \leqslant d(j) + 1$ verifica-se trivialmente em qualquer arco $(i,j) \in E[x]$ para $j \in W$.

Suponha-se agora que as propriedades 1. e 2. são verdadeiras no início de uma passagem genérica do ciclo interno. Seja $i \in W$ o vértice activo identificado.

Se existe um arco admissível (i,j), então é invocado o procedimento *Empurrar* (i,j). A operação de modificação do prefluxo x não envolve arcos que toquem D pelo que, após essa operação, a propriedade 2. mantém-se trivialmente válida. A operação de modificação do prefluxo x faz apenas com que as seguintes modificações ocorram no grafo residual: o arco (i,j) deixa de ser residual (quando esse arco fica *saturado*) ou mantém-se residual. Em qualquer dos casos, a propriedade 1. permanece válida. (Note-se que também pode acontecer que (j,i) não fosse residual e o passe a ser, mas, neste caso, $d(j) = d(i) - 1 \leqslant d(i) + 1$ verifica-se trivialmente).

Se não existe qualquer arco admissível (i,j), então é invocado o procedimento *RotularMod (i)*. Distinguem-se três situações.

Primeiro, i é o único vértice em W com rótulo $d(i)$. Como não existem arcos $(i,j) \in E[x]$ para $i \in D$ e $j \in R$, a inserção de R em D (com as consequentes redefinições $D_{\max} := D_{\max} + 1$ e $Adormec\ (D_{\max}) := R$) mantém a validade da não existência de arcos residuais de $Adormec\ (l)$ para $Adormec\ (k)$ para $0 \leqslant l < k \leqslant D_{\max} - 1$. Por outro lado, como não existem arcos $(i,j) \in E[x]$ para $i \in R, j \in W \setminus R$ (provado no Lema 4.4) e não existem arcos residuais de $\cup_{k=0}^{D_{\max}-1} Adormec\ (k)$ para W, então $D = \cup_{k=0}^{D_{\max}} Adormec\ (k)$ satisfaz a propriedade de adormecimento. Os rótulos d mantêm a W-validade porque não foram alterados e W ficou com menos arcos.

Segundo, i não é o único vértice em W com rótulo $d(i)$ e existe um arco residual (i,j) para $j \in W$. Então, a consequente redefinição do valor de $d(i)$ não interfere na validade da propriedade 2. Mais, a redefinição de $d(i)$ é feita de forma a que d mantenha a W-validade.

Terceiro, i não é o único vértice em W com rótulo $d(i)$ e não existe qualquer arco residual (i,j) para $j \in W$. Como não existem arcos $(k,i) \in E[x]$ para $k \in D$, a inserção de $\{i\}$ em D (com as consequentes redefinições $D_{\max} := D_{\max}+1$ e $Adormec\ (D_{\max}) := \{i\}$) mantém a validade da não existência de arcos residuais de $Adormec\ (l)$ para $Adormec\ (k)$ para $0 \leqslant l < k \leqslant D_{\max} - 1$. Por outro lado, como não existem arcos $(i,j) \in E[x]$ para $j \in W \setminus R$ e não existem arcos residuais de $\cup_{k=0}^{D_{\max}-1} Adormec\ (k)$ para W, então $D\ \cup_{k=0}^{D_{\max}} Adormec\ (k)$ satisfaz a propriedade de adormecimento. Os rótulos d mantêm a W-validade porque não foram alterados e W ficou com menos arcos.

Conclui-se que, no final desta passagem genérica do ciclo interno e, consequentemente, ao longo de todo o ciclo interno, as propriedades 1. e 2. mantêm-se verdadeiras.

Suponha-se que, no início da subrotina *InicializarMod* (segunda e seguintes passagens), $V = D \cup W$ é uma partição tal que as propriedades 1. e 2. se verificam.

Veja-se que a propriedade 2. continua a verificar-se no final da subrotina *Inicializar-Mod*. A modificação do valor do prefluxo x não envolve as componentes $(i,j) \in E$ para $i,j \in D$ pelo que, no final, também não existem arcos residuais de *Adormec* (l) para *Adormec* (k) para $1 \leqslant l < k \leqslant D_{\max}$. A modificação do valor do prefluxo x *satura* as componentes $(t',j) \in E$ para $j \in V \setminus$ *Adormec* (0), pelo que, no final, também não existem arcos residuais de *Adormec* (0) para *Adormec* (k) para $0 < k \leqslant D_{\max}$ (recorde-se que a inclusão de t' em D decorre através da inclusão de t' em *Adormec* (0)). Para demonstrar que, no final da subrotina *InicializarMod*, D satisfaz a propriedade de adormecimento, é preciso distinguir duas situações.

Primeiro, a transferência de t' para D não faz com que W fique vazio. Como as componentes $(t',j) \in E$, para $j \in V \setminus$ *Adormec* (0), do prefluxo x ficaram *saturadas*, não existe $(t',j) \in G[x]$ para $j \in W$, de onde se conclui que, no final, D satisfaz a propriedade de adormecimento.

Segundo, a transferência de t' para D faz com que W fique vazio. Com as redefinições $W :=$ *Adormec* $(D_{\max})$, $D := D \setminus$ *Adormec* $(D_{\max})$ e $D_{\max} := D_{\max} - 1$, o adormecimento de $D = \cup_{k=0}^{D_{\max}}$ *Adormec* (k) é consequência de não existirem arcos residuais de $\cup_{k=0}^{D_{\max}-1}$ *Adormec* (k) para *Adormec* $(D_{\max})$.

Sumariando, no final dessa passagem por *InicializarMod*, as propriedades 1. e 2. mantêm-se válidas. Conclui-se então que, durante todo o algoritmo, as propriedades 1. e 2. mantêm-se válidas. □

Analogamente ao Lema 4.7, o Lema 4.12 mostra que, no final do ciclo interno, fica identificado um corte $S - t'$ de capacidade mínima.

Lema 4.12 (Hao e Orlin [74]) *Durante o ciclo interno do algoritmo DetCorteMin(s) de Hao e Orlin, quando não existirem vértices activos em $G[x]$, $\delta^+(D)$ é um corte $S - t'$ de custo mínimo.*

Demonstração: Quando não existirem vértices activos em $G[x]$, então

$$e_t(x) = \sum_{j \in W} e_j(x) = \sum_{j \in W} \left(\sum_{i \in \delta^-(j)} x_{ij} - \sum_{i \in \delta^+(j)} x_{ji} \right) = \sum_{j \in W} \left(\sum_{i \in \delta^-(j) \cap D} x_{ij} - \sum_{i \in \delta^+(j) \cap D} x_{ji} \right),$$

o que decorre do cancelamento de parcelas simétricas. Além disso, atendendo a que não há arcos em $G[x]$ direccionados de D para W, tem-se $r_{ij} = (u_{ij} - x_{ij}) + x_{ji} = 0$ para $(i,j) \in \delta^+(D)$ e, por isso,

$$e_t(x) = \sum_{(i,j) \in \delta^+(D)} (x_{ij} - x_{ji}) = \sum_{(i,j) \in \delta^+(D)} u_{ij}.$$

Como x é um prefluxo de S para t', então verificam-se as condições do Corolário C.2 do Apêndice C, pelo que, $\delta^+(D)$ é um corte $S - t'$ de capacidade mínima. □

Lema 4.13 (Hao e Orlin [74]) *Durante todas as execuções do ciclo interno do algoritmo DetCorteMin(s) de Hao e Orlin, os seguintes conjuntos são conjuntos de inteiros consecutivos, possivelmente com repetições:*

$$\begin{aligned} d(W) &\equiv \{d(i)\colon i \in W\}, \\ d(Adormec\ (k)) &\equiv \{d(i)\colon i \in Adormec\ (k)\}, \quad k = 1, \ldots, D_{\max}. \end{aligned}$$

Demonstração: No final da primeira passagem pela rotina *InicializarMod*, definiram-se os conjuntos $D_{\max} = 0$ e $W = V \setminus \{s\}$, e escolheu-se algum $t' \in W$. Definiram-se os rótulos $d(t') := 0$ e $d(j) := 1$, para todo $j \in W \setminus \{t'\}$, e consequentemente $d(W)$ é um conjunto de inteiros consecutivos. Para além disso, $d(t') = d_{\min}(W) \equiv \min\{d(i)\colon i \in W\}$.

Suponha-se que, no início de uma passagem genérica pelo ciclo interno, $d(W)$ e $d(Adormec(k))$, para $k = 1, \ldots, D_{\max}$, são conjuntos de inteiros consecutivos e $d(t') = d_{\min}(W)$. Seguidamente, prova-se que, no final dessa passagem, tudo isto se mantém.

Seja $i \in W$ o vértice activo identificado. Se existe um arco admissível $(i, j) \in G[x]$, então é invocada a subrotina *Empurrar (i, j)*, que não tem influência naquelas propriedades. Se não existe qualquer arco admissível (i, j), então é invocada a subrotina *RotularMod (i)*. Neste último caso, distinguem-se três situações.

Primeiro, i é o único vértice em W com rótulo $d(i)$. Como $d(W)$ é consecutivo, então, para $R := \{j \in W\colon d(j) \geqslant d(i)\}$, tem-se, após o incremento de $D_{\max}$, o conjunto $Adormec\ (D_{\max}) := R$ tal que $d(Adormec\ (D_{\max}))$ é um conjunto de inteiros consecutivos, bem como, após a redefinição $W := W \setminus R$, o conjunto $d(W)$.

Segundo, i não é o único vértice em W com rótulo $d(i)$ e existe pelo menos um arco residual (i, j) para $j \in W$. Como existe um vértice $k \in W$ com $k \neq i$ tal que $d(k) = d(i)$, então $d(W \setminus \{i\})$ é ainda um conjunto de inteiros consecutivos. Além disso, a redefinição $d(i) := d(j) + 1$, para algum $j \in W$, faz com que $d(W)$ permaneça um conjunto de inteiros consecutivos.

Terceiro, i não é o único vértice em W com rótulo $d(i)$ e não existe qualquer arco residual (i, j) para $j \in W$. Como existe um vértice $k \in W$ com $k \neq i$ tal que $d(k) = d(i)$, então após a redefinição $W := W \setminus \{i\}$, $d(W)$ é ainda um conjunto de inteiros consecutivos. Por outro lado, após o incremento de $D_{\max}$, o conjunto $Adormec\ (D_{\max}) := \{i\}$ é, trivialmente, um conjunto de inteiros consecutivos.

Conclui-se que, no final de uma passagem genérica pelo ciclo interno e, portanto,

ao longo de todo o ciclo interno, os conjuntos $d(W)$ e $d(Adormec\ (k))$, para $k = 0, 1, \ldots, D_{\max}$, são constituídos por inteiros consecutivos. Também se mantém $d(t') = d_{\min}(W)$ porque t' mantém-se em W durante todo o ciclo interno, o conjunto W nunca é ampliado com a adição de novos vértices e as componentes dos rótulos d nunca diminuem de valor.

Suponha-se agora que, no início de uma qualquer passagem (segunda e seguintes) pela subrotina *InicializarMod*, $d(W)$ e $d(Adormec\ (k))$, para $k = 1, \ldots, D_{\max}$, são conjuntos de inteiros consecutivos e $d(t') = d_{\min}(W)$. Em primeiro lugar, há lugar à remoção de t' de W e inserção em $d(Adormec\ (0))$. Distinguem-se duas situações.

Primeiro, W fica vazio. Neste caso, após a redefinição $W := Adormec\ (D_{\max})$ e $D_{\max} := D_{\max} - 1$, tem-se, trivialmente, que $d(W)$ e $d(Adormec\ (k))$, para $k = 1, \ldots, D_{\max}$, permanecem conjuntos de inteiros consecutivos.

Segundo, W não fica vazio. Então, W continha pelo menos mais um elemento para além de t'. Como $d(t') = d_{\min}(W)$, após $W := W \setminus \{t'\}$, $d(W)$ permanece um conjunto de inteiros consecutivos, bem como os conjuntos $d(Adormec\ (k))$ para $k = 1, \ldots, D_{\max}$.

Finalmente e em ambas as situações, t' é escolhido de forma a $d(t') = d_{\min}(W)$. Portanto, no final de cada passagem pela subrotina *InicializarMod*, $d(W)$ e $d(Adormec(k))$, para $k = 1, \ldots, D_{\max}$, permanecem conjuntos de inteiros consecutivos e $d(t') = d_{\min}(W)$, o que conclui a prova. □

Lema 4.14 (Hao e Orlin [74]) *Ao longo de todo o algoritmo DetCorteMin(s) de Hao e Orlin, $d(i) \leqslant n - 1$ para cada $i \in V$.*

Demonstração: Primeiro, prove-se que, ao longo de todo o algoritmo, se tem o seguinte:

$$\left.\begin{array}{lcll} d_{\min}(W) & \leqslant & n - |W|, & \\ d_{\min}(Adormec\ (k)) & \leqslant & n - |Adormec\ (k)|, & k = 1, \ldots, D_{\max}, \\ d_{\max}(Adormec\ (0)) & \leqslant & n - 1. & \end{array}\right\} \qquad (4.37)$$

No final da primeira passagem pela subrotina *InicializarMod*, aquelas relações são fáceis de verificar, pois, $Adormec\ (0) := \{s\}$, $D_{\max} := 0$, $W := V \setminus \{t'\}$ e, por definição, $d(t') := 0$ e $d(j) = 1$ para todo $j \in W$. Pelo que, $d_{\min}(W) = 0 \leqslant 1 = n - |W|$ e $d_{\min}(Adormec\ (0)) = 1 \leqslant n - 1$.

Suponha-se que as relações (4.37) se verificam no início de uma passagem genérica pelo ciclo interno do algoritmo. Seguidamente, prova-se que, também no final dessa passagem, essas relações se mantêm.

Seja $i \in W$ o vértice activo identificado. Se existe um arco admissível $(i, j) \in G[x]$,

então é invocada a subrotina *Empurrar* (i, j), que não tem influência em (4.37). Se não existe qualquer arco admissível (i, j), então é invocada a subrotina *RotularMod* (i). Neste último caso distinguem-se três situações.

Primeiro, i é o único vértice em W com rótulo $d(i)$. Como $d(W)$ é um conjunto de inteiros consecutivos, então, para $R := \{j \in W : d(j) \geqslant d(i)\}$, tem-se $d_{\min}(R) = d(i) \leqslant d_{\min}(W) + |W \setminus R| \leqslant n - |W| + |W \setminus R| = n - |R|$. Assim, após o incremento de $D_{\max}$ e a consequente definição *Adormec* $(D_{\max}) := R$, tem-se $d_{\min}(\textit{Adormec}\ (D_{\max})) \leqslant n - |\textit{Adormec}\ (D_{\max})|$. Para além disso, $d_{\min}(W \setminus R) = d_{\min}(W) \leqslant n - |W| \leqslant n - |W \setminus R|$ pelo que, após a redefinição $W := W \setminus R$, tem-se $d_{\min}(W) \leqslant n - |W|$.

Segundo, $i \in W$ não é o único vértice em W com rótulo $d(i)$ e existe pelo menos um arco residual (i, j) para $j \in W$. Como existe um vértice $k \in W \setminus \{i\}$ tal que $d(k) = d(i)$, a alteração no valor da componente $d(i)$ não faz com que $d_{\min}(W)$ fique diferente. Como o conjunto W também não é alterado, então $d_{\min}(W) \leqslant n - |W|$ permanece válido.

Terceiro, $i \in W$ não é o único vértice em W com rótulo $d(i)$ e não existe qualquer arco residual (i, j) para $j \in W$. Dada a consecutividade de $d(W)$,

$$d(i) \leqslant d_{\min}(W) + |W| - 1 \leqslant n - |W| + |W| - 1 = n - 1.$$

Assim, após o incremento de $D_{\max}$ e a consequente definição *Adormec* $(D_{\max}) := \{i\}$, tem-se

$$d_{\min}(\textit{Adormec}\ (D_{\max})) = n - 1 = n - |\textit{Adormec}\ (D_{\max})|.$$

Para além disso, como existe um vértice $k \in W \setminus \{i\}$ tal que $d(k) = d(i)$, então

$$d_{\min}(W \setminus \{i\}) = d_{\min}(W) \leqslant n - |W| \leqslant n - |W \setminus \{i\}|,$$

pelo que, após a redefinição $W := W \setminus \{i\}$, tem-se $d_{\min}(W) \leqslant n - |W|$.

Como o conjunto *Adormec* (0) não é alterado nem os seus rótulos, então pode-se também concluir que $d_{\max}(\textit{Adormec}\ (0)) \leqslant n - 1$ no final dessa passagem.

Conclui-se que, no final de uma passagem genérica pelo ciclo interno e, portanto, ao longo de todo o ciclo interno, as relações (4.37) continuam a verificar-se.

Agora, suponha-se que se verificam as relações (4.37) no início de uma qualquer passagem (segunda e seguintes) pela subrotina *InicializarMod*. Como $t' \in W$ e $d(t') = d_{\min}(W) \leqslant n - |W| \leqslant n - 1$, então as redefinições $W := W \setminus \{t'\}$ e *Adormec* $(0) :=$ *Adormec* $(0) \cup \{t'\}$ fazem com que $d_{\max}(\textit{Adormec}\ (0)) \leqslant n - 1$. Distinguem-se duas situações.

Primeiro, W fica vazio. Neste caso, após a redefinição $W :=$ *Adormec* $(D_{\max})$ e

$D_{\max} := D_{\max} - 1$, tem-se trivialmente a verificação das relações (4.37).

