

Combinatória Poliédrica e Planos-de-Corte Faciais

Carlos E. Ferreira
Yoshiko Wakabayashi

Departamento de Ciência da Computação
IME – USP

Sumário

Prefácio	3
1 Preliminares	7
1.1 Teoria dos Grafos	7
1.2 Álgebra Linear	9
1 Teoria de Poliedros	13
1.1 Conceitos Básicos	14
1.2 Transformação de Poliedros	15
1.3 Projeção de Poliedros	17
1.4 Dualidade em Programação Linear	25
1.5 Descrição de Poliedros	30
1.5.1 Polaridade de cones	31
1.6 Faces	34
1.7 Dimensão	38
1.8 Facetas	40
1.9 Vértices e Extremais	44
2 Métodos de Planos-de-corte	49
2.1 Fecho Inteiro de um Poliedro	49
2.1.1 Matrizes totalmente unimodulares	50
2.2 Descrição dos Métodos de Planos-de-Corte	55
2.2.1 Provas de planos-de-corte	58
2.2.2 Comprimento de provas de planos-de-corte	59
2.2.3 Cortes de Gomory	61

2.2.4	Cortes faciais	65
2.3	Equivalência entre Otimização e Separação	66
3	Poliedros de Problemas de O.C.	69
3.1	Otimização Combinatória e Métodos Poliédricos	69
3.2	Poliedro dos Emparelhamentos	75
3.3	Poliedro dos Caminhos e Poliedro dos Cortes	84
3.4	Poliedro de um Matróide	88
3.5	Poliedro do Problema do Caixeiro Viajante	93
3.6	Poliedro do Problema de Steiner em Grafos	96
4	Implementação	101
4.1	O Método <i>Branch and Cut</i>	102
4.2	Limitante Inferior Inicial	104
4.3	Estratégias de Separação	104
4.3.1	Algoritmos polinomiais para separação	104
4.3.2	Heurísticas de separação	106
4.3.3	Seleção de inequações encontradas	108
4.3.4	Manutenção do <i>pool</i> de inequações	109
4.4	Heurísticas Primais	109
4.4.1	Fixação de variáveis por custo reduzido	110
4.4.2	Fixação de variáveis por implicações lógicas	111
4.5	A Fase <i>Branch and Bound</i>	111
4.5.1	Como efetuar o <i>branching</i>	111
4.5.2	Buscas na árvore de decisão	112
4.6	O Programa de Programação Linear	113
4.6.1	Eliminação de variáveis	113
4.6.2	Eliminação de inequações	113
4.7	Comentários Finais	114
	Bibliografia	117

Lista de Figuras

1.1	Exemplo de um poliedro no \mathbb{R}^2 .	15
1.2	Exemplo de projeção	17
1.3	Cone polar de S	32
1.4	Vértices, raios e retas extremas.	44
2.1	Método dos planos-de-corte	61
2.2	Relação entre os problemas apresentados	67
3.1	Vértice fracionário de $P(G, Z)$	97
3.2	<i>3-odd hole</i>	100
4.1	Método <i>branch and bound</i>	102
4.2	Partição de Steiner	106
4.3	Fluxograma simplificado de um algoritmo <i>branch and cut</i> .	115

Prefácio

Muitos problemas de otimização combinatória exerceram, e continuam exercendo ainda, um grande fascínio sobre os pesquisadores da área. A tentativa de se resolver alguns desses problemas motivou o surgimento de teorias interessantes, muitas das quais se desenvolveram independentemente das aplicações que a motivaram, e acabaram criando áreas relevantes de pesquisa. Um exemplo clássico é o problema do caixeiro viajante, que teve um papel importante na área de Combinatória Poliédrica.

Essa área tem como base de sustentação, a teoria de poliedros e a de programação linear. Em termos bem amplos, pode-se dizer que em Combinatória Poliédrica estuda-se propriedades de poliedros associados a problemas combinatórios. Tais estudos podem conduzir a relações min-max (fazendo uso de dualidade), ao desenvolvimento de algoritmos, ou revelar certas propriedades ou estruturas do problema em foco. Não raro, algumas dessas estruturas se fazem presentes em vários problemas, conduzindo a especulações e conseqüente desenvolvimento da teoria a respeito desses objetos.

Neste texto discutimos que tipos de problemas de otimização combinatória podem ser estudados através de métodos poliédricos. Mostramos como associar um poliedro a um problema, como obter uma descrição (completa ou parcial) de um tal poliedro, e como desenvolver um algoritmo baseado nessa descrição.

Neste ponto, cabe mencionar que há alguns fatos que nos desencorajam e outros que nos motivam a utilizar essa estratégia. Por um lado, sabemos que, para a classe de problemas de otimização \mathcal{NP} -difíceis, obter uma boa descrição do respectivo poliedro, não é uma tarefa fácil (se fosse, teríamos $\mathcal{P} = \mathcal{NP}$). Por outro lado, uma boa razão para se usar essa estratégia no tratamento desses problemas, é o fato de que essa é uma das poucas estratégias que tem-se mostrado eficaz na resolução de instâncias reais de grande porte. Resultados recentes para o problema do caixeiro viajante, em que instâncias de mais de três mil cidades foram resolvidas aplicando métodos poliédricos, mostram a força dessa estratégia. Uma outra vantagem de se usar tais métodos, é o fato de que podemos derivar soluções aproximadas para os problemas, com garantia de qualidade.

Neste texto, focalizamos os resultados teóricos que consideramos mais relevantes para aqueles que trabalham — ou pretendem trabalhar — na área de otimização combinatória e têm interesse no desenvolvimento de algoritmos que fazem uso de técnicas de combinatória poliédrica. Assim, apresentamos primeiramente alguns resultados da teoria de poliedros, essenciais para este estudo. Como pré-requisitos, supomos apenas alguma familiaridade com programação li-

near (inteira), e alguns conceitos básicos da teoria de grafos. No decorrer do texto, referimos a conceitos de complexidade computacional, mas não os julgamos essenciais para a compreensão dos métodos que serão descritos.

O nosso objetivo é prover o material necessário para uma introdução a essa área, fornecendo a fundamentação teórica e ilustrando, através da descrição de alguns poliedros, o uso de certas técnicas de prova que são importantes neste contexto. Mostramos como os resultados a respeito da descrição facial de um poliedro podem ser usados para o desenvolvimento de um algoritmo *branch and cut*, que é uma combinação do método *branch and bound* com métodos de planos-de-corte faciais.

Dividimos este texto em 5 capítulos. No *Capítulo 1* apresentamos alguns conceitos básicos da teoria de grafos e de álgebra linear, com a finalidade de fixar a terminologia e a notação adotada.

O *Capítulo 1* é dedicado à teoria de poliedros. Diversos resultados importantes são apresentados, como projeção de poliedros, lemas alternativos sobre existência de solução de sistemas lineares, polaridade cônica, formas de descrição de poliedros e caracterização de facetas,

No *Capítulo 2* apresentamos a teoria sobre métodos de planos-de-corte, mostrando alguns resultados clássicos sobre fecho inteiro de poliedros, matrizes unimodulares, sistemas TDI, cortes de Gomory e posto de Chvátal. Mencionamos, sem provar, o resultado fundamental devido a Grötschel, Lovász e Schrijver, mostrando a equivalência entre problemas de otimização e de separação.

Exemplos de poliedros associados a problemas clássicos de otimização combinatória são apresentados no *Capítulo 3*. Incluímos alguns poliedros “bem caracterizados” (poliedro dos emparelhamentos de um grafo, poliedro de um de um matróide), assim como descrições parciais de poliedros de problemas \mathcal{NP} -difíceis (problema do caixeiro viajante, problema de Steiner em grafos).

No *Capítulo 4* discutimos em detalhe a implementação de um algoritmo *branch and cut*. Essa discussão é feita em torno do problema de Steiner em grafos, porém, incluindo considerações gerais válidas para outros problemas também. São discutidos tópicos como separação, heurísticas primais, manutenção do problema de programação linear, *etc.*

A realização da *X Escola de Computação* motivou-nos, ou melhor, apressou-nos a escrever este texto. Na verdade, a idéia de produzir um texto desta natureza era algo que vínhamos acalentando há algum tempo. Assim, este texto surgiu de versões preliminares — espalhadas em dezenas de folhas avulsas e anotações mentais de ambos os autores — do material que tem sido utilizado em disciplinas e palestras no Departamento de Computação do IME-USP nos últimos anos. Ainda assim, mudanças de percurso ocorreram, em relação ao que incluir, excluir, enfatizar e referenciar. Agora, tendo finalmente que tornar público este trabalho, sentimos que expomos um texto preliminar, ainda longe daquilo que idealizamos. Neste sentido, gostaríamos de contar com sugestões da parte dos leitores, para incorporá-las numa versão futura deste

texto. O leitor interessado pode acessar por `ftp`¹ o arquivo de referências e o código de uma implementação do algoritmo discutido no Capítulo 4.

A principal motivação para escrever este texto é que ele seja usado em cursos de pós-graduação, no ensino de uma disciplina sobre Combinatória Poliédrica. Neste sentido, este texto cobre mais tópicos do que o estritamente necessário para a apresentação do método de planos-de-corte faciais. Além disso, os dois primeiros capítulos podem ser usados como uma introdução “poliédrica” a programação linear e dualidade. Quanto ao capítulo sobre implementação, o objetivo é que ele sirva de base para um projeto a ser desenvolvido pelos alunos da disciplina. Este texto, devidamente condensado, pode também ser usado por alunos de final da graduação em Ciência da Computação ou Matemática Aplicada.

Antes de finalizar, gostaríamos de expressar o nosso *Dankeschön* muito especial ao Professor Martin Grötschel, que guiou os nossos primeiros passos na área de Combinatória Poliédrica e muito contribuiu para a nossa formação. Gostaríamos também de agradecer muitíssimo aos nossos alunos de doutorado Flávio Keidi Miyazawa e Orlando Lee, que participaram pacientemente da revisão de versões preliminares deste texto, contribuindo com sugestões e apontando erros. Certamente eles não podem ser responsabilizados pelos erros que eventualmente ainda restaram nesta versão final. Estendemos os nossos agradecimentos ao Marko Loparić, pelas observações feitas durante as aulas, que foram incorporadas a este texto. Agradecemos também aos docentes do nosso departamento, em especial ao Arnaldo Mandel, José Augusto R. Soares, José Coelho de Pina Jr., Maria Angela Gurgel, Nami Kobayashi e Yoshiharu Kohayakawa, pelo apoio moral e T_EXnico.

Durante a elaboração deste texto contamos com apoio do CNPq (projetos de pesquisa ProComb/ProTeM-CC-II, proc. no. 680065/94-6; e bolsas de pesquisa, proc. nos. 300752/94-6 e 304527/89-0) e da CAPES/DAAD (projeto ProBral, proc. no. 001/94). Expressamos aqui os nossos agradecimentos a essas agências e também à coordenação da *X Escola de Computação* pela oportunidade de divulgar este texto.

São Paulo, maio de 1996

Carlos E. Ferreira

Yoshiko Wakabayashi

¹Para maiores informações, escreva para `[yw, cef]@ime.usp.br`

Capítulo 1

Preliminares

Neste capítulo definimos alguns conceitos básicos da teoria dos grafos e de álgebra linear. Assumimos que o leitor tenha relativa familiaridade com esses conceitos, de modo que, o nosso objetivo é mais o de estabelecer a notação e a terminologia que serão adotadas.

Este material não precisa necessariamente ser lido antes dos demais capítulos. A terminologia que utilizamos é relativamente padrão, podendo o leitor consultar apenas alguma notação não convencional, à medida que isso se fizer necessário.

Neste texto utilizamos também conceitos fundamentais da teoria de complexidade computacional, mas estes não serão apresentados neste capítulo. Definições destes conceitos e resultados dessa teoria podem ser encontrados em Garey e Johnson [38] ou em Aho, Hopcroft e Ullman [1].

1.1 Teoria dos Grafos

No Capítulo 3 estudamos diversos poliedros associados a problemas de otimização em grafos. Definimos aqui, apenas os conceitos relativos a esses problemas. A terminologia adotada, a menos de pequenas variações, é a mesma utilizada por Bondy e Murty [12], Berge [9] e Bollobás [11]).

Grafos

Um **grafo** $G = (V, A)$ consiste de um conjunto finito não-vazio V de elementos chamados **vértices** e um conjunto de pares não-ordenados de elementos distintos de V , chamados **arestas**, isto é, $A \subseteq \{\{u, v\} \mid u, v \in V, u \neq v\}$. Note que não são permitidos “laços” e nem arestas “múltiplas”.

Se $\alpha = \{u, v\}$ é uma aresta, dizemos que α **incide** em u e em v ; que u e v são seus **extremos**; e que u e v são **adjacentes**. Algumas vezes, para simplificar a notação, denotamos uma aresta $\{u, v\}$ por uv . O **grau** de um vértice v é o número de arestas incidentes a v .

Se S é um conjunto de vértices de um grafo $G = (V, A)$, denotamos por $A(S)$ o conjunto

das arestas de G com ambos os extremos em S e por $\delta(S)$ o conjunto das arestas de G com um extremo em S e o outro em $V \setminus S$, isto é, $A(S) := \{uv \in A \mid u, v \in S\}$ e $\delta(S) := \{uv \in A \mid u \in S \text{ e } v \in V \setminus S\}$. Um conjunto da forma $\delta(S)$, onde $\emptyset \neq S \neq V$, é chamado **corte**. Se s é um vértice em S e t é um vértice em $V \setminus S$, então dizemos que $\delta(S)$ é um (s, t) -**corte**. Para $v \in V$, a notação $\delta(v)$ abrevia $\delta(\{v\})$.

Se dois grafos $G = (V, A)$ e $H = (W, B)$ são tais que $V \subseteq W$ e $A \subseteq B$, então G é dito um **subgrafo** de H . Neste caso, dizemos que H **contém** G ou que G está contido em H . Se $|V| = |W|$ então dizemos que G é um subgrafo **gerador** de H . Se $E \subseteq A$, então o subgrafo de G com conjunto de arestas E e vértices consistindo dos extremos das arestas de E é o subgrafo **gerado** ou **induzido** por E e denotado por $G[E]$. Se $W \subseteq V$, então $G = (W, A(W))$ é o subgrafo **gerado** por W e denotado por $G[W]$.

Se W é um conjunto de vértices de um grafo $G = (V, A)$, então $G - W$ denota o subgrafo obtido de G removendo-se W , definido com conjunto de vértices $V \setminus W$ e arestas $A(V \setminus W)$. Observe que $G - W = G[V \setminus W]$. Escrevemos $G - v$ para denotar o subgrafo $G - \{v\}$, se v é um vértice de G .

Dizemos que um grafo é **completo** se quaisquer dois de seus vértices são adjacentes. O (único a menos de isomorfismo) grafo completo com n vértices é denotado por $K_n = (V_n, A_n)$.

Um grafo $G = (V, A)$ é dito **bipartido** se V pode ser particionado em dois conjuntos disjuntos não-vazios V_1 e V_2 tais que $V_1 \cup V_2 = V$ e toda aresta em G tem um extremo em V_1 e outro em V_2 .

Um **emparelhamento** em um grafo é um conjunto de arestas duas a duas não-incidentes a um mesmo vértice. Se um vértice v é extremo de alguma aresta de um emparelhamento E , então dizemos que E **cobre** v , ou que v é coberto por E . Dizemos que um emparelhamento E é **perfeito** se cobre todos os vértices do grafo. Um grafo G é **hipo-emparelhável** se não tem um emparelhamento perfeito, mas ao remover qualquer um de seus vértices o grafo que resulta tem um emparelhamento perfeito. Se E é um emparelhamento em G , então um (s, t) -caminho em G cujas arestas pertencem alternadamente a E e a $A \setminus E$ (ao percorrê-lo de s a t) é chamado **alternante** (em relação a E).

A **matriz de incidência** de um grafo $G = (V, A)$ é uma matriz com linhas e colunas indexadas por V e A , respectivamente, onde a entrada na posição (v, α) é 1 se a aresta α incide no vértice v ; e é 0, em caso contrário.

Grafos orientados

Um **grafo orientado** $D = (V, A)$ consiste de um conjunto finito não-vazio V de elementos chamados **vértices**, e um conjunto A de elementos chamados **arcos** ou **arestas**, que são pares ordenados de vértices distintos, isto é, $A \subseteq \{(u, v) \mid u, v \in V, u \neq v\}$. Se $\alpha = (u, v)$ é um arco, dizemos que α **incide** em u e em v , e que u e v são seus **extremos**. Também nos referimos a u como **início** e a v como **término** do arco α .

Todos os conceitos definidos para grafos que não envolvem a noção de orientação se aplicam

analogamente para grafos orientados.

Se W é um conjunto de vértices de um grafo orientado $D = (V, A)$, denotamos por $A(W)$ o conjunto das arcos de D com ambos os extremos em W e por $\delta^+(W)$ o conjunto dos arcos de D com início em W e término em $V \setminus W$, isto é, $A(W) := \{(u, v) \in A \mid u, v \in W\}$ e $\delta^+(W) := \{(u, v) \in A \mid u \in W \text{ e } v \in V \setminus W\}$. Um conjunto da forma $\delta^+(W)$, onde $\emptyset \neq W \neq V$, é chamado **corte**. Se $s \in W$ e $t \in V \setminus W$, então dizemos que $\delta^+(W)$ é um (s, t) - **corte**. Para $v \in V$, a notação $\delta^+(v)$ abrevia $\delta^+(\{v\})$.

A **matriz de incidência** de um grafo orientado $D = (V, A)$ é uma matriz com linhas e colunas indexadas por V e A , respectivamente, onde a entrada na posição (v, α) é $-1, +1$, ou 0 , se o vértice v é o término de α , o início de α , ou α não incide em v , respectivamente.

Caminhos, circuitos e conexidade

Num grafo orientado $D = (V, A)$, um conjunto não-vazio de arcos $P = \{\alpha_1, \alpha_2, \dots, \alpha_k\}$, onde $\alpha_i = (v_i, v_{i+1}) \in A$, e $v_i \neq v_j$ para $i \neq j$, é chamado **caminho**. Dizemos que P é um (v_1, v_k) - **caminho**; e que tem **comprimento** k . Se $(v_k, v_1) \in A$ então $C := P \cup \{(v_k, v_1)\}$ é chamado **circuito**. Esquecendo-se a orientação (dos arcos) temos analogamente o conceito de caminho e circuito em um grafo. A **distância** entre dois vértices u e v em um grafo (não-orientado) é o comprimento de um (u, v) -caminho de menor comprimento possível; se não existe um tal caminho, então dizemos que a distância é infinita.

Num grafo (orientado ou não) com n vértices, um circuito com exatamente n arestas é chamado **hamiltoniano**.

Um grafo é **conexo** se possui um caminho entre quaisquer dois de seus vértices; caso contrário, é **desconexo**. Denominamos **componentes** os subgrafos conexos maximais de um grafo. Um grafo G com pelo menos 2 vértices é **2-conexo** se é conexo e $G - v$ é conexo para qualquer vértice v em G . Para grafos orientados, esses conceitos são definidos analogamente, ignorando-se a orientação dos arcos.

Um grafo é **acíclico** se não contém circuitos. Um grafo acíclico conexo é chamado **árvore**. Grafos acíclicos não necessariamente conexos são chamados **florestas**.

1.2 Álgebra Linear

Denotamos por \mathbb{R} , \mathbb{Z} e \mathbb{N} os conjuntos dos números reais, inteiros e naturais, respectivamente. O conjunto \mathbb{N} não contém zero. Denotamos por \mathbb{R}_+ o conjunto dos números reais não-negativos e por \mathbb{Z}_+ o dos inteiros não-negativos. Para $n \in \mathbb{N}$, o símbolo \mathbb{R}^n denota o conjunto de todos os vetores com n componentes reais. Definimos \mathbb{Z}^n analogamente. Denotamos por $\mathbb{R}^{m \times n}$ o conjunto das matrizes reais com m linhas e n colunas.

Se $A \in \mathbb{R}^{m \times n}$, a menos de menção contrária, deve ficar subentendido que A é da forma

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}.$$

A j -ésima coluna de A é um vetor com m elementos, denotado por A_{*j} , ou seja,

$$A_{*j} = \begin{pmatrix} a_{1j} \\ \vdots \\ a_{mj} \end{pmatrix}.$$

A i -ésima linha da matriz A é um vetor-linha de comprimento n , denotado por A_{i*} . Ou seja, $A_{i*} = (a_{i1} \dots a_{in})$.

Se $I \subseteq \{1, \dots, m\} =: M$ e $J \subseteq \{1, \dots, n\} =: N$ denotamos por A_{IJ} a submatriz de A formada pela eliminação das linhas de A cujos índices não estão em I e eliminação dos elementos das colunas de A cujos índices não estão em J . Note que, se $I = \{i\}$ e $J = \{j\}$, então $A_{IN} = A_{i*}$ e $A_{MJ} = A_{*j}$. Escrevemos A_{I*} em vez de A_{IN} , e A_{*J} em vez de A_{MJ} .

Se E é um conjunto finito ordenado e W um conjunto qualquer, W^E denota o conjunto dos vetores com $|E|$ componentes indexadas pelos elementos de E , com entradas em W . Similarmente, se A é uma matriz, $A \in \mathbb{R}^{E \times F}$ significa que A é real, possui $|E|$ linhas e $|F|$ colunas, indexadas pelos elementos dos conjuntos finitos ordenados E e F , respectivamente.

Se x é um número real, então $\lfloor x \rfloor$ denota o maior inteiro menor ou igual a x ; e o símbolo $\lceil x \rceil$ denota o menor inteiro maior ou igual a x . Sendo A e B conjuntos, denotamos por $A \triangle B$ a **diferença simétrica** de A e B , isto é, $A \triangle B := (A \setminus B) \cup (B \setminus A)$. A **soma** de A e B é definida como $A + B := \{a + b \mid a \in A \text{ e } b \in B\}$.

A menos de especificação em contrário, vetores serão vetores-coluna. Assim, se x é um vetor, x^\top denota seu transposto. O **produto interno** de dois vetores x e y em \mathbb{R}^n é o escalar $x^\top y := \sum_{i=1}^n x_i y_i$. Um vetor de uns é denotado por $\mathbf{1}$. Um vetor ou uma matriz de zeros é denotada por 0 (ficará claro pelo contexto a sua dimensão). Da mesma forma a matriz identidade é denotada simplesmente por I . Um vetor (x_1, x_2, \dots, x_n) é referido como **unitário** se existe i ($1 \leq i \leq n$), tal que $x_i = 1$ e $x_j = 0$ para $j \neq i$. Utilizamos a notação e_i para designar o vetor unitário cuja componente 1 está na posição i .

Se E é um conjunto finito ordenado e $A \subseteq E$, denotamos por $\chi^A = (\chi_e^A : e \in E)$ o **vetor de incidência** ou vetor **característico** de A , definido como

$$\chi_e^A = \begin{cases} 1, & \text{se } e \in A; \\ 0, & \text{se } e \in E \setminus A. \end{cases}$$

Neste texto, identificamos funções $c : A \rightarrow \mathbb{R}$ com vetores em \mathbb{R}^A . Assim, se $a \in A$, c_a pode ser a componente do vetor c indexada por $a \in A$, ou o valor $c(a)$ da função c . Se $B \subseteq A$ e $c : A \rightarrow \mathbb{R}$ (ou $c \in \mathbb{R}^A$), então $c(B)$ denota a soma $c(B) := \sum_{b \in B} c_b$.

Um vetor $x \in \mathbb{R}^n$ é uma **combinação linear** dos vetores $x_1, x_2, \dots, x_t \in \mathbb{R}^n$, se, para algum $\alpha = (\alpha_1, \dots, \alpha_t) \in \mathbb{R}^t$,

$$x = \sum_{i=1}^t \alpha_i x_i.$$

Uma tal combinação linear é chamada

afim, se $\alpha_1 + \dots + \alpha_t = 1$;
cônica, se $\alpha_1, \dots, \alpha_t \geq 0$;
convexa, se for afim e cônica.

Para um conjunto não-vazio $S \subseteq \mathbb{R}^n$, denotamos por $\text{lin}(S)$ o **fecho linear** dos elementos de S ; isto é, o conjunto de todos os vetores que são combinação linear de um número finito de vetores de S . Analogamente, definimos os fechos **afim**, **cônico** e **convexo**, denotados por $\text{afim}(S)$, $\text{cone}(S)$ e $\text{conv}(S)$, considerando a combinação correspondente. Se S for o conjunto vazio, então $\text{lin}(S) := \text{cone}(S) := \{0\}$ e $\text{conv}(S) = \text{afim}(S) := \emptyset$. Dizemos que $S \subseteq \mathbb{R}^n$ é um **subespaço linear** se $S = \text{lin}(S)$. Analogamente, definimos **subespaço afim**, **cone** e **conjunto convexo**. Se A é uma matriz, então cada um dos conjuntos $\text{lin}(A)$, $\text{afim}(A)$, $\text{cone}(A)$ e $\text{conv}(A)$ são definidos analogamente, considerando-se os vetores-coluna de A .

Um conjunto $S \subseteq \mathbb{R}^n$ é **linearmente independente** se para qualquer subconjunto finito $\{x_1, \dots, x_t\}$ de S , sempre que $\sum_{i=1}^t \alpha_i x_i = 0$, onde $\alpha_i \in \mathbb{R}$, temos que $\alpha_1 = \dots = \alpha_t = 0$. Dizemos que S é **afim-independente** se para qualquer subconjunto finito $\{x_1, \dots, x_t\}$ de S , $\sum_{i=1}^t \alpha_i x_i = 0$, onde $\alpha_i \in \mathbb{R}$ e $\sum_{i=1}^t \alpha_i = 0$, implica $\alpha_1 = \dots = \alpha_t = 0$. Um conjunto S é **linearmente dependente** se ele não for linearmente independente. Definimos **afim-dependente** de forma análoga.

Para $S \subseteq \mathbb{R}^n$, o **posto** de S , denotado por $\text{posto}(S)$, é a cardinalidade de um maior subconjunto de S que é linearmente independente. Analogamente, definimos o **posto afim** de S , denotado por $\text{posto} - \text{afim}(S)$, como sendo a cardinalidade de um maior conjunto **afim independente** contido em S . Observe que $\{0\}$ é afim-independente, mas não é linearmente independente. Por outro lado, se S é um conjunto linearmente independente, então S é afim-independente. Prova-se que para todo $S \subseteq \mathbb{R}^n$, se $0 \in \text{afim}(S)$, então $\text{posto} - \text{afim}(S) = \text{posto}(S) + 1$; e se $0 \notin \text{afim}(S)$, então $\text{posto} - \text{afim}(S) = \text{posto}(S)$.

O posto de uma matriz A , denotado por $\text{posto}(A)$, é o posto do conjunto de vetores-coluna de A , que prova-se ser igual ao posto do conjunto de vetores-linha de A . Uma matriz $A \in \mathbb{R}^{m \times n}$ tem **posto-linha completo** se $\text{posto}(A) = m$, e tem **posto-coluna completo** se $\text{posto}(A) = n$.

Para $S \subseteq \mathbb{R}^n$, a **dimensão** $\text{dim}(S)$ de S é definida como $\text{dim}(S) := \text{posto} - \text{afim}(S) - 1$. Dizemos que S tem dimensão **plena** se $\text{dim}(S) = n$.

Capítulo 1

Teoria de Poliedros

Neste capítulo estudamos conceitos e resultados básicos da teoria de poliedros. Um dos resultados fundamentais que apresentamos diz respeito às formas de se descrever um poliedro: como intersecção de um número finito de semi-espacos (a chamada *representação externa*) e como soma do fecho convexo dos seus vértices com os pontos gerados pelo cone de seus raios extremos (a *representação interna*). Esse resultado não é imediato, e exige alguns preparativos, o que é feito no capítulo.

Uma operação muito importante sobre poliedros é a *projeção*. Apresentamos um algoritmo para projetar um poliedro sobre um conjunto, numa dada direção. Uma aplicação interessante desse algoritmo é o método de eliminação de Fourier–Motzkin. Mostramos como esse método pode ser usado para verificar se um dado poliedro é vazio; ou, em sua versão dual, para achar, a partir da representação externa de um poliedro, os seus vértices e raios extremos.

Uma consequência direta do método de eliminação de Fourier–Motzkin é o chamado Lema de Farkas (também chamado de Lemas Alternativos em suas várias versões). Resultados importantes da teoria de programação linear, como o teorema forte de dualidade, podem ser derivados a partir do Lema de Farkas.

Outro conceito importante que discutimos é o de polaridade cônica. O resultado mais interessante é que pode-se mostrar que para todo cone poliédrico (dado através de sua representação externa), existe uma matriz tal que o conjunto de pontos desse cone coincide com o fecho cônico gerado por essa matriz. Além disso, mostramos que dada uma matriz, tanto o fecho linear, cônico, convexo e afim dessa matriz são poliedros. Nesse ponto, apresentamos o resultado sobre a equivalência entre a representação externa e a representação interna de um poliedro.

Apresentamos caracterizações para faces, facetas e vértices de poliedros. O conceito de inequação que define (ou induz) uma face é introduzido. Mostramos que uma descrição de um poliedro dada por inequações que definem suas facetas é a melhor descrição possível. Além disso, mostramos que se uma inequação define uma faceta, então existe uma forma canônica única de representá-la. Tal resultado, aparentemente simples, permite uma simplificação muito grande nas demonstrações que apresentamos no Capítulo 3, a respeito de facetas de alguns poliedros.

Os resultados apresentados aqui são baseados em vários textos, e segue de perto os tópicos cobertos em [46]. Recomendamos ao leitor os textos de Schrijver [99], Chvátal [14], Grünbaum [62] e Ziegler [105].

1.1 Conceitos Básicos

Um subconjunto $P \subseteq \mathbb{R}^n$ é chamado **poliedro** se

$$P = \{x \in \mathbb{R}^n \mid Ax \leq b\},$$

para alguma matriz $A \in \mathbb{R}^{m \times n}$ e algum vetor $b \in \mathbb{R}^m$.

Um poliedro P definido por uma matriz A e por um vetor b é denotado por $P(A, b)$. Se P é um poliedro limitado (ou seja, existe um real α tal que todo ponto $x \in P$ satisfaz $\|x\| \leq \alpha$), então P é chamado **politopo**.