Segundo, W não fica vazio. Então, W continha pelo menos mais um elemento para além de t'. Como W é um conjunto de inteiros consecutivos, tem-se

$$d_{\min}(W) \leqslant d_{\min}(W \cup \{t'\}) + 1 \leqslant n - |W \cup \{t'\}| + 1 = n - |W|.$$

Portanto, as relações (4.37) verificam-se ainda no final qualquer passagem pela subrotina *InicializarMod* (segunda e seguintes) o que permite concluir que durante todo o algoritmo também se verificam.

Em qualquer passo do algoritmo, um vértice i ou está em *Adormec* (0) ou está em W ou está em *Adormec* (k) para algum $k = 1, 2, \ldots, D_{\max}$. Se $i \in$ *Adormec* (0), então, conforme provado, $d(i) \leqslant n - 1$; se $i \in W$, então $d(i) \leqslant d_{\min}(W) \leqslant n - |W| \leqslant n - 1$; se $i \in$ *Adormec* (k) para algum $k = 1, 2, \ldots, D_{\max}$, então $d(i) \leqslant d_{\min}($*Adormec* $(k)) \leqslant n - |$*Adormec* $(k)| \leqslant n - 1$. Conclui-se que, para todo $i \in V$, $d(i) \leqslant n - 1$ ao longo de todo o algoritmo DetCorteMin(s) de Hao e Orlin. □

Uma interessante consequência do Lema 4.14 é a de que, para cada $i \in V$, o valor da componente $d(i)$ é incrementado, no máximo, $n - 1$ vezes e, por isso, o número máximo de modificações efectuadas em componentes de d, durante todo o algoritmo de Hao e Orlin, é inferior a n^2.

Seguidamente, mostra-se que o número máximo de execuções do procedimento *Empurrar* (i, j) que *saturam* componentes do prefluxo é inferior ou igual a nm.

Suponha-se que, em determinado instante do algoritmo, o procedimento *Empurrar* (i, j) é invocado com i um vértice activo e (i, j) um arco admissível, e a modificação da componente x_{ij} do prefluxo x provoca a *saturação* do arco (i, j). Denote-se, o rótulo do vértice i nesse instante, por $d'_i \equiv d(i) = d(j) + 1$. Se o mesmo arco voltar a ser *saturado*, então é porque, entretanto, foi enviado fluxo ao longo do arco (j, i) durante a execução do algoritmo. Isso pode ter acontecido em duas ocasiões: durante o procedimento *InicializarMod* quando $t' = j$ - o que só pode acontecer uma vez durante todo o algoritmo; ou o procedimento *Empurrar* (i, j) foi executado usando o arco (j, i) como argumento. Neste último caso, ter-se-ia $d'_j \equiv d(j) = d(i) + 1 \geqslant d'_i + 1$ e no instante da nova *saturação* do arco (i, j) ter-se-ia $d(i) = d(j) + 1 \geqslant d'_j + 1 \geqslant d'_i + 2$. Como $d(i) \leqslant n - 1$, então não pode haver mais do que n *saturações* do arco genérico (i, j). O resultado desejado decorre então de haver m arcos.

Seguidamente, mostra-se que o número máximo de execuções do procedimento *Empurrar* (i, j) que *não saturam* componentes do prefluxo é $\mathcal{O}(n^2 m)$.

Considere-se o escalar ϕ definido através de $\phi := \sum_{i \in I} d(i)$, onde $I := \{i \in W \setminus \{t\} \colon e_i(x) > 0\}$ denota o conjunto dos vértices activos, e $\phi := \phi^+ - \phi^-$ com ϕ^+ um

acumulador de incrementos ao valor de ϕ (inicializado ao valor do ϕ inicial) e ϕ^- um acumulador de decrementos ao valor de ϕ (inicializado a zero).

No início da primeira passagem pelo ciclo interno, o valor inicial de ϕ decorre da primeira passagem pelo procedimento *InicializarMod* e é inferior a n. No final de cada ciclo interno, o valor de ϕ é igual a zero. No início das passagens seguintes pelo ciclo interno, o valor de ϕ decorre do procedimento *InicializarMod* e é inferior a n^2.

Analise-se o que pode acontecer a estes valores durante uma passagem genérica pelo ciclo interno do algoritmo. Seja $i \in W \backslash \{t\}$ o vértice activo identificado. Considerem-se duas situações:

1. Se existe um arco admissível (i, j), então é invocado o procedimento *Empurrar* (i, j). Neste procedimento há que considerar duas situações distintas:

   (a) Se $e(i) = \min\{e(i), r_{ij}\}$, então, após a modificação da componente x_{ij} do prefluxo x, tem-se $e(i) = 0$. Se o vértice j também era activo, então ϕ é decrementado $d(i)$ unidades, caso contrário, ϕ é decrementado $d(i) - d(j) = 1$ unidades. Em ambos os casos, o valor de ϕ é reduzido pelo menos uma unidade.

   (b) Se $e(i) > \min\{e(i), r_{ij}\}$, então, após a modificação da componente x_{ij} do prefluxo x, o arco (i, j) fica *saturado* e o valor de ϕ pode permanecer o mesmo (se j também era vértice activo) ou aumentar $d(j)$ unidades (se j não era vértice activo). Portanto, ϕ^+ aumenta, no máximo, n unidades.

2. Se não existe qualquer arco admissível (i, j), então é invocado o procedimento *Rotular* (i). Neste procedimento há que considerar três situações distintas:

   (a) Se i é o único vértice em W com rótulo $d(i)$, então o valor de ϕ é decrementado em pelo menos $d(i) \geqslant 1$ unidades. Portanto, o valor de ϕ é reduzido pelo menos uma unidade.

   (b) Se i não é o único vértice em W com rótulo $d(i)$ e existe um arco residual (i, j) para $j \in W$, então a consequente redefinição de $d(i)$ implica um incremento do valor de ϕ. Portanto, o valor de ϕ é aumentado e ϕ^+ aumenta, no máximo, n unidades.

   (c) Se i não é o único vértice em W com rótulo $d(i)$ e não existe qualquer arco residual (i, j) para $j \in W$, então o valor de ϕ é decrementado em pelo menos $d(i) \geqslant 1$ unidades. Portanto, o valor de ϕ é reduzido pelo menos uma unidade.

Conforme observado num parágrafo anterior, os passos 1b e 2b não ocorrem mais de nm e n^2 vezes, respectivamente, durante todo o algoritmo. O incremento no

valor de ϕ^+ devido à passagem pelo passo 1b é, no máximo, $n(nm)$, enquanto que o incremento no valor de ϕ^+ devido à passagem pelo passo 2b é, no máximo, n^2. Por isso, $\phi^+ < n + n^2m + n^2$ sempre.

Por outro lado, no final de todo o algoritmo, observa-se $\phi = 0$ dada a inexistência de vértices activos. Por isso, os passos nos quais ϕ é decrementado pelo menos uma unidade, isto é, os passos 1a, 2a e 2c ocorrem, no máximo, $n + n^2 + n^2m = \mathcal{O}(n^2m)$ vezes.

Como se viu num parágrafo anterior, os passos 1a e 2b ocorrem, no máximo, $nm + n^2$ vezes. Assim, juntando todo o recente desenvolvimento, conclui-se que as rotinas *RotularMod (i)* e *Empurrar (i, j)* são chamadas, no máximo, $\mathcal{O}(n^2m)$. Hao e Orlin em [74] demonstraram que esse valor também é a ordem de complexidade em termos do número de operações.

Teorema 4.3 (Hao e Orlin [74]) *Seja $G = (V, E)$ um grafo dirigido com n vértices, m arcos e capacidades u_{ij}, para $(i, j) \in E$, definidas por números não negativos. O algoritmo de Hao e Orlin determina o corte global de capacidade mínima em $\mathcal{O}(n^2m)$ operações.*

À semelhança do que aconteceu anteriormente para o algoritmo de Goldberg e Tarjan também é possível efectuar uma implementação com um melhoramento substancial na ordem de complexidade.

Teorema 4.4 (Hao e Orlin [74]) *Nas condições do Teorema 4.3, o algoritmo de Hao e Orlin determina o corte global de capacidade mínima em $\mathcal{O}(nm \log(n^2/m))$ operações.*

4.4.3 Experiência computacional

Nesta secção, apresentam-se alguns resultados computacionais obtidos pela aplicação do algoritmo de Hao e Orlin num conjunto de problemas teste disponíveis em `http://theory.lcs.mit.edu/~mslevine/mincut/`. Cada um desses problemas teste corresponde a um subproblema de identificação de restrições de eliminação de subpercursos em problemas de caixeiro viajante simétrico. As características dos problemas teste podem observar-se na tabela de resultados 4.2.

Usou-se a implementação de domínio público `GMINDCUT` do algoritmo de Hao e Orlin, escrita em linguagem `C` por Georg Skorobohatyj do ZIB (Konrad-Zuse-Zentrum für Informationstechnik Berlin) e disponível em `ftp://ftp.zib.de/pub/Packages/mathprog/mincut/global-dir`.

A implementação realizada consiste num programa em MATLAB 6.0 que lê o problema teste, substitui cada aresta por dois arcos com direcções opostas e a mesma capacidade e executa o código `GMINDCUT`.

O problema teste é lido no formato *DIMACS* [47], que corresponde a um ficheiro ASCII que contém, na primeira linha, os valores de $n = |V|$ e $m = |E|$ de um grafo dirigido $G = (V, E)$ e, em cada uma das linhas seguintes, os valores das capacidades u_{ij} para cada arco (i, j), no formato i, j e u_{ij}. Se o grafo dirigido subjacente $G = (V, E)$ com custos $u_{ij} \geqslant 0$, para $(i, j) \in E$, for fortemente conexo, então o código `GMINDCUT` determina o conjunto $S^* \subset V$ tal que $u(S^*) = \sum_{(i,j)\in\delta^+(S^*)} u_{ij}$ seja mínimo; caso contrário, determina um conjunto $S^* \subset V$ tal que $\delta^+(S^*)$ seja vazio.

Os resultados computacionais foram obtidos num PC com processador Intel Pentium II a 350MHz com 128Mb de RAM e 256K de cache. A Tabela 4.2 contém um resumo dos resultados obtidos. As primeiras três colunas caracterizam o problema, a quarta coluna indica a cardinalidade de S^*, a quinta coluna apresenta o tempo total de leitura, execução de `GDMINCUT` e impressão de resultados, e a sexta coluna, o tempo de execução do algoritmo de Hao e Orlin.

Problema	$\|V\|$	$\|E\|$	$\|S^*\|$	**Tempo**$[s'10ms]$	**Tempo HO**$[s'10ms]$
tspdantzig42x1	42	100	1	0′0	0′0
tspdantzig42x2	42	84	1	0′0	0′0
tsppr76x1	76	184	58	0′0	0′0
tsppr439x1	439	1050	1	0′2	0′2
tsppr439x2	439	1026	437	0′3	0′2
tsp.att532.x.1	532	1574	514	0′50	0′50
tspgr666x1	666	1638	656	0′11	0′11
tspgr666x2	666	1624	1	0′18	0′18
tsppr1002x1	1002	2004	1	0′55	0′55
tsp.vm1084.x.1	1084	2504	1	0′29	0′29
tsp.d1291.x.1	1219	3884	1219	0′42	0′42
tsp.fl1400.x.1	1400	4462	1253	0′27	0′27
tsp.vm1748.x.1	1748	4672	1734	0′70	0′69
usa13509.xo.15631	13509	31262	13422	24′76′	27′73
usa13509.xo.17048	13509	34098	13366	25′16′	25′14
usa13509.xo.17111	13509	34222	13500	21′38′	21′36
d15112.xo.19057	15112	38114	15105	22′47	22′45
pla33810.xo.39367	33810	78734	31715	263′96	263′94
pla33810.xo.39456	33810	78912	33631	144′83	144′79
pla85900.xo.102934	85900	205868	82586	890′89	890′80

Tabela 4.2: *Resultados computacionais da implementação do algoritmo de Hao e Orlin.*

4.5 O problema de afectação

O problema de afectação é um dos tópicos introdutórios em Optimização Combinatória por servir de ilustração da Unimodularidade Total e por possuir algoritmos que o resolvem de modo elegante e eficiente. O problema possui uma descrição simples: como afectar n indivíduos a n tarefas de modo a minimizar o tempo total na execução dessas tarefas, sabendo, antecipadamente, o tempo exacto que cada indivíduo demora a executar cada tarefa. A formulação clássica deste problema usa as variáveis binárias x_{ij} para $i, j = 1, \ldots, n$ e é a seguinte:

$$\begin{aligned}
\max \quad & \sum_{i,j=1}^{n} c_{ij} x_{ij} \\
\text{s.a} \quad & \sum_{i=1}^{n} x_{ij} = 1, \quad j = 1, 2, \ldots, n, \\
& \sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, 2, \ldots, n, \\
& x_{ij} \geqslant 0, \qquad i, j = 1, 2, \ldots, n, \\
& x_{ij} \in \mathbb{Z}, \qquad i, j = 1, 2, \ldots, n,
\end{aligned} \tag{4.38}$$

ou, na forma matricial, $\max\{cx\colon Ax = 1, x \geqslant 0, x \in \mathbb{Z}^{n \times n}\}$, onde A é a matriz de incidência vértice-aresta de um grafo bipartido completo.

Sendo A uma matriz totalmente unimodular, o problema (4.38) pode ser resolvido ignorando as restrições de integralidade nas variáveis através do método Simplex. Mais, o problema de afectação é, simultaneamente, um problema de fluxo máximo e um problema de fluxo de custo mínimo para os quais são conhecidas implementações específicas do método Simplex que funcionam em tempo polinomial.

No entanto, o problema de afectação pode ser resolvido por uma outra classe de métodos, ditos combinatórios por serem métodos que, na sua génese, não usam a formulação algébrica do problema - esse não é o caso, por exemplo, do método Simplex e do método elipsóide [84]. Um desses métodos é o método Húngaro, descrito na Secção 4.5.2, proposto por Kuhn em [86] na década de 50, que pode ser implementado em $\mathcal{O}(n^3)$ operações aritméticas e comparações (veja-se Lawler [89], por exemplo), conforme se demonstrará. Como também se verá, o método reúne as características de um algoritmo primal-dual: mantém a dual admissibilidade e complementaridade durante todo o seu desenrolar e termina quando for alcançada a primal admissibilidade. Portanto, apesar de não ser um método de Simplex clássico, não lhe é muito diferente.

Uma consequência imediata do Lema 4.15, é que se pode assumir, sem perda de generalidade, que $c_{ij} \geqslant 0$ para todo i, j. Outra consequência é o facto de a função objectivo

de (4.38) poder ser substituída por $\sum_{i,j=1}^{n}(c_{ij}-u_i-v_j)\,x_{ij}$ sem que o conjunto de soluções óptimas fique diferente.

Lema 4.15 (Wolsey [129, página 57]) *Seja $\overline{x}$ admissível para o problema (4.38), e sejam $u, v \in \mathbb{R}^n$ quaisquer. Então,*

$$\sum_{i,j=1}^{n} c_{ij}\overline{x}_{ij} = \sum_{i,j=1}^{n}(c_{ij}-u_i-v_j)\,\overline{x}_{ij} + \sum_{i=1}^{n} u_i + \sum_{j=1}^{n} v_j.$$

Demonstração:

$$\begin{aligned}\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}\overline{x}_{ij} &= \sum_{i=1}^{n}\sum_{j=1}^{n}(c_{ij}-u_i-v_j)\overline{x}_{ij} + \sum_{i=1}^{n} u_i\left(\sum_{j=1}^{n}\overline{x}_{ij}\right) + \sum_{j=1}^{n} v_j\left(\sum_{i=1}^{n}\overline{x}_{ij}\right)\\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}(c_{ij}-u_i-v_j)\overline{x}_{ij} + \sum_{i=1}^{n} u_i + \sum_{j=1}^{n} v_j.\end{aligned}$$

□

O Lema 4.15 permanece verdadeiro para qualquer problema de optimização cuja região admissível esteja contida na região admissível do problema de afectação. Esse é o caso, por exemplo, do problema do caixeiro viajante assimétrico ou de problemas de afectação com restrições adicionais (quando, por exemplo, algumas variáveis x_{ij} estão fixas a zero).

Como a matriz A é totalmente unimodular, o valor óptimo de (4.38) não muda se se ignorarem as restrições "$x_{ij} \in \mathbb{Z}$". Por isso, o conjunto de soluções óptimas do problema linear de afectação $\max\{cx\colon Ax = 1, x \geqslant 0\}$, que se designa por problema relaxado, é o invólucro convexo do conjunto de soluções óptimas do problema de afectação (4.38). O dual do problema relaxado é

$$\begin{aligned}\min\quad & \sum_{i=1}^{n} u_i + \sum_{j=1}^{n} v_j\\ \text{s.a}\quad & u_i + v_j \geqslant c_{ij}, \qquad i,j = 1,\ldots,n,\end{aligned} \tag{4.39}$$

com a ajuda do qual se pode demonstrar o seguinte resultado.

Lema 4.16 (Wolsey [129, página 58]) *Uma solução $\overline{x}$, admissível para o problema de afectação(4.38), é óptima se e só se existirem $u, v \in \mathbb{R}^n$ tais que*

$$\overline{c}_{ij} \equiv c_{ij} - u_i - v_j \leqslant 0, \quad \overline{x}_{ij} = 1 \Rightarrow \overline{c}_{ij} = 0, \quad i,j = 1,2,\ldots,n.$$

Nesse caso, o valor óptimo é $\sum_{i=1}^{n} u_i + \sum_{j=1}^{n} v_j$.

Demonstração: Se $\overline{x}$ é solução óptima para (4.38), então $\overline{x}$ também é solução óptima para o problema relaxado. A condição necessária é então consequência da dualidade forte da programação linear. Reciprocamente, se $u, v \in \mathbb{R}^n$ satisfazem as condições do lema, então, relativamente ao problema relaxado, tem-se uma solução dual admissível (u, v) e uma solução primal admissível $\overline{x}$ tais que

$$\sum_{i,j=1}^{n} (c_{ij} - u_i - v_j)\,\overline{x}_{ij} = 0.$$

Atendendo ao Lema 4.15, verifica-se dualidade forte da programação linear, pelo que (u, v) é óptima para o dual (4.39), e $\overline{x}$ é óptima para o problema relaxado. Como $\overline{x}$ é inteira, $\overline{x}$ é óptima para (4.38). □

O problema de afectação pode também interpretar-se como um problema de emparelhamento perfeito de peso máximo num grafo bipartido completo da maneira que se explica a seguir. É esta característica que coloca o problema de afectação entre o leque de problemas analisados em Optimização Combinatória. Reciprocamente, os problemas de emparelhamento perfeito em grafos bipartidos podem ser interpretados como problemas de fluxo em redes de custo mínimo, nos quais todas as capacidades e requerimentos são unitários.

Seja $G = (V_1, V_2, E)$ um grafo bipartido com pesos $c_{ij} \geqslant 0$ para $(i, j) \in E \subseteq V_1 \times V_2$. Considere-se o problema de determinar em G o emparelhamento perfeito de peso máximo. Entenda-se o *peso* de um emparelhamento como sendo a soma dos pesos das arestas que constituem esse emparelhamento. Este problema pode formular-se algebricamente por

$$\begin{array}{lll} \max & \displaystyle\sum_{(i,j)\in E} c_{ij}x_{ij} & \\ \text{s.a} & \displaystyle\sum_{(i,j)\in\delta^+(i)} x_{ij} = 1, & i \in V_1, \\ & \displaystyle\sum_{(i,j)\in\delta^-(j)} x_{ij} = 1, & j \in V_2, \\ & x_{ij} \geqslant 0, & (i,j) \in E, \\ & x_{ij} \in \mathbb{Z}, & (i,j) \in E, \end{array} \tag{4.40}$$

ou, na forma matricial, $\max\{cx\colon Ax = 1, x \geqslant 0, x \in \mathbb{Z}^m\}$, onde A é a matriz de incidência vértice-aresta do grafo G e $m = |E|$. Novamente, como A é totalmente unimodular, existe dualidade fraca e forte entre o problema (4.40) e o problema dual

$$\begin{array}{lll} \min & \displaystyle\sum_{i\in V_1} u_i + \sum_{j\in V_2} v_j & \\ \text{s.a} & u_i + v_j \geqslant c_{ij}, & (i,j) \in E. \end{array} \tag{4.41}$$

Como uma condição necessária para que o problema (4.40) seja admissível é $|V_1| = |V_2|$, se $E = V_1 \times V_2$, então o problema (4.40) é apenas uma maneira diferente de descrever o problema de afectação (4.38). Em geral, interessa a resolução do problema (4.40) quando $E \subset V_1 \times V_2$. Isso é equivalente a resolver o problema de afectação, impedindo determinadas afectações indivíduo-tarefa.

Na Secção 4.5.2, é apresentado o método Húngaro que determina o emparelhamento perfeito de peso máximo num grafo bipartido em $\mathcal{O}(n^3)$ operações, com $n = |V_1| = |V_2|$. O método usa, em cada iteração, o algoritmo para determinação do emparelhamento de máxima cardinalidade num grafo bipartido, recordado na próxima secção. Todo o desenvolvimento algorítmico que se segue é, essencialmente, baseado no Capítulo 4 do livro de Wolsey [129].

4.5.1 Emparelhamentos Perfeitos

Considere-se o problema de determinar o emparelhamento de máxima cardinalidade num grafo bipartido $G = (V_1, V_2, E)$. A resolução deste problema permite assegurar ou não a existência de uma solução admissível para (4.40). O problema pode formular-se algebricamente do seguinte modo

$$\begin{array}{lll} \max & \displaystyle\sum_{(i,j)\in E} x_{ij} & \\ \text{s.a} & \displaystyle\sum_{(i,j)\in\delta^+(i)} x_{ij} \leqslant 1, & (i \in V_1), \\ & \displaystyle\sum_{(i,j)\in\delta^-(j)} x_{ij} \leqslant 1, & (j \in V_2), \\ & x_{ij} \geqslant 0, & (i,j) \in E, \\ & x_{ij} \in \mathbb{Z}, & (i,j) \in E, \end{array} \tag{4.42}$$

ou, na forma matricial, $\max\{1x\colon Ax \leqslant 1, x \geqslant 0, x \in \mathbb{Z}^m\}$, onde A é a matriz de incidência vértice-aresta de G. A abordagem natural, para resolução deste problema, é a de obter, sistematicamente, caminhos aumentados no grafo G, como decorre da próxima proposição que é válida para qualquer grafo não orientado.

Lema 4.17 (e.g., Wolsey [129, página 54]) *Seja $G = (V, E)$ um grafo não orientado e $\mathcal{M}$ um emparelhamento de G.*

1. *Se P é um caminho aumentado relativamente a $\mathcal{M}$, então $\mathcal{M}' = (\mathcal{M} \cup P) \setminus (\mathcal{M} \cap P)$ é um emparelhamento de G e $|\mathcal{M}'| = |\mathcal{M}| + 1$.*
2. *Se não existe nenhum caminho aumentado relativamente a $\mathcal{M}$, então $\mathcal{M}$ é um emparelhamento de máxima cardinalidade de G.*

No caso de G ser um grafo bipartido, a determinação de um caminho aumentado pode ser efectuada de forma eficiente, usando o algoritmo enumerativo, descrito na Figura 4.18.

Algoritmo de determinação do emparelhamento de máxima cardinalidade num grafo bipartido $G = (V_1, V_2, E)$

Passo 0*: Inicializar*

Seja $\mathcal{M}$ um emparelhamento de G.
Atribuir o rótulo de *não explorado* a todo o vértice $i \in V_1 \cup V_2$.
Atribuir o rótulo de * a todo vértice exposto $i \in V_1$.

Passo 1*: Rotular*

Seja i um vértice com rótulo *não explorado*. Se não existir,
então segue para Passo 3.
Se $i \in V_1$,
então atribuir o rótulo i a todo o vértice $j \in V_2$ tal que
$(i, j) \in E \setminus \mathcal{M}$ e j possui o rótulo de *não explorado*.
Se $i \in V_2$,
então se i é um vértice exposto,
então segue para Passo 2,
senão atribuir o rótulo i ao vértice $j \in V_1$ tal que $(j, i) \in \mathcal{M}$.
Repetir o Passo 1.

Passo 2*: Caminho aumentado encontrado*

Usar os rótulos para identificar o caminho aumentado P,
$\mathcal{M} \leftarrow (\mathcal{M} \cup P) \setminus (\mathcal{M} \cap P)$.
Atribuir o rótulo de *não explorado* a todo o vértice $i \in V_1 \cup V_2$.
Atribuir o rótulo de * a todo vértice exposto $i \in V_1$.
Segue para Passo 1.

Passo 3*: Solução óptima encontrada*

Não existe caminho aumentado relativamente a $\mathcal{M}$.

Figura 4.18: *Algoritmo de determinação do emparelhamento de máxima cardinalidade num grafo bipartido.*

O algoritmo começa com um primeiro emparelhamento $\mathcal{M}$ conhecido - uma aresta apenas, por exemplo. No início, todos os vértices de $V = V_1 \cup V_2$ são rotulados com *não explorado*, à excepção dos vértices expostos de V_1 que recebem o rótulo * - a existir caminho aumentado relativamente a M, uma das extremidades desse caminho será um vértice exposto de V_1 e a outra extremidade será um vértice exposto de V_2. As

sucessivas passagens pelo Passo 1 permitem construir (por ampliação) uma floresta de caminhos alternados parciais, isto é, um conjunto de árvores de caminhos alternados parciais (parciais no sentido em que só uma das extremidades é um vértice exposto) emanando de cada um dos vértices expostos de V_1. Esse processo de construção é interrompido quando um caminho alternado é identificado ou quando já não é possível ampliar essa floresta. No primeiro caso, sucede-se o Passo 2, onde o emparelhamento $\mathcal{M}$ é redefinido de modo a conter mais uma aresta, de acordo com o Teorema 4.17, após o que, se dá início ao Passo 1 novamente. No segundo caso, fica verificada a inexistência de um caminho alternado por enumeração exaustiva.

O Teorema 4.5 mostra que, no final do algoritmo, $\mathcal{M}$ é o emparelhamento de máxima cardinalidade.

Teorema 4.5 (Wolsey [129, página 56]) *No Passo 3 do algoritmo, sejam*

$$\begin{aligned} V_1^- &\equiv \{i \in V_1 \colon \textit{rótulo de } i \textit{ é } = \textit{não explorado}\}, \quad V_1^+ \equiv V_1 \setminus V_1^-, \\ V_2^- &\equiv \{i \in V_2 \colon \textit{rótulo de } i \textit{ é } = \textit{não explorado}\}, \quad V_2^+ \equiv V_2 \setminus V_2^-. \end{aligned}$$

O conjunto $R = V_1^- \cup V_2^+$ *é uma cobertura por vértices de* G *tal que* $|\mathcal{M}| = |R|$.

Demonstração: Quando o Passo 3 é executado:

1. não há arestas ligando vértices de V_1^+ a vértices de V_2^-. Por isso, toda a aresta (i, j) do grafo G é da forma $i \in V_1^+, j \in V_2^+$ ou $i \in V_1^-, j \in V_2^+$ ou $i \in V_1^-, j \in V_2^-$. Consequentemente, $R \equiv V_1^- \cup V_2^+$ é uma cobertura por vértices de G.

2. Todos os vértices de V_2^+ são incidentes em alguma aresta $(i, j) \in \mathcal{M}$ com $i \in V_1^+$. Por outro lado, todos os vértices i de V_1^- são extremidades de arestas $(i, j) \in \mathcal{M}$ tais que $j \in V_2^-$ (caso contrário, se i fosse exposto, então teria recebido o rótulo * ou se não fosse exposto, então teria recebido algum rótulo durante o algoritmo). Logo, $|R| \leqslant |\mathcal{M}|$.

3. Por dualidade fraca entre a cardinalidade de um emparelhamento e a cardinalidade de uma cobertura por vértices, sabe-se que $|R| \geqslant |\mathcal{M}|$. Por isso, $|\mathcal{M}| = |R|$.

□

Teorema 4.6 (e.g., Ahuja et al. [1, páginas 478–494]) *O algoritmo de determinação do emparelhamento de máxima cardinalidade num grafo bipartido* $G = (V_1, V_2, E)$ *requer* $\mathcal{O}(n^3)$ *operações aritméticas e comparações.*

Demonstração: Entre duas passagens sucessivas pelo Passo 2, o processo de rotulação requer $\mathcal{O}(n^2)$ operacões aritméticas e comparações com $n = \max\{|V_1|, |V_2|\}$, pois, durante duas passagens sucessivas pelo Passo 2, cada vértice é explorado uma única vez no máximo e, a partir deste, n vértices são rotulados no máximo. Em cada passagem pelo Passo 2, $|\mathcal{M}|$ aumenta pelo menos uma unidade. O resultado pretendido decorre de $|\mathcal{M}| \leqslant n$. □

Naturalmente, uma implementação prática deste algoritmo não inicializaria os rótulos de todos os vértices do modo que é descrito no Passo 2. De facto, apenas os rótulos dos vértices pertencentes à árvore dos caminhos alternados parciais onde ocorreu o caminho alternado identificado necessitam ser modificados. Não se abordará este aspecto por ser irrelevante no contexto do método que se descreve na próxima secção.

4.5.2 O Método Húngaro

Nesta secção, apresenta-se o método Húngaro que resolve o problema (4.40) em $\mathcal{O}(n^3)$ operações aritméticas e comparações. O algoritmo é descrito na Figura 4.19 e usa, em cada iteração, o algoritmo de determinação do emparelhamento de máxima cardinalidade num grafo bipartido, descrito na secção anterior.

Método Húngaro [86]

Passo 0*: Inicialização*

Sejam $u, v \in \mathbb{R}^n$ tais que $\overline{c}_{ij} \equiv c_{ij} - u_i - v_j \leqslant 0$ para todo $(i,j) \in E$.

Passo 1*: Primal*

Seja $\mathcal{M}$ um emparelhamento de cardinalidade máxima em $\overline{G} = (V_1, V_2, \overline{E})$, onde $\overline{E} = \{(i,j) \in E : \overline{c}_{ij} = 0\}$.
Sejam V_1^+, V_2^- os conjuntos obtidos no Passo 3 do algoritmo descrito antes.
Se $|\mathcal{M}| = n$, então $\mathcal{M}$ é solução óptima de (4.40), parar.

Passo 2*: Dual*

Se $S \equiv \{(i,j) \in E : i \in V_1^+, j \in V_2^-\} = \emptyset$,
então não existe solução admissível para (4.40), parar;
senão calcular $\delta = \min\{-\overline{c}_{ij} : (i,j) \in S\}$.
Definir $u_i \leftarrow u_i - \delta$ para $i \in V_1^+$ e $v_j \leftarrow v_j + \delta$ para $j \notin V_2^-$.
Segue para o *Passo* 1.

Figura 4.19: *Método Húngaro.*

Em cada passagem pelo Passo 1, (u, v) é dual admissível (isto é, $\overline{c}_{ij} \equiv c_{ij} - u_i - v_j \leqslant 0$ para todo $(i, j) \in E$), e o vector característico, denotado $\overline{x}$, do emparelhamento $\mathcal{M}$ satisfaz a condição de complementaridade "$\overline{x}_{ij} = 1 \Rightarrow \overline{c}_{ij} = 0$", mas pode não ser primal admissível. A primal admissibilidade é alcançada quando $\mathcal{M}$ for um emparelhamento perfeito. Se $\mathcal{M}$ não é perfeito, o Passo 2 visa a modificação das variáveis duais de forma a proporcionar um emparelhamento com mais uma aresta.

Lema 4.18 (Wolsey [129, página 59]) *No final do Passo 1 do método Húngaro,* $|V_1^+| > |V_2^+|$.

Demonstração: Cada vértice j de V_2^+ é extremidade de uma aresta $(i, j) \in \mathcal{M}$ com $i \in V_1^+$. Portanto, $|V_1^+| \geqslant |V_2^+|$. No entanto, V_1^+ contém pelo menos um vértice exposto que recebeu o rótulo *. Portanto, $|V_1^+| > |V_2^+|$. □

Lema 4.19 (Wolsey [129, páginas 59-60]) *No final do Passo 2 do método Húngaro, se* $S \neq \emptyset$, *então:*

1. $\delta > 0$;
2. $\overline{c}_{ij} \equiv c_{ij} - u_i - v_j \leqslant 0$ *para todo* $(i, j) \in E$;
3. *os rótulos* $|V_1^+|$ *e* $|V_2^+|$ *permanecem válidos.*

Demonstração:

1. Como foi demonstrado no Teorema 4.5, não existem arestas $(i, j) \in \overline{E}$ tais que $i \in V_1^+$ e $j \in V_2^-$. Portanto, $\overline{c}_{ij} < 0$ para todo $(i, j) \in S$, o que implica que $\delta \equiv \min\{-\overline{c}_{ij} : (i, j) \in S\} > 0$.

2. Sejam u^{old} e v^{old} os valores u e v antes do Passo 2. Então, para cada $(i, j) \in E$,

$$\overline{c}_{ij} = \begin{cases} c_{ij} - u_i - v_j &= c_{ij} - u_i^{\text{old}} - v_j^{\text{old}}, & \text{se } i \in V_1^+ \text{ e } j \in V_2^+, \\ c_{ij} - u_i - v_j &= c_{ij} - u_i^{\text{old}} - v_j^{\text{old}} + \delta, & \text{se } i \in V_1^+ \text{ e } j \in V_2^-, \\ c_{ij} - u_i - v_j &= c_{ij} - u_i^{\text{old}} - v_j^{\text{old}} - \delta, & \text{se } i \in V_1^- \text{ e } j \in V_2^+, \\ c_{ij} - u_i - v_j &= c_{ij} - u_i^{\text{old}} - v_j^{\text{old}}, & \text{se } i \in V_1^- \text{ e } j \in V_2^-, \end{cases}$$

   de onde facilmente se conclui o resultado desejado. Note-se que, se $i \in V_1^+$ e $j \in V_2^-$, então $\overline{c}_{ij} \leqslant 0$, pela definição de δ.

3. Para as arestas $(i, j) \in \overline{E}$ tais que $i \in V_1^+$ e $j \in V_2^+$, em particular as arestas de $\mathcal{M}$, a actualização das variáveis duais não altera o valor de $\overline{c}_{ij}$. Pelo que,

permanecendo aquele conjunto de arestas em $\overline{E}$, os rótulos permanecem válidos. Consequentemente, o Passo Primal pode iniciar-se com os conjuntos de rótulos anteriores.