Se $a \in \mathbb{R}^n \setminus 0$ e $\alpha \in \mathbb{R}$, então o poliedro $\{x \in \mathbb{R}^n \mid a^\top x \leq \alpha\}$ é chamado **semi-espaço**. Um conjunto de pontos que divide o espaço em dois semi-espaços, ou seja, um conjunto da forma $\{x \in \mathbb{R}^n \mid a^\top x = \alpha\}$, onde a é não nulo, é chamado **hiperplano**.

Observando que cada uma das linhas da matriz A juntamente com a correspondente entrada do vetor b define um semi-espaço, segue imediatamente que $P(A, b)$ é intersecção de um número finito de semi-espaços.

Exemplo 1.1.1 Na Figura 1.1 a região mais escura corresponde ao poliedro $P(A, b)$, onde

$$A = \begin{pmatrix} 1 & -1 \\ -1 & -1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \quad e \quad b = \begin{pmatrix} -2 \\ -2 \\ 4 \\ 4 \end{pmatrix}.$$

Note que o conjunto vazio é um poliedro, assim como o espaço \mathbb{R}^n . Certos tipos de cones também são poliedros. Lembramos que um conjunto $C \subseteq \mathbb{R}^n$ é um **cone** se $C = \text{cone}(C)$. Um cone C é **poliédrico** se C é um poliedro.

Exercício 1.1.2 Mostre que todo cone $C \subseteq \mathbb{R}^n$ é poliédrico.

Exercício 1.1.3 Dê um exemplo de um cone que não é poliédrico.

Exercício 1.1.4 Mostre que todo poliedro é um conjunto convexo.

Proposição 1.1.5 Um cone $C \subseteq \mathbb{R}^n$ é poliédrico se e somente se existe uma matriz $A \in \mathbb{R}^{m \times n}$ tal que $C = P(A, 0)$.

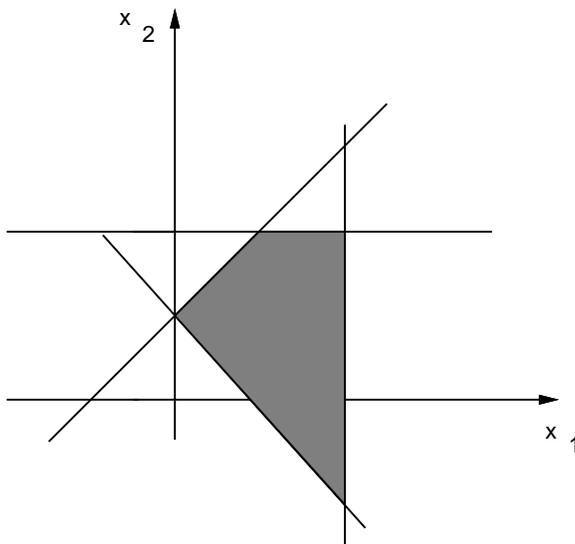


Figura 1.1: Exemplo de um poliedro no \mathbb{R}^2 .

Prova. Seja C um cone. Se $C = P(A, 0)$, então claramente C é um poliedro, e portanto é um cone poliédrico.

Suponhamos agora que C seja poliédrico. Como C é um poliedro, então existe uma matriz A e um vetor b tais que $C = P(A, b)$. Suponhamos que exista $x' \in C$ tal que $Ax' \not\leq 0$. Então, existe uma linha i da matriz A tal que $t := A_{i*}x' > 0$. Como C é um cone, então $\lambda x' \in C$ para todo $\lambda \geq 0$. Tome $\lambda' := b_i/t + 1$ e observe que $\lambda' > 0$ e $\lambda'x' \in C$. Por outro lado, $A_{i*}(\lambda'x') = \lambda't > b_i$, uma contradição. Assim, para todo $x \in C$ temos que $Ax \leq 0$. Logo, $C \subseteq P(A, 0)$. Como $0 \in C$, segue que $b \geq 0$, ou seja, $P(A, 0) \subseteq C$. Portanto, $C = P(A, 0)$. \square

1.2 Transformação de Poliedros

Nem sempre os poliedros surgem na forma de um sistema de inequações. Entretanto, não é difícil mostrar que sistemas constituídos por equações e inequações podem ser transformados em sistemas equivalentes só constituídos por inequações.

Proposição 1.2.1 *O conjunto de soluções do sistema abaixo é um poliedro.*

$$\begin{aligned} Bx + Cy &= c \\ Dx + Ey &\leq d \\ x &\geq 0 \\ x \in \mathbb{R}^p, y &\in \mathbb{R}^q. \end{aligned}$$

Prova. Basta notar que o conjunto das soluções do sistema acima é igual ao conjunto dos pontos do poliedro $P(A, b)$, onde

$$A := \begin{pmatrix} B & C \\ -B & -C \\ D & E \\ -I & 0 \end{pmatrix} \quad e \quad b := \begin{pmatrix} c \\ -c \\ d \\ 0 \end{pmatrix}.$$

□

Um tipo especial de poliedros é aquele definido apenas por equações. Dada uma matriz $A \in \mathbb{R}^{m \times n}$ e um vetor $b \in \mathbb{R}^m$, seja

$$P^=(A, b) := \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}.$$

Note que $P^=(A, b)$ é um poliedro (basta aplicar a proposição anterior). Claramente, nem todo poliedro pode ser descrito dessa forma. De fato, o poliedro $\{x \in \mathbb{R} \mid x \leq 1\}$ não pode ser descrito por um sistema de equações.

Podemos, no entanto, sempre tomar um poliedro descrito de uma forma e transformá-lo em um outro poliedro “equivalente”, descrito de uma outra forma. Para isso, algumas vezes, precisamos trabalhar em um outro espaço, de dimensão diferente.

Assim, vejamos como transformar $P(A, b)$ em um poliedro “equivalente” $P^=(A', b')$. Inicialmente, transformamos todas as inequações em equações. Isso é feito acrescentando-se, para cada inequação, uma variável de folga (*slack variable*) não-negativa.

Ou seja, uma inequação da forma

$$a^\top x \leq \alpha$$

pode ser inicialmente transformada em

$$a^\top x + y = \alpha, \quad y \geq 0.$$

Como a variável x é livre de sinal, podemos substituí-la pela diferença de duas variáveis não-negativas x^+ e x^- . Ou seja, substituímos as ocorrências de x por $x^+ - x^-$ e acrescentamos as restrições $x^+ \geq 0$ e $x^- \geq 0$. Assim, neste caso, o sistema “equivalente” que se obtém é

$$a^\top x^+ - a^\top x^- + y = \alpha, \quad x^+ \geq 0, \quad x^- \geq 0, \quad y \geq 0,$$

que é da forma

$$a'^\top z = \alpha, \quad z \geq 0.$$

Note que, fazendo tais transformações, um poliedro $P(A, b)$, onde $A \in \mathbb{R}^{m \times n}$, é transformado em um poliedro $P' = P^=(A', b)$, onde $A' \in \mathbb{R}^{m \times (n+m)}$. Ambos são ditos “equivalentes”, no sentido de que a cada ponto $x \in P(A, b)$ corresponde um ponto $x' \in P' = P^=(A', b)$ e vice-versa.

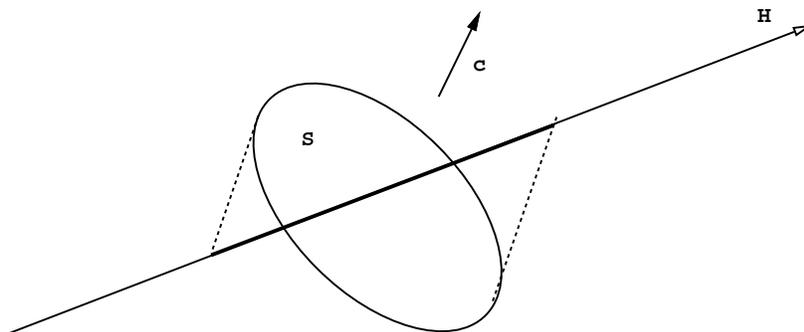


Figura 1.2: Exemplo de projeção

1.3 Projeção de Poliedros

Uma ferramenta que pode ser muito útil no tratamento de certos problemas é considerar projeção de poliedros. Veremos a seguir que o problema de verificar se um poliedro é ou não vazio pode ser resolvido usando projeção. Da mesma forma, pode-se resolver o problema de encontrar todos os vértices ou as inequações que definem um poliedro. Mais adiante faremos alguns comentários a esse respeito.

Sejam $S \subseteq \mathbb{R}^n$ e $H \subseteq \mathbb{R}^n$ dois conjuntos e $c \in \mathbb{R}^n$ um vetor não-nulo. O conjunto

$$\{x \in \mathbb{R}^n \mid x \in H, \text{ existe } \lambda \in \mathbb{R} \text{ tal que } x + \lambda c \in S\}$$

é chamado de **projeção de S sobre H na direção c** . Na Figura 1.2 mostramos um exemplo de projeção.

Interessa-nos, em especial, projeções sobre espaços afins. Veremos que, neste caso, todos os resultados podem ser expressos como projeções sobre hiperplanos, em particular, *projeções ortogonais*. Assim, se H é o hiperplano

$$H = \{x \in \mathbb{R}^n \mid c^\top x = \lambda\}, \text{ para algum } \lambda \in \mathbb{R}^n,$$

então dizemos que a projeção de S sobre H na direção c é **ortogonal**.

Exercício 1.3.1 Seja $H \subseteq \mathbb{R}^n$ um subconjunto qualquer, $c \in \mathbb{R}^n \setminus \{0\}$ e $H' := \{x \in \mathbb{R}^n \mid a^\top x \leq \alpha\}$ um semi-espaço. Seja P_H a projeção de H' sobre H na direção c . Prove que

- (a) Se a é ortogonal a c , então $P_H = H \cap H'$.
- (b) Se a não é ortogonal a c , então $P_H = H$.

Agora vejamos como calcular a projeção de um poliedro $P(A, b)$ sobre um conjunto (qualquer) H , numa dada direção c . O método calcula um poliedro $P(D, d)$ tal que a projeção de $P(A, b)$ sobre H na direção c é dada por

$$H \cap \{x \in \mathbb{R}^n \mid Dx \leq d\}.$$

Algoritmo PROJEÇÃO DE UM POLIEDRO NUMA DADA DIREÇÃO

Entrada: Um poliedro $P(A, b)$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$,
um vetor $c \in \mathbb{R}^n$ (a direção de projeção).

Saída: Um poliedro $P(D, d)$, $D \in \mathbb{R}^{r \times n}$, $d \in \mathbb{R}^r$
(r é calculado no algoritmo) tais que, para qualquer
 $H \subseteq \mathbb{R}^n$, a projeção de $P(A, b)$ sobre H
na direção c é dada por $H \cap P(D, d)$.

Seja $M = \{1, 2, \dots, m\}$ o conjunto dos índices das linhas de A ;

Particione M em 3 partes:

$$N := \{i \in M \mid A_{i*}c < 0\},$$

$$Z := \{i \in M \mid A_{i*}c = 0\},$$

$$P := \{i \in M \mid A_{i*}c > 0\};$$

Faça $r := |Z \cup (N \times P)|$;

Defina $R := \{1, \dots, r\}$;

Construa uma bijeção $p : R \longrightarrow Z \cup (N \times P)$;

Para cada $i := 1, \dots, r$ **faça**

se $p(i) \in Z$

então $D_{i*} := A_{p(i)*}$; $d_i := b_{p(i)}$;

senão seja $p(i) = (s, t) \in N \times P$;

$$D_{i*} := (A_{t*}c)A_{s*} - (A_{s*}c)A_{t*};$$

$$d_i := (A_{t*}c)b_s - (A_{s*}c)b_t;$$

Devolva D e d .

Provaremos a seguir que o algoritmo acima calcula corretamente a projeção de um poliedro. Sabemos que

$$P(A, b) = \bigcap_{i=1}^m H_i, \text{ onde } H_i := P(A_{i*}, b_i).$$

Equivalentemente, podemos escrever que

$$P(A, b) = \bigcap_{(i,j) \in M \times M} H_i \cap H_j.$$

Vamos mostrar que a projeção P_H , de $P(A, b)$ sobre H na direção c , pode ser calculada da seguinte forma:

$$P_H := H \cap \bigcap_{(i,j) \in M \times M} H_{ij},$$

onde H_{ij} é a projeção de $H_i \cap H_j$ sobre \mathbb{R}^n na direção c .

Proposição 1.3.2 *Seja $P(A, b)$ um poliedro em \mathbb{R}^n , $H \subseteq \mathbb{R}^n$ e $c \in \mathbb{R}^n$. Então, a projeção P_H , de $P(A, b)$ sobre H na direção c , pode ser calculada da seguinte forma:*

$$P_H := H \cap \bigcap_{(i,j) \in M \times M} H_{ij},$$

onde H_{ij} é a projeção de $H_i \cap H_j$ sobre \mathbb{R}^n na direção c .

Prova. Temos que

$$\begin{aligned} P_H &= \{x \in \mathbb{R}^n \mid x \in H, \text{ existe } \lambda \in \mathbb{R} \text{ tal que } x + \lambda c \in P(A, b)\} \\ &= \{x \in \mathbb{R}^n \mid x \in H, \text{ existe } \lambda \in \mathbb{R} \text{ tal que } x + \lambda c \in H_i \cap H_j \\ &\quad \text{para todo } (i, j) \in M \times M\} \\ &= H \cap \bigcap_{(i,j) \in M \times M} \{x \in \mathbb{R}^n \mid \text{existe } \lambda \in \mathbb{R} \text{ tal que } x + \lambda c \in H_i \cap H_j\} \\ &= H \cap \bigcap_{(i,j) \in M \times M} H_{ij}. \end{aligned}$$

□

Com isso, concluímos que basta estudar como projetar sobre um conjunto a intersecção de dois semi-espacos numa direção c . Isso é feito no teorema a seguir.

Teorema 1.3.3 *Sejam $c \in \mathbb{R}^n \setminus \{0\}$, H um conjunto qualquer, $S_1 := \{x \in \mathbb{R}^n \mid a^\top x \leq \alpha\}$ e $S_2 := \{x \in \mathbb{R}^n \mid b^\top x \leq \beta\}$. Seja P_H a projeção de $S_1 \cap S_2$ sobre H na direção c .*

(a) *Se ambos os vetores a, b são ortogonais a c , então*

$$P_H = S_1 \cap S_2 \cap H.$$

(b) *Se um dos vetores a, b é ortogonal a c e o outro não, digamos $a^\top c = 0$, $b^\top c \neq 0$, então*

$$P_H = S_1 \cap H.$$

(c) *Se ambos os vetores a, b não são ortogonais a c e $a^\top c$ e $b^\top c$ têm o mesmo sinal, então*

$$P_H = H.$$

(d) *Se ambos os vetores a, b não são ortogonais a c , e $a^\top c$ e $b^\top c$ têm sinais diferentes, digamos, $a^\top c < 0$ e $b^\top c > 0$, então tomando-se*

$$d = (b^\top c)a - (a^\top c)b, \quad \delta = (b^\top c)\alpha - (a^\top c)\beta,$$

temos que

$$P_H = H \cap \{x \in \mathbb{R}^n \mid d^\top x \leq \delta\}.$$

Adicionalmente, temos que a inequação $d^\top x \leq \delta$ é uma combinação cônica das inequações $a^\top x \leq \alpha$ e $b^\top x \leq \beta$, e d é ortogonal a c .

Prova. (a) Da definição de projeção, temos que

$$P_H = \{x \in \mathbb{R}^n \mid x \in H, \text{ existe } \lambda \in \mathbb{R} \text{ tal que } x + \lambda c \in S_1 \cap S_2\}.$$

Ou seja,

$$P_H = H \cap \{x \in \mathbb{R}^n \mid \text{ existe } \lambda \in \mathbb{R} : a^\top(x + \lambda c) \leq \alpha \text{ e } b^\top(x + \lambda c) \leq \beta\}.$$

Mas, como a e b são ortogonais a c ,

$$P_H = H \cap S_1 \cap S_2.$$

(b) A demonstração é análoga à do item (a) (exercício para o leitor).

(c) Claramente, $P_H \subseteq H$. Vamos mostrar que a outra inclusão também é válida. Seja $x' \in H$. Sem perda de generalidade podemos supor que $a^\top c > 0$ e $b^\top c > 0$. Tome

$$\lambda := \min \left\{ \frac{1}{a^\top c}(\alpha - a^\top x'), \frac{1}{b^\top c}(\beta - b^\top x') \right\},$$

e observe que $x' + \lambda c \in S_1 \cap S_2$. Então, $x' \in P_H$.

(d) Defina o conjunto $Q := \{x \in \mathbb{R}^n \mid d^\top x \leq \delta\}$. Claramente, por construção, temos que $d^\top x \leq \delta$ é uma combinação cônica de $a^\top x \leq \alpha$ e $b^\top x \leq \beta$. Daí, é fácil ver que $S_1 \cap S_2 \subseteq Q$. Além disso, $d^\top c = 0$, e portanto, a projeção Q_H , de Q sobre H na direção c , é dada por $Q \cap H$. Assim,

$$P_H \subseteq Q_H = Q \cap H.$$

Resta mostrar que $Q \cap H \subseteq P_H$. Seja $x' \in Q \cap H$. Por hipótese, temos que $a^\top c < 0$ e $b^\top c > 0$. Pela construção de d e δ , temos

$$((b^\top c)a - (a^\top c)b)^\top x' = d^\top x' \leq \delta = (b^\top c)\alpha - (a^\top c)\beta.$$

Assim,

$$(a^\top c)(\beta - b^\top x') \leq (b^\top c)(\alpha - a^\top x'),$$

ou ainda,

$$\lambda_b := \frac{1}{b^\top c}(\beta - b^\top x') \geq \frac{1}{a^\top c}(\alpha - a^\top x') =: \lambda_a.$$

Basta agora tomar qualquer λ tal que $\lambda_a \leq \lambda \leq \lambda_b$, e observar que

$$a^\top(x' + \lambda c) = a^\top x' + \lambda a^\top c \leq a^\top x' + \lambda_a a^\top c = \alpha.$$

Da mesma forma,

$$b^\top(x' + \lambda c) = b^\top x' + \lambda b^\top c \leq b^\top x' + \lambda_b b^\top c = \beta.$$

Com isso, $x' \in P_H$. □

Sabemos que a projeção P_H do poliedro $P(A, b)$ sobre H na direção c é dada por

$$P_H = H \cap \bigcap_{(i,j) \in M \times M} H_{ij}.$$

Usando a partição de M em Z , N e P como definido no algoritmo, temos,

$$P_H = H \cap \bigcap_{(i,j)} H_{ij},$$

onde a intersecção é sobre todos os pares (i, j) em $(M \times Z) \cup (P \times P) \cup (N \times N) \cup (N \times P)$.

Aplicando o teorema anterior, temos

$$P_H = H \cap \bigcap_{i \in Z} H_i \cap \bigcap_{(i,j) \in N \times P} H_{ij}.$$

Mas, $P(D, d) = \bigcap_{i \in Z} H_i \cap \bigcap_{(i,j) \in N \times P} H_{ij}$, o que completa a demonstração de que o algoritmo de projeção está correto.

No teorema seguinte, formalizamos algumas das idéias vistas.

Teorema 1.3.4 *Sejam $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ e $H \subseteq \mathbb{R}^n$ um conjunto qualquer. Considere o sistema de inequações $Dx \leq d$ construído aplicando-se o algoritmo de projeção e seja P_H a projeção de $P(A, b)$ sobre H na direção c . Então,*

- (a) *Cada inequação $D_{i*}x \leq d_i$ é combinação linear não-negativa de inequações do sistema $Ax \leq b$, i.e., para cada $i \in \{1, \dots, r\}$ existe um vetor $u \geq 0$ com $D_{i*} = u^\top A$ e $d_i = u^\top b$.*
- (b) *$D_{i*}c = 0$ para todo $i = 1, \dots, r$.*
- (c) *$P_H = H \cap P(D, d)$.*
- (d) *Tome $x' \in H$ e N , Z e P definidos como no algoritmo. Defina*

$$\lambda_i := \frac{1}{A_{i*}c}(b_i - A_{i*}x'), \quad \text{para todo } i \in P \cup N,$$

$$L := \begin{cases} -\infty, & \text{se } N = \emptyset, \\ \max\{\lambda_i \mid i \in N\}, & \text{caso contrário;} \end{cases}$$

$$U := \begin{cases} +\infty, & \text{se } P = \emptyset, \\ \min\{\lambda_i \mid i \in P\}, & \text{caso contrário.} \end{cases}$$

Então,

- (d₁) *$x' \in P(D, d)$ implica que $L \leq U$ e $x' + \lambda c \in P(A, b)$ para todo λ tal que $L \leq \lambda \leq U$.*
- (d₂) *$x' + \lambda c \in P(A, b)$ implica que $L \leq \lambda \leq U$ e $x' \in P(D, d)$.*

Prova. Os itens (a) e (b) são conseqüências diretas da definição da matriz D e vetor d . O item (c) é implicado pelo item (d). Resta-nos então, provar esse último item.

(d₁) Seja $x' \in P(D, d)$. Mostremos inicialmente que $L \leq U$. Se algum dos conjuntos N ou P for vazio, então a afirmação é claramente verdadeira. Suponhamos, então, que $N \neq \emptyset$ e

$P \neq \emptyset$. Sejam $s \in N$, $t \in P$ e $r \in R$ com $p(r) = (s, t)$, e além disso, $\lambda_s = L$ e $\lambda_t = U$. Com isso,

$$(A_{t^*}c)A_{s^*}x' - (A_{s^*}c)A_{t^*}x' = D_{r^*}x' \leq d_r = (A_{t^*}c)b_s - (A_{s^*}c)b_t.$$

Daí,

$$(A_{s^*}c)(b_t - A_{t^*}x') \leq (A_{t^*}c)(b_s - A_{s^*}x'),$$

de onde segue que,

$$U = \lambda_t = \frac{1}{A_{t^*}c}(b_t - A_{t^*}x') \geq \frac{1}{A_{s^*}c}(b_s - A_{s^*}x') = \lambda_s = L.$$

Vamos agora mostrar que $A_{i^*}(x' + \lambda c) \leq b_i$ para todo $i \in M$ e todo λ tal que $L \leq \lambda \leq U$. Temos três casos a considerar.

- $i \in Z$. Nesse caso, existe $j \in R$ tal que $p(j) = i$, $D_{j^*} = A_{i^*}$ e $d_j = b_i$. Como $A_{i^*}c = 0$, então $A_{i^*}(x' + \lambda c) = A_{i^*}x' + \lambda A_{i^*}c = A_{i^*}x' = D_{j^*}x' \leq d_j = b_i$.
- $i \in P$. Nesse caso, $U < \infty$, e com isso,

$$A_{i^*}(x' + \lambda c) = A_{i^*}x' + \lambda A_{i^*}c \leq A_{i^*}x' + U A_{i^*}c \leq A_{i^*}x' + \lambda_i A_{i^*}c = b_i.$$

- $i \in N$. A demonstração é análoga.

(d₂) Tomemos agora $x' + \lambda c \in P(A, b)$. Suponhamos que $\lambda \notin [L, U]$, por exemplo, $\lambda < L$. Então, existe $i \in N$ com $\lambda < \lambda_i$, e com isso,

$$A_{i^*}(x' + \lambda c) = A_{i^*}x' + \lambda A_{i^*}c > A_{i^*}x' + \lambda_i A_{i^*}c = b_i,$$

contradizendo o fato de que $x' + \lambda c \in P(A, b)$. Da mesma forma, $\lambda > U$ leva a um absurdo. Resta, então, mostrar que $x' \in P(D, d)$. Mas, isso é consequência do fato de que o sistema $Dx \leq d$ é combinação cônica de $Ax \leq b$; e assim, $A(x' + \lambda c) \leq b$ implica $D(x' + \lambda c) \leq d$. Como $Dc = 0$, segue que $Dx' = D(x' + \lambda c) \leq d$. \square

Do teorema acima, segue imediatamente o seguinte resultado.

Proposição 1.3.5 *Se $H = P(A', b')$ é um poliedro, então a projeção P_H , de $P(A, b)$ sobre H em uma direção c é um poliedro.*

Prova. Do Teorema 1.3.4 (c) temos que $P_H = P(A', b') \cap P(D, d)$, ou seja,

$$P_H = P\left(\left(\begin{array}{c} A' \\ D \end{array}\right), \left(\begin{array}{c} b' \\ d \end{array}\right)\right).$$

\square

Uma aplicação interessante do algoritmo de projeção é o método de eliminação da j -ésima variável de Fourier–Motzkin. Esse método permite, através da projeção de um poliedro no hiperplano $\{x \in \mathbb{R}^n \mid x_j = 0\}$, a eliminação de uma variável. Em seguida veremos uma motivação para o método.

Algoritmo ELIMINAÇÃO DE FOURIER–MOTZKIN

Entrada: Um poliedro $P(A, b)$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$,
variável j (a direção c será e_j).

Saída: Um poliedro $P(D, d)$, $D \in \mathbb{R}^{r \times n}$, $d \in \mathbb{R}^r$
(r é calculado no algoritmo) tais que, a projeção de
 $P(A, b)$ sobre $\{x \in \mathbb{R}^n \mid x_j = 0\}$, na direção e_j , é o
conjunto $\{x \in P(D, d) \mid x_j = 0\}$.

Particione M , o conjunto dos índices das linhas de A , nas partes

$$N := \{i \in M \mid a_{ij} < 0\},$$

$$Z := \{i \in M \mid a_{ij} = 0\},$$

$$P := \{i \in M \mid a_{ij} > 0\};$$

Faça $r := |Z \cup (N \times P)|$; defina $R := \{1, \dots, r\}$;

Construa uma bijeção $p : R \rightarrow Z \cup (N \times P)$;

Para cada $i := 1, \dots, r$ **faça**

Se $p(i) \in Z$

então $D_{i*} := A_{p(i)*}$; $d_i := b_{p(i)}$;

senão seja $p(i) = (s, t) \in N \times P$;

$$D_{i*} := a_{tj}A_{s*} - a_{sj}A_{t*};$$

$$d_i := a_{tj}b_s - a_{sj}b_t;$$

Devolva D e d .

O método descrito acima pode ser usado para resolver o problema de determinar se um dado poliedro é ou não-vazio. A proposição seguinte mostra como podemos usar o método de eliminação de Fourier–Motzkin para resolver esse problema.

Proposição 1.3.6 *Seja $P(A, b)$ um poliedro e $P(D, d)$ o poliedro obtido aplicando o algoritmo de projeção descrito acima. Então,*

$$P(A, b) \neq \emptyset \text{ se e só se } P(D, d) \neq \emptyset.$$

Prova. Inicialmente suponha que $P(D, d) = \emptyset$. Como a projeção de $P(A, b)$ sobre si mesmo é o próprio $P(A, b)$, então usando o fato de que $P_H = H \cap P(D, d)$, temos que $P(A, b) = P(A, b) \cap P(D, d)$; e portanto, $P(A, b) = \emptyset$.

Agora suponhamos que exista $x \in P(D, d)$. Pelo item (d_1) do Teorema 1.3.4, segue que o poliedro $P(A, b)$ também é não-vazio. \square

Observe que podemos aplicar iterativamente o método de eliminação de Fourier–Motzkin, começando com o poliedro $P(A, b)$ e fazendo projeções em sucessivas direções (nos hiperplanos

correspondentes). Seja $\{c_1, c_2, \dots, c_n\}$ uma base ortogonal de \mathbb{R}^n (por exemplo a base ortogonal canônica) e projete o poliedro $P(A, b)$ na direção c_1 . Obtemos, então, segundo a proposição anterior, uma matriz D_1 e um vetor d_1 tais que:

- (i) $D_1 c_1 = 0$,
- (ii) existe $U_1 \in \mathbb{R}_+^{r_1 \times m}$ com $U_1 A = D_1$ e $U_1 b = d_1$ e
- (iii) $P(A, b) \neq \emptyset$ se e só se $P(D_1, d_1) \neq \emptyset$.

Aplicando o método novamente, agora projetando o poliedro $P(D_1, d_1)$ na direção c_2 obtemos analogamente uma matriz D_2 e um vetor d_2 tais que:

- (i) $D_2 c_2 = 0$,
- (ii) existe $U_2 \in \mathbb{R}_+^{r_2 \times r_1}$ com $U_2 D_1 = D_2$ e $U_2 d_1 = d_2$ e
- (iii) $P(D_1, d_1) \neq \emptyset$ se e só se $P(D_2, d_2) \neq \emptyset$.

Combinando o conhecimento que temos da matriz D_1 e da matriz D_2 , obtemos que $D_2 c_1 = U_2 D_1 c_1 = 0$. Além disso, $d_2 = U_2 d_1 = U_2 U_1 b$. Assim, temos as seguintes relações (para não sobrecarregar a notação, chamamos a matriz $U_2 U_1$ de U_2):

- (i) $D_2 c_1 = D_2 c_2 = 0$,
- (ii) existe $U_2 \in \mathbb{R}_+^{r_2 \times m}$ com $U_2 A = D_2$ e $U_2 b = d_2$ e
- (iii) $P(A, b) \neq \emptyset$ se e só se $P(D_2, d_2) \neq \emptyset$.

Aplicando o mesmo método para todos os vetores da base ortogonal, obtemos

- (i) $D_n c_1 = D_n c_2 = \dots = D_n c_n = 0$,
- (ii) existe $U_n \in \mathbb{R}_+^{r_n \times m}$ com $U_n A = D_n$ e $U_n b = d_n$ e
- (iii) $P(A, b) \neq \emptyset$ se e só se $P(D_n, d_n) \neq \emptyset$.

Como $\{c_1, \dots, c_n\}$ é uma base de \mathbb{R}^n , então $D_n = 0$. Assim, é fácil verificar se o poliedro $P(D_n, d_n)$ é ou não-vazio. Note que tal poliedro é vazio se e só se $d_n \not\geq 0$.

Vale aqui ressaltar que, considerando a base ortogonal canônica $\{e_1, e_2, \dots, e_p\}$, cada vez que fazemos uma projeção em uma direção e_j , obtemos um conjunto de “dimensão” menor (no sentido de que, a cada projeção que efetuamos, obtemos conjuntos em que mais uma coordenada x_j passa a ser igual a zero). Dessa forma, alguns problemas relativos a um dado poliedro (inicial) podem ser transformados em problemas relativos a um conjunto muito mais simples, obtido por projeção.