□

Lema 4.20 *Se, no Passo 2 do método Húngaro, $S = \emptyset$, então não existe solução admissível para o problema (4.40).*

Demonstração: Se $S = \emptyset$, então não existe nenhuma aresta de E orientada de V_1^+ para V_2^-. Por isso, como se pode verificar na demonstração do Lema 4.19, o vector $(\overline{u}, \overline{v})$ definido por

$$\overline{u}_i = \begin{cases} u_i & \text{se } i \notin V_1^+, \\ u_i - \delta & \text{se } i \in V_1^+, \end{cases} \qquad \overline{v}_i = \begin{cases} v_i & \text{se } i \notin V_2^-, \\ v_i + \delta & \text{se } i \notin V_2^-, \end{cases}$$

é admissível para o dual (4.41), para todo $\delta > 0$. Mas, isso significa que o dual (4.41) é ilimitado e, portanto, não existe solução admissível para (4.40). □

Teorema 4.7 (Lawler [89] e Derigs [46, página 135]) *O método Húngaro requer $\mathcal{O}(n^3)$ operações aritméticas e comparações.*

Demonstração: Na primeira passagem pelo Passo 1 do método Húngaro, a determinação do emparelhamento de cardinalidade máxima requer $\mathcal{O}(n^3)$ operações aritméticas e comparações, conforme foi demonstrado no Teorema 4.6.

Prove-se agora que, entre duas execuções sucessivas do Passo 1 sem que haja entre elas um incremento no valor de $|\mathcal{M}|$, o número operações aritméticas e comparações é $\mathcal{O}(n^2)$. Seja k o número de vértices de V_2, e de V_1 em igual número, que durante essas iterações passaram de *não rotulados* a *rotulados* sem que haja alteração no valor de $|\mathcal{M}|$. O esforço computacional envolvido na confirmação de que não existe um emparelhamento de maior cardinalidade nos sucessivos $\overline{G}$ corresponde, simplesmente, a averiguar se, entre os vizinhos daqueles vértices em V_1, existe algum em V_2^-. Para cada um daqueles vértices, o esforço computacional é $\mathcal{O}(n)$. O resultado pretendido decorre de $k \leqslant n$.

Suponha-se que numa dada iteração do método Húngaro, a redefinição das variáveis duais permite a inserção de um arco (i, j) com $i \in V_1^+$ e $j \in V_2^-$ tal que $(k, j) \in \mathcal{M}$ com $k \in V_1^-$ e (k, l) pertencente ao novo grafo $\overline{G}$ com $l \in V_2^- \setminus \{j\}$ - ver Figura 4.20. Esta é a situação que permite o incremento de $|\mathcal{M}|$ no subsequente Passo 1. À semelhança do que foi demonstrado para o Teorema 4.6, são necessárias $\mathcal{O}(n^2)$ operações aritméticas

e comparações até concluir que o novo emparelhamento $\mathcal{M}$ é de máxima cardinalidade no novo grafo $\overline{G}$ ou encontrar um emparelhamento de cardinalidade superior. Como existem, no máximo, n incrementos de $\mathcal{M}$, conclui-se que o esforço total envolvido em iterações em que haja incremento de $\mathcal{M}$ é $\mathcal{O}(n^3)$.

Sumariando os três parágrafos anteriores, o esforço computacional total com o método Húngaro é $\mathcal{O}(n^3)$. □

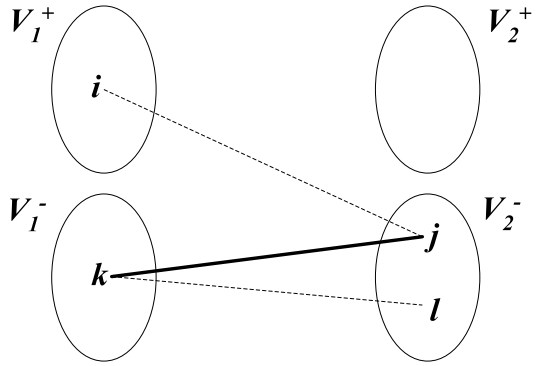


Figura 4.20: *Redefinição das variáveis duais no método Húngaro.*

4.5.3 Histórico

4.5.3.1 Emparelhamento de máxima cardinalidade

Bons trabalhos de síntese sobre o problema de determinar o emparelhamento de máxima cardinalidade num grafo não orientado (bipartido ou não) incluem Tarjan [123, Capítulo 9], Gibbons [61, Capítulo 5], Ahuja et et al. [1, Capítulo 12], Papadimitriou e Steiglitz [107, Capítulo 10], Lawler [89, Capítulo 5], Lovasz [92], Lovasz e Plummer [93], Cook et al. [41, Capítulo 5] e o artigo de Galil [59]. Essencialmente, todas estas referências descrevem o algoritmo proposto por Edmonds em [49, 50] para determinar o emparelhamento de peso máximo (do qual o problema de determinar o emparelhamento de máxima cardinalidade é um caso particular).

No caso do grafo ser bipartido, o problema de determinar o emparelhamento de peso máximo pode ser resolvido em $\mathcal{O}\left(nm \log_{2+m/n} n\right)$ operações aritméticas e comparações, conforme descrito por Tarjan em [123, Secção 8.4]. Se o problema for o de determinar o emparelhamento de máxima cardinalidade, Hopcroft e Karp em [79] e Even e Tarjan em [53] mostraram que o algoritmo descrito na Secção 4.5.1 pode ser implementado com complexidade computacional $\mathcal{O}\left(n^{1/2}m\right)$. Em Ahuja et al. [1, páginas 478–494], pode-se ver uma explicação alternativa da complexidade de $\mathcal{O}(n^3)$, demonstrada nessa secção. Alt et al. em [3] propuseram um algoritmo que funciona em $\mathcal{O}\left(n^{1.5}(m/\log n)^{1/2}\right)$.

O problema de determinar o emparelhamento de peso máximo num grafo não orientado (bipartido ou não) pode ser resolvido pelo algoritmo de Edmonds, cuja complexidade é simples de provar ser $\mathcal{O}(n^4)$ (ver [107, Capítulo 10], e.g.). Lawler em [89, Capítulo 5] e Gabow em [58] mostraram que o algoritmo de Edmonds pode ser implementado em $\mathcal{O}(n^3)$. Galil, Micali e Gabow em [60] mostraram ser possível uma implementação em $\mathcal{O}(nm \log n)$, portanto, superior em grafos esparsos. Se o problema for o de determinar o emparelhamento de máxima cardinalidade, Even e Kariv em [52] generalizaram o algoritmo de Hopcroft e Karp para funcionar em $\mathcal{O}\left(\min(n^{2.5}, n^{1/2}m \log\log n)\right)$ e Micali e Vazirani em [97] introduziram-lhe modificações que permitiram chegar a $\mathcal{O}\left(n^{1/2}m\right)$.

Em 1991, o *Center for Discrete Mathematics and Theoretical Computer Science* (DIMACS, `http://dimacs.rutgers.edu/`) propôs a sua primeira competição de implementações, precisamente em problemas fluxos em redes e emparelhamentos - o resultado é documentado em [47]. Geradores de instâncias de problemas de emparelhamentos de máxima cardinalidade (bem como problemas de afectação, fluxo máximo e fluxo de custo mínimo) e algumas implementações de algoritmos podem ser obtidos em `ftp://dimacs.rutgers.edu/` no directório `pub/netflow/matching`. Destaca-se a implementação em `Fortran 77` do algoritmo de Micali e Vazirani por Mattingly e Ritchey [96] para o emparelhamento de máxima cardinalidade e a implementação em `C` do algoritmo de Gabow por Rothberg [111], em ambos os casos aplicáveis a grafos não necessariamente bipartidos.

A implementação em `C` do algoritmo de Edmonds por Cook e Rohe [40], denominada `Blossom IV`, para o problema de emparelhamento de peso máximo, encontra-se disponível em `http://www.or.uni-bonn.de/home/rohe/matching.html`.

A biblioteca de algoritmos combinatórios para matemática discreta e teoria de grafos `Combinatorica`, documentada em [118], contém implementações em `Mathematica` para o problema do emparelhamento de cardinalidade máxima num grafo bipartido. A biblioteca é acessível em `ftp://ftp.cs.sunysb.edu` no directório `pub/Combinatorica` e contém implementações para muitos outros problemas de matemática discreta.

A biblioteca de implementações algorítmicas `LEDA` (abreviatura de *Library of Efficient Data types and Algorithms*) contém implementações em `C++` de algoritmos para o emparelhamento de máxima cardinalidade em grafos bipartidos. Nomeadamente, as implementações do algoritmo de Hopcroft e Karp e do algoritmo de Alt, Blum, Mehlhorn e Paul podem ser obtidas em `ftp://ftp.mpi-sb.mpg.de` no directório `/pub/LEDA` ou em `http://www.mpi-sb.mpg.de/LEDA/leda.html`.

A aplicação *Stanford GraphBase*, desenvolvida por Knuth e documentada em [85], contém implementações de vários algoritmos para problemas de optimização combinatória, incluindo o algoritmo de Alt, Blum, Mehlhorn e Paul (para o emparelhamentos

de máxima cardinalidade em grafos bipartidos) e o algoritmo de Goldberg e Tarjan [66] (para o fluxo máximo), escritas por Setubal [115]. Podem ser obtidas em `http://www.cs.sunysb.edu/~algorith/implement/bipm/implement.shtml`.

Implementações em `Pascal` de vários algoritmos para o emparelhamento de máxima cardinalidade por Syslo, Deo e Kowalik, descritas em [122], estão disponíveis em `http://www.cs.sunysb.edu/~algorith/implement/syslo/`; e, para o emparelhamento de máxima cardinalidade em grafos bipartidos ou não por Moret e Shapiro, descritas em [99], estão disponíveis em `http://www.cs.sunysb.edu/~algorith/implement/moret/`.

4.5.3.2 Afectação

Bons trabalhos de síntese sobre o problema de afectação incluem Derigs [46, Capítulo 10], Ahuja et al. [1, Capítulo 12], Bazaraa, Jarvis e Sherali [21, Capítulo 10.7], Papadimitriou e Steiglitz [107, Capítulo 11], Bertsekas [22, Capítulo 7.1] e os artigos de Dell'Amico e Toth [45], Martello e Toth [95], Akgul [2] e Burkard e Çela [28].

As abordagens propostas para a solução do problema de afectação podem ser agrupadas em três classes: algoritmos primal-duais (baseados na identificação de caminhos aumentados), algoritmos primais puros (basicamente, são implementações especializadas do Algoritmo Simplex para redes) e algoritmos duais puros (abordagens em que a solução primal admissível é encontrada apenas no último passo do algoritmo sem requerer complementaridade).

O método Húngaro, assim designado em homenagem ao trabalho de Konig e Egerváry no qual se baseia e desenvolvido por Kuhn [86, 87], enquadra-se na primeira classe. O método foi motivado pela dualidade forte entre o problema do emparelhamento de cardinalidade máxima e o problema da cobertura (por vértices) de cardinalidade mínima em grafos bipartidos. A complexidade polinomial do método Húngaro foi primeiro considerada $\mathcal{O}(n^4)$, tendo sido reduzida para $\mathcal{O}(n^3)$ por Lawler [89] (ver também Derigs [46, página 135]).

Das implementações de domínio público disponíveis para o problema de afectação, refira-se a implementação de Dell'Amico, Carpaneto e Toth escrita em linguagem `Fortran` e disponível na `Netlib` na biblioteca de algoritmos da `ACM`, em `http://www.netlib.org/toms/750`. O código completo destina-se a determinar a resolução do problema do caixeiro viajante assimétrico, mas inclui rotinas que resolvem o problema de afectação, usando o método Húngaro, as quais se encontram descritas por Carpaneto and Toth em [33].

Refira-se, também para o problema de afectação, a implementação `CSA` de Goldberg

e Kennedy [65] escrita em linguagem `C` que se baseia no algoritmo *preflow-push* com *cost-scaling* para o problema de fluxo de custo máximo e se encontra disponível em `http://www.avglab.com/andrew/soft.html`.

Capítulo 5

Conclusões e perspectivas de trabalho futuro

Neste trabalho, abordámos a resolução de uma classe de modelos de optimização, conhecidos por modelos de *programação linear disjuntiva*, que se caracterizam por ser problemas de programação linear cuja região de admissibilidade é união de poliedros. Os problemas de programação linear disjuntiva (incluindo muitos dos problemas de optimização combinatória) são melhor resolvidos por métodos do tipo enumerativo, eventualmente combinados com métodos de planos cortantes. No caso de problemas de optimização combinatória, os cortes mais eficazes são os que constituem facetas do poliedro associado. No caso de problemas sem estrutura identificável, os *cortes mais profundos* propostos em [17] são vistos, em geral, como uma boa opção. Contudo, a sua caracterização obriga à resolução de um adequado problema de optimização cuja resolução algébrica obriga à duplicação de estruturas de dados. Em geral, não são conhecidas simplificações se o problema possuir estrutura.

Mostrámos como é possível gerar cortes mais profundos para problemas com estrutura como é o caso, por exemplo, dos problemas de optimização combinatória. Para a resolução do problema de optimização subjacente à caracterização do corte mais profundo, usámos algoritmos de feixe. O desenvolvimento que efectuámos encontra-se fundamentalmente relacionado com o algoritmo proposto por P. Wolfe na década de 70 [128], que determina o ponto pertencente ao invólucro convexo de um conjunto finito de pontos que está mais próximo da origem, no sentido da norma Euclidiana.

Assim, no Capítulo 2, propôs-se uma generalização do algoritmo de P. Wolfe [128] para qualquer métrica, diferenciável ou não. Tal como aquele, o algoritmo proposto é composto por um ciclo externo e um ciclo interno, termina ao fim de um número finito de iterações e gera a mesma sucessão de pontos quando a métrica é a Euclidiana. A implementação é constituída essencialmente por um conjunto de rotinas escritas em

MATLAB e um conjunto de rotinas escritas em C que inclui chamadas a rotinas do *software* de optimização CPLEX. Utilizaram-se dois conjuntos distintos de aplicações, o primeiro visando a resolução do problema, não necessáriamente proveniente de um problema de geração de cortes disjuntivos, com as normas l_1, l_2 e l_∞ e o segundo, visando a resolução de problemas lineares 0-1 e de partição, que possuem grande relevância prática, como gerador de cortes e com diversos tipos de disjunções. Foi testada a qualidade de diversos cortes disjuntivos, a partir de disjunções mais complexas que a usual disjunção de dois termos $(x_j = 0) \vee (x_j = 1)$. Para a resolução de problemas lineares 0-1, testaram-se cortes que separam a solução óptima da relaxação linear do invólucro convexo da região admissível do problema original e cortes baseados nas restrições de integralidade das variáveis e em cada uma das restrições do problema. Para a resolução de problemas de partição, foram testados cortes baseados na disjunção de Ryan-Foster.

No Capítulo 3, foi elaborado um trabalho de síntese inspirado nos trabalhos de Cornuéjols e Li [43], sobre fechos elementares para programação inteira, e de Letchford [91], sobre cortes disjuntivos para optimização combinatória. No trabalho efectuado demonstrou-se que alguns cortes combinatórios bem conhecidos são válidos para determinadas relaxações disjuntivas do problema de optimização combinatória subjacente. A identificação dessas relaxações permite obter cortes mais profundos de acordo com a metodologia que se apresentou no Capítulo 2. Os cortes combinatórios analisados dizem respeito aos problemas de partição por cliques, corte máximo, subdigrafo acíclico, ordenamento linear, cobertura de conjuntos e caixeiro viajante assimétrico. As provas apresentadas na Secção 3.3 usam argumentos primais e constituem uma alternativa às provas apresentadas em [91], cujo autor usou argumentos duais.

No Capítulo 4, procurou-se pôr em prática algumas das metodologias sugeridas num problema de optimização combinatória, especificamente, na versão assimétrica do problema do caixeiro viajante (ATSP). O facto deste problema ser frequentemente utilizado como plataforma de teste de inúmeros métodos de resolução e a importância da determinação de bons limites inferiores do seu valor óptimo motivaram a sua escolha para testar a implementação do algoritmo de planos cortantes proposto. Assim, foi proposta uma estratégia para obter um limite inferior para o valor óptimo do problema do caixeiro viajante assimétrico, melhor do que o que advém do problema de afectação. O algoritmo proposto é um método de primeira ordem baseado na função de penalidade exponencial que requer apenas a resolução sucessiva de problemas de afectação como subproblemas e baseia-se numa abordagem Lagrangeana e em relaxações lineares disjuntivas definidas pelo invólucro convexo das soluções admissíveis do ATSP. As relaxações baseiam-se no conhecimento de um circuito não Hamiltoniano ou de um clique, que permitem a identificação de um hiperplano separador. Explicou-se como determinar esse circuito ou o clique e como reoptimizar a resolução dos problemas de

afectação inerentes à identificação desse hiperplano, apresentando o desenvolvimento algorítmico necessário á resolução dos problemas de corte mínimo global num grafo dirigido e de afectação. Para a identificação desse hiperplano separador ou corte foi usado o algoritmo modificado de P. Wolfe e a norma l_∞. A implementação do algoritmo é essencialmente constituída por um conjunto de rotinas escritas em MATLAB e um conjunto de rotinas escritas em C, que inclui chamadas a rotinas do *software* de optimização CPLEX, à implementação `GMINDCUT` do algoritmo de Hao e Orlin [74] (escrita em linguagem `C` por Georg Skorobohatyj), que determina o corte global de capacidade mínima num grafo dirigido, e às rotinas que resolvem o problema de afectação, usando o método Húngaro. Estas últimas fazem parte da implementação de Dell'Amico, Carpaneto e Toth (escrita em linguagem `Fortran`) que se destina à resolução do problema do caixeiro viajante assimétrico.

Descreveu-se o problema do corte mínimo global num grafo dirigido, recordando-se o algoritmo de *empurrar-prefluxo* de Goldberg e Tarjan [66] para o problema de determinar o fluxo máximo entre dois vértices designados e o algoritmo de Hao e Orlin [74], que determina o corte global de capacidade mínima num grafo dirigido. Este último baseia-se no algoritmo *DetCorteMin(s)* de Padberg e Rinaldi [105], também recordado, que resolve a variante do problema de corte mínimo onde um vértice está fixo, e que é, essencialmente, uma versão adaptada do algoritmo de *empurrar-prefluxo* de Goldberg e Tarjan.

Descreveu-se também o problema de afectação conjuntamente com o método clássico utilizado para a sua resolução, o método Húngaro (Kuhn [86]). O método usa o algoritmo para determinação do emparelhamento de máxima cardinalidade num grafo bipartido, também recordado, e cujo desenvolvimento algorítmico se baseou em Wolsey [129, Capítulo 4].

Os resultados desta dissertação conduzem-nos às seguintes questões que estabelecem continuidade ao presente trabalho.

A experiência computacional apresentada no Capítulo 2 visou a comparação do comportamento do algoritmo para diferentes normas. No algoritmo generalizado de P. Wolfe, existe grande liberdade na determinação do ponto $\widehat{p}$ (no ínicio do Ciclo Maior). Pensamos que a utilização de diferentes regras heurísticas podem reduzir o número de ciclos maiores do algoritmo, contribuindo para a aceleração da convergência do método. Por exemplo, pensamos que podem ser aplicadas as seguintes regras heurísticas: $\widehat{p} = \arg\min\{\alpha p\colon p \in K^0\}$, $\widehat{p} = \arg\min\left\{\frac{\alpha p - \alpha X}{\|p\|}\colon p \in K^0\right\}$ e $\widehat{p} = \arg\min\{\varphi(p)\colon p \in K^0, \alpha p < \alpha X\}$, em que $\varphi(p)$ é uma estimativa da redução do valor de $\|X\|$ com a inserção de p.

As implementações efectuadas com o objectivo de testar a qualidade de diversos cortes disjuntivos deixam em aberto possibilidades de investigação futura tais como a

modificação dos procedimentos com vista à geração de cortes mais profundos a partir de disjunções mais complexas.

Pensamos que a consideração de melhores regras de fortalecimento para alguns problemas disjuntivos com variáveis inteiras constitui um tópico de investigação importante. As regras de fortalecimento dos coeficientes dos cortes disjuntivos, que foram implementadas nas aplicações descritas no Capítulo 2, somente requerem a informação dual, na forma de custos reduzidos óptimos e variáveis duais óptimas relativas ao grupo de restrições que definem a disjunção. Pensamos que estes procedimentos devem ser melhorados, nomeadamente, os referentes aos cortes *split* [126] e aos cortes disjuntivos derivados a partir de regras de ramificação, em especial os que envolvem as regras propostas por Ryan e Foster [112] para problemas de partição e por Smith, Srinivasan e Thompson [119] e Carpaneto e Toth [32] para o problema do caixeiro viajante assimétrico.

No Capítulo 4, foi proposta uma abordagem Lagrangeana para obter um limite inferior melhorado para o valor óptimo do ATSP relativamente ao obtido unicamente com a resolução do problema de afectação e que usa apenas a resolução de problemas de afectação como subprogramas. O algoritmo proposto requer um estudo computacional mais aprofundado. Na experiência computacional realizada, utilizou-se a regra de redução do parâmetro de penalidade $\rho := \min\{\rho/10, \rho^{1.5}\}$ e a norma l_∞ na determinação do corte mais profundo, que permitiram identificar diversas limitações.

Em primeiro lugar, observou-se que o algoritmo pode tornar-se lento, progredindo com λ's demasiado pequenos. Pensamos que a escolha do parâmetro de penalidade, tem impacto directo no número de iterações. Optou-se pela aplicação da variante da regra de redução desse parâmetro $\rho := \min\{\rho/10, \rho^{1.5}\}$, inspirada em Cominetti e Dussault [39], Dussault [48] e Mongeau e Sartenaer [98], em alternativa à abordagem usual, que consiste na aplicação de um factor constante, por exemplo, 2 ou 10. Pensamos que a aplicação de outras regras de redução baseadas em medidas relativas poderá melhorar os resultados obtidos, acelerando a convergência para melhores limites inferiores do valor óptimo do problema.

Em segundo lugar, observou-se que, em geral, o corte mais profundo entre q_k e P_S^{ASS} é uma desigualdade de ciclo - portanto, menos profunda que a correspondente desigualdade de clique em S. Fica em aberto a questão de identificar para que normas $\|\cdot\|$ essa desigualdade será sempre o corte mais profundo em (4.26) quando $\overline{x}$ é o vector característico de um subcircuito. Analisou-se também o comportamento do algoritmo se, no último passo da iteração genérica k, o vector q_k for substituído por x_k. Neste caso, obtiveram-se algumas desigualdades que não são de circuito, mas também ainda não são desigualdades de clique. Fica também em aberto a questão de saber que tipo de desigualdades se conseguirão gerar se x^k satisfizer todas as restrições de clique e for

ponto extremo de uma relaxação linear de P^{ATSP}.

Como perspectivas de trabalho futuro sugere-se ainda a aplicação directa desta abordagem de geração de planos cortantes baseados em argumentos disjuntivos, em modelos como o problema linear complementar, o modelo de sequenciamento/escalonamento de tarefas e problemas com restrições de acaso (ou distribuição de suporte finito), em alternativa à abordagem que obriga à necessidade de efectuar a sua reformulação como programas lineares inteiros mistos e que implica a introdução de um grande número de variáveis adicionais.

Apêndice A

Fortalecimento do corte $\alpha x \geq \beta$

Toda a desigualdade "$\alpha x \geq \beta$" válida para o poliedro $\overline{K} \equiv \text{cl conv } \left(\cup_{i \in I} K_i\right)$, em que I é algum conjunto finito e todo $K_i \subseteq \mathbb{R}^n$ denota um poliedro, pode ser fortalecida pela desigualdade válida do poliedro

$$P \equiv \text{cl conv } \left(\overline{K} \cap \{x\colon \ x_i \text{ é inteiro}, i = 1, 2, \ldots, p\}\right) \tag{A.1}$$

usando os procedimentos descritos por E. Balas e R. Jeroslow em [18].

Seja "$\alpha x \geqslant \beta$" uma desigualdade válida conhecida para $\overline{K}$, uma relaxação disjuntiva de P. Geralmente, esta desigualdade é um corte disjuntivo que separa P de um dado ponto $\overline{x}$ da relaxação linear de P, denotada por $\overline{P}$. As regras de fortalecimento somente requerem a informação dual, na forma de custos reduzidos óptimos e variáveis duais óptimas relativas a um pequeno grupo de restrições - as restrições que definem a disjunção, e aplicam-se mesmo na presença de um número exponencial de restrições originais.

O próximo lema é uma forma particular de apresentar um resultado conhecido por *princípio disjuntivo*, introduzido num relatório técnico de E. Balas de 1979, publicado em [14], para a geração de desigualdades válidas para poliedros disjuntivos quando existem limites nas variáveis que não só os de não negatividade.

Lema A.1 (Soares [121]) *Seja $K^i \subseteq \mathbb{R}^n$, para $i \in I$ finito, um poliedro, e sejam $L \cup \overline{L} = U \cup \overline{U}$ duas partições do conjunto de índices $\{1, 2, \ldots, n\}$ tais que, para cada $i \in I$, existem vectores l^i_L, u^i_U que satisfazem*

$$l^i_L \leqslant x_L \ e \ x_U \leqslant u^i_U, \ \textit{para todo } x \in K^i.$$

Se $(\alpha^i, \beta_i) \in$ polar (K^i) para $i \in I$, então, para $\overline{K} =$ cl conv $\left(\cup_{i \in I} K^i\right)$,

1. *se* $\alpha^i_{\overline{L}} = \alpha_{\overline{L}}$ *para* $i \in I$ *e* $\alpha_L = \max_{i\in I}(\alpha^i_L)$, *então* $(\alpha, \min_{i\in I}(\beta_i + (\alpha_L - \alpha^i_L)l^i_L)) \in$ *polar* $(\overline{K})$;

2. *se* $\alpha^i_{\overline{U}} = \alpha_{\overline{U}}$ *para* $i \in I$ *e* $\alpha_U = \min_{i\in I}(\alpha^i_U)$, *então* $(\alpha, \min_{i\in I}(\beta_i - (\alpha^i_U - \alpha_U)u^i_U)) \in$ *polar* $(\overline{K})$.

Demonstração:

1. Para todo $k \in I$ e $x \in K^k$ qualquer,

$$\begin{aligned}
\alpha x &= \alpha_{\overline{L}} x_{\overline{L}} + \alpha_L x_L \\
&= \alpha^k_{\overline{L}} x_{\overline{L}} + \alpha^k_L x_L + (\alpha_L - \alpha^k_L)x_L \\
&= \alpha^k x + (\alpha_L - \alpha^k_L)l^k_L + (\alpha_L - \alpha^k_L)(x_L - l^k_L) \\
&\geqslant \beta_k + (\alpha_L - \alpha^k_L)l^k_L \\
&\geqslant \min_{i\in I}\{\beta_i + (\alpha_L - \alpha^i_L)l^i_L\}.
\end{aligned}$$

2. Para todo $k \in I$ e $x \in K^k$ qualquer,

$$\begin{aligned}
\alpha x &= \alpha_{\overline{U}} x_{\overline{U}} + \alpha_U x_U \\
&= \alpha^k_{\overline{U}} x_{\overline{U}} + \alpha^k_U x_U + (\alpha^k_U - \alpha_U)(-x_U) \\
&= \alpha^k x - (\alpha^k_U - \alpha_U)u^k_U + (\alpha^k_U - \alpha_U)(u^k_U - x_L) \\
&\geqslant \beta_k - (\alpha^k_U - \alpha_U)u^k_U \\
&\geqslant \min_{i\in I}\{\beta_i - (\alpha^i_U - \alpha_U)u^i_U\}.
\end{aligned}$$

□

Se $L = \{1, 2, \ldots, n\}$ e $l^i = 0$ para todo $i \in I$, então o Item 1 do Lema A.1 assume a forma do princípio disjuntivo: $(\max_{i\in I}(\alpha^i), \min_{i\in I}(\beta_i)) \in$ polar $(\overline{K})$.

O teorema seguinte generaliza [13, Teorema 2.2] para problemas lineares 0-1 mistos e evidencia a possibilidade de fortalecer a desigualdade válida para $\overline{K}$ em duas direcções distintas. Neste resultado, o conjunto $\overline{P}$ denota a relaxação linear de P que se assume ser definida por

$$\overline{P} \equiv \{x \in \mathbb{R}^n \colon Ax - b \in M^*, l \leqslant x \leqslant u\},$$

em que M^* é um cone poliedral. O conjunto $\overline{K}$ é definido por

$$\overline{K} \equiv \text{cl conv}\left(\left(\overline{P} \cap \{x \colon x \leqslant z_0\}\right) \cup \left(\overline{P} \cap \{x \colon x \geqslant z_0 + 1\}\right)\right),$$

em que z_0 é um escalar inteiro satisfazendo $l_j \leqslant z_0 < z_0 + 1 \leqslant u_j$ para algum $j \in \{1, 2, \ldots, p\}$.

Teorema A.1 (Soares [121]) *Sejam $\overline{c}^0, \overline{c}^1 \in \mathbb{R}^n$ os custos reduzidos do último quadro Simplex na resolução dos seguintes problemas lineares, supostos admissíveis e com valor óptimo finito,*

$$\begin{array}{rlll} \beta_0 = & \min & \alpha x & \\ & s.a & x \in \overline{P}, & \text{(A.2)} \\ & & x \leqslant z_0, & \end{array} \qquad \begin{array}{rlll} \beta_1 = & \min & \alpha x & \\ & s.a & x \in \overline{P}, & \text{(A.3)} \\ & & x \geqslant z_0 + 1, & \end{array}$$

A desigualdade $\overline{\alpha} x \geqslant \overline{\beta}$ caracterizada por

$$\begin{array}{rcl} \overline{\alpha}_{\overline{L}} & = & \alpha_{\overline{L}}, \\ \overline{\alpha}_{L_j} & = & \alpha_{L_j} - \max\left\{\max\{\overline{c}^0_{L_j}, 0\} + \min(\overline{c}^0_j, 0)\lceil \overline{m}_{L_j} \rceil, \max\{\overline{c}^1_{L_j}, 0\} + \max(\overline{c}^1_j, 0)\lfloor \overline{m}_{L_j} \rfloor\right\}, \\ \overline{\alpha}_j & = & \alpha_j - \max\left\{\max\{\overline{c}^0_j, 0\} + \min(\overline{c}^0_j, 0)\lceil \overline{m}_j \rceil, \max(\overline{c}^1_j, 0)\lfloor \overline{m}_j \rfloor\right\}, \\ \overline{\beta} & = & \min\{\beta_0, \beta_1\} - (\alpha_L - \overline{\alpha}_L) l_L, \end{array}$$

e a desigualdade $\widetilde{\alpha} x \geqslant \widetilde{\beta}$ caracterizada por

$$\begin{array}{rcl} \widetilde{\alpha}_{\overline{U}} & = & \alpha_{\overline{U}}, \\ \widetilde{\alpha}_{U_j} & = & \alpha_{U_j} - \min\left\{\min\{\overline{c}^0_{U_j}, 0\} + \min(\overline{c}^0_j, 0)\lfloor \widetilde{m}_{U_j} \rfloor, \min\{\overline{c}^1_{U_j}, 0\} + \max(\overline{c}^1_j, 0)\lceil \widetilde{m}_{U_j} \rceil\right\}, \\ \widetilde{\alpha}_j & = & \alpha_j - \min\left\{\min(\overline{c}^0_j, 0)\lfloor \widetilde{m}_j \rfloor, \min\{\overline{c}^1_j, 0\} + \max(\overline{c}^1_j, 0)\lceil \widetilde{m}_j \rceil\right\}, \\ \widetilde{\beta} & = & \min\{\beta_0, \beta_1\} + (\alpha_U - \widetilde{\alpha}_U) u_U, \end{array}$$

onde

$$\begin{array}{rclrcl} L & = & \{1, 2, \ldots, p\} \cap \{k \colon l_k > -\infty\}, & U & = & \{1, 2, \ldots, p\} \cap \{k \colon u_k < +\infty\}, \\ L_j & = & L \setminus \{j\}, & U_j & = & U \setminus \{j\}, \\ \overline{L} & = & \{1, 2, \ldots, n\} \setminus L, & \overline{U} & = & \{1, 2, \ldots, n\} \setminus U, \\ \overline{m}_{L_j} & = & \frac{\max(\overline{c}^0_{L_j}, 0) - \max(\overline{c}^1_{L_j}, 0)}{\max(\overline{c}^1_j, 0) - \min(\overline{c}^0_j, 0)}, & \widetilde{m}_{U_j} & = & \frac{\min(\overline{c}^0_{U_j}, 0) - \min(\overline{c}^1_{U_j}, 0)}{\max(\overline{c}^1_j, 0) - \min(\overline{c}^0_j, 0)}, \\ \overline{m}_j & = & \frac{\max(\overline{c}^1_j, 0)}{\max(\overline{c}^1_j, 0) - \min(\overline{c}^0_j, 0)}, & \widetilde{m}_j & = & \frac{-\min(\overline{c}^1_j, 0)}{\max(\overline{c}^1_j, 0) - \min(\overline{c}^0_j, 0)}, \end{array}$$

são válidas para P.