Para trabalharmos com tais conjuntos, é conveniente identificarmos $\{x \in \mathbb{R}^n \mid x_n = 0\}$ com \mathbb{R}^{n-1} , o que nos permite falar de projeção de um poliedro $P \subseteq \mathbb{R}^n$ sobre \mathbb{R}^k , onde $0 < k < n$.

Seja $P(A, b) \subseteq \mathbb{R}^n$ um poliedro e k um inteiro positivo tal que $k + p = n$. A **projeção de $P(A, b)$ sobre \mathbb{R}^k** é definida como sendo o conjunto

$$Q := \left\{ x \in \mathbb{R}^k \mid \text{existe } y \in \mathbb{R}^p \text{ com } \begin{pmatrix} x \\ y \end{pmatrix} \in P(A, b) \right\}.$$

Basicamente, para se obter tal projeção, basta considerar a projeção de $P(A, b)$ sobre \mathbb{R}^n , nas direções $\{e_n, e_{n-1}, \dots, e_{n-p+1}\}$, e depois considerar o conjunto dos vetores $x = (x_1, x_2, \dots, x_k)$ tal que o vetor n -dimensional $(x_1, x_2, \dots, x_k, 0, \dots, 0)$ pertence a essa projeção.

O método de Fourier–Motzkin pode ser aplicado para resolver problemas de pequeno porte, mas requer muito tempo quando a matriz é grande, já que o número de operações pode crescer exponencialmente. O dual do método de eliminação de Fourier–Motzkin consiste em eliminar uma inequação por iteração, introduzindo novas variáveis. Este método pode ser usado para achar todos os vértices e raios extremais de um cone (veja definições a seguir). De fato, alguns programas de computador implementam as idéias descritas aqui, e são muito úteis para se determinar a estrutura de poliedros de dimensão pequena (veja, por exemplo, o programa PORTA, disponível por ftp¹).

1.4 Dualidade em Programação Linear

Um resultado interessante que pode ser obtido como consequência do método de eliminação de Fourier–Motzkin é o **Lema de Farkas**.

Lema 1.4.1 *Seja $A \in \mathbb{R}^{m \times n}$ uma matriz e $b \in \mathbb{R}^m$ um vetor. Então, existe um vetor x com $Ax \leq b$ se e somente se para cada vetor $y \geq 0$ com $y^\top A = 0$ temos $y^\top b \geq 0$.*

Prova. Suponha que $P(A, b)$ seja não-vazio. Então, existe x' tal que $Ax' \leq b$. Tome agora $y' \geq 0$ com $y'^\top A = 0$. Então, multiplicando $Ax' \leq b$ por y' , obtemos $y'^\top Ax' \leq y'^\top b$. Mas, como $y'^\top A = 0$, obtemos $y'^\top b \geq 0$.

Para provar a implicação reversa, suponha por absurdo que $P(A, b)$ seja vazio. Aplicando o método de projeção visto na última seção, esse poliedro é vazio se e somente se existe um vetor $y \geq 0$ em \mathbb{R}^m com $y^\top A = 0$ e $y^\top b < 0$ (uma linha da matriz U^n), contrariando a hipótese acima. \square

Outras versões do Lema de Farkas podem ser encontradas na literatura. Em alguns textos, esses lemas são também chamados de **lemas alternativos**.

¹Endereço: <http://www.iwr.uni-heidelberg.de/iwr/comopt/soft/PORTA/readme.html>

Lema 1.4.2 (Lema de Farkas – caso geral) Para matrizes A, B, C e D e vetores a e b de dimensões apropriadas temos que

$$\left. \begin{array}{l} \text{Existem } x, y \text{ tais que} \\ Ax + By \leq a \\ Cx + Dy = b \\ x \geq 0 \end{array} \right\} \vee \left\{ \begin{array}{l} \text{Existem } u, v \text{ tais que} \\ u^\top A + v^\top C \geq 0 \\ u^\top B + v^\top D = 0 \\ u \geq 0 \\ u^\top a + v^\top b < 0 \end{array} \right.$$

O símbolo \vee (ou exclusivo) indica que uma das alternativas ocorre, mas nunca ambas ao mesmo tempo.

Prova. Segue usando o Lema 1.4.1 (deixamos como exercício para o leitor). \square

Considerando casos especiais do Lema 1.4.2, obtemos os seguintes resultados.

Lema 1.4.3 (Lema de Farkas – casos especiais) Sejam $A \in \mathbb{R}^{m \times n}$ e $b \in \mathbb{R}^m$. Então, valem as seguintes afirmações:

- (a) Existe x tal que $Ax \leq b$ \vee existe $u \geq 0$ tal que $u^\top A = 0$ e $u^\top b < 0$;
- (b) Existe $x \geq 0$ tal que $Ax \leq b$ \vee existe $u \geq 0$ tal que $u^\top A \geq 0$ e $u^\top b < 0$;
- (c) Existe $x \geq 0$ tal que $Ax = b$ \vee existe u tal que $u^\top A \geq 0$ e $u^\top b < 0$;
- (d) Existe x tal que $Ax = b$ \vee existe u tal que $u^\top A = 0$ e $u^\top b < 0$.

Prova. Imediata a partir do Lema 1.4.2. \square

Em alguns casos é conveniente formular a versão poliédrica dos lemas alternativos. No próximo lema mostramos dois desses exemplos.

Lema 1.4.4 (Lema de Farkas – versão poliédrica) As seguintes asserções são válidas:

- (i) $P(A, b) \neq \emptyset \vee P^= \left(\left(\begin{array}{c} A^\top \\ b^\top \end{array} \right), \left(\begin{array}{c} 0 \\ -1 \end{array} \right) \right) \neq \emptyset$;
- (ii) $P^=(A, b) \neq \emptyset \vee P \left(\left(\begin{array}{c} -A^\top \\ b^\top \end{array} \right), \left(\begin{array}{c} 0 \\ -1 \end{array} \right) \right) \neq \emptyset$.

Prova. O item (i) Segue do Lema 1.4.3, fazendo uma mudança de escala do valor dos vetores solução da alternativa do lado direito (precisamos transformá-los em $\mathbf{1}$). A prova do item (ii) é análoga à do item (i). \square

Exercício 1.4.5 Dê a versão poliédrica do Lema 1.4.3 (b) e (d).

Exercício 1.4.6 Dê a versão poliédrica do Lema 1.4.2.

Observe que o Lema de Farkas nos dá um certificado para a não-existência de solução de sistemas de inequações (ou equações) lineares. Podemos, por exemplo, ler o Lema 1.4.3 (d) da seguinte forma: para provar que um sistema $Ax = b$ não tem solução, basta exibir um vetor u tal que $u^\top A = 0$ e $u^\top b < 0$.

Vamos usar esse resultado para demonstrar algumas propriedades a respeito de dualidade em programação linear. Como veremos, dualidade em programação linear pode ser visto como a versão de otimização do Lema de Farkas.

Considere o seguinte problema de programação linear

$$(P) \quad \max \quad c^\top x \\ Ax \leq b$$

Dizemos que um vetor x é uma **solução viável** do problema acima, se x satisfaz $Ax \leq b$. Uma solução viável x^* tal que $c^\top x^* \geq c^\top x$ para toda solução viável x é chamada de **solução ótima** do problema.

Podemos reescrever o problema (P) da seguinte maneira:

$$\max \quad z \\ z \leq c^\top x \\ Ax \leq b$$

Ou seja,

$$\max \quad z \\ \begin{pmatrix} 1 & -c^\top \\ 0 & A \end{pmatrix} \begin{pmatrix} z \\ x \end{pmatrix} \leq \begin{pmatrix} 0 \\ b \end{pmatrix}$$

Ou ainda, transpondo z para o lado direito da desigualdade,

$$\max \quad z \\ \begin{pmatrix} -c^\top \\ A \end{pmatrix} x \leq \begin{pmatrix} -z \\ b \end{pmatrix}$$

Para $z \in \mathbb{R}$ fixo, o conjunto solução do problema acima é um poliedro em \mathbb{R}^n . Para cada $z \in \mathbb{R}$, seja

$$P_z := \left\{ x \in \mathbb{R}^n \mid \begin{pmatrix} -c^\top \\ A \end{pmatrix} x \leq \begin{pmatrix} -z \\ b \end{pmatrix} \right\}.$$

Com isso, resolver o problema acima é equivalente a resolver

$$\max\{z \mid P_z \neq \emptyset\}.$$

Note que P_z pode ser vazio para todo $z \in \mathbb{R}$. Nesse caso (P) não tem solução ótima. Caso contrário, podemos tentar limitar superiormente o valor da solução resolvendo o seguinte problema:

$$\min\{z \mid P_z = \emptyset\}.$$

Aplicando o Lema de Farkas (Lema 1.4.3(a)), $P_z = \emptyset$ se e somente se o sistema

$$\begin{aligned} y^\top A &= \lambda c^\top \\ y^\top b &< \lambda z \\ y &\geq 0 \\ \lambda &\geq 0 \end{aligned}$$

tem solução. Sob a hipótese de que (P) tem solução, é fácil ver (podemos dividir as equações por λ) que o sistema acima tem solução se e somente se

$$\begin{aligned} y^\top A &= c^\top \\ y^\top b &< z \\ y &\geq 0 \end{aligned}$$

tem solução. Assim, determinar o menor z tal que P_z é vazio é equivalente a resolver o seguinte problema:

$$(D) \quad \begin{aligned} \min \quad & y^\top b \\ & y^\top A = c^\top \\ & y \geq 0. \end{aligned}$$

Note que, partimos de um problema de programação linear — o problema (P) — e recaímos em um outro problema de programação linear — o problema (D). Esses dois pares de problemas são chamados de **primal** e **dual**, respectivamente. Mais especificamente, (D) é o dual do problema (P). É imediato que se x' é uma solução viável de (P) e y' é uma solução viável de (D), então $y'^\top b \geq y'^\top Ax' \geq c^\top x'$.

O fato de que “se (P) e (D) têm soluções viáveis, então o valor ótimo de (D) é sempre maior ou igual ao valor ótimo de (P)”, é referido como **Teorema Fraco de Dualidade**. Podemos, na verdade, mostrar um resultado mais forte do que esse. Se (P) e (D) têm soluções ótimas, então os valores ótimos dos dois problemas são iguais. Vamos mostrar esse resultado usando o Lema de Farkas.

Teorema 1.4.7 (Teorema Forte de Dualidade Linear) *Sejam (P) e (D) os problemas primal e dual enunciados acima. Então, ambos os problemas têm solução ótima e os valores ótimos são iguais se e somente se esses problemas têm soluções viáveis.*

Prova. Claramente, se (P) e (D) têm soluções ótimas, então também têm soluções viáveis.

Para mostrar o outro sentido da implicação, observe que se (P) e (D) são viáveis, então o valor ótimo de (D) é maior do que ou igual ao valor ótimo de (P), como observamos acima. Resta mostrar que existem x' e y' tais que $Ax' \leq b$, $y' \geq 0$, $y'A = c$ e ainda $c^\top x' \geq y'^\top b$. Reescrevendo utilizando a linguagem de poliedros, devemos mostrar que o poliedro $P(A', b')$ é não-vazio, onde,

$$A' := \begin{pmatrix} A & 0 \\ 0 & -I \\ 0 & A^\top \\ 0 & -A^\top \\ -c & b^\top \end{pmatrix} \quad \text{e} \quad b' := \begin{pmatrix} b \\ 0 \\ c \\ -c \\ 0 \end{pmatrix}.$$

Aplicando o Lema de Farkas, isso ocorre se e somente se para cada vetor-linha $w \geq 0$ com $wA' = 0$, temos $wb' \geq 0$. Tome $w = (w_1, w_2, w_3, w_4, w_5)$ que satisfaz $w \geq 0$ e $wA' = 0$, ou seja, $w_1A - w_5c = 0$ e $w_5b^\top + w_3A^\top - w_4A^\top \geq 0$ (eliminando w_2). Queremos mostrar que $w_1b + (w_3 - w_4)c \geq 0$. Supondo $w_5 > 0$, $w_1b = b^\top w_1^\top = w_5^{-1}w_5b^\top w_1^\top \geq w_5^{-1}(w_4 - w_3)A^\top w_1^\top = w_5^{-1}(w_4 - w_3)w_1A = w_5^{-1}(w_4 - w_3)w_5c = (w_4 - w_3)c$.

Caso $w_5 = 0$, tome x' e y' pontos viáveis de (P) e (D), respectivamente. Então, $w_1b \geq w_1Ax' = 0 \geq (w_4 - w_3)A^\top y' = (w_4 - w_3)c$. \square

A seguir enunciamos um resultado que caracteriza completamente a relação entre o valor ótimo primal e dual, mesmo quando algum dos problemas é ilimitado ou não tem solução viável. A demonstração fica por conta dos leitores.

Teorema 1.4.8 *Sejam (P) e (D) os problemas primal e dual como enunciados acima. Além disso, sejam P e z^* o conjunto de soluções viáveis de (P) e o valor ótimo primal, respectivamente. Defina, D e u^* analogamente. Caso (P) não tenha solução viável, defina $z^* = -\infty$, e caso (P) não tenha solução ótima (seja ilimitado), $z^* = +\infty$ (análogo para (D)). Assim, (P) e (D) têm soluções ótimas com o mesmo valor se e somente se $P \neq \emptyset$ e $D \neq \emptyset$. Além disso,*

- (a) *Se $z^* = +\infty$ então $D = \emptyset$;*
- (b) *Se $u^* = -\infty$ então $P = \emptyset$;*
- (c) *Se $P = \emptyset$, então $D = \emptyset$ ou $u^* = -\infty$;*
- (d) *Se $D = \emptyset$, então $P = \emptyset$ ou $z^* = +\infty$.*

\square

No teorema a seguir, caracterizamos quando duas soluções viáveis para o problema primal e dual são ótimas.

Teorema 1.4.9 (Teorema de Folgas Complementares) *Sejam (P) e (D) os problemas primal e dual como enunciados acima e suponha que x e y são soluções viáveis de (P) e (D), respectivamente. Então, as soluções são ótimas se e somente se, para todo j ,*

$$y_j > 0 \text{ implica que } A_{j*}x = b_j.$$

\square

Exercício 1.4.10 *Repita o raciocínio mostrado nesta seção para construir o problema dual do seguinte problema $\min\{c^\top x \mid Ax \geq b\}$.*

1.5 Descrição de Poliedros

Até agora vimos um poliedro como a intersecção de um número finito de semi-espacos (ou o conjunto de soluções de um sistema de inequações lineares). Uma tal descrição é chamada *representação externa* de um poliedro. Há outras formas equivalentes para se descrever os pontos de um poliedro. Por exemplo, podemos representar um poliedro como a soma do fecho convexo de seus vértices com o conjunto de pontos do cone gerado pelos seus raios extremais. Esta representação é conhecida como *representação interna* do poliedro. A equivalência entre estas duas representações será o tema desta seção.

Inicialmente tomemos uma matriz $A \in \mathbb{R}^{m \times n}$. No próximo teorema, devido a Weyl, mostramos que o espaço linear, o espaço afim, o fecho convexo e o cone gerados por A são poliedros.

Teorema 1.5.1 (Teorema de Weyl, 1935) *Para toda matriz $A \in \mathbb{R}^{m \times n}$, temos que $\text{lin}(A)$, $\text{afim}(A)$, $\text{conv}(A)$ e $\text{cone}(A)$ são poliedros.*

Prova. Vamos mostrar que $\text{conv}(A)$ é um poliedro; os outros casos são análogos (deixaremos como exercício). Temos que

$$\begin{aligned} \text{conv}(A) &= \{x \in \mathbb{R}^m \mid \text{existe } y \geq 0, \sum_i y_i = 1 \text{ com } x = Ay\} \\ &= \left\{ x \in \mathbb{R}^m \mid \text{existe } y \in \mathbb{R}^n, \begin{pmatrix} 0 & -I \\ 0 & \mathbf{1} \\ 0 & -\mathbf{1} \\ I & -A \\ -I & A \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \leq \begin{pmatrix} 0 \\ \mathbf{1} \\ -\mathbf{1} \\ 0 \\ 0 \end{pmatrix} \right\}, \end{aligned}$$

que é a projeção de um poliedro em \mathbb{R}^{n+m} sobre \mathbb{R}^m . □

O próximo teorema nos diz que a soma de dois poliedros também é um poliedro. Lembramos que, se P_1 e P_2 são poliedros, então

$$P_1 + P_2 := \{x \mid x = x_1 + x_2, \ x_1 \in P_1 \text{ e } x_2 \in P_2\}.$$

Teorema 1.5.2 *Se P_1 e P_2 são poliedros, então $P_1 + P_2$ é também um poliedro.*

Prova. Exercício. □

Corolário 1.5.3 *Se $A \in \mathbb{R}^{m \times n}$ e $B \in \mathbb{R}^{m \times n'}$, então*

$$\text{conv}(A) + \text{cone}(B)$$

é um poliedro. □

1.5.1 Polaridade de cones

Sabemos da seção anterior que se $A \in \mathbb{R}^{m \times n}$ e $B \in \mathbb{R}^{m \times n'}$, então $\text{conv}(A) + \text{cone}(B)$ é um poliedro. Na verdade, vale um resultado muito importante, que é de certa forma, a recíproca da afirmação acima. Mais precisamente, veremos que

$$\text{todo poliedro é da forma } \text{conv}(A) + \text{cone}(B).$$

Para provar isso, vamos introduzir o conceito de **polaridade de cones** e usar uma versão geométrica do Lema de Farkas.

Consideremos inicialmente a seguinte versão do Lema de Farkas (Lema 1.4.3(c)):

$$\text{existe } x \geq 0 \text{ tal que } Ax = b \text{ se e só se } A^\top u \geq 0 \text{ implica } u^\top b \geq 0 \text{ para todo } u.$$

Note que o resultado acima caracteriza todos os vetores b para os quais o sistema $Ax \leq b$, $x \geq 0$ tem solução.

Lembrando que, por definição,

$$\text{cone}(A) = \{b \in \mathbb{R}^m \mid \text{existe } x \geq 0 \text{ tal que } Ax = b\},$$

usando a equivalência acima, conclui-se imediatamente a seguinte proposição.

Proposição 1.5.4 *Para toda matriz $A \in \mathbb{R}^{m \times n}$,*

$$\text{cone}(A) = \{b \in \mathbb{R}^m \mid u^\top b \leq 0 \text{ para todo } u \in P(A^\top, 0)\}.$$

□

Geometricamente, a Proposição 1.5.4 pode ser assim descrita. O conjunto dos vetores b para os quais o sistema $Ax = b$, $x \geq 0$ tem solução é precisamente o conjunto de todos os vetores b que formam um ângulo obtuso (maior que 90°) com cada um dos vetores do cone $P(A^\top, 0)$.

Mais genericamente, vamos dar um nome especial ao conjunto dos vetores que formam um ângulo obtuso com cada um dos vetores de um conjunto dado. Seja $S \subseteq \mathbb{R}^n$ e defina

$$S^\circ := \{y \in \mathbb{R}^n \mid y^\top x \leq 0 \text{ para todo } x \in S\}.$$

O conjunto S° é chamado de **cone polar** de S . Denotamos por $S^{\circ\circ}$ o conjunto $(S^\circ)^\circ$. Nos exercícios a seguir, citamos algumas relações entre os conjuntos S , S° e $S^{\circ\circ}$. Observe que $S^\perp \subseteq S^\circ$, onde $S^\perp := \{y \in \mathbb{R}^n \mid x^\top y = 0 \text{ para todo } x \in S\}$ é o **complemento ortogonal** de S .

Fazendo uso da definição de cone polar, a Proposição 1.5.4 pode ser assim reescrita.

Proposição 1.5.5 *Para toda matriz $A \in \mathbb{R}^{m \times n}$,*

$$P(A, 0)^\circ = \text{cone}(A^\top).$$

□

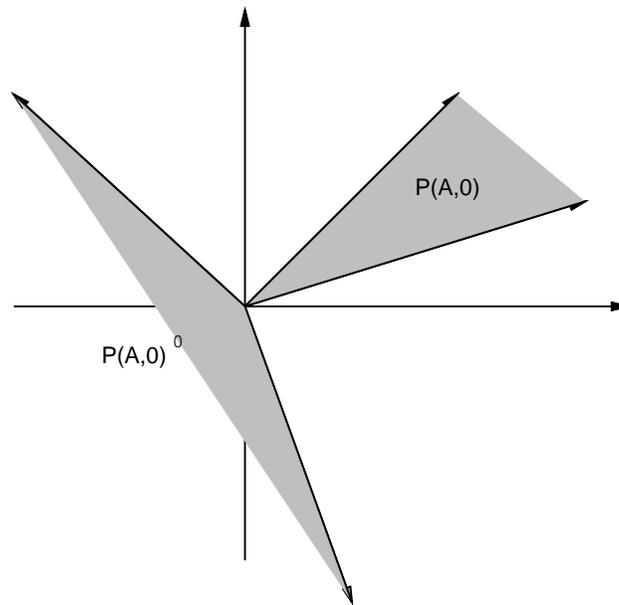


Figura 1.3: Cone polar de S

Exemplo 1.5.6 Seja $A = \begin{pmatrix} -2 & 2 \\ 1 & -3 \end{pmatrix}$. Os cones $P(A, 0)$ e $P(A, 0)^\circ$ estão indicados na figura abaixo.

Exercício 1.5.7 Prove as seguintes afirmações para $S, S_i \subseteq \mathbb{R}^n$:

- (a) $S_i \subseteq S_j$ implica $S_j^\circ \subseteq S_i^\circ$.
- (b) $S \subseteq S^{\circ\circ}$.
- (c) $\left(\bigcup_{i=1}^k S_i\right)^\circ = \bigcap_{i=1}^k S_i^\circ$.
- (d) $S^\circ = \text{cone}(S^\circ) = (\text{cone}(S))^\circ$.
- (e) $S = \text{lin}(S)$ implica que $S^\circ = S^\perp$.

Exercício 1.5.8 Vale a recíproca da afirmação (e) acima?

Exercício 1.5.9 Se nas afirmações (a) a (e) substituirmos “ \circ ” por “ \perp ” elas continuam válidas?

Exercício 1.5.10 Para quais conjuntos $S \subseteq \mathbb{R}^n$ vale

- (i) $S^\circ = S^{\circ\circ}$?
- (ii) $S = S^\circ$?

Usando a afirmação (d) do Exercício 1.5.7, pode se provar facilmente o seguinte fato.

Proposição 1.5.11 Para toda matriz $A \in \mathbb{R}^{m \times n}$,

$$\text{cone}(A^\top)^\circ = P(A, 0).$$

Prova. Temos que $\text{cone}(A^\top)^\circ = \text{cone}((A^\top)^\circ) = (A^\top)^\circ = \{x \mid Ax \leq 0\} = P(A, 0)$. \square

O seguinte resultado segue diretamente das proposições 1.5.5 e 1.5.11.

Teorema 1.5.12 Para toda matriz $A \in \mathbb{R}^{m \times n}$,

$$P(A, 0)^{\circ\circ} = P(A, 0) \quad e \quad \text{cone}(A)^{\circ\circ} = \text{cone}(A).$$

Prova. Temos que $P(A, 0) = (\text{cone}(A^\top))^\circ = (P(A, 0)^\circ)^\circ = P(A, 0)^{\circ\circ}$. Ademais, $\text{cone}(A) = P(A^\top, 0)^\circ = (\text{cone}(A)^\circ)^\circ = \text{cone}(A)^{\circ\circ}$. \square

Com isso, podemos enunciar o Teorema de Minkowski.

Teorema 1.5.13 (Minkowski, 1896) Um subconjunto $K \subseteq \mathbb{R}^n$ é um cone poliédrico se e só se K é o fecho cônico de um número finito de vetores. Em outras palavras, para cada matriz $A \in \mathbb{R}^{m \times n}$, existe uma matriz $B \in \mathbb{R}^{n \times p}$ tal que

$$P(A, 0) = \text{cone}(B);$$

e, reciprocamente, para cada matriz $B \in \mathbb{R}^{n \times p}$, existe uma matriz $A \in \mathbb{R}^{m \times n}$ tal que $P(A, 0) = \text{cone}(B)$.

Prova. Temos que $P(A, 0) = \text{cone}(A^\top)^\circ = P(B^\top, 0)^\circ = \text{cone}(B)$. \square

Estamos agora em condições de provar o resultado central desta seção.

Teorema 1.5.14 Considere $A \in \mathbb{R}^{m \times n}$ e $b \in \mathbb{R}^m$. Então existem conjuntos finitos $V, E \subseteq \mathbb{R}^n$ tais que

$$P(A, b) = \text{conv}(V) + \text{cone}(E).$$

Prova. Tome o cone

$$H := P\left(\left(\begin{array}{cc} A & -b \\ 0 & -1 \end{array}\right), \left(\begin{array}{c} 0 \\ 0 \end{array}\right)\right).$$

Observe que $x \in P(A, b)$ se e só se $\begin{pmatrix} x \\ 1 \end{pmatrix} \in H$. Como H é poliédrico, existe uma matriz B tal que $H = \text{cone}(B)$. Da definição de H , segue que a última linha da matriz B só tem elementos

não negativos. Usando mudança de escala e permutação na matriz B , podemos transformá-la em $\tilde{B} = \begin{pmatrix} V & E \\ \mathbf{1}^\top & 0^\top \end{pmatrix}$, sendo $\text{cone}(\tilde{B}) = H$. Com isso,

$$\begin{aligned} x \in P(A, b) &\iff \begin{pmatrix} x \\ 1 \end{pmatrix} \in H \iff \begin{pmatrix} x \\ 1 \end{pmatrix} \in \text{cone}(\tilde{B}) \iff \\ &x = V\lambda + E\nu, \lambda \mathbf{1} = 1, \lambda, \nu \geq 0 \iff x \in \text{conv}(V) + \text{cone}(E). \end{aligned}$$

□

Corolário 1.5.15 *Um subconjunto $P \subseteq \mathbb{R}^n$ é um polítopo se e só se P é o fecho convexo de um número finito de vetores.*

Prova. Se P é um polítopo, sabemos que existem conjuntos finitos V e E tais que $P = \text{conv}(V) + \text{cone}(E)$. Caso $E \setminus \{0\} \neq \emptyset$, então P não é limitado, e portanto não é um polítopo. Logo, $P = \text{conv}(V)$.

Para mostrar a implicação contrária, basta notar que se P é o fecho convexo de um número finito de vetores, então é um poliedro. Além disso, é fácil ver que P é limitado. □

Juntando todos os resultados dessa seção, podemos enunciar o seguinte resultado.

Teorema 1.5.16 *Um subconjunto $P \subseteq \mathbb{R}^n$ é um poliedro se e só se P é a soma de um polítopo e um cone poliédrico, ou seja, existem conjuntos finitos $V, E \subseteq \mathbb{R}^n$ tais que*

$$P = \text{conv}(V) + \text{cone}(E).$$

Prova. Exercício. □

1.6 Faces

Nesta seção estudaremos formas de caracterizar as faces de um poliedro. Para isso, precisamos de algumas definições.

Seja $S \subseteq \mathbb{R}^n$, $a \in \mathbb{R}^n$ e $\alpha \in \mathbb{R}$. Uma inequação $a^\top x \leq \alpha$ é **válida** em relação a S se $S \subseteq \{x \in \mathbb{R}^n \mid a^\top x \leq \alpha\}$. Quando o conjunto S está claro pelo contexto, dizemos simplesmente que a inequação é válida. Um hiperplano $H = \{x \mid a^\top x = \alpha\}$ é um **hiperplano suporte** de S se $a^\top x \leq \alpha$ para todo $x \in S$ e $S \cap H \neq \emptyset$.

Seja $P \subseteq \mathbb{R}^n$ um poliedro. Um conjunto $F \subseteq P$ é uma **face** de P , se existe uma inequação $a^\top x \leq \alpha$ válida em relação a P , tal que

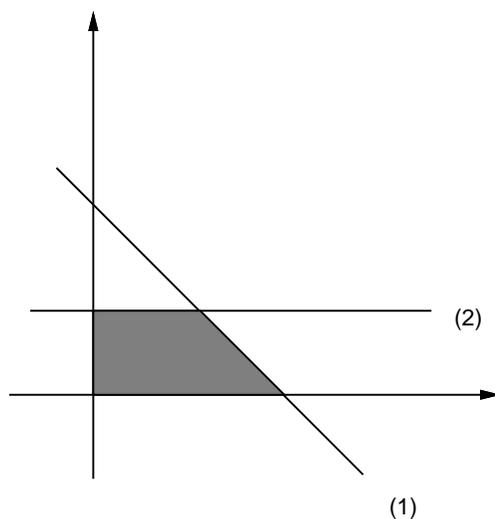
$$F = P \cap \{x \mid a^\top x = \alpha\}.$$

Dizemos que F é uma face **própria** se $F \neq P$; e F é **não-trivial** se $\emptyset \neq F \neq P$. Se $a^\top x \leq \alpha$ é válida em relação a P , então dizemos que $P \cap \{x \mid a^\top x = \alpha\}$ é a face **induzida** ou **definida** por $a^\top x \leq \alpha$.

Claramente, faces de poliedros são também poliedros. Assim, podem ser definidas de diversas maneiras.

Exemplo 1.6.1 Considere o poliedro $P = P(A, b)$, onde

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{pmatrix} \quad e \quad b = \begin{pmatrix} 2 \\ 1 \\ 0 \\ 0 \end{pmatrix}.$$



O ponto $(2, 0)$ é a face induzida pela inequação $x \leq 2$ e também pela inequação $x - y \leq 2$. O segmento $[(1, 1), (2, 0)]$ é a face induzida pela inequação $x + y \leq 2$.

Proposição 1.6.2 Se $P \subseteq \mathbb{R}^n$ é um poliedro, então

- (a) P é uma face de si mesmo;
- (b) \emptyset é uma face de P ;
- (c) Se $F = \{x \in P \mid a^\top x = \alpha\}$ é uma face não-trivial de P , então $a \neq 0$.

Prova. Exercício. □

No próximo teorema mostramos que o conjunto das soluções ótimas de um problema de maximizar ou minimizar uma função linear sobre um poliedro P (ou seja, de um problema de programação linear), é sempre uma face de P . Esse resultado é de fundamental importância para justificar o funcionamento do algoritmo simplex para programação linear.