Demonstração: Começa-se por obter uma caracterização de α, β_0 e β_1 em termos das variáveis duais óptimas dos problemas lineares (A.2) e (A.3). Sejam $u, v \in M$, o cone dual de M^*, os vectores das variáveis duais óptimas relativamente às restrições "$Ax - b \in M^*$" nos problemas (A.2) e (A.3), respectivamente. As restantes variáveis duais óptimas são as seguintes. Para o problema (A.2): $u_0 \equiv \min(\overline{c}^0_j, 0)$ relativamente

a "$x_j \leqslant z_0$"; $\gamma^1 \equiv \max(\overline{c}^0, 0)$ relativamente a "$l \leqslant x$"; $\gamma^2 \equiv \min(\overline{c}^0, 0)$ relativamente a "$x \leqslant u$", excepto $\gamma^2_j \equiv 0$ (porque é variável dual relativamente à restrição redundante "$x_j \leqslant u_j$"). E, para o problema (A.3): $v_0 \equiv \max(\overline{c}^1_j, 0)$ relativamente a "$x_j \geqslant z_0 + 1$"; $\rho^1 \equiv \max(\overline{c}^1, 0)$ relativamente a "$l \leqslant x$", excepto $\rho^1_j \equiv 0$ (porque é variável dual relativamente à restrição redundante "$x_j \geqslant l_j$"); $\rho^2 \equiv \min(\overline{c}^1, 0)$ relativamente a "$x \leqslant u$". Portanto, a desigualdade $\alpha x \geqslant \min\{\beta_0, \beta_1\}$, que é válida para $\overline{P}_j$, é caracterizada por

$$\begin{aligned} \alpha &= uA + u_0 e^j + \gamma^1 + \gamma^2 \quad (= vA + v_0 e^j + \rho^1 + \rho^2), \\ \beta_0 &= ub + u_0 z_0 + \gamma^1 l + \gamma^2 u, \\ \beta_1 &= vb + v_0(z_0 + 1) + \rho^1 l + \rho^2 u. \end{aligned}$$

Considere-se o conjunto de índices $L \equiv \{1, 2, \ldots, p\} \cap \{k\colon l_k > -\infty\}$ e $\overline{L} \equiv \{1, 2, \ldots, n\} \setminus L$. Seja $d \in \mathbb{Z}^n$ tal que $d_{\overline{L}} = 0$ e

$$-\frac{\rho^1_L}{v_0} \leqslant d_L \leqslant -\frac{\gamma^1_L}{u_0}. \tag{A.4}$$

A desigualdade $\overline{\alpha}^0 x \geqslant \overline{\beta}_0$ definida por $\overline{\alpha}^0_{\overline{L}} = \alpha_{\overline{L}}$ e

$$\begin{array}{lll} \overline{\alpha}^0_L & = uA_{\cdot L} + u_0(e^j_L - d_L) + \gamma^2_L & = \alpha_L - (\gamma^1_L + u_0 d_L) \\ \overline{\beta}_0 & = ub + u_0(z_0 - d_L l_L) + \gamma^1_{\overline{L}} l_{\overline{L}} + \gamma^2 u & = \beta_0 - (\gamma^1_L + u_0 d_L) l_L \end{array}$$

é válida para o poliedro

$$\overline{P} \cap \{x\colon x_j - dx \leqslant z_0 - d_L l_L\}, \tag{A.5}$$

e a desigualdade $\overline{\alpha}^1 x \geqslant \overline{\beta}_1$ definida por $\overline{\alpha}^1_{\overline{L}} = \alpha_{\overline{L}}$ e

$$\begin{array}{lll} \overline{\alpha}^1_L & = vA_{\cdot L} + v_0(e^j_L - d_L) + \rho^2_L & = \alpha_L - (\rho^1_L + v_0 d_L) \\ \overline{\beta}_1 & = vb + v_0(z_0 + 1 - d_L l_L) + \rho^1_{\overline{L}} l_{\overline{L}} + \rho^2 u & = \beta_1 - (\rho^1_L + v_0 d_L) l_L \end{array}$$

é válida para o poliedro

$$\overline{P} \cap \{x\colon x_j - dx \geqslant z_0 + 1 - d_L l_L\}. \tag{A.6}$$

Atendendo a (A.4), $\alpha_L - \max(\overline{\alpha}^0_L, \overline{\alpha}^1_L) = \min(\gamma^1_L + u_0 d_L, \rho^1_L + v_0 d_L) \geqslant 0$. Por isso, usando o Item 1 do Lema A.1, a desigualdade $\overline{\alpha} x \geqslant \overline{\beta}$ definida por $\overline{\alpha}_{\overline{L}} = \alpha_{\overline{L}}$ e

$$\begin{aligned}
\overline{\alpha}_L &= \max(\overline{\alpha}^0_L, \overline{\alpha}^1_L) \\
&= \alpha_L - \min(\gamma^1_L + u_0 d_L, \rho^1_L + v_0 d_L) \\
\overline{\beta} &= \min\{\overline{\beta}_0 + (\overline{\alpha}_L - \overline{\alpha}^0_L) l_L, \overline{\beta}_1 + (\overline{\alpha}_L - \overline{\alpha}^1_L) l_L\} \\
&= \min\{\beta_0 - (\gamma^1_L + u_0 d_L) l_L + (\overline{\alpha}_L - \overline{\alpha}^0_L) l_L, \beta_1 - (\rho^1_L + v_0 d_L) l_L + (\overline{\alpha}_L - \overline{\alpha}^0_L) l_L\} \\
&= \min\{\beta_0 - (\overline{\alpha}^0_L - \alpha_L + \overline{\alpha}_L - \overline{\alpha}^0_L) l_L, \beta_1 - (\overline{\alpha}^1_L - \alpha_L + \overline{\alpha}_L - \overline{\alpha}^1_L) l_L\} \\
&= \min\{\beta_0, \beta_1\} - (\overline{\alpha}_L - \alpha_L) l_L
\end{aligned} \tag{A.7}$$

é válida para ambos os poliedros (A.5) e (A.6). Para cada $k \in L$, pode-se escolher o valor para d_k que torna máximo o valor de $\min(\gamma^1_k + u_0 d_k, \rho^1_k + v_0 d_k)$. Pode mostrar-se que esse valor maximizante satisfaz (A.4) e é $\lceil \overline{m}_k \rceil$ ou $\lfloor \overline{m}_k \rfloor$ com

$$\overline{m}_k = \frac{\gamma^1_k - \rho^1_k}{v_0 - u_0},$$

onde se destaca que $v_0 - u_0 > 0$ se as restrições "$x_j \leqslant z_0$" e "$x_j \geqslant z_0 + 1$" não forem ambas redundantes nos problemas (A.2) e (A.3), respectivamente. O valor máximo é $\max\{\gamma^1_k + u_0 \lceil \overline{m}_k \rceil, \rho^1_k + v_0 \lfloor \overline{m}_k \rfloor\}$, terminando assim a primeira parte da demonstração.

Considere-se o conjunto de índices $U \equiv \{1, 2, \ldots, p\} \cap \{k\colon u_k < +\infty\}$ e $\overline{U} \equiv \{1, 2, \ldots, n\} \setminus U$. Seja $d \in \mathbb{Z}^n$ tal que $d_{\overline{U}} = 0$ e

$$-\frac{\gamma^2_U}{u_0} \leqslant d_U \leqslant -\frac{\rho^2_U}{v_0}. \tag{A.8}$$

A desigualdade $\widetilde{\alpha}^0 x \geqslant \widetilde{\beta}_0$ definida por $\widetilde{\alpha}^0_{\overline{U}} = \alpha_{\overline{U}}$ e

$$\begin{array}{lll}
\widetilde{\alpha}^0_U & = uA_{.U} + u_0(e^j_U - d_U) + \gamma^1_U & = \alpha_U - (\gamma^2_U + u_0 d_U) \\
\widetilde{\beta}_0 & = ub + u_0(z_0 - d_U u_U) + \gamma^1 l + \gamma^2_{\overline{U}} u_{\overline{U}} & = \beta_0 - (\gamma^2_U + u_0 d_U) u_U
\end{array}$$

é válida para o poliedro

$$\overline{P} \cap \{x\colon x_j - dx \leqslant z_0 - d_U u_U\}, \tag{A.9}$$

e a desigualdade $\widetilde{\alpha}^1 x \geqslant \widetilde{\beta}_1$ definida por $\widetilde{\alpha}^1_{\overline{U}} = \alpha_{\overline{U}}$ e

$$\begin{array}{lll}
\widetilde{\alpha}^1_U & = vA_{.U} + v_0(e^j_U - d_U) + \rho^1_L & = \alpha_U - (\rho^2_U + v_0 d_U) \\
\widetilde{\beta}_1 & = vb + v_0(z_0 + 1 - d_U u_U) + \rho^1 l + \rho^2_{\overline{U}} u_{\overline{U}} & = \beta_1 - (\rho^2_U + v_0 d_U) u_U
\end{array}$$

é válida para o poliedro

$$\overline{P} \cap \{x\colon x_j - dx \leqslant z_0 + 1 - d_U u_U\}. \tag{A.10}$$

Atendendo a (A.8),

$$\alpha_U - \min\left(\widetilde{\alpha}_U^0, \widetilde{\alpha}_U^1\right) = \max\left(\gamma_U^2 + u_0 d_U, \rho_U^2 + v_0 d_U\right) \leqslant 0.$$

Por isso, usando o Item 2 do Lema A.1, a desigualdade $\widetilde{\alpha}x \geqslant \widetilde{x}$ definida por $\widetilde{\alpha}_{\overline{U}} = \alpha_{\overline{U}}$ e

$$\begin{aligned}
\widetilde{\alpha}_U &= \min(\widetilde{\alpha}_U^0, \widetilde{\alpha}_U^1) \\
&= \alpha_U - \max\left(\gamma_U^2 + u_0 d_U, \rho_U^2 + v_0 d_U\right) \\
\widetilde{\beta} &= \min\{\widetilde{\beta}_0 + (\widetilde{\alpha}_U - \widetilde{\alpha}_U^0)u_U, \widetilde{\beta}_1 + (\widetilde{\alpha}_U - \widetilde{\alpha}_U^1)u_U\} \\
&= \min\{\widetilde{\beta}_0 - (\gamma_U^2 + u_0 d_U)u_U + (\widetilde{\alpha}_U - \widetilde{\alpha}_U^0)u_U, \widetilde{\beta}_1 - (\rho_U^2 + v_0 d_U)u_U + (\widetilde{\alpha}_U - \widetilde{\alpha}_U^1)u_U\} \\
&= \min\{\widetilde{\beta}_0 - (\widetilde{\alpha}_U^0 - \alpha_U + \widetilde{\alpha}_U - \widetilde{\alpha}_U^0)u_U, \widetilde{\beta}_1 - (\widetilde{\alpha}_U^1 - \alpha_U + \widetilde{\alpha}_U - \widetilde{\alpha}_U^1)u_U\} \\
&= \min\{\beta_0, \beta_1\} + (\alpha_U - \widetilde{\alpha}_U)u_U
\end{aligned}$$

é válida para ambos os poliedros (A.9) e (A.10). Para cada $k \in U$, pode-se escolher o valor para d_k que torna mínimo o valor de $\max\left(\gamma_k^2 + u_0 d_k, \rho_k^2 + v_0 d_k\right)$. Pode mostrar-se que esse valor minimizante satisfaz (A.8) e é $\lceil \widetilde{m}_k \rceil$ ou $\lfloor \widetilde{m}_k \rfloor$, com

$$\widetilde{m}_k = \frac{\gamma_k^2 - \rho_k^2}{v_0 - u_0}.$$

O valor mínimo é $\min\left\{\gamma_k^2 + u_0 \lfloor \widetilde{m}_k \rfloor, \rho_k^2 + v_0 \lceil \widetilde{m}_k \rceil\right\}$, que termina assim a demonstração. □

Após o fortalecimento descrito, o valor de β no corte "$\alpha x \geqslant \beta$" pode ainda ser mais fortalecido, usando um procedimento semelhante ao procedimento "*strong branching*" (ver [80, pg. 252]) através de

$$\beta := \min_{i=0,1} \left(\max_{k=1,2,\ldots,p} \left(\begin{array}{ll} \min & \alpha x \\ \text{t.q.} & x \in \overline{P}, x_j = i, \\ & x_k \in \{0,1\} \end{array} \right) \right). \tag{A.11}$$

Realça-se este facto porque a resolução de todos estes problemas lineares pode ser feita de forma muito eficiente, uma vez que há conhecimento de bases avançadas em cada reoptimização através do método Dual-Simplex.

Apêndice B

Regra de Ryan-Foster

Na resolução de problemas de partição através do método *Branch-and-bound*, Ryan e Foster [112] sugeriram uma regra específica de ramificação, e Ceria e Pataki [35] usaram o mesmo princípio para gerar cortes disjuntivos. Os problemas de partição têm o seguinte aspecto:

$$\begin{array}{lllll} \min & cx & = & \min & \displaystyle\sum_{j=1}^{n} c_j x_j \\ \text{s.a} & Ax = e, & & \text{s.a} & \displaystyle\sum_{j \in S_i} x_j = 1, \quad i = 1, 2, \ldots, m, \\ & x \in \{0,1\}^n, & & & x_j \in \{0,1\}, \quad j = 1, 2, \ldots, n, \end{array} \tag{B.1}$$

onde $A \in \{0,1\}^{m \times n}$, e denota um vector-coluna de tudo-uns e $S_i = \{j \colon a_{ij} = 1\}$ denota o *conjunto de suporte da linha i da matriz A*.

A regra de Ryan-Foster [112] baseia-se na observação de que, dada uma solução fraccionária $\overline{x}$, ponto extremo de uma qualquer relaxação linear $\overline{P} \subseteq [0,1]^n$ da região admissível P do problema (B.1), é sempre possível determinar um conjunto não singular de restrições que são cobertas fraccionariamente pelas variáveis de dois conjuntos complementares entre si. Isto é, se $\overline{x}$ é um ponto extremo fraccionário de $\overline{P}$, então existe $I \subseteq \{1, 2, \ldots, m\}$ tal que

$$0 < \sum_{k \in T_I} \overline{x}_k < 1, \tag{B.2}$$

onde $T_I \equiv \cap_{i \in I} S_i$. Então, $\overline{x}$ não pertence ao seguinte poliedro, que contém P, definido por

$$\begin{aligned} & \text{conv } \left(\left(\overline{P} \cap \{x\colon \sum_{k \in T_I} x_k = 0\} \right) \cup \left(\overline{P} \cap \{x\colon \sum_{k \in T_I} x_k = 1\} \right) \right) \\ = \; & \text{conv } \left(\left(\overline{P} \cap \{x\colon x_k = 0, k \in T_I\} \right) \cup \left(\overline{P} \cap \{x\colon x_k = 0, k \in \overline{T}_I\} \right) \right), \end{aligned} \tag{B.3}$$

onde $\overline{T}_I \equiv (\cup_{i \in I} S_i) \setminus T_I$. Pode-se ver a demonstração da existência de pelo menos um conjunto I de cardinalidade dois em Vance [124, Proposição 2.2], por exemplo.

Para ilustração da regra de Ryan-Foster, considere-se o seguinte exemplo com 9 variáveis e 5 restrições:

$$\begin{array}{lccccccccccc} \min & x_1 & +x_2 & +x_3 & +x_4 & +x_5 & +x_6 & +x_7 & +x_8 & +x_9 & & \\ \text{s.a} & x_1 & +x_2 & & +x_4 & +x_5 & & & +x_8 & & = & 1 \\ & & x_2 & +x_3 & +x_4 & & & & +x_8 & +x_9 & = & 1 \\ & x_1 & & +x_3 & & & +x_6 & +x_7 & & & = & 1 \\ & x_1 & & +x_3 & & +x_5 & +x_6 & +x_7 & & +x_9 & = & 1 \\ & x_1 & +x_2 & +x_3 & & +x_5 & +x_6 & +x_7 & & & = & 1 \\ & x_1, & x_2, & x_3, & x_4, & x_5, & x_6, & x_7, & x_8, & x_9 & \in & \{0,1\} \end{array}$$

A solução óptima $\overline{x}$ da relaxação linear do problema é caracterizada por zeros, excepto para $\overline{x}_1 = \overline{x}_3 = \overline{x}_4 = 1/2$. Existem sete pares de linhas que verificam a relação (B.2): $(1,3)$, $(2,3)$, $(1,4)$, $(2,4)$, $(2,5)$, $(1,2)$ e $(1,5)$. Se se escolher, por exemplo, o par de linhas $(1,5)$, então as restrições de ramificação são $x_1 + x_2 + x_5 = 0$ e $x_1 + x_2 + x_5 = 1$. A implementação dessas restrições por eliminação de variáveis equivale a eliminar x_1, x_2, x_5 e x_3, x_4, x_6, x_7, x_8, respectivamente.

Em geral, dada uma solução fraccionária $\overline{x}$ da relaxação linear do problema (B.1), existem vários conjuntos de restrições que podem ser utilizados para definir a disjunção em (B.3). A regra de prioridade estabelecida consiste em ordenar os conjuntos identificados pelo maior valor da função definida por

$$f(I) = \sum_{k \in T_I} \overline{x}_k < 1,$$

em que $T_I \equiv \cap_{i \in I} S_i$ para $I \subseteq \{1, 2, \ldots, m\}$.

Apêndice C

Fluxos e prefluxos

Teorema C.1 (Teorema do Fluxo Máximo-Corte Mínimo, Ford e Fulkerson, 1956 [57]) *Seja $G = (V, E)$ um grafo dirigido com capacidades $u_{ij} \geqslant 0$ associadas a cada arco $(i, j) \in E$, e sejam $s, t \in V$ dois vértices distintos. O valor do fluxo máximo de s para t é igual à capacidade do corte $s - t$ de capacidade mínima.*

O Teorema do Fluxo Máximo-Corte Mínimo pode demonstrar-se usando a dualidade forte da Programação Linear e a teoria da unimodularidade total. De facto, o problema do fluxo máximo de s para t é o seguinte problema linear

$$\begin{array}{rlll} f(s,t) = & \max & v & \\ & \text{s.a} & \displaystyle\sum_{j \in \delta^-(i)} x_{ji} - \sum_{j \in \delta^+(i)} x_{ij} + \left\{ \begin{array}{rl} v & \text{se } i = s \\ -v & \text{se } i = t \end{array} \right\} = 0, & i \in V, \\ & & 0 \leqslant x \leqslant u, & \end{array} \tag{C.1}$$

e o seu dual é o seguinte problema linear

$$\begin{array}{rlll} f(s,t) = & \min & \displaystyle\sum_{(i,j) \in E} u_{ij} w_{ij} & \\ & \text{s.a} & -\gamma_i + \gamma_j + w_{ij} \geqslant 0, & (i,j) \in E, \\ & & \gamma_s - \gamma_t = 1, & \\ & & w \geqslant 0. & \end{array} \tag{C.2}$$

Usando a teoria da unimodularidade total, mostra-se que este último problema é uma formulação exacta (isto é, sem fosso dual) do problema do corte mínimo entre s e t (veja-se [129, página 43], por exemplo) definido por

$$\begin{array}{rlll} f(s,t) = & \min & u(D) \equiv \displaystyle\sum_{(i,j)\in\delta^+(D)} u_{ij} & \\ & \text{s.a} & s \in D, \quad t \in V \setminus D, & \end{array} \tag{C.3}$$

onde $\delta^+(D)$ designa um corte $s-t$ de G. Daqui se pode concluir o Teorema C.1.