Teorema 1.6.3 *Sejam $P = P(A, b) \neq \emptyset$ um poliedro, $c \in \mathbb{R}^n$, $\gamma \in \mathbb{R}$ e*

$$z^* = \begin{cases} +\infty, & \text{caso } c^\top x \text{ seja ilimitado sobre } P, \\ \max\{c^\top x \mid x \in P\}, & \text{caso contrário.} \end{cases}$$

Então,

- (a) a inequação $c^\top x \leq \gamma$ é válida em relação a P se e só se $\gamma \geq z^*$;
- (b) $z^* < +\infty$ implica que $F = \{x \in P \mid c^\top x = z^*\}$ é uma face não-vazia de P e $\{x \mid c^\top x = z^*\}$ é um hiperplano suporte de P (caso $c \neq 0$).

Prova. (a) Suponha que $\gamma < z^*$. Então, existe $x^* \in P$ tal que $c^\top x^* = z^* > \gamma$, ou seja, a inequação $c^\top x \leq \gamma$ não é válida em relação a P .

Por outro lado, se $\gamma \geq z^*$, então $c^\top x \leq z^* \leq \gamma$ para todo $x \in P$. Logo, $c^\top x \leq \gamma$ é válida.

(b) Se $z^* < \infty$ então, por (a), segue que $c^\top x \leq z^*$ é válida em relação a P . Além disso, como z^* é limitado, existe $x^* \in P$ tal que $z^* = c^\top x^*$. Logo, F é uma face não-vazia de P . Se $c \neq 0$, então $\{x \mid c^\top x = z^*\}$ é um hiperplano suporte de P . \square

Dada a descrição de um poliedro $P = P(A, b)$ através de um sistema de inequações, e uma face F desse poliedro, é interessante determinar um conjunto de inequações desse sistema que definem F . Denotamos por $\text{ind}(A)$ o conjunto dos índices das linhas da matriz A . Definimos o **conjunto igualdade** de F como sendo

$$\text{ig}(F) := \{i \in \text{ind}(A) \mid A_{i*}x = b_i \text{ para todo } x \in F\}.$$

Para $I \subseteq \text{ind}(A)$, seja

$$\text{fa}(I) := \{x \in P \mid A_{I*}x = b_I\}.$$

O conjunto $\text{fa}(I)$ é chamado de **face induzida por I** .

Exercício 1.6.4 *Considere $P = P(A, b)$ e $I \subseteq \text{ind}(A)$. Prove que $\text{fa}(I)$ é uma face de P .*

Teorema 1.6.5 *Sejam $P = P(A, b)$ um poliedro e $F = \{x \in P \mid c^\top x = \gamma\}$ uma face não-vazia de P . Então,*

$$\text{ig}(F) = \{i \in \text{ind}(A) \mid \text{existe } u \geq 0 \text{ tal que } u_i > 0 \text{ e } u^\top A = c^\top, u^\top b = \gamma\}.$$

Prova. Seja $\widetilde{M} := \{i \in \text{ind}(A) \mid \text{existe } u \geq 0 \text{ tal que } u_i > 0 \text{ e } u^\top A = c^\top, u^\top b = \gamma\}$. Considere agora os pares de problemas primal e dual:

$$(P) \quad \max \quad c^\top x \quad \text{e} \quad (D) \quad \min \quad u^\top b \\ \text{Ax} \leq b \quad \quad \quad u^\top A = c^\top \\ u \geq 0$$

Sob a hipótese de que F é uma face não-vazia de P , os dois problemas têm solução ótima com o mesmo valor objetivo γ e F é o conjunto de soluções ótimas de P . Usando o Teorema das Folgas Complementares (Teorema 1.4.9), existem soluções ótimas x^* e u^* de (P) e (D) tais que

$$u_j^* > 0 \text{ se e só se } A_{j*}x^* = b_j.$$

Tomemos $i \in \text{ig}(F)$. Como $x^* \in F$, segue que $A_{i*}x^* = b_i$. Então, $i \in \widetilde{M}$.

Para mostrar a outra direção, seja $i \in \widetilde{M}$. Então, existe $u \geq 0$ tal que $u_i > 0$ e $u^\top A = c^\top, u^\top b = \gamma$. Neste caso, u é solução ótima de (D) . Usando o Teorema de Folgas Complementares, $A_{i*}x = b_i$ para todo $x \in F$; ou seja, $i \in \text{ig}(F)$. \square

O próximo teorema caracteriza as faces de um poliedro P .

Teorema 1.6.6 *Sejam $P = P(A, b)$ um poliedro e $F \neq \emptyset$ um subconjunto de P . As seguintes afirmações são equivalentes:*

- (a) F é uma face de P ;
- (b) existe $I \subseteq \text{ind}(A)$ tal que $F = \text{fa}(I)$;
- (c) $F = \text{fa}(\text{ig}(F))$.

Prova. [(a) \Rightarrow (b)] Seja $F = \{x \in P \mid c^\top x = \gamma\}$ uma face de P , e $I = \text{ig}(F)$. Então, sabemos que $F \subseteq \{x \in P \mid A_{I*}x = b_I\} := F'$. Vamos provar que $F' \subseteq F$. Seja $x \in F'$. Como $x \in P$, resta mostrar que $c^\top x = \gamma$. Para cada $i \in I$, pelo Teorema 1.6.5, existe um vetor $u^{(i)}$ tal que $u^{(i)} \geq 0$, $u_j^{(i)} > 0$ para algum j , $(u^{(i)})^\top A = c^\top$ e $(u^{(i)})^\top b = \gamma$. Considere

$$u := \sum_{i \in I} \frac{1}{|I|} u^{(i)}.$$

Por construção, $u_i > 0$ para todo $i \in I$ e além disso, $u_i = 0$ para todo $i \in \text{ind}(A) \setminus I$. Então, os vetores x e u são viáveis para os problemas (P) e (D) , definidos na prova do Teorema 1.6.5. Pelo Teorema de Folgas Complementares segue que tais soluções são ótimas. Logo, $c^\top x = u^\top b = \gamma$ e $x \in F$.

[(b) \Rightarrow (c)] Se $F = \text{fa}(I)$, então F é uma face de P . Da prova acima temos que $F = \text{fa}(J)$, onde $J = \text{ig}(F)$. Ou seja, $F = \text{fa}(\text{ig}(F))$.

[(c) \Rightarrow (a)] Segue do fato de que se $J \subseteq \text{ind}(A)$, então $\text{fa}(J)$ é uma face de P . \square

Note que o teorema anterior nos dá uma forma de obter faces de um poliedro $P(A, b)$. Para isso, basta escolhermos algumas inequações do sistema $Ax \leq b$ e exigirmos igualdade nessas inequações. Essa caracterização nos dá um limitante superior para o número de faces de um poliedro.

Corolário 1.6.7 *Se $P = P(A, b)$ com $A \in \mathbb{R}^{m \times n}$ e $b \in \mathbb{R}^m$, então P tem no máximo $2^m + 1$ faces.*

Prova. Pelo Teorema 1.6.6 o número máximo de faces não-vazias é dado pelo número de subconjuntos de $\text{ind}(A)$. \square

1.7 Dimensão

Nesta seção tratamos de um conceito importante na teoria de poliedros, que é a de **dimensão** de um conjunto. Algumas faces receberão nomes especiais, dependendo de sua dimensão. Inicialmente, lembramos que a **dimensão** de um conjunto F , denotada por $\dim(F)$ é d se F contém exatamente $d + 1$ vetores afim-independentes. Podemos relacionar a dimensão de uma face com seus pontos interiores, como vemos a seguir.

Um elemento x de um poliedro P é um **ponto interior** de P se x não está contido em nenhuma face própria de P .

Teorema 1.7.1 *Todo poliedro não-vazio tem um ponto interior.*

Prova. Seja $P = P(A, b)$, $I := \text{ig}(P)$ e $J := \text{ind}(A) \setminus I$. Se $I = \text{ind}(A)$, então P não tem nenhuma face própria e assim todo elemento de P é um ponto interior.

Caso $I \neq \text{ind}(A)$, então o sistema $Ax \leq b$ é equivalente a

$$\begin{cases} A_{I*}x = b_I \\ A_{J*}x \leq b_J \end{cases}$$

e para cada $j \in J$ existe $x^j \in P$ com $A_{j*}x^j < b_j$. Tome, então,

$$x' := \frac{1}{|J|} \sum_{j \in J} x^j.$$

Observe que $x' \in P$ e como $A_{j*}x' < b_j$, x' é ponto interior de P . \square

Exercício 1.7.2 *Seja F uma face do poliedro $P(A, b)$ e $x' \in F$. Então, x' é um ponto interior de F se e só se $\text{ig}(\{x'\}) = \text{ig}(F)$.*

Com isso, podemos mostrar o seguinte teorema que relaciona a dimensão de uma face F com o posto da matriz $A_{\text{ig}(F)*}$.

Teorema 1.7.3 *Se $F \neq \emptyset$ é uma face do poliedro $P(A, b) \subseteq \mathbb{R}^n$, então*

$$\dim(F) = n - \text{posto}(A_{\text{ig}(F)*}),$$

onde o $\text{posto}(A)$ é o número máximo de colunas linearmente independentes de A .

Prova. Seja $I := \text{ig}(F)$. Como

$$n = \text{posto}(A_{I^*}) + \dim(\text{núcleo}(A_{I^*})),$$

resta provar que $r := \dim(F) = \dim(\text{núcleo}(A_{I^*})) := s$.

Primeiramente, vamos mostrar que $r \leq s$. Como $\dim(F) = r$, então existem $r + 1$ vetores afim-independentes x_0, x_1, \dots, x_r em F . Com isso, $x_1 - x_0, x_2 - x_0, \dots, x_r - x_0$ são linearmente independentes e satisfazem $A_{I^*}(x_j - x_0) = 0$ para todo $j = 1, \dots, r$. Logo, $\text{núcleo}(A_{I^*})$ contém pelo menos $r + 1$ vetores afim-independentes. Além disso, $0 \in \text{núcleo}(A_{I^*})$, e portanto $s \geq r$.

Mostremos agora que $s \leq r$. Pelo Teorema 1.7.1, F contém um ponto interior x' , e aplicando o Exercício 1.7.2, $\text{ig}(\{x'\}) = \text{ig}(F) = I$. logo, $A_{I^*}x' = b_I$ e $A_{J^*}x' < b_J$ para $J := \text{ind}(A) \setminus I$.

Se $s = 0$, segue imediatamente que $s \leq r$. Assim, suponhamos que $s \geq 1$ e tomemos uma base $\{x^1, x^2, \dots, x^s\}$ do $\text{núcleo}(A_{I^*})$. Defina agora

$$\alpha_{jp} := \begin{cases} \infty, & \text{se } A_{j^*}x_p = 0, \\ (b_j - A_{j^*}x')/A_{j^*}x_p, & \text{caso contrário;} \end{cases}$$

e $\lambda := \min\{\alpha_{jp} \mid j \in J, p \in \{1, \dots, s\}\}$ (se $\alpha_{jp} = \infty$ para todo j, p , defina $\lambda \neq 0$ arbitrariamente). Seja $p \in \{1, \dots, s\}$ e note agora que para todo $i \in I$, temos

$$A_{i^*}(x' + \lambda x^p) = A_{i^*}x' + \lambda A_{i^*}x^p = A_{i^*}x' = b_i.$$

Para todo $j \in J$, temos que

$$\begin{aligned} A_{j^*}(x' + \lambda x^p) &= A_{j^*}x' + \lambda A_{j^*}x^p \\ &\leq A_{j^*}x' + \alpha_{jp}A_{j^*}x^p \\ &= A_{j^*}x' + b_j - A_{j^*}x' \\ &= b_j \end{aligned}$$

Vemos então que $x' + \lambda x^p \in F$ para todo $p \in \{1, \dots, s\}$. Como os vetores $\lambda x^1, \dots, \lambda x^s$ são linearmente independentes, os vetores $x', x' + \lambda x^1, \dots, x' + \lambda x^s$ são afim-independentes, e com isso $r \geq s$. \square

O próximo corolário reúne alguns resultados importantes a respeito da dimensão de faces de poliedros. Deixamos a prova a cargo do leitor.

Corolário 1.7.4 *Seja $P = P(A, b) \subseteq \mathbb{R}^n$ um poliedro não-vazio. Então,*

- (a) $\dim(P) = n - \text{posto}(A_{\text{ig}(P)^*})$;
- (b) se $\text{ig}(P) = \emptyset$, então P tem dimensão plena (i.e. $\dim(P) = n$);
- (c) se F é uma face própria de P , então $\dim(F) \leq \dim(P) - 1$.

\square

Faces de dimensão 0 são chamadas de **vértices** de um poliedro. Faces próprias maximais de um poliedro são chamadas de **facetras**. Nas próximas seções deste capítulo discutimos algumas propriedades dessas faces.

Exercício 1.7.5 *Sejam X e Y subconjuntos do \mathbb{R}^n tais que $X \subseteq Y$. Mostre que, se X é um conjunto afim, então $\dim(X) \leq \dim(Y)$.*

1.8 Facetas

Uma face não-trivial F de $P(A, b)$ é uma **faceta** se F não está contida em nenhuma outra face própria de $P(A, b)$. Algumas vezes, por simplicidade, se uma inequação induz uma faceta então chamamos tal inequação de faceta. Nesta seção mostramos que as facetras constituem um sistema “minimal” de inequações para representar um poliedro.

Seja $Ax \leq b$ um sistema de inequações. Para simplificar a notação, o conjunto $\text{ind}(A)$ será denotado por M . Seja $I \subseteq M$. O sistema $A_{I*}x \leq b_I$ é **redundante** em relação a $Ax \leq b$, se $P(A, b) = P(A_{M \setminus I*}, b_{M \setminus I})$. Se um sistema contém um subsistema redundante, então dizemos que $Ax \leq b$ é **redundante**; caso contrário, o sistema é **irredundante**.

Uma inequação $A_{i*}x \leq b_i$ irredundante em relação a $Ax \leq b$ é dita **equação implícita** se $i \in \text{ig}(P(A, b))$.

Um sistema $Ax \leq b$, $Cx = d$ é **irredundante** se $Ax \leq b$ não contém nenhuma inequação redundante em relação ao sistema $Ax \leq b$, $Cx \leq d$, $-Cx \leq -d$ e C tem posto-linha completo.

Observe que redundância ou irredundância não é uma propriedade de poliedros, mas de sistemas de inequações. Um poliedro pode ter várias descrições irredundantes (e várias descrições redundantes). Note que, dado um sistema redundante, não é verdade que, ao removermos as inequações redundantes desse sistema, obtemos um sistema irredundante.

Exercício 1.8.1 *Mostre que se a inequação $c^\top x \leq \delta$ é redundante com o sistema $Ax \leq b$, então a inequação pode ser escrita como combinação cônica das inequações de $Ax \leq b$.*

No restante desta seção mostramos algumas formas de caracterizar facetras de um poliedro. Mostramos também que, um sistema constituído pelas facetras de um poliedro é uma descrição irredundante desse poliedro.

Teorema 1.8.2 *Seja F uma faceta de $P = P(A, b)$. Então*

- (a) $\text{ig}(P) \subseteq \text{ig}(F)$;
- (b) *para todo $i \in \text{ig}(F) \setminus \text{ig}(P)$, $F = \text{fa}(\{i\})$.*

Prova. (a) Imediato.

(b) Para todo $i \in \text{ig}(F) \setminus \text{ig}(P)$, $F \subseteq \text{fa}(\{i\})$. Mas, como $\text{fa}(\{i\})$ é uma face, e F é faceta, então $\text{fa}(\{i\}) = F$. \square

Este teorema, apesar de simples, nos permite determinar quem são as facetas de um poliedro $P(A, b)$. Considere o seguinte corolário.

Corolário 1.8.3 *Seja $P = P(A, b)$ um poliedro e \mathcal{F} o conjunto das facetas de P . Então,*

- (a) $F_1, F_2 \in \mathcal{F}$, $F_1 \neq F_2$ então $\text{ig}(F_1) \cap \text{ig}(F_2) = \text{ig}(P)$;
- (b) $|\mathcal{F}| \leq |M| - |\text{ig}(P)|$;
- (c) *existe $I \subseteq M$ tal que $I \subseteq M \setminus \text{ig}(P)$, $|I| = |\mathcal{F}|$ e $F \in \mathcal{F}$ se e só se existe precisamente um $i \in I$ tal que $F = \text{fa}(\{i\})$.* \square

Um conjunto $I \subseteq M$ com as propriedades descritas em (c) é chamado **conjunto dos índices-faceta**. Pelo Teorema 1.8.2(b), temos que as facetas de P são obtidas impondo-se igualdade em apenas uma inequação $A_i x \leq b_i$ do sistema $Ax \leq b$. Note que isso não implica que para todo $i \in M$ $\text{fa}(\{i\})$ é uma faceta de P ! Isso só vale para os $i \in M$ que estão contidos em um conjunto de índices-faceta de P . A seguir, caracterizamos descrições irredundantes de poliedros.

Teorema 1.8.4 *Sejam $P = P(A, b) \neq \emptyset$ um poliedro, $I \subseteq M \setminus \text{ig}(P)$, $J \subseteq \text{ig}(P)$ e suponha que $P = \{x \mid A_{J^*}x = b_J, A_{I^*}x \leq b_I\}$. Essa descrição de P é irredundante se e só se*

- (a) I é um conjunto de índices-faceta de P ;
- (b) A_{J^*} é uma matriz $p \times n$ de posto-linha completo.

Prova. Segue dos resultados anteriores e da definição de descrição irredundante. \square

Corolário 1.8.5 *Seja $P = P(A, b) \subseteq \mathbb{R}^n$ um poliedro de dimensão plena. Então, para todo $I \subseteq M$, $A_{I^*}x \leq b_I$ é uma descrição irredundante de P se e só se I é o conjunto de índices-faceta de P .* \square

O resultado central dessa seção está apresentado no próximo teorema.

Teorema 1.8.6 *Seja $P = P(A, b)$ um poliedro e F uma face não trivial de P . As seguintes afirmações são equivalentes:*

- (a) F é uma faceta de P ;
- (b) F é uma face própria maximal de P ;

- (c) $\dim(F) = \dim(P) - 1$;
- (d) F contém $\dim(P)$ vetores afim-independentes;
- (e) Seja $c^\top x \leq \gamma$ uma inequação válida em relação a P tal que $F = \{x \in P \mid c^\top x = \gamma\}$. Então, para toda inequação $d^\top x \leq \delta$ válida em relação a P , tal que $F \subseteq \{x \in P \mid d^\top x = \delta\}$, temos que existe um vetor $u \in \mathbb{R}^{\text{ig}(P)}$ e $\alpha \in \mathbb{R}$, $\alpha \geq 0$ tal que

$$\begin{aligned} d^\top &= \alpha c^\top + u^\top A_{\text{ig}(P)*}, \\ \delta &= \alpha \gamma + u^\top b_{\text{ig}(P)}. \end{aligned}$$

Prova. Da definição de faceta, temos (a) \iff (b). É imediato que (c) \iff d. Vamos mostrar as equivalências em três partes, mostrando que (c) \implies (b); (a) \implies (e) e (e) \implies (c).

[(c) \implies (b)] Suponha que $\dim(F) = \dim(P) - 1$. Se F não é uma face própria maximal de P , então existe uma face própria G tal que $F \subset G \subset P$. Do Corolário 1.7.4(c) temos que $\dim(F) \leq \dim(G) - 1 \leq \dim(P) - 2$; uma contradição.

[(a) \implies (e)] Seja F uma face arbitrária de P . Vamos assumir que $A_{I*} \leq B_I$, $A_{J*} = b_J$ é uma descrição irredundante de $P(A, b)$ com $1 \in I$ e $F := \{x \in P \mid A_{1*}x = b_1\}$.

Seja $d^\top x \leq \delta$ uma inequação válida tal que $F \subseteq \{x \in P \mid d^\top x = \delta\}$. Então (veja Exercício 1.8.9 a seguir), existe $v \geq 0$ e $w \geq 0$ tais que

$$v^\top A_{I*} + w^\top A_{J*} = d^\top \quad \text{e} \quad v^\top b_I + w^\top b_J \leq \delta.$$

Suponha que existe $i \in I \setminus \{1\}$ tal que $v_i > 0$. Então, pelo Teorema 1.6.5, $i \in \text{ig}(F)$. Mas, isto contradiz o fato de que I é um conjunto de índices-faceta. Portanto $v_1 \geq 0$ e $v_i = 0$ para todo $i \in I \setminus \{1\}$. Com isso, $v_1 A_{1*} + w^\top A_{J*} = d^\top$ e $v_1 b_1 + w^\top b_J \leq \delta$. Como $v_1 A_{1*} = c^\top$ e $v_1 b_1 = \gamma$, segue a afirmação (e).

[(e) \implies] (c) Como F é uma face própria de P , então $\dim(F) \leq \dim(P) - 1$. Sem perda de generalidade podemos supor que $F = \{x \in P \mid A_{1*}x = b_1\}$. Supondo que $\dim(F) \leq \dim(P) - 1$, do Teorema 1.7.3 temos que

$$\text{posto}(A_{\text{ig}(F)*}) \geq \text{posto}(A_{\text{ig}(P)*}) + 2.$$

Então, existe $i \in \text{ig}(F) \setminus (\text{ig}(P) \cup \{1\})$ tal que o vetor A_{i*} e os A_{j*} , $j \in \text{ig}(P) \cup \{1\}$, são linearmente independentes. Isto significa que o sistema

$$A_{i*} = \alpha A_{1*} + u^\top A_{\text{ig}(P)*}$$

não tem solução α, u .

Como $F \subseteq \{x \in P \mid A_{i*}x = b_i\}$, isto contraria a hipótese de (e). \square

No próximo corolário mostramos o caso em que $\text{ig}(P) = \emptyset$, ou seja, o poliedro é de dimensão plena.

Corolário 1.8.7 *Seja $P \subseteq \mathbb{R}^n$ um poliedro de dimensão plena e $F = \{x \in P \mid c^\top x = \gamma\}$ uma face de P . São equivalentes:*

- (a) F é uma faceta de P ;
- (b) $\dim(F) = n - 1$;
- (c) se $d^\top x = \delta$, $d \neq 0$, é uma inequação válida tal que $F \subseteq \{x \in P \mid d^\top x = \delta\}$, então existe $\alpha \in \mathbb{R}$, $\alpha > 0$ tal que $d^\top = \alpha c^\top$ e $\delta = \alpha \gamma$.

□

Exercício 1.8.8 *Considere o poliedro $P = P(A, b)$ dado por*

$$A = \begin{pmatrix} -1 & 1 \\ 1 & 0 \\ -1 & 0 \\ 3 & -3 \\ 1 & 1 \end{pmatrix} \quad e \quad b = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 2 \end{pmatrix}.$$

Determine $\text{ig}(P)$ e quais das inequações definem facetas de P .

Exercício 1.8.9 *Considere $A \in \mathbb{R}^{p \times n}$, $C \in \mathbb{R}^{q \times n}$, $b \in \mathbb{R}^p$ e $d \in \mathbb{R}^q$. Então ocorre **exatamente uma** das alternativas:*

- (a) existe $x \in \mathbb{R}^n$ tal que $Ax \leq b$, $Cx < d$;
- (b) (b₁) existe $u \in \mathbb{R}_+^p$, $v \in \mathbb{R}_+^q \setminus \{0\}$ tal que $u^\top A + v^\top C = 0$ e $u^\top b + v^\top d \leq 0$, ou
(b₂) existe $u \in \mathbb{R}_+^p$ tal que $u^\top A = 0$, $u^\top b < 0$.

Exercício 1.8.10 *Se $P(A, b) \neq \emptyset$ então ocorre exatamente uma das alternativas:*

- (a) existe $x \in \mathbb{R}^n$ tal que $Ax \leq b$, $Cx < d$;
- (b) existe $\begin{pmatrix} u \\ v \end{pmatrix} \in \mathbb{R}_+^{p+q}$, $v \neq 0$, tal que $u^\top A + v^\top C = 0$ e $u^\top b + v^\top d \leq 0$.

Exercício 1.8.11 (Gordan, 1873) *Ocorre exatamente uma das alternativas:*

- (a) existe x tal que $Ax < 0$;
- (b) existe $u \geq 0$, $u \neq 0$ tal que $u^\top A = 0$.

Exercício 1.8.12 (Stiemke, 1915) *Ocorre exatamente uma das alternativas:*

- (a) existe $x > 0$ tal que $Ax = 0$;
- (b) existe u tal que $u^\top A \geq 0$, $u^\top b \neq 0$.

1.9 Vértices e Extremais

Estudamos aqui outras faces especiais de poliedros: as de dimensão 0 e as de dimensão 1.

Seja $P \subseteq \mathbb{R}^n$ um poliedro e F uma face de P . Se $F = \{x\}$ para algum $x \in \mathbb{R}^n$, então dizemos que F é um **vértice** de P . Se existem $x \in \mathbb{R}^n$ e $z \in \mathbb{R}^n \setminus \{0\}$ tais que $F = x + \text{lin}(\{z\})$, então chamamos F de **reta extremal**; se $F = x + \text{cone}(\{z\})$, então dizemos que F é um **raio extremal** de P .

Se $F = \{x\}$ é um vértice de P , dizemos simplesmente que x é um vértice de P . Vértices são faces de dimensão 0, enquanto retas e raios extremais são faces de dimensão 1. Faces de dimensão 1 também são chamadas de **arestas**. Além de raio ou reta extremal, uma aresta pode ser um segmento de reta ligando dois vértices. Veja a Figura 1.4.

Se x, y são vértices de um poliedro P que são ligados por uma aresta (i.e. $\text{conv}\{x, y\}$ é uma face de P), então dizemos que x, y são **adjacentes** em P . Na Figura 1.4, os vértices x e y são adjacentes; já os vértices y e z não são adjacentes.

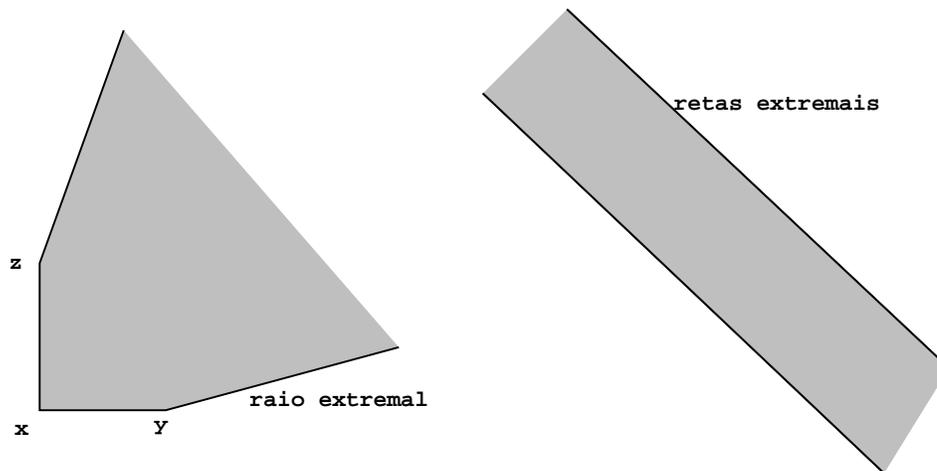


Figura 1.4: Vértices, raios e retas extremais.

No restante desta seção apresentamos caracterizações de vértices, raios e retas extremais de poliedros.

Teorema 1.9.1 *Seja $C \subseteq \mathbb{R}^n$ um cone poliédrico. Neste caso,*

- (a) *se x é um vértice de C , então $x = 0$.*
- (b) *F é raio extremal de C se e só se existe $z \in \mathbb{R}^n \setminus \{0\}$ tal que $F = \text{cone}(\{z\})$ é face de C .*

Prova. Como C é cone poliédrico, pela Proposição 1.1.5 existe uma matriz A tal que $C = P(A, 0)$. Pelo Teorema 1.6.6, segue imediatamente que, toda face não-vazia de C é um cone que contém o vetor 0.

- (a) Se x é um vértice, então $0 \in \{x\}$, e portanto $x = 0$.
- (b) A prova de uma das implicações segue diretamente da definição. Vamos mostrar que se F é raio extremal, então existe z tal que $F = \text{cone}(\{z\})$ é face de C .

Como F é raio extremal de C , então existe $x \in \mathbb{R}^n$ e $z \in \mathbb{R}^n \setminus \{0\}$ tais que $F = x + \text{cone}(\{z\})$. Mas, da observação acima segue que $0 \in F$. Logo, existe $\alpha \geq 0$ tal que $0 = x + \alpha z$, ou seja, $x = -\alpha z$. Como F é um cone, então

$$\{\lambda x \mid \lambda \in \mathbb{R}_+\} = \{\lambda(-\alpha z) \mid \lambda \in \mathbb{R}_+\} \subseteq F.$$

Caso $\alpha \neq 0$, então $\{\lambda(-\alpha z) \mid \lambda \in \mathbb{R}_+\} = \{-\lambda z \mid \lambda \in \mathbb{R}_+\} \not\subseteq x + \text{cone}(\{z\})$. Assim, $\alpha = 0$, e neste caso, $F = \text{cone}(\{z\})$. \square

Lema 1.9.2 *Seja F uma face não-vazia de $P = P(A, b)$, $I = \text{ig}(F)$ e $B = \{y^1, \dots, y^k\}$ uma base do núcleo de A_I . Então, para cada ponto interior $x \in F$, existe $\epsilon > 0$ tal que $x + \epsilon y^j$ e $x - \epsilon y^j \in P$ para $j = 1, \dots, k$.*

Prova. Seja x um ponto interior de F e $J := \{1, \dots, m\} \setminus I$. Pelo Exercício 1.7.2, temos que $\text{ig}(\{x\}) = I$. Logo, $A_{J*}x < b_J$. Além disso, $A_{I*}(x + \epsilon y^j) = b_I$ e $A_{I*}(x - \epsilon y^j) = b_I$ para todo $\epsilon \in \mathbb{R}$. Para ϵ suficientemente pequeno valem também as desigualdades $A_{J*}(x + \epsilon y^j) \leq b_J$ e $A_{J*}(x - \epsilon y^j) \leq b_J$. \square

O teorema a seguir contém o resultado principal desta seção, e caracteriza quando um ponto é vértice de um poliedro.

Teorema 1.9.3 *Seja $P = P(A, b) \subseteq \mathbb{R}^n$ um poliedro e $x \in P$. São equivalentes:*

- (a) x é vértice de P ;
- (b) $\{x\}$ é uma face de P de dimensão 0;
- (c) x não é combinação convexa própria de elementos de P , ou seja, para quaisquer $y, z \in P$, $y \neq z$, $0 < \lambda < 1$, então $x \neq \lambda y + (1 - \lambda)z$;
- (d) $P \setminus \{x\}$ é convexo;
- (e) $\text{posto}(A_{\text{ig}(\{x\})*}) = n$;
- (f) existe $c \in \mathbb{R}^n \setminus \{0\}$ tal que x é a única solução ótima do problema $\max\{c^\top y \mid y \in P\}$.

Prova. Por definição, temos que (a) \iff (b). Deixamos como exercício para o leitor, mostrar que (c) \iff (d) e (c) \implies (e) (basta usar o Lema 1.9.2). Para completar a prova, vamos mostrar as implicações (a) \implies (f), (f) \implies (c) e (e) \implies (b).

[(a) \Rightarrow (f)] Por definição, $\{x\}$ é uma face. Logo, existe uma inequação válida $c^\top y \leq \gamma$ tal que $\{x\} = \{y \in P \mid c^\top y = \gamma\}$.

[(f) \Rightarrow (c)] Seja $c \in \mathbb{R}^n \setminus \{0\}$ e x a única solução ótima de $\max\{c^\top y \mid y \in P\}$. Considere γ o valor ótimo desse problema, ou seja, $\gamma := c^\top x$. Se $x = \lambda y + (1 - \lambda)z$ para $y, z \in P$, $y \neq z$ e $0 < \lambda < 1$, então segue que $\gamma = c^\top x = c^\top(\lambda y + (1 - \lambda)z) = \lambda c^\top y + (1 - \lambda)c^\top z \leq \lambda\gamma + (1 - \lambda)\gamma = \gamma$. Com isso, $c^\top y = \gamma = c^\top z$, contradizendo o fato de que x é a única solução ótima do problema considerado.

[(e) \Rightarrow (b)] Seja $I = \text{ig}(\{x\})$. Então, $A_{I^*}y = b_I$ tem uma solução única que é precisamente x . Disso segue que $F = \{y \in P \mid A_{I^*}y = b_I\} = \{x\}$. Ou seja, $\{x\}$ é uma face de dimensão 0. \square

O teorema a seguir caracteriza vértices de poliedros da forma $P = P(A, b)$. Lembramos que, se $x \in \mathbb{R}^n$ então o **suporte** de x , denotado por $\text{sup}(x)$ é o conjunto dos índices correspondentes às entradas não nulas de x . Ou seja,

$$\text{sup}(x) := \{i \in \{1, \dots, n\} \mid x_i \neq 0\}.$$

Teorema 1.9.4 *Para $x \in P = P(A, b) \subseteq \mathbb{R}^n$ as seguintes afirmações são equivalentes:*

- (a) x é um vértice de $P = P(A, b)$;
- (b) $\text{posto}(A_{*\text{sup}(x)}) = |\text{sup}(x)|$;
- (c) os vetores-coluna A_{*j} , $j \in \text{sup}(x)$, são linearmente independentes. \square

Nem todos os poliedros possuem vértices. Basta tomar, por exemplo, a intersecção de menos do que n semi-espacos no \mathbb{R}^n . Dizemos que um poliedro é **pontudo**, se possui pelo menos um vértice.

Um conceito relacionado com poliedros pontudos é o de **cone recessional**. Se um poliedro P não é limitado, ele contém alguma semi-reta. O conjunto de todas as direções de semi-retas contidas em P formam um cone, chamado de **cone recessional** de P . Mais precisamente,

$$\text{rec}(P) := \{y \in \mathbb{R}^n \mid x + \lambda y \in P \text{ para todo } x \in P, \lambda \geq 0\}.$$

O teorema a seguir, e os corolários subseqüentes caracterizam poliedros pontudos.

Teorema 1.9.5 *Seja $P = P(A, b) \subseteq \mathbb{R}^n$ um poliedro não-vazio. São equivalentes:*

- (a) P é pontudo;
- (b) $\text{posto}(A) = n$;
- (c) toda face não-vazia de P é um poliedro pontudo;

- (d) P não contém nenhuma reta;
- (e) $\text{rec}(P)$ é pontudo, isto é, 0 é vértice de $\text{rec}(P)$;
- (f) $\text{rec}(P)$ não contém nenhuma reta.

Prova. Exercício. □

Corolário 1.9.6 *Se $P = P^=(A, b)$, então $P \neq \emptyset$ se e só se P é pontudo.* □

Corolário 1.9.7 *Se P é um politopo, então $P \neq \emptyset$ se e só se P é pontudo.* □

Corolário 1.9.8 *Se P é pontudo e $\max\{c^\top x \mid x \in P\}$ tem uma solução ótima, então existe um vértice de P que é solução ótima.* □

Vale observar que o resultado acima é fundamental para justificar as idéias que estão por trás do método simplex.

Capítulo 2

Métodos de Planos-de-corte

Neste capítulo apresentamos a teoria sobre os chamados **métodos de planos-de-corte**. Esses métodos foram introduzidos por Gomory na década de 50 para resolver problemas de programação linear inteira. Essencialmente, esses métodos se baseiam na seguinte idéia: começa-se com uma solução do problema (PL) relaxado, depois, de uma maneira sistemática, enquanto a solução obtida não é inteira, procura-se eliminá-la acrescentando-se novas inequações (chamadas *planos-de-corte*) ao sistema linear corrente.

Descrevemos aqui a base geométrica dessa teoria, que diz respeito à garantia de que esse processo de acrescentar planos-de-corte, nos conduz a uma seqüência finita de poliedros $P \supseteq P' \supseteq P'' \supseteq \dots \supseteq P_I$, onde P_I é precisamente o fecho inteiro do poliedro original P sobre qual queremos otimizar. Ou seja, esse processo é finito, e de fato, nos permite encontrar um vértice inteiro que é uma solução do problema (PLI) original.

Começamos introduzindo o conceito de poliedro inteiro, matriz totalmente unimodular e sistema totalmente dual integral (TDI). Conceitos como *provas de planos-de-corte* e *posto de Chvátal* de uma inequação também são discutidos. Descrevemos o método de planos-de-corte devido a Gomory e mencionamos a questão da equivalência entre problemas de otimização e separação.

2.1 Fecho Inteiro de um Poliedro

Dado um poliedro P , define-se o **fecho inteiro** de P , denotado por P_I como sendo o fecho convexo dos vetores inteiros em P , isto é,

$$P_I := \text{conv}\{x \in P \mid x \text{ é inteiro}\}.$$

Com isso, resolver um problema de programação linear inteira do tipo

$$\begin{aligned} \max \quad & c^\top x \\ & Ax \leq b \\ & x \text{ inteiro} \end{aligned}$$

é equivalente a resolver o problema $\max\{c^\top x \mid x \in P(A, b)_I\}$. Assim, se formos capazes de encontrar o fecho inteiro de um poliedro, seremos capazes de resolver problemas de programação linear inteira. Considerando que problemas de programação linear inteira são \mathcal{NP} -difíceis [38], concluímos que não deve existir algoritmos eficientes para encontrar fecho inteiro de poliedros (a que $\mathcal{P} = \mathcal{NP}$).

Assim como no caso de problemas de programação linear, ainda que o problema geral seja difícil, há casos especiais que são tratáveis. Veremos aqui, alguns desses casos. Para isso, estudaremos algumas propriedades do fecho inteiro de um poliedro.

Mostraremos que se $P = P(A, b)$ é um poliedro racional (isto é, com A e b só contendo entradas racionais), então P_I é um poliedro, ou seja, existe uma matriz A' e um vetor b' tais que $P(A, b)_I = P(A', b')$. Este é o principal resultado desta seção, e envolve conceitos como matrizes totalmente unimodulares e sistemas totalmente dual inteiros. Antes, definamos o conceito de **poliedro inteiro**.

Dizemos que um poliedro P é **inteiro** se $P = P_I$. Pode-se mostrar facilmente que as seguintes afirmações são equivalentes sobre um poliedro racional P .

- (a) P é inteiro;
- (b) cada face não vazia de P contém vetores inteiros;
- (c) cada face não vazia minimal de P contém vetores inteiros;
- (d) $\max\{c^\top x \mid x \in P\}$ é atingido por um vértice inteiro para todo c para o qual o máximo é finito.

2.1.1 Matrizes totalmente unimodulares

Uma matriz A é **totalmente unimodular** (TU) se o valor de cada subdeterminante de A é 0, +1 ou -1. Exemplos bastante conhecidos de matrizes TU são a matriz de incidência de um grafo orientado e a matriz de incidência de um grafo bipartido.

Um resultado importante a respeito de matrizes TU (aliás que motivou sua descoberta) está expresso no próximo teorema, devido a Hoffman e Kruskal [65].

Teorema 2.1.1 *Seja A uma matriz TU e b um vetor inteiro. Então o poliedro $P(A, b)$ é inteiro.*

Prova. Seja $F = \{x \mid A'x = b'\}$ uma face minimal de $P(A, b)$, onde $A'x \leq b'$ é um subsistema de $Ax \leq b$ com A' de posto completo. Então, $A' = [B \ C]$ para alguma matriz B com determinante +1 ou -1. Claramente, $x = [B^{-1}b' \ 0]$ é um vetor inteiro em F , e portanto, $P(A, b)$ é inteiro.

□

Outros resultados importantes a respeito de matrizes totalmente unimodulares podem ser encontrados em Schrijver [99]. Aqui nos restringiremos a enunciar alguns desses resultados.

Lema 2.1.2 *Se A e B são matrizes TU, então*

(a) A^\top é TU;

(b) $\begin{pmatrix} A \\ B \end{pmatrix}$ é TU. □

Corolário 2.1.3 *Seja A uma matriz TU e b, c vetores inteiros. Então os seguintes problemas de programação linear têm solução inteira:*

$$\max\{c^\top x \mid Ax \leq b\} = \min\{y^\top b \mid y^\top A = c, y \geq 0\}.$$

Prova. Segue do Teorema 2.1.1 e do fato da matriz

$$\begin{pmatrix} A^\top \\ -A^\top \\ -I \end{pmatrix}$$

ser totalmente unimodular. □

Em vista dos resultados acima, a pergunta natural que surge é *como descobrir se uma dada matriz é TU*.

Diversas caracterizações de matrizes TU foram obtidas nas décadas de 60 e 70 (veja Teorema 2.1.8), nenhuma das quais fornece um algoritmo polinomial para testar total-unimodularidade. Em 1980, Seymour [100] caracterizou matrizes totalmente unimodulares usando decomposição de matróides. Estes resultados foram utilizados por Cunningham e Edmonds [19] que apresentaram um algoritmo de reconhecimento de matrizes TU. O algoritmo mais rápido que se conhece é devido a Truemper [102].

Dizemos que uma matriz A de posto-linha completo é **unimodular** se A é inteira e cada base de A tem determinante igual a $+1$ ou -1 . Lembramos que uma **base** de uma matriz $m \times n$ A de posto-linha completo é uma submatriz de A inversível de ordem m .

Deixamos a cargo do leitor a prova do seguinte exercício, a respeito da relação existente entre matrizes unimodulares e totalmente unimodulares.

Exercício 2.1.4 *A é TU se e somente se $[I \ A]$ é unimodular.*

O próximo teorema mostra uma relação entre matrizes unimodulares e certos poliedros.

Teorema 2.1.5 *Seja A uma matriz inteira de posto-linha completo. Então, $P^=(A, b)$ é inteiro para cada vetor inteiro b se e somente se A é unimodular.*

Prova. Inicialmente suponha que a matriz $A \in \mathbb{Z}^{m \times n}$ seja unimodular. Seja b um vetor inteiro e x' um vértice de $P^=(A, b)$. Como x' é vértice, existem m restrições linearmente independentes de A que são satisfeitas por x' com igualdade. Assim, as colunas de A correspondentes aos índices do suporte de x' são linearmente independentes. Com isso, podemos estender tais colunas à uma base B de A . Como x' restrito aos índices de B é da forma $B^{-1}b$, e os seus demais componentes são nulos, é fácil ver que x' só tem componentes inteiros.

Para provar a outra direção, suponha que $P^=(A, b)$ é inteiro para cada vetor inteiro b . Seja B uma base de A . Para mostrar que A é unimodular, basta provar que $B^{-1}z$ é inteiro para cada vetor inteiro z . Seja z um vetor inteiro. Então existe um vetor inteiro y tal que $w := y + B^{-1}z \geq 0$. Assim, $b = Bw$ é inteiro. Mas, estendendo w de forma a ter componentes nulas fora da base B , obtemos um vetor w' tal que $Aw' = Bw = b$. Com isso, w' é vértice de $P^=(A, b)$, e portanto w' é inteiro. Logo, $w - y = B^{-1}z$ também é inteiro. \square

Uma consequência imediata do Teorema 2.1.5 é o seguinte resultado devido a Hoffman e Kruskal [65].

Corolário 2.1.6 (Hoffman e Kruskal, 1956) *Seja A uma matriz inteira. Então, a matriz A é TU se e somente se para cada vetor inteiro b o poliedro $\{x \mid Ax \leq b, x \geq 0\}$ é inteiro.*

Prova. Temos que A é TU se e só se $[I \ A]$ é unimodular. Por sua vez, esta última propriedade é equivalente a $P^=([I \ A], b)$ ser inteiro para cada inteiro b , donde segue o resultado. \square

Outra consequência do Teorema 2.1.5 está expressa no exercício abaixo, que pode ser entendido como um teorema de “dualidade” inteira.

Exercício 2.1.7 *Uma matriz A é TU se e somente se para quaisquer vetores inteiros b e c , ambos os lados da equação (de dualidade linear):*

$$\max\{c^\top x \mid Ax \leq b, x \geq 0\} = \min\{y^\top b \mid y^\top A \geq c, y \geq 0\}$$

são atingidos por vetores inteiros x e y (se forem finitos).

No teorema a seguir enunciamos várias formas equivalentes de caracterizar matrizes TU. Ao leitor interessado na prova desse teorema ou em outros resultados a respeito de matrizes TU sugerimos consultar Schrijver [99].

Teorema 2.1.8 *Seja A uma matriz com entradas 0, +1 e -1. As seguintes afirmações são equivalentes.*

- (a) A é TU;
- (b) para cada vetor inteiro b o poliedro $\{x \mid Ax \leq b, x \geq 0\}$ só tem vértices inteiros;

- (c) cada coleção de colunas de A pode ser particionada em duas partes, de modo que a soma das colunas em uma parte menos a soma das colunas na outra parte é um vetor com entradas 0, +1 e -1;
- (d) cada submatriz não singular de A tem uma linha com um número ímpar de componentes não nulas;
- (e) nenhuma submatriz quadrada de A tem determinante +2 ou -2. □

Exercício 2.1.9 *Seja G um grafo orientado e A a matriz de incidência de G . Mostre que A é TU.*

Exercício 2.1.10 *Seja G um grafo e A a matriz de incidência de G . Mostre que A é TU se e só se G é bipartido.*

Exercício 2.1.11 *Utilize os resultados mencionados nos exercícios 2.1.7 e 2.1.9 para provar o Teorema de König: Num grafo bipartido a cardinalidade de um emparelhamento máximo é igual à cardinalidade de uma cobertura mínima de vértices.*

Já vimos que matrizes TU e poliedros inteiros têm uma relação estreita entre si. A seguir, mostramos outros resultados relacionando esses dois conceitos. Antes, vejamos uma caracterização de poliedros inteiros devida a Edmonds e Giles.

Teorema 2.1.12 (Edmonds e Giles, 1977) *Um poliedro racional P é inteiro se e somente se cada hiperplano suporte racional de P contém vetores inteiros.*

Prova. Uma das implicações segue diretamente da definição de poliedros inteiros. Vamos mostrar que, se cada hiperplano racional de P contém vetores inteiros, então P é inteiro. Tomemos $P = P(A, b)$ racional e suponhamos A e b inteiros. Seja $F := \{x \mid A'x = b'\}$ uma face não vazia de P . Sabemos que F contém vetores inteiros se e só se $y^\top b'$ é inteiro para todo vetor y tal que $y^\top A'$ é inteiro (veja Exercício 2.1.7). Assim, se F não contém vetores inteiros, então existe y tal que $c := y^\top A'$ é inteiro e $\gamma := y^\top b$ é não inteiro. Claramente, $H := \{x \mid c^\top x \leq \gamma\}$ é um hiperplano suporte de P e, pela hipótese, contém vetores inteiros, uma contradição. □

Um corolário imediato do teorema acima está enunciado a seguir, e caracteriza quando um problema de programação linear tem solução inteira, qualquer que seja a função objetivo.

Corolário 2.1.13 *Seja $Ax \leq b$ um sistema racional. Então, $\max\{c^\top x \mid Ax \leq b\}$ é atingido por um vetor inteiro x^* , para cada vetor c para o qual o máximo é finito, se e só se, $\max\{c^\top x \mid Ax \leq b\}$ é inteiro para cada vetor inteiro c para o qual o máximo é finito.*

Prova. Uma das implicações é imediata. Vamos mostrar que, se o máximo é inteiro (sempre que finito), então existe um vetor inteiro que atinge esse máximo. Tome c' um vetor racional, e $\hat{c} = Mc'$ um vetor inteiro. Então, $\max\{\hat{c}^\top x \mid Ax \leq b\}$ é inteiro, se for finito. Seja $H := \{x \mid \hat{c}^\top x = \gamma\}$ um hiperplano suporte de $P = P(A, b)$, onde $\gamma := \max\{\hat{c}^\top x \mid Ax \leq b\}$. Então, H contém vetores inteiros, e pelo Teorema 2.1.12, o poliedro P é inteiro. Pela definição de poliedro inteiro temos que, para todo c , esse poliedro tem face ótima com vértice inteiro. \square

Lembramos que pelo teorema forte de dualidade,

$$\max\{c^\top x \mid Ax \leq b\} = \min\{y^\top b \mid y^\top A = c, y \geq 0\}.$$

De certa forma, gostaríamos de estender o resultado acima, que vale em programação linear, para programação linear inteira. Infelizmente, nem sempre isso é verdade.

Dizemos que um sistema racional $Ax \leq b$ é **totalmente dual integral** (TDI) se o mínimo na expressão acima é atingido por uma solução ótima y inteira para cada vetor inteiro c para o qual o mínimo é finito.

De certa forma, esse conceito nos fornece um resultado “equivalente” para a teoria de dualidade para programação inteira. Lembramos que, pelo Corolário 2.1.3, se uma matriz A é TU então $Ax \leq b$ é TDI para cada vetor racional b .

Exercício 2.1.14 *Seja $Ax \leq b$ um sistema TDI. Mostre que se b é inteiro e o máximo na expressão acima é finito, então o problema de maximização tem solução ótima inteira.*

Exercício 2.1.15 *Mostre que se $Ax \leq b$ é um sistema TDI, então $P(A, b)$ é inteiro.*

Vale ressaltar que a propriedade de “ser TDI” não se aplica a poliedros, mas a sistemas de inequações. Veja o exemplo abaixo.

Exemplo 2.1.16 *Os sistemas*

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \leq \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad e \quad \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \leq \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

definem o mesmo poliedro, mas o primeiro sistema é TDI, enquanto que o segundo sistema não é. Em geral, sistemas TDI são redundantes.

Exercício 2.1.17 *Se A é uma matriz inteira e b é um vetor racional, então o sistema $Ax \leq b, x \geq 0$ é TDI se e só se o mínimo em*

$$\max\{c^\top x \mid Ax \leq b, x \geq 0\} = \min\{y^\top b \mid y^\top A \geq c, y \geq 0\}$$

tem uma solução ótima inteira y para cada vetor inteiro c para o qual o mínimo é finito.

Dizemos que um sistema $Ax \leq b$ é **TDI minimal** ou **minimalmente TDI** se esse sistema é TDI, mas qualquer seu subsistema próprio que define o mesmo poliedro que $Ax \leq b$ não é TDI. Pode-se provar o seguinte teorema a respeito de sistemas minimalmente TDI.

Teorema 2.1.18 *Para cada poliedro racional P existe um sistema TDI $Ax \leq b$ com A inteiro e $P = P(A, b)$. O vetor b pode ser escolhido inteiro se e só se P for inteiro. Além disso, se P é de dimensão plena, existe um único sistema TDI minimal $Ax \leq b$ com A inteiro e $P = P(A, b)$. Nesse caso, b é inteiro se e só se P é inteiro.* \square

O seguinte resultado é um corolário imediato do teorema acima.

Corolário 2.1.19 *Um poliedro racional P é inteiro se e só se existe um sistema TDI $Ax \leq b$ com $P = P(A, b)$ e b inteiro.* \square

Assim, se fôssemos capazes de descobrir um sistema TDI que caracteriza um poliedro inteiro, poderíamos resolver qualquer problema de programação linear inteira com função objetivo racional. Porém, em geral, um sistema TDI minimal com matriz inteira pode ter tamanho exponencial.

Exemplo 2.1.20 *Seja k um poliedro inteiro e P o poliedro $P := \{(x, y) \mid x \geq 0, x + ky \geq 0\}$, k inteiro. Um sistema $Ax \leq b$ TDI minimal tal que $P = P(A, b)$ é dado pelas inequações $x + jy \geq 0$ para $j = 0, \dots, k$. Tal sistema tem tamanho $O(k \log k)$, que é portanto exponencial no tamanho da instância.*

Para encerrar essa seção faremos alguns comentários a respeito da complexidade computacional de alguns problemas referentes a reconhecimento de sistemas TDI e poliedros inteiros.

Dado um sistema racional $Ax \leq b$, o problema de decidir se tal sistema determina um poliedro inteiro está em $\text{co-}\mathcal{NP}$. Não se sabe se esse problema está em \mathcal{NP} . Se o posto de A é fixo, então o problema pode ser resolvido em tempo polinomial.

Por outro lado, decidir se um dado sistema $Ax \leq b$ é TDI está em $\text{co-}\mathcal{NP}$ quando a matriz A é inteira. Quando a matriz não é inteira não se sabe se o problema está em $\text{co-}\mathcal{NP}$.

2.2 Descrição dos Métodos de Planos-de-Corte

Na seção anterior definimos o conceito de fecho inteiro de um poliedro P . Uma pergunta natural que surge é como determinar esse fecho inteiro. Note que se $H = \{x \mid c^\top x \leq \gamma\}$ é um semi-espço afim racional, onde c tem componentes inteiros primos entre si, então é fácil achar o fecho inteiro de H :

$$H_I = \{x \mid c^\top x \leq \lfloor \gamma \rfloor\}.$$

Assim, para qualquer poliedro P , defina

$$P' := \bigcap_{H \supseteq P} H_I.$$

Note que $P_I \subseteq P'$, já que $P \subseteq H$ implica $P_I \subseteq H_I$.

Poderíamos, então, seguir obtendo inequações que definem semi-espacos H que contêm P e repetindo o processo, até, eventualmente, atingirmos P_I . Se $P^{(0)} = P$ e $P^{(t+1)} = (P^{(t)})'$, temos

$$P \supseteq P' \supseteq P'' \supseteq \dots \supseteq P_I.$$

A idéia do método é obter aproximações cada vez melhores de P_I através da introdução de tais inequações, chamados de **planos-de-corte**.

No final da década de 50, Gomory desenvolveu o chamado **método dos planos-de-corte** (MPC) para resolver problemas de programação linear inteira usando o método simplex. A teoria que justifica tal método é o resultado acima mencionado.

O MPC resolve problemas lineares P, P', P'', \dots até que uma solução inteira seja encontrada. Veremos a seguir que, para encontrar uma solução ótima, podemos nos restringir a um número finito de semi-espacos H . Apesar de limitado, em geral o número de inequações pode crescer exponencialmente.

Teorema 2.2.1 *Seja $P = P(A, b)$ com $Ax \leq b$ TDI e A inteiro. Então,*

$$P' = \{x \mid Ax \leq \lfloor b \rfloor\}.$$

Em particular, para qualquer poliedro racional P , tem-se que P' é um poliedro.

Prova. Suponhamos que $P \neq \emptyset$. É claro que $P' \subseteq \{x \mid Ax \leq \lfloor b \rfloor\}$. Vamos provar que $\{x \mid Ax \leq \lfloor b \rfloor\} \subseteq P'$.

Seja $H := \{x \mid c^\top x \leq \delta\}$ um semi-espaco afim racional tal que $P \subseteq H$. Sem perda de generalidade podemos assumir que os componentes de c são inteiros primos entre si. Então, $H_I = \{x \mid c^\top x \leq \delta\}$ e

$$\delta \geq \max\{c^\top x \mid Ax \leq b\} = \min\{y^\top b \mid y^\top A = c, y \geq 0\}.$$

Como $Ax \leq b$ é TDI, o mínimo na equação acima é atingido por um valor inteiro, digamos y^* . Seja x tal que $Ax \leq \lfloor b \rfloor$. Então,

$$c^\top x = y^{*\top} Ax \leq y^{*\top} \lfloor b \rfloor \leq \lfloor y^{*\top} b \rfloor \leq \lfloor \delta \rfloor.$$

Portanto, $\{x \mid Ax \leq \lfloor b \rfloor\} \subseteq H_I$. Como isso vale para qualquer semi-espaco afim H que contém P , temos que $\{x \mid Ax \leq \lfloor b \rfloor\} \subseteq \bigcap H_I = P'$. \square

Lema 2.2.2 *Seja F uma face não-vazia de um poliedro racional P . Então, $F' = P' \cap F$.*

Prova. Considere P definido por $Ax \leq b$, um sistema TDI, onde a matriz A é inteira. Tome $F := \{x \in P \mid a^\top x = \beta\}$ uma face não-vazia de P com a e β inteiros. Como o sistema $\{Ax \leq b, a^\top x \leq \beta\}$ é TDI, o sistema $\{Ax \leq b, a^\top x = \beta\}$ também é TDI. Sendo β um inteiro, segue que

$$P' \cap F = \{x \mid Ax \leq \lfloor b \rfloor, ax = \beta\} = \{x \mid Ax \leq \lfloor b \rfloor, ax \leq \lfloor \beta \rfloor, ax \geq \lfloor \beta \rfloor\} = F'.$$

□

Pelo acima exposto, temos que se F' é uma face não-vazia de um poliedro racional P , então

- (a) F' é uma face de P' ;
- (b) para todo natural t , temos que $F^{(t)} = P^{(t)} \cap F$.

O próximo teorema (que tem uma prova bastante técnica) nos dá o embasamento teórico para os métodos de planos-de-cortes. Nesse teorema mostramos que para cada poliedro racional P existe um natural t tal que $P^{(t)} = P_I$.

Teorema 2.2.3 *Para cada poliedro racional P , existe um natural t tal que $P^{(t)} = P_I$.*

Prova. Vamos demonstrar esse resultado por indução na dimensão do poliedro P . Se o poliedro é vazio (dimensão -1) ou tem dimensão 0 , o resultado é trivial.

Vamos analisar dois casos. Se $\text{afim}(P)$ não contém vetores inteiros, existe um vetor inteiro c e um escalar δ tal que $P \subseteq \{x \mid c^\top x = \delta\}$. Logo,

$$P' \subseteq \{x \mid c^\top x \leq \lfloor \delta \rfloor \text{ e } c^\top x \geq \lceil \delta \rceil\} = \emptyset,$$

e portanto, $P' = P_I$.

Suponhamos agora que $\text{afim}(P)$ contenha vetores inteiros, e que a dimensão de $\text{afim}(P)$ seja $n-d$. Existe uma transformação do poliedro $P \subseteq \mathbb{R}^n$ para um poliedro P' de dimensão completa em \mathbb{R}^d . Podemos, então, supor sem perda de generalidade que P tem dimensão completa.

Para um poliedro P de dimensão completa, podemos mostrar que existe uma matriz A e vetores b e b' tais que $P = P(A, b)$ e $P_I = P(A, b')$. Vamos mostrar que, para cada inequação $a^\top x \leq \beta'$ de $Ax \leq b'$, existe um natural s tal que $P^{(s)} \subseteq \{x \mid a^\top x \leq \beta'\}$. Com isso, o teorema ficará provado,

Seja $H := \{x \mid a^\top x \leq \beta'\}$ e $a^\top x \leq \beta$ a inequação correspondente (a $a^\top x \leq \beta'$) em $Ax \leq b$. Suponha que $P^{(s)} \not\subseteq H$ para todo s . Como $P' \subseteq \{x \mid a^\top x \leq \lfloor \beta \rfloor\}$, existem inteiros β'' e r tais que

$$\beta' < \beta'' \leq \lfloor \beta \rfloor \text{ e } P^{(s)} \subseteq \{x \mid a^\top x \leq \beta''\} \text{ para todo } s > r,$$

*(2.1) e além disso,

$$P^{(s)} \not\subseteq \{x \mid a^\top x \leq \beta'' - 1\} \text{ para todo } s > r.$$

Considere $F := P^{(r)} \cap \{x \mid a^\top x = \beta''\}$. Como P tem dimensão completa, $\dim(F) < \dim(P)$. Além disso, F não contém vetores inteiros. Assim, pela hipótese de indução, $F^{(u)} = \emptyset$ para algum inteiro u . Aplicando o Lema 2.2.2,

$$\emptyset = F^{(u)} = P^{(r+u)} \cap F = P^{(r+u)} \cap \{x \mid a^\top x = \beta''\}.$$

Então, $P^{(r+u)} \subseteq \{x \mid ax < \beta''\}$, e portanto $P^{(r+u+1)} \subseteq \{x \mid a^\top x \leq \beta'' - 1\}$, contrariando (*). □

Algumas propriedades apresentadas no início deste capítulo, a respeito de poliedros inteiros (por exemplo, o Teorema 2.1.12), podem ser obtidas como consequência do resultado apresentado acima. Um outro resultado que segue do Teorema 2.2.3 é o seguinte resultado, devido a Chvátal 1973, para politopos não necessariamente racionais.