Teorema C.2 *Seja $G=(V,E)$ um grafo dirigido com capacidades $u_{ij} \geqslant 0$ associadas a cada arco $(i,j) \in E$, e sejam $s,t \in V$ dois vértices distintos. O valor do prefluxo máximo de s para t é igual ao valor do fluxo máximo de s para t.*

Demonstração: Usando o conceito de excesso de fluxo do vértice $i \in V$ definido na Secção 4.4.1, pode-se escrever o valor $f(s,t)$ do fluxo máximo de s para t em G como

$$\begin{array}{rlll} f(s,t) = & \max & e_t(x) & \\ & \text{s.a} & e_i(x) = 0, \quad i \in V \setminus \{s,t\}, & \\ & & 0 \leqslant x \leqslant u, & \end{array} \tag{C.4}$$

enquanto que o valor $p(s,t)$ do prefluxo máximo de s para t em G se pode escrever como

$$\begin{array}{rlll} p(s,t) = & \max & e_t(x) & \\ & \text{s.a} & e_i(x) \geqslant 0, \quad i \in V \setminus \{s,t\}, & \\ & & 0 \leqslant x \leqslant u. & \end{array} \tag{C.5}$$

Trivialmente, tem-se $f(s,t) \leqslant p(s,t)$ porque a região admissível do problema (C.4) está contida na de (C.5).

Seja $\overline{x}$ o prefluxo máximo de s para t. Construir um novo grafo $G'=(V,E')$, expandindo G com a inserção dos arcos (i,s) para todo $i \neq t$ com capacidade $u_{is} = +\infty$. O novo grafo pode ter arcos duplicados. Agora, considerar o problema de fluxo máximo de s para t em G'

$$\begin{array}{rlll} g(s,t) = & \max & e_t(y) & \\ & \text{s.a} & e_i(y) - y_{is} = 0, \quad i \in V \setminus \{s,t\}, & \\ & & 0 \leqslant y \leqslant u. & \end{array} \tag{C.6}$$

Defina-se $\overline{y}$ tal que $\overline{y}_{ij} = \overline{x}_{ij}$ para todo $(i,j) \in E$ e $\overline{y}_{ij} = e_i(\overline{x})$ para todo $(i,j) \in \overline{E} \setminus E$. Como $\overline{y}$ é admissível no problema (C.6), conclui-se que $p(s,t) \leqslant g(s,t)$. Reciprocamente, qualquer solução admissível y do problema (C.6) se converte numa solução admissível x do problema (C.5) definida por $x_{ij} = y_{ij}$ para todo $(i,j) \in E$. Portanto, $g(s,t) \leqslant p(s,t)$, pelo que $g(s,t) = p(s,t)$.

Pelo Teorema do Fluxo Máximo-Corte Mínimo, o corte $s-t$ mínimo $\delta^+(D)$, correspondente ao problema (C.4), possui valor $f(s,t) = \sum_{(i,j)\in\delta^+(D)} u_{ij}$. Mas, o corte $\delta^+(D)$

de G também é um corte $s-t$ de G' porque $s \in D, t \in V \setminus D$. Por isso, tem-se $p(s,t) = g(s,t) \leqslant f(s,t)$. □

De seguida, apresenta-se uma demonstração construtiva do Teorema C.2. Na demonstração que se fez, não se explica como obter um fluxo óptimo para o problema (C.4) a partir do conhecimento de um prefluxo óptimo para o problema (C.5). A próxima demonstração inclui um algoritmo que é uma adaptação do algoritmo descrito em [1, pág. 79] e usado na demonstração do Teorema da Decomposição de Fluxo.

Demonstração: (alternativa do Teorema C.2) Seja x um prefluxo óptimo de s para t que não é fluxo; isto é, existe pelo menos um $i \in V \setminus \{s,t\}$ tal que $e_i(x) > 0$. Note-se que $x_{ti} = 0$ para todo $(t,i) \in E$, pois, caso contrário, x não seria óptimo. Considere-se o seguinte algoritmo de modificação de x:

Passo 0: Se x for um fluxo de s para t, então parar; senão faça-se $i_0 = s$ e $k = 1$.

Passo 1: Seja $i_k \in V \setminus \{t\}$ tal que $(i_{k-1}, i_k) \in E$ e $x_{i_{k-1},i_k} > 0$. Se $i_k = i_l$ para algum $l < k$, então seguir para o Passo 3 com

$$W := \{(i_l, i_{l+1}), (i_{l+1}, i_{l+2}), \ldots, (i_{k-1}, i_k)\},$$

um ciclo orientado simples. Se $e_{i_k}(x) > 0$, então seguir para o Passo 2 com

$$P := \{(i_0, i_1), (i_1, i_2), \ldots, (i_{k-1}, i_k)\},$$

um caminho orientado de s para i_k; senão, para as situações anteriores, faça-se $k := k+1$ e repita-se o Passo 1.

Passo 2: Defina-se $f(P) := \min\{e_{i_k}(x), \min\{x_{ij} : (i,j) \in P\}\}$, e redefina-se o fluxo x através de $x_{ij} := x_{ij} - f(P)$ para todo $(i,j) \in P$. Voltar ao Passo 0.

Passo 3: Defina-se $f(W) := \min\{x_{ij} : (i,j) \in W\}$, e redefina-se o fluxo x através de $x_{ij} := x_{ij} - f(W)$ para todo $(i,j) \in W$. Voltar ao Passo 0.

Neste algoritmo, de cada vez que se encontra um caminho dirigido P, $e_{i_k}(x)$ fica igual a zero (quando $f(P) = e_{i_k}(x)$) ou x_{ij} fica igual a zero para algum $(i,j) \in P$ (quando $f(P) < e_{i_k}(x)$). Por outro lado, de cada vez que se identifica um ciclo dirigido W, x_{ij} fica igual a zero para algum $(i,j) \in W$. A validade deste algoritmo e sua terminação finita demonstram-se recorrendo ao Teorema da Decomposição de Fluxo (ver [1, Teorema 3.5, página 80], por exemplo). Após um número finito de passagens pelo Passo 0, alcança-se um fluxo x sem que se tenha modificado o valor de $e_t(x)$, pelo que o fluxo alcançado x é óptimo quer para o problema (C.5) quer para o problema (C.4). □

Uma consequência imediata dos Teoremas C.1 e C.2 é o reconhecimento de optimalidade formalizado no corolário seguinte.

Corolário C.1 *Seja $G = (V, E)$ um grafo dirigido com capacidades $u_{ij} \geqslant 0$ associadas a cada arco $(i, j) \in E$, e sejam $s, t \in V$ vértices distintos. Se existir um prefluxo x em G de s para t tal que*

$$e_t(x) = \sum_{(i,j)\in\delta^+(D)} u_{ij}$$

para algum $D \subseteq V$ tal que $s \in D$ e $t \in V \setminus D$, então $\delta^+(D)$ é o corte $s-t$ mínimo.

Considere-se agora uma generalização do problema do corte $s-t$ mínimo. Seja $S \subseteq V$ um conjunto de vértices (origem) de G e $t \in V \setminus S$ um vértice (terminal). O problema de determinar o corte $S-t$ mínimo é o seguinte problema de optimização combinatória

$$\begin{array}{rlll} f(S,t) = & \min & u(D) \equiv \displaystyle\sum_{(i,j)\in\delta^+(D)} u_{ij} & \\ & \text{s.a} & S \subseteq D, & \\ & & t \in V \setminus D. & \end{array} \tag{C.7}$$

Mais uma vez, usando unimodularidade total, mostra-se que uma formulação exacta (isto é, sem fosso dual) deste problema de corte mínimo é

$$\begin{array}{rlll} f(S,t) = & \min & \displaystyle\sum_{(i,j)\in E} u_{ij} w_{ij} & \\ & \text{s.a} & -\gamma_i + \gamma_j + w_{ij} \geqslant 0, & (i,j) \in E, \\ & & \gamma_i - \gamma_t = 1, & i \in S, \\ & & w \geqslant 0. & \end{array} \tag{C.8}$$

cujo dual é o seguinte problema de programação linear

$$\begin{array}{rll} f(S,t) = & \max & \displaystyle\sum_{k\in S} v_k \\ & \text{s.a} & \displaystyle\sum_{j\in\delta^-(i)} x_{ji} - \sum_{j\in\delta^+(i)} x_{ij} + \left\{ \begin{array}{ll} v_i, & \text{se } i \in S \\ -\displaystyle\sum_{k\in S} v_k, & \text{se } i = t \end{array} \right\} = 0, \quad i \in V, \\ & & 0 \leqslant x \leqslant u. \end{array} \tag{C.9}$$

Um *fluxo em G de S para t* é uma solução admissível x para o problema linear (C.9) para algum valor de $v \geqslant 0$. Pode-se escrever o problema (C.9) sem o vector de variáveis v como

$$\begin{array}{lll} \max & e_t(x) & \\ \text{s.a} & e_i(x) = 0, & i \in V \setminus (S \cup \{t\}), \\ & 0 \leqslant x \leqslant u. & \end{array} \tag{C.10}$$

Este desenvolvimento prova o seguinte resultado.

Teorema C.3 *Seja $G = (V, E)$ um grafo dirigido com capacidades $u_{ij} \geqslant 0$ associadas a cada arco $(i, j) \in E$. Sejam $S \subseteq V$ um conjunto de vértices e $t \in V \setminus S$ um vértice. O maior valor do fluxo de S para t em G é igual ao valor do corte $S - t$ mínimo em G.*

O problema do fluxo máximo de S para t em G é redutível a um (clássico) problema do fluxo máximo entre dois vértices. Construa-se $G' = (V', E')$ um grafo dirigido tal que $V' = V \cup \{s'\}$, $E' = E \cup \{(s', i), i \in S\}$, com as mesmas capacidades $u_{ij} \geqslant 0$ para $(i, j) \in E$ e capacidades $u_{ij} = +\infty$ para $(i, j) \in E' \setminus E$. Como é simples de verificar, o maior valor do fluxo de S para t em G coincide com o maior valor do fluxo de s' para t em G'. Mais, todo o corte $s' - t$ mínimo de G' é corte $S - t$ mínimo de G e reciprocamente.

Um *prefluxo em G de S para t* é um vector $x \in \mathbb{R}^m$, satisfazendo $0 \leqslant x \leqslant u$ e tal que $e_i(x) \geqslant 0$ para todo $i \in V \setminus S$. Analogamente ao Teorema C.2, tem-se o seguinte resultado cuja demonstração é trivial, tendo em consideração a redução anterior.

Teorema C.4 *Seja $G = (V, E)$ um grafo dirigido com capacidades $u_{ij} \geqslant 0$ associadas a cada arco $(i, j) \in E$. Sejam $S \subseteq V$ um conjunto de vértices e $t \in V \setminus S$ um vértice. O valor do prefluxo máximo de S para t é igual ao valor do fluxo máximo de S para t.*

Demonstração: O valor $p(S, t)$ do prefluxo máximo de S para t em G é definido por

$$\begin{array}{rlll} p(S,t) = & \max & e_t(x) & \\ & \text{s.a} & e_i(x) \geqslant 0, & i \in V \setminus (S \cup \{t\}), \\ & & 0 \leqslant x \leqslant u. & \end{array} \tag{C.11}$$

Se se introduzirem novas variáveis v_i, para $i \in S$, e novas restrições "$e_i(x) + v_i \geqslant 0$, para $i \in S$", então

$$\begin{array}{rlll} p(S,t) = & \max & e_t(x) & \\ & \text{s.a} & e_i(x) \geqslant 0, & i \in V \setminus (S \cup \{t\}), \\ & & e_i(x) + v_i \geqslant 0, & i \in S, \\ & & 0 \leqslant x \leqslant u, & \\ & & 0 \leqslant v \leqslant +\infty, & \end{array} \tag{C.12}$$

porque as restrições introduzidas são redundantes. Mas, esta é uma formulação do problema de prefluxo máximo de s' para t no grafo $G' = (V', E')$, introduzido num parágrafo anterior (interpretem-se as variáveis v_i como sendo $x_{s'i}$). Pelo Teorema C.2,

pode-se substituir "$\geqslant$" por "$=$", em algumas das restrições do problema (C.12), o que permite concluir que $p(S,t) = f(S,t)$. □

Analogamente ao Corolário C.1, tem-se o seguinte resultado que permite reconhecer a optimalidade.

Corolário C.2 *Seja $G = (V, E)$ um grafo dirigido com capacidades $u_{ij} \geqslant 0$ associadas a cada arco $(i,j) \in E$. Sejam $S \subseteq V$ um conjunto de vértices e $t \in V \setminus S$ um vértice. Se existir um prefluxo x em G de S para t tal que*

$$e_t(x) = \sum_{(i,j)\in\delta^+(D)} u_{ij}$$

para algum $D \subseteq V$ tal que $S \subseteq D$ e $t \in V \setminus D$, então $\delta^+(D)$ é um corte $S-t$ mínimo.

Demonstração: Seja $v_i = \max(0, -e_i(x))$ para $i \in S$. Sabe-se então que (x, v) é um prefluxo de s' para t no grafo $G' = (V', E')$, introduzido num parágrafo anterior. Pelo Corolário C.1, (x, v) é um prefluxo máximo de s' para t em G'. Pelos Teoremas C.1 e C.2, $\delta^+(D)$ é um corte $s'-t$ mínimo em G'. Então, $\delta^+(D)$ é um corte $S-t$ mínimo em G, pois, se não fosse, isso contrariaria o facto de $\delta^+(D)$ ser um corte $s'-t$ mínimo em G'. □

Referências

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Inc., 1993.

[2] M. Akgül. The linear assignment problem. In M. Akgül, H. Hamacher, and S. Tüfekçi, editors, *Combinatorial Optimization: New Frontiers in the Theory and Practice*, number 82 in NATO ASI Series F: Computer and Systems Sciences, pages 85–122. Springer-Verlag, Berlin, 1992.

[3] H. Alt, N. Blum, K. Mehlhorn, and M. Paul. Computing a maximum cardinality matching in a bipartite graph in time $O(n^{1.5}\sqrt{m/\log n})$. *Information Processing Letters*, 37(4):237–240, 1991.

[4] K. Andersen, G. Cornuéjols, and Y. Li. Split closure and intersection cuts. In B. Cook and A. S. Schulz, editors, *Integer Programming and Combinatorial Optimization, 9th International IPCO Conference*, volume 2337 of *Lecture Notes in Computer Science*, pages 127–144, MIT, Boston, USA, May 2002. Springer.

[5] D. Applegate, R. Bixby, V. Chvátal, and W. Cook. Implementing the Dantzig-Fulkerson-Johnson algorithm for large traveling salesman problems. *Mathematical Programming*, 97(1-2, Ser. B):91–153, 2003. ISMP 2003 (Copenhagen).

[6] E. Balas. Intersection cuts - a new type of cutting-planes for integer programming. *Operations Research*, 19:19–39, 1971.

[7] E. Balas. Integer programming and convex analysis: intersection cuts from outer polars. *Mathematical Programming*, 2:330–382, 1972.

[8] E. Balas. Intersection cuts from disjunctive constraints. Technical Report 330, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, PA, Aug 1973. Revised and expanded, Feb 1974.

[9] E. Balas. Disjunctive programming: Properties of the convex hull of feasible points. Technical Report 348, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, PA, Jul 1974.

[10] E. Balas. Disjunctive programming: Cutting planes from logical conditions. In *Nonlinear Programming*, volume 2, pages 279–312. Academic Press, 1975.

[11] E. Balas. Disjunctive programming. *Annals of Discrete Mathematics*, 5:3–51, 1979.

[12] E. Balas. Disjunctive programming and a hierarchy of relaxations for discrete optimization problems. *SIAM Journal on Algebraic and Discrete Methods*, 6(3):466–486, 1985.

[13] E. Balas. Enhancements of lift-and-project. Technical report, Graduated School of Industrial Administration, Carnegie Mellon University, 1997.

[14] E. Balas. Disjunctive programming: Properties of the convex hull of feasible points. *Discrete Applied Mathematics*, 89:3–44, 1998.

[15] E. Balas, V. Bowman, F. Glover, and D. Sommer. An intersection cut from the dual of the unit hypercube. *Operations Research*, 19:40–44, 1971.

[16] E. Balas, S. Ceria, and G. Cornuéjols. A lift-and-project cutting plane algorithm for mixed 0-1 programs. *Mathematical Programming*, 58:295–323, 1993.

[17] E. Balas, S. Ceria, and G. Cornuéjols. Mixed 0-1 programming by lift-and-project in a branch-and-cut framework. *Management Science*, 42(9):1229–1246, 1996.

[18] E. Balas and R. G. Jeroslow. Strengthening cuts for mixed integer programs. *European Journal of Operations Research*, 4:224–234, 1980.

[19] E. Balas and P. Toth. Branch and bound methods. In E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, editors, *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. John Wiley & Sons, 1985.

[20] F. Barahona and A. R. Mahjoub. On the cut polytope. *Mathematical Programming*, 36:157–173, 1986.

[21] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali. *Linear Programming and Network Flows*. John Wiley & Sons, Inc., 2nd edition, 1990.

[22] D. P. Bertsekas. *Network Optimization: Continuous and Discrete Models*. Athena Scientific, Belmont, MA, 1998.

[23] R. E. Bixby, E. A. Boyd, S. S. Dadmerhr, and R. R. Indovina. The MIPLIB Mixed Integer Programming Library. *COAL Bulletin*, 22, 1993.

[24] R. E. Bixby, S. Ceria, C. M. McZeal, and M. W. P. Savelsbergh. An updated mixed integer programming library: MIPLIB 3.0. *Optima*, 58:12–15, 1998.

[25] E. Boyd. Solving integer programs with Fenchel cutting planes and preprocessing. *Neural, Parallel and Scientific Computations*, 1:453–466, 1993.

[26] E. Boyd. Solving 0/1 integer programs with enumeration cutting planes. *Annals of Operations Research*, 50:61–72, 1994.

[27] C. Burdet. Enumerative cuts: I. *Operations Research*, 21:61–89, 1973.