Corolário 2.2.4 *Para cada politopo P , existe um natural t tal que $P^{(t)} = P_I$.* □

2.2.1 Provas de planos-de-corte

Seja $Ax \leq b$ um sistema de inequações lineares, e $c^\top x \leq \delta$ uma inequação. Dizemos que uma seqüência de inequações lineares

$$c_1^\top x \leq \delta_1, \quad c_2^\top x \leq \delta_2, \dots, \quad c_m^\top x \leq \delta_m$$

é uma **prova de planos-de-corte** de $c^\top x \leq \delta$ (a partir de $Ax \leq b$), se

- (a) cada c_1, c_2, \dots, c_m são inteiros, $c_m = c$, $\delta_m = \delta$ e,
- (b) para cada $i = 1, \dots, m$, a inequação $c_i^\top x \leq \delta'_i$ é uma combinação cônica das inequações $Ax \leq b$, $c_1^\top x \leq \delta_1, \dots, c_{i-1}^\top x \leq \delta_{i-1}$ para algum δ'_i com $\lfloor \delta'_i \rfloor \leq \delta_i$.

Dizemos que uma tal prova tem **comprimento m** .

Claramente, se uma inequação $c^\top x \leq \delta$ tem uma prova de planos-de-corte a partir de $Ax \leq b$, então essa inequação é válida para qualquer solução inteira de $Ax \leq b$. A recíproca dessa afirmação também vale, se assumirmos que $Ax \leq b$ possui pontos inteiros (isto segue do Teorema 2.2.3).

Lema 2.2.5 *Seja $P = P(A, b)$ um poliedro não-vazio e $c^\top x \leq \delta$ uma inequação válida para P . Então, existe $\delta' \leq \delta$ tal que $c^\top x \leq \delta'$ é uma combinação cônica das inequações do sistema $Ax \leq b$.* □

Deixamos a cargo do leitor provar o seguinte resultado a respeito de existência de provas de planos-de-cortes.

Corolário 2.2.6 *Seja $P = P(A, b)$ um poliedro não-vazio, racional e limitado.*

- (a) *se $P_I \neq \emptyset$ e a inequação $c^\top x \leq \delta$ (c inteiro) é válida para P_I , então existe uma prova de planos-de-cortes de $c^\top x \leq \delta$ a partir de $Ax \leq b$.*
- (b) *se $P_I = \emptyset$, então existe uma prova de planos-de-cortes de $0^\top x \leq -1$ a partir de $Ax \leq b$.*

□

2.2.2 Comprimento de provas de planos-de-corte

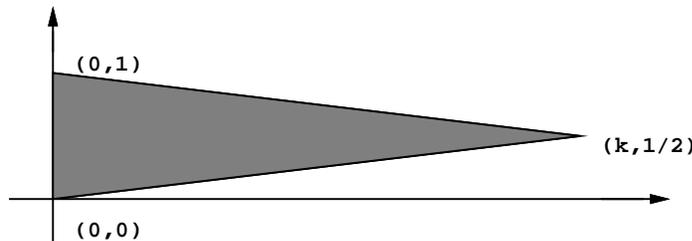
O Teorema 2.1.18 sugere o seguinte procedimento para achar o fecho inteiro de um poliedro P_I (supondo de dimensão plena).

- Encontre um sistema TDI minimal $Ax \leq b$ com A inteiro que define P ;
- Seja $P' := \{x \mid Ax \leq \lfloor b \rfloor\}$;
- Encontre um sistema TDI minimal $A'x \leq b'$ com A' inteiro que define P' ;
- Seja $P'' := \{x \mid A'x \leq \lfloor b' \rfloor\}$;
- Repita o processo acima até obter um poliedro, digamos $P^{(t)}$, para o qual o sistema TDI minimal é inteiro e tem lado direito também inteiro. Neste caso, $P^{(t)} = P_I$.

O menor t para o qual $P^{(t)} = P_I$ dá uma “medida” da complexidade do problema de achar o fecho inteiro do poliedro P . Pode-se mostrar que não existe um limite superior para t em função da dimensão do poliedro. Observe o exemplo a seguir.

Exemplo 2.2.7 *Considere P o seguinte poliedro no \mathbb{R}^2 :*

$$P := \text{conv}\{(0, 0), (0, 1), (k, 1/2)\}.$$



Note que $P_I = \text{conv}\{(0, 0), (0, 1)\}$, e que $(k - t, 1/2) \in P^{(t)}$ para todo $t < k$. Assim, $P^{(t)} \neq P_I$ para $t \leq k$.

No caso do Exemplo 2.2.7, note que t não é polinomialmente limitado pelo tamanho do sistema que define o poliedro P . Vale notar que o comprimento das provas de planos-de-corte também pode ser bem grande em relação ao tamanho do sistema.

São conhecidos alguns resultados a respeito de limitantes superiores para o comprimento de provas de planos-de-corte. Para o caso em que o poliedro P não contém vetores inteiros, em 1985, Cook, Coullard e Turán [15] mostraram que o número de planos-de-corte pode ser limitado em termos da dimensão de P , como mostra o resultado que enunciamos a seguir.

Teorema 2.2.8 *Se P é um poliedro racional de dimensão d , tal que $P_I = \emptyset$, então existe um natural $t(d)$ tal que $P^{(t(d))} = \emptyset$.* \square

Cook, Gerards, Schrijver e Tardos [16] mostraram em 1986 que o número de planos-de-cortes para um problema de programação linear inteira tem um limite superior que só depende da matriz de restrições.

Teorema 2.2.9 *Para cada matriz racional A , existe um número t tal que, para cada vetor b , tem-se que*

$$\{x \mid Ax \leq b\}^{(t)} = \{x \mid Ax \leq b\}_I.$$

\square

O Teorema 2.2.9 motiva as seguintes definições. Seja A uma matriz racional. Chamamos de **posto de Chvátal** ao menor número t tal que $\{x \mid Ax \leq b\}^{(t)} = \{x \mid Ax \leq b\}_I$, para cada vetor inteiro b . O **posto forte de Chvátal** é o posto da matriz

$$\begin{pmatrix} I \\ -I \\ A \\ -A \end{pmatrix}.$$

Alguns resultados que listamos a seguir seguem diretamente do Teorema 2.1.6 (de Hoffman e Kruskal). Esses resultados, a respeito de matrizes inteiras, mostram como o posto de Chvátal se relaciona com outros conceitos apresentados neste capítulo.

- Se A é uma matriz inteira, então A tem posto forte de Chvátal 0 se e somente se A é TU;
- Se A é uma matriz inteira, então o posto de Chvátal de A é 0 se e somente se A^\top é unimodular;
- Se A é TU então o posto forte de Chvátal de A é menor ou igual a 1, e o posto forte de Chvátal da matriz $2A$ também é menor ou igual a 1.

No próximo capítulo, mostramos que a matriz do sistema que descreve o poliedro dos emparelhamentos de um grafo tem posto de Chvátal 1.

2.2.3 Cortes de Gomory

Estamos prontos para apresentar o método de planos-de-cortes sugerido em 1958 por Gomory [43].

Considere o seguinte problema:

$$(P) \quad \begin{array}{l} \max \quad c^\top x \\ Ax = b \\ x \geq 0 \\ x \text{ inteiro.} \end{array}$$

De forma geral, o método dos planos-de-corte para resolver o problema acima pode ser assim descrito.

1. Resolva a relaxação linear (PL) correspondente ao problema (P), que é gerada eliminando a condição de integralidade da solução x . Se o (PL) não tem solução ótima, então (P) é inviável ou ilimitado. Caso (PL) seja limitado, tome x^* uma solução ótima de (PL).
2. Se x^* for inteiro, então pare (e devolva a solução x^*). Caso contrário, encontre uma inequação $d^\top x \leq d_0$ com as seguintes propriedades:

$$d^\top x^* > d_0 \text{ e } d^\top x \leq d_0, \text{ para todo } x \in P^=(A, b)_I.$$

3. Junte essa nova inequação ao sistema de equações (usando uma variável de folga) e volte ao passo 1.

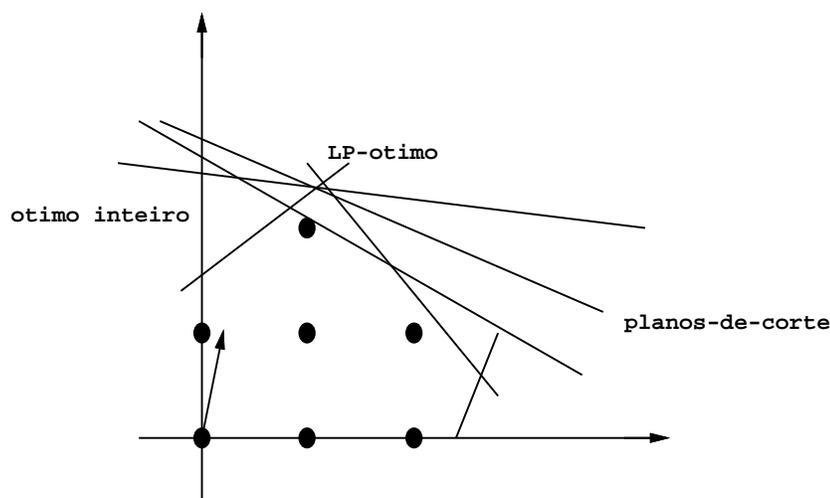


Figura 2.1: Método dos planos-de-corte

A inequação $d^\top x \leq d_0$ definida no passo 2 é chamada **plano-de-corte** (para a solução x^*).

Os vários “métodos de planos-de-corte” diferem entre si pelo tipo de planos-de-cortes utilizados no passo 2. O nome “plano-de-corte” vem do fato de que a solução ótima da relaxação atual vai sendo “cortada”, como vemos na Figura 2.1.

No próximo teorema apresentamos os cortes sugeridos por Gomory. Antes disso, vejamos a notação que será usada. Denotamos por A_B a base da matriz A associada à uma solução ótima x^* da relaxação linear. As variáveis correspondentes às colunas da base estão em B . As outras variáveis (não-básicas) estão em N . Denotamos por c^* o valor da solução ótima, ou seja, $c^* := c^\top x^*$. Além disso, definimos $\bar{A} := A_B^{-1} A_N$, $\bar{b} := A_B^{-1} b$, $\bar{c}^\top := (c_N^\top - c_B^\top \bar{A})$ e $\tilde{c} := -\bar{c}$.

Teorema 2.2.10 *Considere o problema $\max\{c^\top x \mid x \in P^=(A, b), x \text{ inteiro}\}$, onde A e b são inteiros. Seja x^* uma solução ótima do programa linear relaxado e A_B a base associada à solução x^* . Seja $h \neq 0$ um inteiro. Considerando a notação acima definida, temos que as inequações*

$$\sum_{j \in N} ([h\bar{a}_{ij}] - h\bar{a}_{ij})x_j \leq [h\bar{b}_i] - h\bar{b}_i, \text{ para todo } i \in B; a \quad (2.2)$$

$$\sum_{j \in N} ([h\tilde{c}_j] - h\tilde{c}_j)x_j \leq [hc^*] - hc^*, b \quad (2.3)$$

são planos-de-corte para a solução x^* , caso $h\bar{b}_i$ ou hc^* não sejam inteiros.

Prova. Sabemos que uma solução básica x , relativa à base A_B , pode ser escrita da seguinte forma:

$$x_B = A_B^{-1}b - A_B^{-1}A_N x_N = \bar{b} - \bar{A}x_N.$$

Com isso, $x_i = \bar{b}_i - \bar{A}_i x_N$ para todo $i \in B$. Ou seja,

$$x_i + \sum_{j \in N} \bar{a}_{ij}x_j = \bar{b}_i, \text{ para todo } i \in B.1 \quad (2.4)$$

Multiplicando (1) por h obtemos

$$hx_i + \sum_{j \in N} h\bar{a}_{ij}x_j = h\bar{b}_i, \text{ para todo } i \in B.2 \quad (2.5)$$

Como $x \geq 0$, podemos truncar os coeficientes do lado esquerdo da equação e obtemos a seguinte inequação válida

$$hx_i + \sum_{j \in N} [h\bar{a}_{ij}]x_j \leq h\bar{b}_i, \text{ para todo } i \in B.3 \quad (2.6)$$

Para as soluções inteiras, o lado esquerdo de (3) tem valor inteiro. Com isso, podemos truncar também o lado direito da inequação, obtendo-se a seguinte inequação válida:

$$hx_i + \sum_{j \in N} [h\bar{a}_{ij}]x_j \leq [h\bar{b}_i], \text{ para todo } i \in B.4 \quad (2.7)$$

Subtraindo (2) de (4) obtemos a inequação válida

$$\sum_{j \in N} ([h\bar{a}_{ij}] - h\bar{a}_{ij})x_j \leq [h\bar{b}_i] - h\bar{b}_i, \text{ para todo } i \in B.$$

Assim, se a solução ótima x^* não é inteira, então \bar{b}_i não é inteiro. Com isso, o lado esquerdo da inequação acima deve ser negativo. Como a solução básica x^* satisfaz $x_N^* = 0$, essa solução será cortada pela inequação acima.

Para a segunda inequação a prova é análoga. Sabemos que o valor objetivo $c_0 := c^\top x$ de um ponto x pode ser escrito em função de uma base A_B da seguinte forma:

$$c_0 = c^\top x = c_B^\top x_B + (c_N^\top - c_B^\top A_B^{-1} A_N) x_N = c_B^\top \bar{b} + \bar{c}^\top x_N.$$

Assim, $c_0 - \bar{c}^\top x_N = c_0 + \tilde{c}^\top x_N = c_B^\top \bar{b} = c^*$. Multiplicando a inequação acima por h , obtemos

$$hc_0 + \sum_{j \in N} h\tilde{c}_j x_j = hc^*. \quad (2.8)$$

Sabendo que $x \geq 0$, como acima, podemos truncar o lado esquerdo da equação, obtendo a inequação

$$hc_0 + \sum_{j \in N} [h\tilde{c}_j x_j] \leq hc^*. \quad (2.9)$$

Considerando que, para pontos inteiros o valor do lado esquerdo da inequação é um inteiro, podemos truncar também o lado direito, e obtemos

$$hc_0 + \sum_{j \in N} [h\tilde{c}_j x_j] \leq [hc^*]. \quad (2.10)$$

Subtraindo (5) de (7) obtemos a inequação

$$\sum_{j \in N} ([h\tilde{c}_j] - h\tilde{c}_j) x_j \leq [hc^*] - hc^*.$$

Com isso, fica completa a prova do teorema. □

Os planos-de-corte definidos no Teorema 2.2.10 são conhecidos como **cortes de Gomory**

Note que os dois cortes do Teorema 2.2.10 são suficientes para “cortar” soluções da relaxação linear. Caso o valor objetivo não seja inteiro, então uma inequação do tipo (b) pode ser usada para “cortar” a solução. Se o valor objetivo for inteiro, mas a solução não for, então uma inequação do tipo (a) pode ser usada. Ao acrescentar inequações desse tipo nenhum ponto inteiro do poliedro é excluído, e a base continua viável no simplex dual. Esta observação nos leva a um algoritmo em que acrescentamos inequações violadas do tipo (a) ou (b), resolvendo no passo seguinte o simplex dual, até que a solução da relaxação linear seja inteira. Este algoritmo é chamado **dual fracionário**. Uma descrição sucinta desse algoritmo é dada a seguir.

Algoritmo DUAL FRACIONÁRIO

Encontre uma solução ótima x^* do problema relaxado;

viável := *verdadeiro*;

enquanto x^* não é inteiro e viável = *verdadeiro*

faça escolha uma linha i ;

 adicione um corte de Gomory gerado pela linha i ;

 aplique o método simplex dual;

se o problema dual é ilimitado

então viável := *falso*

senão seja x^* a nova solução ótima.

Pode-se mostrar que, usando este algoritmo e uma versão especial do método simplex dual para a solução da relaxação linear, o algoritmo termina com uma solução inteira (caso exista) em um número finito de passos. Para discutir isso, vamos introduzimos algumas definições.

Um vetor não nulo $x \in \mathbb{R}^n$ é **lexicograficamente positivo** se seu primeiro componente não nulo é maior que zero. Se $x = 0$, dizemos que x é **lexicograficamente nulo**. Caso $-x$ seja lexicograficamente positivo, então x é **lexicograficamente negativo**. Denotamos esses casos por

$$x \stackrel{L}{>} 0, \quad x \stackrel{L}{=} 0 \quad \text{e} \quad x \stackrel{L}{<} 0.$$

Dizemos que um vetor $x \in \mathbb{R}^n$ é **lexicograficamente maior** (respectivamente, **menor**, **igual**) que $w \in \mathbb{R}^n$, caso $x - w \stackrel{L}{>} 0$ (respectivamente, $x - w \stackrel{L}{<} 0$, $x - w \stackrel{L}{=} 0$). Definimos também **lexicograficamente mínimo** e **lexicograficamente máximo** de forma análoga.

Teorema 2.2.11 *Aplique as seguintes regras no algoritmo dual fracionário descrito acima:*

- (a) *escolha a linha fonte como sendo a primeira linha com a_{i0} não-inteiro;*
- (b) *use a versão lexicográfica para o simplex dual.*

Então, assumindo que o problema original tem um limitante superior z para o valor da solução ótima, o algoritmo termina com uma solução inteira em um número finito de passos, ou descobre que não existe solução inteira viável para o problema original. □

O método de Gomory, entretanto, não é prático, pois o seu tempo de execução, apesar pode ser muito grande. Além disso, as inequações introduzidas no sistema trazem instabilidade numérica, e a convergência do método só é garantida usando-se precisão infinita.

2.2.4 Cortes faciais

Como já mencionamos na descrição geral dos métodos de planos-de-cortes, os vários métodos diferem de acordo com o tipo de plano-de-corte utilizado. Nesta seção discutimos como utilizar os chamados *cortes faciais*.

Considere o problema $\max\{c^\top x \mid x \in P(A, b), x \text{ inteiro}\}$. Sabemos (veja seção 2.1 deste capítulo) que este problema é equivalente ao problema

$$(P) \quad \max\{c^\top x \mid x \in P(A, b)_I\}.$$

Assim, se conhecemos inequações que definem facetas do poliedro $P(A, b)_I$, podemos utilizá-las como planos-de-corte.

Dizemos que tais planos-de-corte são **faciais**. Lembramos que as inequações que definem facetas do poliedro são as de melhor qualidade possível, pois estas inequações compõem uma descrição completa e irredundante do poliedro.

Suponha que $A'x \leq b'$ seja um sistema de inequações que definem facetas de $P(A, b)_I$. Seja

$$(P') \quad \max\{c^\top x \mid x \in P(A', b')\},$$

e seja (PR) o problema definido por um subsistema de $A'x \leq b'$.

Para encontrar uma solução ótima de (P') , proceda da seguinte forma:

1. Resolva (PR) .
2. Se (PR) não tem solução ótima, o problema (P') é inviável ou ilimitado. Como (P') é uma relaxação de (P) , então este também é inviável ou ilimitado. Pare.
3. Se (PR) é limitado, considere uma solução ótima x^* de (PR) .
 - Se x^* é um ponto inteiro viável de (P) , então é certamente uma solução ótima de (P') (e portanto de (P)).
 - Caso contrário, encontre uma inequação $d^\top x \leq d_0$ em $A'x \leq b'$ que corta a solução x^* . Junte essa inequação ao sistema atual, formando um novo (PR) .
 - Vá para o passo 1.

Como vimos, o método só difere do que foi apresentado na seção anterior pela escolha do tipo de plano-de-corte utilizado para cortar a solução ótima do problema relaxado. Da mesma forma, após introduzirmos novos cortes a base ótima desse novo problema continua viável para o dual. Este fato pode ser utilizado na implementação do método para melhorar a sua eficiência.

Na realidade, nem sempre é possível encontrar inequações que definem facetas do poliedro para adicionarmos ao problema relaxado. Em geral, nas implementações práticas tentamos encontrar tais facetas, mas nos damos por satisfeitos com uma inequação qualquer que corta a solução fracionária obtida.

2.3 Equivalência entre Otimização e Separação

Nesta seção discutiremos, de forma resumida e simplificada, este resultado fundamental na teoria de combinatória poliédrica. Uma apresentação precisa dos resultados exigiria a introdução de vários resultados, fugindo das pretensões deste texto. Ao leitor interessado, sugerimos consultar Grötschel, Lovász e Schrijver [53], no qual esta seção está baseada.

A seguir introduzimos alguns problemas que serão usados no decorrer desta seção. Seja P um poliedro (na verdade poderíamos tomar qualquer corpo convexo e compacto) e defina os seguintes problemas:

Problema de Otimização (OPT)

Dado um vetor $c \in \mathbb{R}^n$, encontre $x \in P$ que maximiza $c^\top x$, ou certifique que P é vazio.

Problema de Violação (VIOL)

Dado um vetor $c \in \mathbb{R}^n$ e $\gamma \in \mathbb{R}$, decida se $c^\top x \leq \gamma$ vale para todo $x \in P$; se não, encontre um vetor $y \in \mathbb{R}^n$ tal que $c^\top y > \gamma$.

Problema de Validade (VAL)

Dado um vetor $c \in \mathbb{R}^n$ e $\gamma \in \mathbb{R}$, decida se $c^\top x \leq \gamma$ vale para todo $x \in P$.

Problema da Separação (SEP)

Dado um vetor $y \in \mathbb{R}^n$, decida se $y \in P$; se $y \notin P$ encontre um hiperplano que separa y de P ; mais precisamente, encontre $c \in \mathbb{R}^n$ tal que $c^\top y > c^\top x$ para todo $x \in P$.

Problema da Pertinência (PERT)

Dado um vetor $y \in \mathbb{R}^n$, decida se $y \in P$.

Claramente, se sabemos resolver (OPT), então também sabemos resolver (VIOL). Também é imediato que resolver (VIOL) implica em resolver (VAL). Da mesma forma, (PERT) pode ser resolvido se (SEP) o for. Na figura abaixo mostramos de forma esquemática essas implicações através de flechas tracejadas (omitimos as implicadas por transitividade).

Note que a complexidade de cada um dos problemas apresentados depende essencialmente da forma como o poliedro P é dado. Por exemplo, (SEP) é trivial se o poliedro é dado através de sua representação externa (sistema de equações e inequações), mas não é trivial se a representação é a interna (conjunto de vértices e raios extremais). Para o problema (OPT) ocorre exatamente o oposto.

Outras implicações quanto à solubilidade de um problema em relação a outro, estão demonstradas em Grötschel, Lovász e Schrijver [53]. Tais implicações estão indicadas na Figura 2.2 com flechas contínuas.

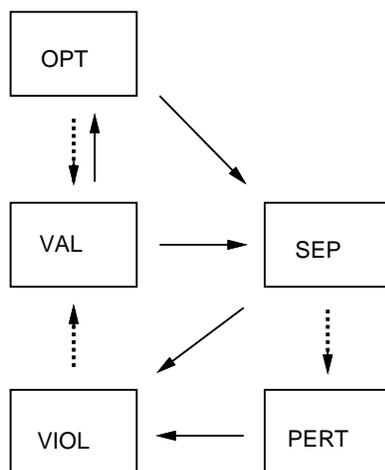


Figura 2.2: Relação entre os problemas apresentados

No contexto deste texto, o resultado mais importante é a equivalência entre os problemas (OPT) e (SEP). Tal resultado, que envolve o método elipsóide para programação linear, garante que os problemas de otimização e separação têm a mesma complexidade computacional. Assim, para poliedros associados a problemas de otimização \mathcal{NP} -difíceis, não podemos esperar que algoritmos polinomiais sejam encontrados para o problema da separação correspondente, a menos que $\mathcal{P} = \mathcal{NP}$. No próximo capítulo veremos alguns exemplos de problemas formulados como PL (de tamanho exponencial em relação à entrada do problema) para os quais o problema da separação é polinomial. Com isso, podemos concluir que existe um algoritmo polinomial para o problema (ainda que um tal algoritmo não seja de interesse prático, por envolver o uso do método elipsóide).

Capítulo 3

Poliedros de Problemas de Otimização Combinatória

Neste capítulo definimos formalmente os tipos de problemas de otimização combinatória que podem ser tratados por métodos poliédricos. Fazemos um apanhado geral a respeito da evolução dos métodos poliédricos na solução de problemas de otimização combinatória. Mostramos como formular esses problemas como problemas de otimização sobre um poliedro, e como desenvolver um algoritmo baseado nessa formulação.

Um passo fundamental nessa abordagem é o estudo do poliedro que associamos ao problema. Mais precisamente, estamos interessados numa descrição desse poliedro, dada por um sistema de inequações que definem facetas. Para ilustrar como isso pode ser feito, apresentamos diversos poliedros bem-estudados associados a problemas clássicos de otimização combinatória. Dentre esses problemas, destacamos o problema do emparelhamento máximo, o problema do caminho mínimo, o problema do caixeiro viajante e o problema de Steiner em grafos.

3.1 Otimização Combinatória e Métodos Poliédricos

Problemas de otimização, na sua forma geral, têm como objetivo maximizar ou minimizar uma função definida sobre um certo domínio. A teoria clássica de otimização trata do caso em que o domínio é infinito. Já no caso dos chamados problemas de *otimização combinatória*, o domínio é finito; e tipicamente, seus elementos podem ser “facilmente” gerados ou testados quanto à sua pertinência a este domínio. Posto dessa forma, tais problemas parecerem relativamente simples, pois em princípio, bastaria testarmos todos os elementos do domínio na busca de um que seja ótimo. De fato, essa idéia seria perfeitamente válida se a questão fosse simplesmente encontrar um algoritmo.

Na maioria dos problemas de otimização combinatória, a questão da existência de um algoritmo que resolve o problema não é difícil, mas sim o desenvolvimento de algoritmos que, de alguma forma esperta, restrinjam a busca a um subconjunto do domínio. Essa restrição é não apenas desejável, mas imprescindível, se queremos resolver instâncias de tamanho relativamen-

te grande do problema. Vale notar que, na maioria dos casos que nos interessam, o domínio tem tamanho não-polinomial em relação à entrada do problema (essa é a razão que nos leva a descartar a idéia ingênua de se testar todo o domínio).

Veremos neste capítulo exemplos de problemas de otimização combinatória que são fáceis e outros que são \mathcal{NP} -difíceis, para ilustrar que ambos os tipos podem ser tratados através de métodos poliédricos. Aliás, cabe aqui mencionar que instâncias bem grandes de certos problemas \mathcal{NP} -difíceis, como o problema do caixeiro viajante, só puderam ser resolvidas com o uso do método *branch and cut* que descrevemos no Capítulo 4.

Para expor nossas idéias, vamos definir formalmente a classe de problemas que serão nosso objeto de estudo.

Considere uma tripla da forma (E, c, \mathcal{S}) , onde E é um conjunto finito, $c : E \rightarrow \mathbb{R}$ é uma função custo que associa a cada elemento de E um valor real, e \mathcal{S} é um conjunto de soluções viáveis (domínio), formado por subconjuntos de E . Dada uma tal tripla, deseja-se encontrar $S \in \mathcal{S}$ que maximiza (ou minimiza) $c(S) := \sum_{e \in S} c(e)$. Problemas que podem ser formulados dessa maneira são chamados de **Problemas de Otimização Combinatória Linear**.

Note que estamos considerando uma classe especial de problemas de otimização onde a função objetivo é linear (o que não necessariamente precisaria ocorrer). Para os nossos propósitos (aplicação de métodos poliédricos), vamos nos restringir a essa classe de problemas, denotando-os simplesmente por **POC**.

Ainda que, de certa forma, tal classe possa parecer restrita, veremos que vários problemas interessantes podem ser formulados dessa maneira. Tipicamente, os problemas que trataremos aqui são problemas sobre grafos, que têm a seguinte forma geral:

Dado um grafo G , com custos nas arestas (ou nos vértices), encontrar em G um subgrafo com propriedade π (por exemplo, ser hamiltoniano, ser uma árvore geradora) que seja de custo mínimo (ou máximo).

Em particular, focalizaremos os seguintes problemas.

1. **Problema do Emparelhamento Máximo.** Dado um grafo $G = (V, A)$ e uma função custo $c : A \rightarrow \mathbb{R}$ definida nas arestas de G , achar um emparelhamento F em G tal que $c(F)$ seja máximo.
2. **Problema do Caminho Mínimo.** Dado um grafo orientado $D = (V, A)$, dois vértices distintos s, t em V e uma função custo $c : A \rightarrow \mathbb{R}$ definida nos arcos de D , encontrar em D um caminho de s a t que seja de custo mínimo.
3. **Problema do (s, t) -Corte Mínimo.** Dado um grafo orientado $D = (V, A)$, dois vértices distintos s, t em V e uma função capacidade $c : A \rightarrow \mathbb{R}$, encontrar em D um (s, t) -corte de capacidade mínima.
4. **Problema da Árvore Geradora Mínima.** Dado um grafo $G = (V, A)$ e uma função custo $c : A \rightarrow \mathbb{R}$, achar uma árvore geradora F em G tal que $c(F)$ seja mínimo.

5. **Problema do Caixeiro Viajante.** Dado um grafo $G = (V, A)$ e uma função custo $c : A \rightarrow \mathbb{R}$, encontrar em G um circuito hamiltoniano de custo mínimo.
6. **Problema de Steiner em Grafos.** Dado um grafo $G = (V, A)$, um conjunto de vértices $Z \subseteq V$ chamados terminais e uma função custo $c : A \rightarrow \mathbb{R}$, encontrar $S \subseteq A$ tal que $G[S]$ conecta todos os terminais e $c(S)$ seja mínimo.

Note que, todos os problemas acima se caracterizam pelo fato de o domínio ser finito, e de que podemos testar facilmente se um subconjunto de arestas de G tem a propriedade desejada. É imediato também que esses problemas podem ser formulados como um POC. Por exemplo, no caso do problema 1, basta tomar E como sendo o conjunto das arestas de G e $\mathcal{S} := \{F \subseteq E \mid F \text{ é um emparelhamento em } G\}$.

Para fazer uso de métodos poliédricos, vejamos como associar um poliedro a um POC constituído por uma tripla (E, c, \mathcal{S}) . Seja \mathbb{R}^E o espaço vetorial real $|E|$ -dimensional onde cada vetor $x \in \mathbb{R}^E$ tem seus componentes indexados pelos elementos de E , ou seja, $x = (x_e)_{e \in E}$. Para cada elemento $S \in \mathcal{S}$, associamos a esse elemento um ponto em \mathbb{R}^E , correspondente ao seu *vetor de incidência* $\chi^S = (\chi_e^S)_{e \in E}$, definido como

$$\chi_e^S := \begin{cases} 1, & \text{se } e \in S; \\ 0, & \text{caso contrário.} \end{cases}$$

Consideramos agora o politopo $P \subseteq \mathbb{R}^E$, definido como o fecho convexo dos vetores de incidência dos subconjuntos em \mathcal{S} , isto é,

$$P := \text{conv}\{\chi^S \in \mathbb{R}^E \mid S \in \mathcal{S}\}.$$

Exercício 3.1.1 *Seja P o politopo definido acima. Mostre que o conjunto dos vértices de P é precisamente o conjunto dos vetores de incidência dos elementos $S \in \mathcal{S}$.*

Dessa forma, o problema original é equivalente ao problema (considerando c como um vetor em \mathbb{R}^E)

$$\max\{c^\top x \mid x \in P\}.$$

Pelo Exercício 3.1.1, sabemos que o conjunto dos vértices do politopo P é precisamente o conjunto dos vetores de incidência dos elementos em \mathcal{S} . Ou seja, existe uma correspondência biunívoca entre os vértices de P e as soluções viáveis (o domínio) do problema. Ademais, como P é um politopo, existe um sistema de inequações lineares que define P . Assim, aparentemente, transformamos o problema original em um problema de programação linear (PL). Ainda que isso possa parecer bastante satisfatório, na verdade, as coisas não são tão simples assim.

Uma das dificuldades está em encontrar um tal sistema. A outra, reside no fato de que esse sistema pode ter tamanho exponencial (em relação às entradas do problema original). Apesar dessas dificuldades, veremos que, mesmo tendo apenas uma descrição parcial do politopo, é possível desenvolvermos um algoritmo para o problema. Além disso, esta abordagem permite

eventualmente descobrir casos especiais de problemas difíceis que podem ser resolvidos em tempo polinomial. Por exemplo, no caso do problema do corte máximo, foi provado [7] que este problema, que é \mathcal{NP} -difícil no caso geral, pode ser resolvido em tempo polinomial para grafos planares (ou mais geralmente, para grafos que não são contrátiveis ao K_5).

A redução que apresentamos mostra que todo POC pode ser formulado como um PL 0/1. A recíproca dessa afirmação também é verdadeira. De fato, dado um PL 0/1 da forma $\max\{c^\top x \mid Ax \leq b, x \in \{0, 1\}\}$, onde $A \in \mathbb{R}^{m,n}$ e $b \in \mathbb{R}^m$, basta tomarmos $E := \{1, 2, \dots, n\}$ e $\mathcal{S} := \{S \subseteq E \mid A\chi^S \leq b\}$.

Essas observações nos mostram que os POC constituem uma classe equivalente à classe dos PL 0/1. A técnica que veremos também se aplica a problemas de otimização combinatória que podem ser formulados como PLI não necessariamente 0/1. Ou seja, problemas do tipo

$$\max\{c^\top x \mid Ax \leq b, x \text{ inteiro}\}.$$

Neste caso, considerando o fecho inteiro Q_I do poliedro Q definido pelo sistema $Ax \leq b$, o problema acima é equivalente ao problema

$$\max\{c^\top x \mid x \in Q_I\}.$$

Como já vimos no Capítulo 2, se Q é um poliedro racional, então existe um sistema linear que descreve Q_I .

Resumindo, por enquanto só vimos que certos problemas de otimização combinatória podem ser reduzidos a problemas de programação linear sobre um poliedro inteiro, definido implicitamente. Interessa-nos agora, mostrar como obter descrições de tais poliedros. Lembramos que no Capítulo 2 já mencionamos como podemos fazer uso de uma tal descrição para desenvolver um algoritmo baseado em métodos de planos-de-corte faciais.

Sabemos que uma descrição não-redundante de um poliedro é dada por um sistema de inequações que definem facetas. Nem sempre porém, conseguimos encontrar uma descrição completa desse tipo. Aliás, sabe-se que problemas que são \mathcal{NP} -difíceis são pouco prováveis de terem uma descrição completa e “boa” (veja [72] e [94]). Sabe-se também que, se fosse encontrada uma descrição completa de um politopo associado a um problema \mathcal{NP} -difícil e o correspondente problema da separação pudesse ser resolvido em tempo polinomial, então teríamos que $\mathcal{P} = \mathcal{NP}$.

Neste ponto, cabe ressaltar que essa estratégia de se associar um politopo a um problema, e mais especificamente, a tarefa de encontrar um sistema de inequações que descreve um tal poliedro (definido implicitamente pelos seus vértices), ou provar que certos sistemas lineares descrevem poliedros inteiros, constituem tópicos centrais de pesquisa em combinatória poliédrica.

O desenvolvimento desta subárea, que surgiu da tentativa de se desenvolver algoritmos para problemas de otimização combinatória, pode ser dividido em 3 períodos, caracterizados por resultados marcantes em termos de abordagens algorítmicas.

O primeiro uso explícito de métodos de combinatória poliédrica parece ter ocorrido em 1943 num artigo de R. Rado [96], que fazia uso do Lema de Farkas, ainda que de maneira indireta. Entretanto, o *primeiro período* de desenvolvimento de combinatória poliédrica teve início nos anos 50, após o descobrimento do método simplex para programação linear. Data deste período a observação de que muitos problemas de otimização combinatória, em particular, vários problemas de fluxos em redes, podem ser formulados como problemas de programação linear inteira, mas que devido à total-unimodularidade da matriz de restrições, tais programas lineares inteiros podem ser tratados simplesmente como programas lineares. Conseqüentemente, o método simplex — e mesmo dualidade em programação linear — podem ser aplicados. Tal observação constituiu um marco e direcionou muitas pesquisas que foram realizadas neste período, quando então vários resultados mini-max foram descobertos (veja [21, 29, 34, 35, 64, 66, 74, 75, 76, 98]).

O *segundo período* de desenvolvimento teve início em meados da década de 60 quando J. Edmonds [24] obteve uma descrição linear completa — de tamanho exponencial — para o politopo dos emparelhamentos de um grafo. O fato interessante foi a descoberta de que mesmo uma descrição de tamanho exponencial pode ser útil para o desenvolvimento de algoritmos polinomiais. Ademais, tais algoritmos podem fazer uso, ou mesmo provar, relações de dualidade em programação linear. Isso levou a tentativas “infrutíferas” de se obter descrições lineares completas de problemas que mais tarde foram provados ser \mathcal{NP} -difíceis.

O *terceiro período* teve início em 1979 com o desenvolvimento do método elipsóide para programação linear, devido a Khachiyan [73]. A conseqüência importante para a área de combinatória poliédrica não foi simplesmente a constatação de que programas lineares podem ser resolvidos em tempo polinomial, mas o princípio no qual tal método se baseia.

Tal princípio básico reside no fato de que é possível resolver eficientemente um programa linear desde que se saiba resolver eficientemente o problema da separação. Ademais, descobriu-se que vale não apenas o fato de que é possível se encontrar uma solução ótima sabendo-se resolver o problema da separação, mas também é verdadeira a recíproca dessa afirmação. Ou seja, se sabemos encontrar eficientemente uma solução ótima então também sabemos resolver eficientemente o problema da separação.

A “equivalência entre otimização e separação” (mencionada no Capítulo 2) foi uma descoberta que se tornou um marco no que se refere ao desenvolvimento de algoritmos na área de combinatória poliédrica, trazendo como conseqüência novas perspectivas com relação ao uso de métodos poliédricos no tratamento de problemas de otimização combinatória. Ficou transparente que associar poliedros a problemas de otimização combinatória e resolver o problema da separação correspondente, poderia ser uma estratégia para se conseguir algoritmos polinomiais para certos problemas. E que, no caso de problemas \mathcal{NP} -difíceis, tal abordagem também era interessante, principalmente no tratamento de problemas de grande porte, para os quais a busca de soluções exatas através de outros métodos mostravam-se inviáveis.

Essa descoberta e as conseqüências advindas daí no tratamento de problemas de otimização combinatória constituem a essência do artigo “The Ellipsoid Method and its Consequences in Combinatorial Optimization” de Grötschel, Lovász e Schrijver [51, 52] que recebeu o Fulkerson Prize em 1982 pela sua relevância na área de programação linear e otimização combinatória.

Mais tarde, em 1988, esses mesmos autores vieram a publicar o livro “Geometric Algorithms and Combinatorial Optimization” onde essas idéias germinais foram estendidas e investigadas a fundo [53].

O conseqüente reflorescimento do uso de técnicas de combinatória poliédrica, que ao ser utilizada por Dantzig, Fulkerson e Johnson [22] em 1954 recebeu pouca atenção e aceitação na comunidade, pode ser constatado pelos artigos a este respeito que surgiram na literatura na década de 80 (veja [49, 50, 57, 58, 91, 89]).

Para situarmo-nos no contexto histórico, vale aqui lembrar como foi a evolução em termos algorítmicos do uso da abordagem poliédrica na resolução de instâncias reais de problemas \mathcal{NP} -difíceis.

O já mencionado artigo de Dantzig, Fulkerson e Johnson [22] de 1954 foi o primeiro a usar tal abordagem, relatando como uma instância do problema do caixeiro viajante de 49 cidades foi resolvida. Neste trabalho pioneiro, o uso desta técnica é ilustrada com uma série de exemplos, onde muitos dos passos (para encontrar planos-de-corte) são realizados manualmente, de uma maneira interativa e *ad hoc*. Esta abordagem foi “esquecida” por um longo tempo até que em 1977, Grötschel em sua tese de doutoramento [45] fez um estudo mais aprofundado da estrutura facial dos politopos associados ao problema do caixeiro viajante simétrico e assimétrico e resolveu uma instância deste problema (caso simétrico) com 120 cidades da Alemanha. Isto porém, foi feito de maneira sistemática e totalmente automática (sem nenhuma intervenção manual para encontrar planos-de-corte).

Outros trabalhos seguiram-se na década de 80, muitos deles sobre o problema do caixeiro viajante [57, 80, 91, 89]. Este problema continua desafiando os pesquisadores, e tem sido fonte de inspiração para o desenvolvimento de muitas técnicas, úteis também no tratamento de outros problemas. Além desse, vários outros problemas \mathcal{NP} -difíceis também foram pesquisados sob esta abordagem: o problema do subgrafo acíclico [70], o problema da ordem linear [49, 50], o problema do corte máximo [6], o problema da partição em cliques [58, 61], o problema das mochilas múltiplas [31], *etc.*

Todos esse problemas \mathcal{NP} -difíceis têm em comum o fato de que é conhecida apenas uma descrição parcial dos politopos associados a esses problemas, sendo que para muitos deles, o problema da separação foi resolvido apenas para classes especiais de facetas. Como já observamos, tais resultados têm conseqüências muito interessantes em termos de casos especiais do problema que podem ser resolvidos em tempo polinomial.

Os primeiros algoritmos implementados usando essa abordagem poliédrica faziam uso do método simplex (para resolver os vários PL que são gerados), o que era natural, já que outros métodos ainda não eram conhecidos. Embora em 1979 tenha surgido o algoritmo polinomial para PL devido a Khachiyan [73], este não se mostrou útil do ponto de vista prático, e por essa razão não foi adotado para uso em contextos de planos-de-corte.

Assim, o método simplex, que se celebrou por se mostrar muito eficiente na prática (embora no pior caso seja exponencial), é o método que tem sido usado nas implementações de algoritmos baseados em planos-de-corte. Na falta de outros produtos (*software*) mais convenientes, o

pacote MPSX/370 da IBM [69] (uma implementação em PL/1 do método simplex revisado) foi largamente usado nos algoritmos desenvolvidos na década de 80.

Mais recentemente, foi desenvolvido o CPLEX [17] — uma implementação do método simplex na linguagem C — que tem a vantagem de ser muito conveniente para uso em contextos de planos-de-corte, onde operações como adicionar e eliminar restrições são freqüentes.

Para finalizar, vale lembrar que outros métodos polinomiais para a resolução de programas lineares foram descobertos na década de 80: os chamados métodos de pontos interiores. Esses métodos tiveram suas idéias lançadas num algoritmo desenvolvido por Karmarkar [71] em 1984, não tendo os defeitos do método dos elipsóides em relação ao seu comportamento na prática, considerado muito aquém de satisfatório. Tais métodos, estão sendo acoplados ao CPLEX.

O propósito das considerações acima foi dar uma indicação do estado-da-arte do uso da abordagem poliédrica para o tratamento de problemas de otimização combinatória. Os resultados que serão apresentados nas próximas seções serão mais técnicos, objetivando mostrar ao leitor como se prova que certos sistemas lineares descrevem um dado poliedro, ou como se prova que certas inequações definem facetas.

Cada uma das seções seguintes focalizam um problema ou uma classe de problemas que estão interligados. Para cada problema, descrevemos o(s) poliedro(s) associados(s) a esse problema.

3.2 Poliedro dos Emparelhamentos

O *Problema do Emparelhamento Máximo* é o seguinte:

Dado um grafo $G = (V, A)$ e um função custo $c : A \rightarrow \mathbb{R}$ definida nas arestas de G , encontrar um emparelhamento que seja de custo máximo.

Definimos o **poliedro dos emparelhamentos** de um grafo G , denotado por $P_{Emp}(G)$, como o fecho convexo dos vetores de incidência dos emparelhamentos em G . Ou seja,

$$P_{Emp}(G) := \text{conv}\{\chi^E \in \mathbb{R}^A \mid E \text{ é um emparelhamento em } G\}.$$

Primeiramente, observemos que o conjunto vazio, e para cada aresta $a \in A$, o conjunto $\{a\}$ é um emparelhamento. Como os vetores de incidência desses conjuntos são afim-independentes, segue imediatamente o seguinte resultado.

Lema 3.2.1 *O poliedro $P_{Emp}(G)$ tem dimensão plena, ou seja,*

$$\dim(P_{Emp}(G)) = |A|.$$

□

Considere as seguintes inequações:

$$x_a \geq 0, \text{ para todo } a \in A,$$

$$x(\delta(v)) \leq 1, \text{ para todo } v \in V.$$

Lembrando que $\delta(v)$ denota o conjunto das arestas que incidem em v , é imediato que essas inequações são válidas para $P_{Emp}(G)$. Logo,

$$P_{Emp}(G) \subseteq P_1(G) := \{x \in \mathbb{R}^A \mid x_a \geq 0, \text{ para todo } a \in A; \\ x(\delta(v)) \leq 1, \text{ para todo } v \in V\}.$$

É fácil ver que o sistema acima é uma formulação inteira do problema dos emparelhamentos, ou seja, todo ponto inteiro que satisfaz as condições acima é um emparelhamento e vice versa, ou seja, todo vetor de incidência de um emparelhamento satisfaz estas condições. Por outro lado, não é difícil encontrar grafos G para os quais a inclusão acima é estrita. Basta considerarmos, por exemplo, $G = K_3$ e observarmos que $x = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ é um vértice de $P_1(K_3)$ que não pertence a $P_{Emp}(K_3)$. Analisando esse exemplo, imediatamente notamos que a seguinte inequação está sendo violada por um tal vértice:

$$x(A) \leq 1, \text{ onde } A \text{ é o conjunto das arestas de } K_3.$$

De um modo geral, sempre que temos um conjunto $S \subseteq V$, $|S| \geq 3$ e ímpar, é imediato que qualquer emparelhamento em G pode ter no máximo $(|S| - 1)/2$ arestas do conjunto $A(S)$ (arestas com ambos os extremos em S). Ou seja, a seguinte inequação é válida para $P_{Emp}(G)$:

$$x(A(S)) \leq (|S| - 1)/2.$$

Considerando que,

$$P_{Emp}(G) = \text{conv}\{x \in \mathbb{Z}^A \mid x \in P_1(G)\} := \{P_1(G)\}_I,$$

de acordo com a teoria vista no Capítulo 2, sabemos que todas as inequações que definem facetas de $P_{Emp}(G)$ podem ser obtidas por “combinação e arredondamento” das inequações que definem $P_1(G)$.

No caso da inequação acima, vejamos como ela pode ser obtida por esse processo. Primeiramente, somamos as inequações $x(\delta(v)) \leq 1$ para todo $v \in S$, com as inequações $-x_e \leq 0$ para todo $e \in \delta(S)$, obtendo a inequação $2x(A(S)) \leq |S|$. Dividindo essa inequação por 2 temos $x(A(S)) \leq |S|/2$. Arredondando o lado direito (pois $|S|$ é ímpar), obtemos a inequação desejada. Note que essa *prova de plano-de-corte*, nos mostra que essa inequação tem *comprimento de prova* igual a 1. Com o resultado que veremos a seguir, poderemos concluir que o sistema que define o poliedro $P_1(G)$ tem *posto de Chvátal* igual a 1 (veja Capítulo 2).

Incluindo essas últimas inequações na descrição de $P_{Emp}(G)$, temos que

$$P_{Emp}(G) \subseteq \{x \in \mathbb{R}^A \mid x_a \geq 0, \text{ para todo } a \in A, \\ x(\delta(v)) \leq 1, \text{ para todo } v \in V,$$

$$x(A(S)) \leq (|S| - 1)/2, \text{ para todo } |S| \geq 3 \text{ e ímpar}.$$

A prova de que a inclusão acima é de fato uma igualdade, foi obtida em 1965 por Edmonds [24]. Este é um dos resultados mais conhecidos e referenciados em combinatória poliédrica, muito importante pelas consequências que trouxe nessa área. A idéia da prova que apresentamos aqui é devida a Lovász [81].

Teorema 3.2.2 *Para qualquer grafo $G = (V, A)$, o poliedro $P_{Emp}(G)$ é descrito pelo sistema*

- (1) $x_a \geq 0$, para todo $a \in A$;
- (2) $x(\delta(v)) \leq 1$, para todo $v \in V$;
- (3) $x(A(S)) \leq (|S| - 1)/2$, para todo $S \subseteq V$, $|S| \geq 3$ e ímpar.

Prova. Seja $a^\top x \leq \alpha$ uma inequação que induz uma faceta, digamos F_a , de $P_{Emp}(G)$. Ou seja,

$$F_a := \{x \in P_{Emp}(G) \mid a^\top x = \alpha\}.$$

Vamos chamar de F_a -emparelhamento, um emparelhamento cujo vetor de incidência pertence a F_a . Sejam

$$\begin{aligned} F_1(e) &:= \{x \in \mathbb{R}^A \mid x_e = 0\}; \\ F_2(v) &:= \{x \in \mathbb{R}^A \mid x(\delta(v)) = 0\}; \\ F_3(S) &:= \{x \in \mathbb{R}^A \mid x(A(S)) = (|S| - 1)/2\}. \end{aligned}$$

Note que, como F_a é uma faceta, é suficiente provarmos que $F_a \subseteq F_1(e)$ para algum $e \in A$, ou $F_a \subseteq F_2(v)$ para algum $v \in V$, ou $F_a \subseteq F_3(S)$ para algum $S \subseteq V$, $|S| \geq 3$ e ímpar.

Caso 1. Suponha que exista $e \in A$ tal que $a_e < 0$.

Neste caso, podemos concluir que todo F_a -emparelhamento, digamos E , satisfaz $x_e = 0$. De fato, se isso não ocorresse, então $F := E - \{e\}$ seria um emparelhamento em G tal que $a^\top \chi^F > \alpha$, um absurdo.

Caso 2. Suponha que $a_e \geq 0$ para todo $e \in A$, e que exista $v \in V$ tal que v é coberto por todo F_a -emparelhamento.

Seja E um F_a -emparelhamento. Como, por hipótese, v é coberto por E , segue imediatamente que χ^E satisfaz $x(\delta(v)) = 1$. Portanto, $F_a \subseteq F_2(v)$.

Caso 3. Suponha que $a_e \geq 0$ para todo $e \in A$, e que para cada vértice $v \in V$ exista um F_a -emparelhamento que não cobre v .

Seja $A^+ := \{e \in A \mid a_e > 0\}$ e G^+ o subgrafo de G induzido por A^+ .

Vamos mostrar que todo F_a -emparelhamento satisfaz a equação $x(A(S)) = (|S| - 1)/2$, onde S é o conjunto dos vértices de G^+ .

Para não interrompermos o raciocínio, vamos assumir que G^+ é conexo (a prova será feita depois).

Suponha que exista um F_a -emparelhamento que não cobre (pelo menos) dois vértices de G^+ . Dentre todos os tais emparelhamentos, escolha um emparelhamento E_1 , e um par de vértices u, v não cobertos por E_1 , tais que a distância entre u e v em G^+ seja mínima.

Note que u e v são não-adjacentes em G^+ (caso contrário, $F := E_1 \cup \{uv\}$ seria um emparelhamento tal que $a^\top \chi^F > \alpha$). Existe então um vértice, digamos w , num caminho mais curto de u para v .

Seja E_2 um F_a -emparelhamento que não cobre w . Pela escolha de E_1 , podemos concluir que E_2 cobre u e v ; e pela escolha de u e v , temos que E_1 cobre w .

Seja H o grafo formado pelas arestas $E_1 \cup E_2$. Como os vértices u, v e w têm grau 1 em H , existe em H um caminho alternante, digamos Q , que contém u ou v (eventualmente ambos), mas não contém w . Suponha que Q contém u e denote por AQ o conjunto das arestas no caminho Q .

Considere agora os emparelhamentos

$$F_1 := E_1 \Delta AQ \quad \text{e} \quad F_2 := E_2 \Delta AQ.$$

Note que $F_1 \cup F_2 = E_1 \cup E_2$ e $F_1 \cap F_2 = E_1 \cap E_2$. Logo, F_1 e F_2 são F_a -emparelhamentos. Como F_2 não cobre w e nem u , isso contraria a escolha de E_1 .

Tendo concluído que todo F_a -emparelhamento deixa de cobrir no máximo um vértice de S , e lembrando a hipótese que estamos assumindo no caso 3, temos que $|S| \geq 3$ e ímpar, e que todo F_a -emparelhamento satisfaz a equação $x(A(S)) = (|S| - 1)/2$. Logo, $F_a \subseteq F_3(S)$.

Para completar a prova, resta mostrar que G^+ é conexo. Suponhamos que G^+ não seja conexo e que $G^+ = G_1 \cup G_2$, onde G_1 e G_2 são subgrafos disjuntos de G . Vamos mostrar que $a^\top x \leq \alpha$ pode ser escrito como soma de outras duas inequações válidas para $P_{Emp}(G)$.

Para $i = 1, 2$, seja $g^i = (a_e^i : e \in A)$, onde $g_e^i = a_e$, se e é uma aresta de G_i ; e $g_e^i = 0$, caso contrário.

Claramente, $g^1 + g^2 = a$. Seja E_1 um emparelhamento cujo vetor de incidência maximiza a função $g^i x$ e seja α_i o valor desse máximo. Então, $\alpha_1 + \alpha_2 = \alpha$. De fato, se E é um F_a -emparelhamento então

$$\alpha_1 + \alpha_2 \geq g^1 \chi^E + g^2 \chi^E = a \chi^E = \alpha.$$

Por outro lado, podemos assumir que E_i só contém arestas de G_i . Então $E := E_1 \cup E_2$ é um emparelhamento em G , e

$$\alpha_1 + \alpha_2 = g^1 \chi^{E_1} + g^2 \chi^{E_2} = (g^1 + g^2) \chi^E = a \chi^E \leq \alpha.$$

Assim, $\alpha_1 + \alpha_2 = \alpha$, donde segue que $a^\top x \leq \alpha$ é soma das inequações $g^1 x \leq \alpha_1$ e $g^2 x \leq \alpha_2$ (ambas válidas para $P_{Emp}(G)$), o que é uma contradição ao fato de que $a^\top x \leq \alpha$ define uma faceta. \square

A prova do teorema acima permite-nos dizer que as inequações que definem facetas de $P_{Emp}(G)$ são de um dos tipos mencionados no enunciado do teorema. Veremos a seguir que, a menos das inequações de não-negatividade, não é verdade que todas elas definem facetas.

As inequações relativas a um conjunto S de vértices são chamadas **inequações blossom**, em alusão aos *blossoms* introduzidos por Edmonds [25], quando desenvolveu um algoritmo polinomial para resolver o problema do emparelhamento máximo.

O próximos resultados mostram quais são as inequações que realmente definem facetas de $P_{Emp}(G)$.

Teorema 3.2.3 *Para cada aresta $a \in A$ a inequação*

$$x_a \geq 0$$

define uma faceta do poliedro $P_{Emp}(G)$.

Prova. Considere uma aresta $a \in A$ e a correspondente inequação $x_a \geq 0$. Claramente, os vetores de incidência do emparelhamento vazio e dos emparelhamentos $\{e\}$ em G , $e \neq a$, satisfazem essa inequação com igualdade. Temos assim, $|A|$ vetores afim-independentes na face definida por essa inequação, donde se conclui que essa face é maximal, e portanto uma faceta. \square

Teorema 3.2.4 *Seja $G = (V, A)$ um grafo conexo com pelo menos 3 vértices, e seja v um vértice de G . Então a inequação*

$$x(\delta(v)) \leq 1$$

define uma faceta de $P_{Emp}(G)$ se e só se os dois vizinhos de v são não-adjacentes sempre que $\text{grau}(v) = 2$.

Prova. Considere um vértice $v \in V$ e suponha que u e w são os dois únicos vizinhos de v , com u e w adjacentes. Neste caso, a inequação blossom $x_{vu} + x_{vw} + x_{uw} \leq 1$ é válida para $P_{Emp}(G)$. Somando essa inequação com a inequação $-x_{uw} \leq 0$, temos que a inequação $x(\delta(v)) := x_{vu} + x_{vw} \leq 1$ também é válida. Logo, essa inequação não define uma faceta, já que é uma combinação de outras inequações que são válidas.

Para provar a implicação reversa, considere v um vértice nas condições do teorema. Para cada aresta a incidente a v considere o emparelhamento $\{a\}$; e para cada aresta a que não incide em v , escolha uma aresta f incidente a v , tal que f não é adjacente a a , e considere o emparelhamento $\{a, f\}$. Os vetores de incidência desses $|A|$ emparelhamentos satisfazem a

inequação $x(\delta(v)) \leq 1$ com igualdade e são afim-independentes. Logo, essa inequação define uma faceta de $P_{Emp}(G)$. \square

Para finalizar, falta provar para a classe das inequações *blossom*, quais delas definem facetas. Antes de fazermos isso, lembramos que um grafo é **hipo-emparelhável** se não possui um emparelhamento perfeito, mas ao remover qualquer um de seus vértices o grafo que resulta possui um emparelhamento perfeito.

Teorema 3.2.5 *Seja $G = (V, A)$ um grafo conexo com pelo menos 3 vértices, e seja $S \subseteq V$ tal que $|S| \geq 3$ e ímpar. Então a inequação*

$$x(A(S)) \leq (|S| - 1)/2$$

define uma faceta de $P_{Emp}(G)$ se e só se o subgrafo induzido por S é 2-conexo e hipo-emparelhável.

Prova. Para todo $S \subseteq V$, $|S| \geq 3$, a inequação $x(A(S)) \leq (|S| - 1)/2$ é claramente válida para $P_{Emp}(G)$. Fixemos um tal conjunto S e denotemos por G^+ o grafo induzido por S .

Primeiramente, suponhamos que a inequação $x(A(S)) \leq (|S| - 1)/2$ define uma faceta. Então, é imediato que G^+ é conexo. Vamos mostrar que G^+ é 2-conexo. Suponha que isso não ocorra, e que G^+ seja a união de dois grafos não triviais G_1 e G_2 , com exatamente um vértice em comum. Suponha que $G_i = (V_i, A_i)$ e que $|V_i| \geq 3$. Então somando-se para $i = 1, 2$ as inequações

$$x(A_i) \leq (|V_i| - 1)/2,$$

que são válidas para $P_{Emp}(G)$, obtemos a inequação $x(A(S)) \leq (|S| - 1)/2$. Isto é um absurdo, pois estamos supondo que esta última inequação define uma faceta. Caso $|V_1| = 2$ e/ou $|V_2| = 2$, é fácil ver que a inequação $x(A(S)) \leq (|S| - 1)/2$ também pode ser obtida como combinação linear de outras.

Para provar que G^+ é hipo-emparelhável, precisamos mostrar que para todo vértice $v \in G^+$ existe um emparelhamento perfeito em $G^+ - v$. Suponhamos que exista um vértice $v \in G^+$ tal que em $G^+ - v$ não existe um emparelhamento perfeito. Então a inequação $x(A(S - \{v\})) \leq (|S| - 3)/2$ é válida para $P_{Emp}(G)$. Somando essa inequação com a inequação $x(\delta(v)) \leq 1$, obtemos $x(A(S)) \leq (|S| - 1)/2$, um absurdo.

Para provar a implicação reversa, suponhamos que G^+ seja 2-conexo hipo-emparelhável. Claramente, a inequação $x(A(S)) \leq (|S| - 1)/2$ é válida para $P_{Emp}(G)$. Seja F_S a face definida por essa inequação.

Seja $a^\top x \leq \alpha$ uma inequação que define uma faceta de $P_{Emp}(G)$, e suponha que

$$F_S \subseteq F_a := \{x \in \mathbb{R}^A \mid a^\top x = \alpha\}.$$

Fato 1. Se $e \notin A(S)$ então $a_e = 0$.

De fato, como a aresta e tem no máximo um dos seus extremos em S , e G^+ é hipo-emparelhável, existe um emparelhamento E de G^+ tal que $F := E \cup (e)$ é um emparelhamento. Tanto χ^E quanto χ^F pertencem a F_S , e portanto, pertencem a F_a . Logo, $a^\top \chi^F = a^\top \chi^E = \alpha$, ou seja,

$$0 = a^\top \chi^F - a^\top \chi^E = a_e.$$

Fato 2. Se $e, f \in A(S)$, então $a_e = a_f$.