[28] R. E. Burkard and E. Çela. Linear assignment problems and extensions. In D.-Z. Du and P. M. Pardalos, editors, *Handbook of Combinatorial Optimization*, volume 4. Kluwer Academic Publishers, 1999.

[29] J. V. Burke and M. C. Ferris. Characterization of solution sets of convex programs. *Oper. Res. Lett.*, 10(1):57–60, 1991.

[30] A. Caprara and M. Fischetti. Branch-and-cut algorithms. In M. Dell'Amico, F. Maffioli, and S. Martello, editors, *Annotated Bibliographies in Combinatorial Optimization*, pages 45–63. Wiley, Chichester, 1997.

[31] G. Carpaneto, M. Dell'Amico, and P. Toth. Exact solution of large-scale, asymmetric traveling salesman problems. *ACM Transactions on Mathematical Software*, 21(4):394–409, 1995.

[32] G. Carpaneto and P. Toth. Some new branching and bounding criteria for the asymmetric traveling salesman problem. *Management Science*, 26:736–743, 1980.

[33] G. Carpaneto and P. Toth. Primal-dual algorithms for the assignment problem. *Discrete Applied Mathematics*, 18:137–153, 1987.

[34] S. Ceria. *Lift-and-project methods for mixed 0-1 programs*. PhD thesis, Carnegie Mellon University, Pittsburg, Jan 1993.

[35] S. Ceria and G. Pataki. Solving integer and disjunctive programs by lift-and-project. In R. E. Bixby, E. A. Boyd, and R. Z. Rios-Mercado, editors, *Integer Programming and Combinatorial Optimization, 6th International IPCO Conference*, volume 1412 of *Lecture Notes in Computer Science*, pages 22–24, Rice University, Houston, USA, June 1998. Springer.

[36] S. Chopra and M. R. Rao. The partition problem. *Mathematical Programming*, 59:87–115, 1993.

[37] V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4:305–337, 1973.

[38] J. Cirasella, D. S. Johnson, L. A. McGeoch, and W. Zhang. The asymmetric traveling salesman problem: Algorithms, instance generators, and tests. In A. L. Buchsbaum and J. Snoeyink, editors, *Algorithm Engineering and Experimentation, Third International Workshop, ALENEX 2001*, volume 2153 of *Lecture Notes in Computer Science*, pages 32–59, Berlin, 2001. Springer-Verlag.

[39] R. Cominetti and J.-P. Dussault. A stable exponential penalty algorithm with superlinear convergence. *Journal of Optimization Theory and Applications*, 83(2), 1994.

[40] W. Cook and A. Rohe. Computing minimum-weight perfect matchings. *INFORMS Journal on Computing*, 11:138–148, 1999.

[41] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. John Wiley & Sons Inc., 1997.

[42] W. J. Cook, R. Kannan, and A. Schrijver. Chvátal closures for mixed integer programming problems. *Mathematical Programming*, 47:155–174, 1990.

[43] G. Cornuéjols and Y. Li. Elementary closures for integer programs. *Operations Research Letters*, 28:1–8, 2001.

[44] G. Cornuéjols and Y. Li. On the rank of mixed 0-1 polyhedra. *Mathematical Programming*, 91:391–397, 2002.

[45] M. Dell'Amico and P. Toth. Algorithms and codes for dense assignment problems: the state of the art. *Discrete Applied Mathematics*, 100:17–48, 2000.

[46] U. Derigs. *Programming in Networks and Graphs: on the Combinatorial Background and Near–Equivalence of Network Flow and Matching Algorithms*. Springer, Heidelberg, 1988.

[47] DIMACS. *Network Flows and Matching: First DIMACS Implementation Challenge*, volume 12. American Mathematical Society, 1993.

[48] J.-P. Dussault. Augmented penalty algorithms. *IMA Journal on Numerical Analysis*, 18, 1998.

[49] J. Edmonds. Maximum matching and a polyhedron with 0, 1 vertices. *Journal of Research at the National Bureau of Standards (Section B)*, 69B:125–130, 1965.

[50] J. Edmonds. Paths, trees and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.

[51] R. Euler, M. Jünger, and G. Reinelt. Generalizations of cliques, odd cycles and anticycles and their relation to independence system polyhedra. *Mathematics of Operations Research*, 12:451–462, 1987.

[52] S. Even and O. Kariv. An O($n^{2.5}$) algorithm for maximum matching in general graphs. In *16th Annual IEEE Symposium on Foundations of Computer Science*, pages 100–112, Berkeley, 1975. The University of California.

[53] S. Even and R. E. Tarjan. Network flow and testing graph connectivity. *SIAM Journal on Computing*, 4:507–518, 1975.

[54] M. Fischetti. Facets of the asymmetric traveling salesman polytope. *Mathematics of Operations Research*, 16:42–56, 1991.

[55] M. Fischetti, A. Lodi, and P. Toth. Exact methods for the asymmetric traveling salesman problem. In *The traveling salesman problem and its variations*, volume 12 of *Combinatorial Optimization*, pages 169–205. Kluwer Academic Publishers, Dordrecht, 2002.

[56] R. W. Floyd. Algorithm 97: Shortest path. *Communications of the Association for Computing Machinery*, 5(6), 1962.

[57] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[58] H. Gabow. An efficient implementation of Edmonds' maximum matching algorithm. *Journal of the Association for Computing Machinery*, 23:221–234, 1976.

[59] Z. Galil. Efficient algorithms for finding maximum matchings in graphs. *ACM Computing Surveys*, 18(1):23–38, 1986.

[60] Z. Galil, S. Micali, and H. Gabow. An $O(EV\log V)$ algorithm for finding a maximal weighted matching in general graphs. *SIAM Journal on Computing*, 15(1):120–130, 1986.

[61] A. Gibbons. *Algorithmic Graph Theory*. Cambridge University Press, Cambridge, 1985.

[62] F. Glover. Convexity cuts and cut search. *Operations Research*, 21:123–134, 1973.

[63] F. Glover. Polyhedral convexity cuts and negative edge directions. *Zeitschrift fur Operations Research*, 18:181–186, 1974.

[64] F. Glover. Poliedral annexation in mixed integer and combinatorial programming. *Mathematical Programming*, 8:161–188, 1975.

[65] A. Goldberg and R. Kennedy. An efficient cost scaling algorithm for the assignment problem. *Mathematical Programming*, 71:153–178, 1995.

[66] A. V. Goldberg and R. E. Tarjan. A new approach to the maximum-flow problem. *Journal of the Association for Computing Machinery*, 35(4):921–940, 1988.

[67] R. E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64:257–278, 1958.

[68] R. E. Gomory. An algorithm for the mixed integer problem. Technical Report RM-2597, The RAND Corporation, Santa Monica, CA, 1960.

[69] R. E. Gomory. An algorithm for integer solutions to linear programs. In R. Graves and P. Wolfe, editors, *Recents advances in Mathematical Programming*, pages 269–302. McGraw-Hill, 1963.

[70] M. Grötschel, M. Jünger, and G. Reinelt. On the acyclic subgraph polytope. *Mathematical Programming*, 33:28–42, 1985.

[71] M. Grötschel, L. Lovász, and A. Shrijver. *Geometric Algorithms and Combinatorial Optimization*. Algorithms and Combinatorics, 2. Springer-Verlag, 2nd corrected edition, 1993.

[72] M. Grötschel and Y. Wakabayashi. Facets of the clique partitioning polytope. *Mathematical Programming*, 47:367–387, 1990.

[73] G. Gutin and A. P. Punnen, editors. *The Traveling Salesman Problem and Its Variations*. Kluwer Academic Publishers, Boston, 2002.

[74] J. Hao and J. B. Orlin. A faster algorithm for finding the minimum cut in a directed graph. *Journal of Algorithms*, 17:424–446, 1994.

[75] M. Held and R. M. Karp. The traveling salesman problem and minimum spanning trees. *Operations Research*, 18:1138–1162, 1970.

[76] M. Held and R. M. Karp. The traveling salesman problem and minimum spanning trees: Part II. *Mathematical Programming*, 1:6–25, 1971.

[77] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms I: Fundamentals*. Springer-Verlag, Berlin, 1993.

[78] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms II: Advanced Theory and Bundle Methods.* Springer-Verlag, Berlin, 1993.

[79] J. E. Hopcroft and R. M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal of Computing*, 2(4):225–231, 1973.

[80] ILOG Inc. ILOG CPLEX 8.0, User's Manual, 2002.

[81] The Math Works Inc. Using Matlab, 1999.

[82] R. G. Jeroslow. Cutting-plane theory: disjunctive methods. *Annals of Discrete Mathematics*, 1:293–330, 1977.

[83] R. G. Jeroslow. A cutting-plane game for facial disjunctive programs. *SIAM Journal on Control and Optimization*, 18:264–281, 1980.

[84] L. G. Khachiyan. A polynomial algorithm in linear programming (in russian). *Doklady Akademii Nauk SSSR*, 244:1093–1096, 1979. (English translation: Soviet Mathematics Doklady, 20:191-194, 1979).

[85] D. E. Knuth. *The Stanford GraphBase: A Platform for Combinatorial Computing.* Addison-Wesley, 1993. (Graph data available from `ftp://labrea.stanford.edu/pub/sgb/`).

[86] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[87] H. W. Kuhn. Variants of the hungarian method for solving the assignment problem. *Naval Research Logistics Quarterly*, 3:253–258, 1956.

[88] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, editors. *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization.* John Wiley & Sons, Chichester, 1985.

[89] E. S. Lawler. *Combinatorial Optimization: Networks and Matroids.* Dover Publications, Inc., 2001.

[90] G. L. Lawson and R. J. Hanson. *Solving least squares problems.* Society for Industrial & Applied Mathematics, 1995.

[91] A. N. Letchford. Disjunctive cuts for combinatorial optimization. *Journal of Combinatorial Optimization*, 5(3):299–315, 2001.

[92] L. Lovász. *Matching Theory.* Elsevier Science Ltd, 1986.

[93] L. Lovász and M. D. Plummer. *Matching Theory*. North-Holland, Amsterdam, 1986.

[94] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1:166–190, 1991.

[95] S. Martello and P. Toth. Linear assignment problems. *Annals of Discrete Mathematics*, 31:259–282, 1987.

[96] R. B. Mattingly and N. P. Ritchey. Implementing an $O((nm)^{1/2})$ cardinality matching algorithm. In *Network Flows and Matching, DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, volume 12, pages 539–556. American Mathematical Society, 1993.

[97] S. Micali and V. V. Vazirani. An $O((|V|)^{1/2} \cdot |E|)$ algorithm for finding maximum matching in general graphs. In *21st Annual IEEE Symposium on Foundations of Computer Science*, pages 17–27, Syracuse, New York, 1980.

[98] M. Mongeau and A. Sartenaer. Automatic decrease of the penalty parameter in exact penalty function methods. *European Journal of Operational Research*, 83:686–699, 1995.

[99] B. Moret and H. Shapiro. *Algorithms from P to NP - Volume I: Design and Efficiency*. Benjamin/Cummings Publishing Company, Menlo Park, CA, 1991.

[100] R. Müeller and A. S. Schulz. Transitive packing. In W. H. Cunningham, S. T. McCormick, and M. Queyranne, editors, *Integer Programming and Combinatorial Optimization, 5th International IPCO Conference*, volume 1084 of *Lecture Notes in Computer Science*, pages 430–444, University of British Columbia, Vancouver, Canada, June 1996. Springer.

[101] B. A. Murtagh. *Advanced Linear Programming*. McGraw-Hill Book Co., New York, NY, 1981.

[102] J. L. Nazareth. *Computer Solutions of Linear Programs*. Oxford University Press, Oxford, 1987.

[103] G. L. Nemhauser and L. A. Wolsey. A recursive procedure for generating all cuts for 0-1 mixed integer programs. *Mathematical Programming*, 46:379–390, 1990.

[104] J. H. Owen and S. Mehrotra. A disjunctive cutting plane procedure for general mixed-integer linear programs. *Mathematical Programming*, 89(3):437–448, 2001.

[105] M. Padberg and G. Rinaldi. An efficient algorithm for the minimum capacity cut problem. *Mathematical Programming*, 47:19–36, 1990.

[106] M. Padberg and G. Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review*, 33(1):60–100, 1991.

[107] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Dover Pubns, 1998.

[108] S. Poljak and D. Turzík. Max-cut in circulant graphs. *Discrete Mathematics*, 108:379–392, 1992.

[109] A. Ramires and J. Soares. Um melhor limite inferior para o problema do caixeiro viajante assimétrico baseado na afectação. *Investigação Operacional, APDIO*, 2004. Submetido.

[110] A. Ramires and J. Soares. Uma abordagem primal para a geração de planos cortantes disjuntivos mais separadores. *Investigação Operacional, APDIO*, 2004. Submetido.

[111] E. Rothberg. Implementation of Gabow's $O(n^3)$ version of Edmonds' algorithm for unweighted nonbipartite matching. *Computer software*, 1985. `ftp://dimacs.rutgers.edu/pub/netflow/matching/cardinality/solver-1`.

[112] D. M. Ryan and B. A. Foster. An integer programming approach to scheduling. In A. Wren, editor, *Computer scheduling of public transport urban passenger vehicle and crew scheduling,*, pages 269–280, North Holland, Amesterdam, 1981. North-Holland Publishing Company.

[113] A. Schrijver. On cutting planes. In M. Deza and I. G. Rosenberg, editors, *Combinatorics 79 Part II, Annals of Discrete Mathematics 9*, pages 291– 296, North Holland, Amsterdam, 1980.

[114] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, Ltd, 1986.

[115] J. Setubal. Sequential and parallel experimental results with bipartite matching algorithms. Technical Report EC-96-09, Institute of Computing, University of Campinas, Brazil, 1996.

[116] H. Sherali and W. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3:411–430, 1990.

[117] H. Sherali and C. Shetty. Optimization with disjunctive constraints. In M. Beckman and H. Kunzi, editors, *Lecture notes in Economics and Mathematical Systems*, volume 181. Springer-Verlag, 1980.

[118] S. Skiena. *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica.* Addison-Wesley, Redwood City, CA, 1990.

[119] T. H. C. Smith, V. Srinivasan, and G. L. Thompson. Computational performance of three subtour elimination algorithms for solving asymmetric traveling salesman problems. *Annals of Discrete Mathematics*, 1:495–506, 1977.

[120] J. Soares. *Disjunctive Convex Optimization.* PhD thesis, Graduate School of Business, Columbia University, New York, USA, June 1998.

[121] J. Soares, May 2003. Personnal comunication.

[122] M. M. Syslo, N. Deo, and J. S. Kowalik. *Discrete Optimization Algorithms with Pascal Programs.* Prentice Hall, Englewood Cliffs NJ, 1983.

[123] R. E. Tarjan. *Data Structures and Network Algorithms.* Society of Industrial and Applied Mathematics, Philadelphia, PA, 1983.

[124] P. Vance. *Crew Scheduling, Cutting-stock and Column Generation: solving huge integer programs.* PhD thesis, Georgia Institute of Technology, Georgia, Aug 1993.

[125] S. Warshall. A theorem on boolean matrices. *Journal of the Association for Computing Machinery*, 9(1):11–12, 1962.

[126] W.Cook, R. Kannan, and A. Schrijver. Chvátal closures for mixed integer programming problems. *Mathematical Programming*, 47(2):155 – 17, 1990.

[127] D. B. West. *Introduction to graph theory.* Prentice Hall, 2nd edition, 2001.

[128] P. Wolfe. Finding the nearest point in a polytope. *Mathematical Programming*, 11:128–149, 1976.

[129] L. Wolsey. *Integer Programming.* Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, Inc., 1998.

ERRATA

Algoritmos de Feixe para Programação Linear Inteira Mista

Página	**Linha**	**Onde se lê**	**Leia-se**
18	26	subpercursos	subcircuitos
29	14	é elevado.	é elevada.
35	28	$Q^TQ\bar{v} + \lambda e^T$	$Q^TQ\bar{v} - \lambda e^T$
35	30	$\bar{v}^T(Q^TQ\bar{v} + \lambda e^T)$	$\bar{v}^T(Q^TQ\bar{v} - \lambda e^T)$
43	6	o Figura 2.1	o algoritmo da Figura 2.1
61	18	para todo $u_1, u_2 \geq 0$	para todo $u^1, u^2 \geq 0$
67	5	desigualdadee (3.22)	desigualdades (3.22)
68	13-14	$x_{v_5v_1} + x_{v_5v_2} = x_{v_2v_4} + x_{v_4v_1} \leq 2$	$x_{v_5v_1} + x_{v_5v_2} + x_{v_2v_4} + x_{v_4v_1} \leq 2$
69	3	$3p + 2 - x_{v_{2p+3}v_2} \leq 3(p+1)$	$3p + 2 + x_{v_{2p+3}v_2} \leq 3(p+1)$
77	9	subpercursos	subcircuitos
79	9	através relaxações	através de relaxações
79	penúltima	*subpercursos de* G.	*subcircuitos de* G.
80	5	*subpercursos.*	*subcircuitos.*
82	15	$P_k \subset P$.	$P_k \subseteq P$.
84	14	emparelhamento perfeito M,	emparelhamento perfeito M do grafo bipartido $G(V \times V, E)$,
85	2	relaxação disjuntiva	relaxação linear disjuntiva
86	18	poliedros.	poliedros em (4.18).
89	última	emparelhamento perfeito M,	emparelhamento perfeito M do grafo bipartido $G(V \times V, E)$,
109	28	$(i,j)j \in$ para W	(i,j) para $j \in W$
123	26	subpercursos	subcircuitos