Vamos supor que e e f são adjacentes a um vértice comum v . Considere o grafo G^* obtido de G subdividindo v em dois vértices v_e e v_f , onde v_e é incidente a todas as arestas h que eram adjacentes a v e para as quais $a_h = a_e$; e v_f é incidente a todas as demais arestas originalmente incidentes a v . Pode-se provar (exercício), usando o Teorema de Tutte, que G^* tem um emparelhamento perfeito. Seja E^* um tal emparelhamento perfeito e sejam e' e f' as duas arestas de E^* incidentes a v_e e v_f , respectivamente. Claramente, $E - e'$ e $E - f'$ são emparelhamentos perfeitos em $G^+ - v$, e portanto, seus vetores de incidência pertencem a F_S , e portanto a F_a . Neste caso, $a_e = a_f$, pois

$$0 = a^\top (\chi^{E-e'}) - a^\top (\chi^{E-f'}) = a_e - a_f.$$

Tendo mostrado que quaisquer duas arestas e, f em $A(S)$ incidentes a um mesmo vértice são tais que $a_e = a_f$, segue que todas as arestas e em $A(S)$ têm o mesmo coeficiente $a_e := \gamma$. Logo, como $F_S \subseteq F_a$, segue que $x(A(S)) = \gamma x(A)$, e portanto, $(|S| - 1)/2 = \gamma\alpha$. Ou seja, $F_S = F_a$. Logo, F_S é uma faceta de $P_{Emp}(G)$, e a prova do teorema está completa. \square

A demonstração da segunda implicação mencionada na prova acima é basicamente a que consta em Lovász e Plummer [82].

Exercício 3.2.6 *Para provar que a inequação $x(A(S)) \leq (|S| - 1)/2$, $|S| \geq 3$ e ímpar, define uma faceta de $P_{Emp}(G)$ se o grafo induzido por S é 2-conexo hipo-emparelhável, exiba $|A|$ vetores afim-independentes que satisfazem essa inequação. Sugestão: usar o Teorema da decomposição auricular de grafos 2-conexos hipo-emparelháveis (veja Lovász e Plummer [82]).*

Exercício 3.2.7 *Prove que o grafo G^* mencionado na prova do Teorema 3.2.5 tem um emparelhamento perfeito (sob a hipótese de que o grafo G^+ é hipo-emparelhável).*

Voltando à discussão anterior à prova do Teorema 3.2.2, quando observamos que não é verdade que $P_{Emp}(G) = P_1(G)$ para todo grafo G , uma pergunta natural que surge é a seguinte: *Existem classes de grafo para as quais essa igualdade é sempre válida?*

A resposta é que os grafos bipartidos constituem uma tal classe. Mais precisamente, podemos afirmar que

$$P_{Emp}(G) = P_1(G) \iff G \text{ é bipartido.}$$

O lado mais interessante dessa afirmação é o seguinte teorema, cuja autoria é geralmente atribuída a Birkhoff [10] e von Neumann [87]. A prova que faremos não é a que foi apresentada por esses autores. Uma outra prova alternativa, muito interessante, pode ser encontrada em [39].

Teorema 3.2.8 *Seja $G = (V, A)$ um grafo bipartido. Então*

$$P_{Emp}(G) = P_1(G) := \{x \in \mathbb{R}^A \mid x_a \geq 0, \text{ para todo } a \in A, x(\delta(v)) \leq 1, \text{ para todo } v \in V\}.$$

Prova. Considere o sistema $Cx \leq d$ correspondente ao conjunto de inequações $x(\delta(v)) \leq 1$ (para todo $v \in V$) que ocorre na descrição de $P_1(G)$. Note que, C é precisamente a matriz de incidência de G . Como G é bipartido, C é totalmente unimodular (segue do Teorema 2.1.8), e portanto o poliedro $P_1(G)$ é inteiro (pelo Corolário 2.1.6). \square

Um resultado min-max interessante que segue deste teorema, usando dualidade em programação linear, é o conhecido Teorema de König:

Num grafo bipartido a cardinalidade de um emparelhamento máximo é igual ao número mínimo de vértices que cobre todas as arestas do grafo.

Um resultado min-max de mesma natureza pode ser provado com relação à cardinalidade de uma cobertura mínima de arestas e a cardinalidade de um conjunto estável máximo de um grafo bipartido (veja o próximo exercício).

Exercício 3.2.9 *Um subconjunto de arestas F de um grafo é uma **cobertura de arestas** se todo vértice do grafo é extremo de alguma aresta em F . Um conjunto **estável** é um conjunto de vértices, dois a dois não-adjacentes. Considere o poliedro definido como o fecho convexo dos vetores de incidência dos conjuntos estáveis de um grafo. Encontre uma descrição completa desse poliedro no caso em que o grafo é bipartido. Usando dualidade em programação linear, prove que: Se G é um grafo bipartido sem vértices isolados, então a cardinalidade de uma cobertura mínima de arestas é igual à cardinalidade de um conjunto estável máximo. (Veja Nemhauser e Trotter [86].)*

Exercício 3.2.10 *Prove que os vértices do poliedro $P_1(G)$, conhecido como o **poliedro fracionário dos emparelhamentos**, só tem componentes em $\{0, 1/2, 1\}$. (Veja Balinski [5].)*

O Teorema 3.2.2 é equivalente ao seguinte teorema a respeito do sistema de inequações que descreve o poliedro dos emparelhamentos perfeitos de um grafo $G = (V, A)$, com $|V|$ par. Esse poliedro é assim definido.

$$P_{Perf}(G) := \text{conv} \{ \chi^E \in \mathbb{R}^A \mid E \text{ é um emparelhamento perfeito em } G \}.$$

Teorema 3.2.11 *O poliedro $P_{Perf}(G)$ é descrito pelo seguinte sistema:*

- (1) $x_a \geq 0$, para todo $a \in A$;
- (2) $x(\delta(v)) = 1$, para todo $v \in V$;
- (3) $x(\delta(W)) \geq 1$, para todo $W \subseteq V$, W ímpar. □

A prova do teorema acima a partir do Teorema 3.2.2 é imediata. Já a prova do Teorema 3.2.2 a partir do teorema acima não é tão simples. Essa prova pode ser encontrada em [99, 97] Ela mostra como o problema do emparelhamento (geral) pode ser reduzido ao problema de encontrar um emparelhamento perfeito. Assim, o problema geral pode ser resolvido através do poliedro dos emparelhamentos perfeitos.

Em 1985, Grötschel e Holland [48] implementaram um algoritmo baseado em planos-de-corte faciais para o problema do emparelhamento perfeito e compararam os resultados computacionais dessa implementação com os da implementação de um algoritmo combinatório. É interessante notar que os tempos de processamento são competitivos para instâncias com pelo menos 300 vértices.

Considere o problema da separação relativo ao poliedro P_{Perf} :

Dado um vetor $y \in \mathbb{R}^A$, decidir se y satisfaz (1), (2) e (3). Se não for este o caso, encontrar uma inequação violada por y .

É fácil testar se y satisfaz (1) e (2). Vamos então assumir que y satisfaz (1) e (2) e vejamos como testar se y satisfaz (3). Para isso, considere $y = (y_e)_{e \in A}$ como sendo um vetor de custos associados às arestas de G . Assim, o problema consiste em testar se G tem um corte $\delta(W)$ com $|W|$ ímpar, cujo custo $y(\delta(W))$ é menor que 1.

Um corte $\delta(W)$ com W ímpar é chamado **corte ímpar**. O problema que queremos resolver então é o de encontrar um corte ímpar de custo mínimo num grafo com custos não-negativos associados às suas arestas. Em 1982, Padberg e Rao [90] desenvolveram um algoritmo polinomial que resolve este problema. Na verdade, esses autores desenvolveram um algoritmo polinomial para um problema mais geral, chamado problema do T -corte de capacidade mínima (veja abaixo). Sendo $T \subseteq V$, $|T|$ par, um **T -corte** é um corte $\delta(W)$ tal que $|T \cap W|$ é ímpar.

Dado um grafo $G = (V, A)$ com capacidades $c : A \rightarrow \mathbb{R}_+$, $T \subseteq V$, $|T|$ par, encontrar um T -corte de capacidade mínima.

Antes de finalizar esta seção, vamos mostrar que o problema da separação para os inequações *blossom* pode ser reduzido ao problema do corte de capacidade mínima. Para isso, vamos novamente listar as restrições que definem $P_{Emp}(G)$.

- (1) $x_e \geq 0$, para todo $e \in A$.

- (2) $x(\delta(v)) \leq 1$, para todo $v \in V$.
 (3) $x(A(S)) \leq (|S| - 1)/2$, para todo $S \subseteq V$, $|S| \geq 3$ e ímpar.

O problema da separação das inequações *blossom* que queremos resolver é o seguinte:

Dado x' satisfazendo (1) e (2), x' não inteiro, encontrar um subconjunto $S \subseteq V$, $|S|$ ímpar, tal que $x'(A(S)) > (|S| - 1)/2$, ou provar que tal S não existe.

Primeiramente, vamos reescrever (3) de outra forma. Seja $s = (s_v : v \in V)$ o vetor das variáveis de folga das restrições (2). Ou seja,

$$x(\delta(v)) + s_v = 1, \text{ para todo } v \in V.$$

Somando essas equações para todo $v \in S$, obtemos

$$2x(A(S)) + x(\delta(S)) + s(S) = |S|.$$

Obtendo $x(A(S))$ da equação acima e substituindo-a na restrição (3), temos que

$$x(\delta(S)) + s(S) \geq 1.$$

Assim, dado x' (do problema da separação), se definirmos $s' = (s'_v : v \in V)$, onde $s'_v := 1 - x'(\delta(v))$, temos que

$$x'(A(S)) > (|S| - 1)/2 \iff x'(\delta(S)) + s'(S) < 1.$$

Para verificarmos a desigualdade direita acima, procedemos da seguinte forma. Construimos, a partir do grafo original G , um grafo $G^* = (V^*, A^*)$, com $V^* = V \cup \{w^*\}$ e $A^* = A \cup \{vw^* \mid v \in V\}$. Definimos em G^* capacidades x'_e para as arestas $e \in A$, e capacidades s'_v para cada aresta $vw^* \in A^* \setminus A$. Dessa forma, basta agora encontrar em G^* um corte ímpar de capacidade mínima. Se tal corte em G^* , digamos $\delta(W)$, onde $w^* \notin W$, tem capacidade menor do que 1, então W é um subconjunto de V tal que $x'(A(W)) > (|W| - 1)/2$. Caso contrário, x' satisfaz todas as inequações *blossom*.

3.3 Poliedro dos Caminhos e Poliedro dos Cortes

O Problema do Caminho Mínimo é o seguinte:

Dado um grafo orientado $D = (V, A)$, dois vértices distintos s, t em V , uma função custo $c : A \rightarrow \mathbb{R}$ definida nos arcos de D , encontrar em D um caminho de s a t que seja de custo mínimo.

Considere o seguinte politopo, associado ao problema acima.

$$P_{Cam}(D) := \text{conv}\{\chi^P \in \mathbb{R}^A \mid P \text{ é um } (s, t)\text{-caminho em } D\}.$$

É interessante notar que não se conhece uma descrição linear de $P_{Cam}(D)$. Ademais, é pouco provável (de acordo com Karp e Papadimitriou [72] que uma tal descrição completa e tratável exista, pois o problema do caminho mínimo (com custos quaisquer) é \mathcal{NP} -difícil. No caso especial em que os custos são não-negativos (ou mais geralmente, se D não contém nenhum circuito de custo negativo), sabe-se que o problema pode ser resolvido em tempo polinomial (veja [95]). Para estudar este caso, vamos considerar o seguinte politopo.

$$P_{Cam+}(D) := \text{conv}\{\chi^P \in \mathbb{R}^A \mid P \text{ contém um } (s, t)\text{-caminho em } D\}.$$

Note que se os custos são não-negativos, então o problema do caminho mínimo é equivalente ao problema $\min\{c^\top x \mid x \in P_{Cam+}(D)\}$. No próximo teorema damos uma descrição desse poliedro. Deixamos a cargo do leitor provar esse resultado usando técnica análoga à apresentada na descrição do poliedro dos emparelhamentos.

Teorema 3.3.1 *Seja $D = (V, A)$ um grafo orientado e s, t vértices distintos de V . Então o politopo $P_{Cam+}(D)$ é descrito pelo seguinte sistema de inequações.*

$$\begin{aligned} 0 &\leq x_a \leq 1, \text{ para todo } a \in A; \\ x(\delta^+(W)) &\geq 1, \text{ para todo } W \subseteq V, \text{ com } s \in W \text{ e } t \notin W. \end{aligned}$$

□

Um outro poliedro relacionado com $P_{Cam+}(D)$ é o poliedro $P_{Cam}(D) + \mathbb{R}_+^A$, chamado o **dominante** de $P_{Cam}(D)$. Note que, para resolver o problema do caminho de custo mínimo, no caso em que os custos são não-negativos, pode-se considerar esse poliedro. Ao leitor interessado em resultados a respeito de dominantes de poliedros, e problemas em aberto relativos a dominantes de poliedros associados a problemas de caminhos e circuitos com restrição de paridade, sugerimos consultar [53].

Lembramos que, num grafo orientado $D = (V, A)$, um conjunto da forma $\delta^+(W)$ é chamado um **corte**. No caso particular em que $s \in W$ e $t \notin W$, é chamado um (s, t) -**corte**.

Note que, no problema de minimização sobre o politopo $P_{Cam+}(D)$, o problema da separação para a classe das inequações definidas pelos (s, t) -cortes (supondo que as restrições de não-negatividade estejam satisfeitas), pode ser reduzido ao seguinte problema, conhecido como o *problema do (s, t) -corte de capacidade mínima*:

Dado um grafo orientado $D = (V, A)$, dois vértices distintos s, t em V , uma função capacidade $c : A \rightarrow \mathbb{R}_+$, encontrar em D um (s, t) -corte de capacidade mínima.

O problema acima é por si só um problema interessante, mas tem também diversas aplicações, ocorrendo como subproblema de outros. Sabemos que este problema pode ser resolvido em tempo polinomial, usando o conhecido algoritmo de Dijkstra (veja [95]).

Dessa forma, podemos concluir (usando o fato de que problemas de otimização e separação são equivalentes), que o problema do caminho mínimo, no caso de custos não-negativos, pode ser resolvido em tempo polinomial. Note que estamos nos referindo aqui à existência de um algoritmo polinomial baseado nesses resultados poliédricos (e não aos algoritmos combinatórios que sabemos existir).

Vamos considerar agora, o seguinte poliedro associado ao problema do (s, t) -corte mínimo.

$$P_{Cor+}(D) := \text{conv}\{\chi^F \in \mathbb{R}^A \mid F \text{ contém um } (s, t)\text{-corte em } D\}.$$

Deixamos a cargo do leitor fazer uma prova direta do seguinte resultado a respeito da descrição desse politopo.

Teorema 3.3.2 *Seja $D = (V, A)$ um grafo orientado e s, t vértices distintos de V . Então o politopo $P_{Cor+}(D)$ é descrito pelo seguinte sistema de inequações:*

$$\begin{aligned} 0 &\leq x_a \leq 1, \text{ para todo } a \in A; \\ x(P) &\geq 1, \text{ para todo } (s, t)\text{-caminho } P \text{ em } D. \end{aligned}$$

□

Em vista do que foi dito acima, usando os mesmos argumentos, podemos concluir que o problema de minimização sobre o poliedro P_{Cor+} pode ser resolvido em tempo polinomial. Com isso, segue que o problema do (s, t) -corte mínimo, com capacidades não-negativas, também pode ser resolvido em tempo polinomial.

Vamos agora mencionar outros problemas de fluxo, relacionados com os problemas acima considerados.

Um dos problemas mais fundamentais na teoria de fluxos em redes é o seguinte problema, conhecido como o *Problema do Fluxo Máximo*.

Dado um grafo orientado $D = (V, A)$, dois vértices distintos s, t em V , uma função capacidade $c : A \rightarrow \mathbb{R}_+$, encontrar em D um (s, t) -fluxo de valor máximo.

Não definimos o conceito de fluxo, e para fazê-lo, vamos escrever a formulação matemática deste problema:

$$\begin{aligned} \max \quad & x(\delta^+(s)) - x(\delta^-(s)) \\ (D) \quad & x(\delta^+(v)) = x(\delta^-(v)), \text{ para todo } v \in V \setminus \{s, t\}, \\ & 0 \leq x_a \leq c_a, \text{ para todo } a \in A \end{aligned}$$

Um vetor $x \in \mathbb{R}^A$ que satisfaz as restrições acima é chamado um (s, t) -**fluxo**, ou simplesmente um **fluxo**. O seu **valor** é $x(\delta^+(s)) - x(\delta^-(s))$.

Em 1957 Ford e Fulkerson [34, 35] desenvolveram um algoritmo polinomial para este problema. Sabe-se que se as capacidades c_a são inteiras, então esse algoritmo encontra um fluxo máximo inteiro. Mais precisamente, se adicionarmos a restrição de integralidade

$$c_a \in \mathbb{Z} \text{ para todo } a \in A,$$

O algoritmo de Ford e Fulkerson resolve o problema de programação linear inteira que resulta. O mais interessante é que, mesmo sem adicionarmos tais restrições, ao resolver o PL acima, com capacidades inteiras, obtemos uma solução ótima inteira. Ou seja, as restrições de integralidade são supérfluas. De fato, Ford e Fulkerson provaram que o poliedro definido pelo sistema acima é inteiro, quando as capacidades são inteiras. Este resultado pode ser provado usando o Teorema de Hoffman e Kruskal, juntamente com o fato de que a matriz de restrições é totalmente unimodular (veja Capítulo 2).

Considerando-se o dual do PL acima, obtemos um PL, cuja interpretação é precisamente o problema do (s, t) -corte de capacidade mínima. Não vamos mostrar isso aqui, mas sugerimos que o leitor consulte [53] ou [79]. Assim, usando o teorema da dualidade em programação linear, obtemos o seguinte resultado min-max.

Teorema 3.3.3 (Teorema do fluxo máximo–corte mínimo) *Seja $D = (V, A)$ um grafo orientado, s, t dois vértices distintos em V e $c : A \rightarrow \mathbb{R}_+$ uma função capacidade. Então, o valor máximo de um (s, t) -fluxo é igual à capacidade mínima de um (s, t) -corte. \square*

Vale observar que, os algoritmos que encontram um fluxo máximo, fornecem, como subproduto, um corte de capacidade mínima. Dessa forma, temos algoritmos polinomiais para esse último problema. Analogamente ao problema do fluxo máximo, no caso do PL para o problema do corte, o poliedro correspondente é inteiro. Portanto, aqui também, as restrições de integralidade podem ser omitidas.

Consideremos agora, um caso especial do problema do fluxo máximo, onde $c_a \in \mathbb{Z}$, para todo arco $a \in A$. Neste caso, se x é um (s, t) -fluxo inteiro de valor k , então existem (s, t) -caminhos P_1, P_2, \dots, P_k em D , tais que, para cada arco a , o número desses caminhos aos quais esse arco pertence é no máximo c_a (deixamos essa prova a cargo do leitor). Com isso, podemos concluir o seguinte resultado.

Teorema 3.3.4 *Seja $D = (V, A)$ um grafo orientado, s, t dois vértices distintos de D , e $c : A \rightarrow \mathbb{Z}_+$ uma função capacidade. Então, a capacidade mínima de um (s, t) -corte é igual ao número máximo de (s, t) -caminhos, tais que, cada arco a está contido em no máximo c_a desses caminhos. \square*

A partir do teorema acima podemos concluir que o sistema

$$\begin{aligned} x_a &\geq 0, \text{ para todo } a \in A; \\ x(P) &\geq 1, \text{ para todo } (s, t)\text{-caminho } P \text{ em } D. \end{aligned}$$

é TDI, e portanto, o poliedro descrito por essas inequações é inteiro. O sistema acima descreve precisamente o dominante do poliedro $P_{Cor}(D)$, definido como $P_{Cor}(D) + \mathbb{R}_+^A$.

No caso especial em que as capacidades são todas iguais a 1, o teorema acima nos fornece um outro resultado min-max clássico.

Teorema 3.3.5 (Teorema de Menger) *Seja $D = (V, A)$ um grafo orientado, s, t dois vértices distintos de D . Então, a mínima cardinalidade de um (s, t) -corte é igual ao número máximo de (s, t) -caminhos dois a dois disjuntos nos arcos.* \square

Tanto o problema do caminho mínimo quanto o problema do corte mínimo têm as suas versões correspondentes para o caso de grafos (não orientados). Variantes desses problemas, sem fixar pares de vértices, bem como o caso de cortes máximos, também foram amplamente estudados. Considerações a respeito de todos esses problemas, com uma farta indicação de referências podem ser encontradas em [53].

O tipo de relação que existe entre os problemas do (s, t) -caminho mínimo e o problema do (s, t) -corte mínimo também ocorre com outros pares de problemas. Os poliedros $P_{Cam}(D) + \mathbb{R}_+^A$ e $P_{Cor}(D) + \mathbb{R}_+^A$ formam o que se chama um “par de poliedros bloqueadores”. Trata-se de uma relação de polaridade, introduzida por Fulkerson [37] na década de 70, quando desenvolveu a teoria de poliedros bloqueadores e anti-bloqueadores. As relações bloqueadoras e anti-bloqueadoras têm aplicações interessantes em combinatória poliédrica, permitindo derivar certas relações min-max a partir de outras. Sugerimos que o leitor interessado nesse tópico consulte [99].

3.4 Poliedro de um Matróide

O conceito de matróide foi introduzido por Whitney [104] e Van der Waerden [103] como abstração combinatória do conceito de dependência linear em espaços vetoriais e de certas propriedades de grafos. Matróides ocupam um lugar de destaque em otimização combinatória pelo seu papel unificador, que permite enquadrar sob a mesma ótica vários problemas dessa área.

Um **matróide** é um par (E, \mathcal{I}) , onde E é um conjunto finito e \mathcal{I} é uma coleção de subconjuntos de E satisfazendo as seguintes propriedades:

- (i) $\emptyset \in \mathcal{I}$;

- (ii) se $A \in \mathcal{I}$ e $B \subseteq A$, então $B \in \mathcal{I}$;
- (iii) se $A, B \in \mathcal{I}$ e $|A| > |B|$, então existe $e \in A \setminus B$ tal que $B \cup \{e\} \in \mathcal{I}$.

Os elementos em \mathcal{I} são chamados **conjuntos independentes** do matróide. Pela propriedade (iii), para todo $A \subseteq E$, os conjuntos independentes maximais de A têm a mesma cardinalidade. Os conjuntos independentes maximais de A são chamados **bases** de A , e a sua cardinalidade é o **posto** de A , denotado por $r(A)$. Note que, $A \in \mathcal{I}$ se e só se $r(A) = |A|$; assim, um matróide fica univocamente determinado pela sua função posto (definido para todo subconjunto de E). Vejamos dois exemplos de matróides.

Matróides Gráficos. Considere E o conjunto das arestas de um grafo G e \mathcal{I} a coleção de todas as florestas de G .

Matróides Lineares. Considere E um conjunto finito de vetores em um espaço linear, e \mathcal{I} a coleção de subconjuntos de E que são linearmente independentes.

Considere o seguinte problema de otimização:

Dado um matróide $M = (E, \mathcal{I})$ e uma função custo $c : E \rightarrow \mathbb{R}_+$, encontrar um conjunto independente de custo máximo.

Um resultado básico em otimização combinatória é o fato de que esse problema pode ser resolvido em tempo polinomial pelo seguinte método, conhecido como algoritmo *greedy*:

- (a) Faça $S := \{e_1\}$ onde e_1 é um elemento em E tal que $\{e\} \in \mathcal{I}$ e $c(e_1)$ é maximal;
- (b) Supondo que $S := \{e_1, e_2, \dots, e_k\}$ foi escolhido, escolha e_{k+1} em $E \setminus S$ tal que $S \cup \{e_{k+1}\} \in \mathcal{I}$ e $c(e_{k+1})$ seja maximal.
- (c) Repita o passo (b) até que não exista nenhum tal elemento e_{k+1} .

Deixamos a cargo do leitor verificar que o método acima realmente determina um conjunto independente de custo máximo (veja [27]). Mais interessante do que essa afirmação, é o fato de que os matróides são precisamente as estruturas (E, \mathcal{I}) que satisfazem as propriedades (i) e (ii) para as quais o algoritmo *greedy* encontra um conjunto independente de custo máximo em \mathcal{I} para qualquer função custo $c : E \rightarrow \mathbb{R}_+$.

Em vista do que foi dito, o problema acima pode ser considerado bem resolvido, e de certa forma, parece dispensar outras considerações. Vejamos, no entanto, o que podemos obter estudando esse problema sob o enfoque poliédrico.

Seja $M = (E, \mathcal{I})$ um matróide, e seja $P_{Mat}(M)$ o poliedro associado a M , definido como

$$P_{Mat}(M) := \text{conv}\{\chi^A \in \mathbb{R}^E \mid A \in \mathcal{I}\}.$$

O seguinte teorema foi provado em 1970 por Edmonds [26]. A prova desse teorema é devida a Grötschel, e segue basicamente as idéias apresentadas na prova do Teorema 3.2.2 que observamos ser devida a Lovász [81].

Teorema 3.4.1 *Seja $M = (E, \mathcal{I})$ um matróide. Então*

$$P_{Mat}(M) = \{x \in \mathbb{R}^E \mid x(A) \leq r(A) \text{ para todo } A \subseteq E, \ x_e \geq 0 \text{ para todo } e \in E \}.$$

Prova. Podemos assumir que $\{e\} \in \mathcal{I}$, para todo $e \in E$. Logo, todos os vetores unitários pertencem a P_{Mat} . Esses vetores juntamente com o vetor zero formam um conjunto de $|E| + 1$ vetores afim-independentes. Portanto, P_{Mat} tem dimensão plena.

Seja $a^\top x \leq \alpha$ uma inequação que define uma faceta de P_{Mat} , e

$$F_a := \{x \in \mathbb{R}^E \mid a^\top x = \alpha\}.$$

Vamos chamar de F_a -independente um conjunto em \mathcal{I} cujo vetor de incidência pertence a F_a ,

Caso 1. Suponha que exista $e \in E$ tal que $a_e < 0$.

Neste caso, todo conjunto F_a -independente satisfaz $x_e = 0$. De fato, se existisse um conjunto F_a -independente S tal que $e \in S$, então, $S' := S \setminus \{e\}$ seria um conjunto em \mathcal{I} tal que $a^\top \chi^{S'} > \alpha$, contrariando a validade de $a^\top x \leq \alpha$.

Caso 2. Suponha que $a_e \geq 0$ para todo $e \in E$.

Seja $A := \{e \in E \mid a_e > 0\}$. Então, para todo conjunto F_a -independente S temos que $|S \cap A| = r(A)$. De fato, se $|S \cap A| < r(A)$, então pela propriedade (ii) segue que existe $e \in A \setminus S$ tal que $S' := S \cup \{e\} \in \mathcal{I}$. Neste caso, como $a_e > 0$, temos que $a^\top \chi^{S'} > \alpha$, uma contradição. Portanto, todo conjunto F_a -independente satisfaz a equação $x(A) = r(A)$, ou seja, $F_a \subseteq \{x \in \mathbb{R}^E \mid x(A) = r(A)\}$.

Pelo que concluímos nos casos 1 e 2, temos que toda faceta de P_{Mat} é de um dos tipos mencionados na descrição desse poliedro. Logo, essa descrição é completa. \square

Veremos a seguir, quais são os conjuntos A para os quais a inequação $x(A) \leq r(A)$ induz uma faceta de $P_{Mat}(M)$. Para isso, vejamos duas definições.

Um conjunto $A \subseteq E$ é dito **fechado** se $r(A \cup \{e\}) > r(A)$ para todo $e \in E \setminus A$, ou seja, A é maximal para o seu posto. Dizemos que A é **não-separável** se, sempre que $S \subseteq A$ satisfaz $r(S) + r(A \setminus S) = r(A)$, temos que $S = \emptyset$ ou $S = A$.

Teorema 3.4.2 *Seja $M = (E, \mathcal{I})$ um matróide. Então, a inequação $x(A) \leq r(A)$, $A \subseteq E$, define uma faceta de $P_{Mat}(M)$ se e só se A é fechado e não-separável.*

Prova. Considere $A \subseteq E$. Se A não é fechado, então existe $e \in E \setminus A$ tal que $r(A \cup \{e\}) = r(A)$. Neste caso,

$$x(A) + x_e = x(A \cup \{e\}) \leq r(A \cup \{e\}) = r(A).$$

Ou seja, a inequação $x(A) \leq r(A)$ pode ser obtida como combinação linear de inequações válidas para $P_{Mat}(M)$, e portanto, não define uma faceta.

Suponha agora que A tenha um subconjunto próprio S , não-vazio, tal que $r(S) + r(A \setminus S) = r(A)$. Então,

$$x(A) = x(S) + x(A \setminus S) \leq r(S) + r(A \setminus S) = r(A),$$

e portanto, $x(A) \leq r(A)$ não define uma faceta de $P_{Mat}(M)$.

Com isso, provamos que se $x(A) \leq r(A)$ define uma faceta, então A é fechado e não-separável. Vejamos agora a recíproca dessa afirmação.

Suponha que A seja fechado e não-separável. Seja $a^\top x \leq \alpha$ uma inequação que define uma faceta de $P_{Mat}(M)$, e tal que

$$F_A := \{x \in \mathbb{R}^E \mid x(A) = r(A)\} \subseteq \{x \in \mathbb{R}^E \mid a^\top x = \alpha\} =: F_a.$$

Vamos mostrar que

- (a) $a_e = a_f$ para todo $e, f \in A$;
- (b) $a_e = 0$ para todo $e \in E \setminus A$.

Suponha que (a) não ocorra e considere os conjuntos

$$A_1 := \{e \in A \mid a_e \text{ tem o menor valor sobre todos os elementos de } A\} \text{ e}$$

$$A_2 := A \setminus A_1.$$

Seja I_2 uma base de A_2 e I uma base de A que é uma extensão de I_2 . Seja $I_1 := I \setminus I_2$. Como A é não-separável, então $|I_1| < r(A_1)$. Logo, existe $e \in A_1 \setminus I_1$ tal que $I_1 \cup \{e\} \in \mathcal{I}$. Portanto, existe $f \in I_2$ tal que $J := I \cup \{e\} \setminus \{f\} \in \mathcal{I}$. Neste caso, χ^I e χ^J pertencem a $F_A \subseteq F_a$, e portanto, $a^\top \chi^I = a^\top \chi^J$. Por outro lado, como $a_e < a_f$, temos que $a^\top \chi^I > a^\top \chi^J$, uma contradição.

Para provar (b), considere $e \in E \setminus A$ e uma base I de A . Como $\chi^I \in F_A$, temos que $a^\top \chi^I = \alpha$. Por outro lado, como A é fechado, $J := I \cup \{e\} \in \mathcal{I}$ e portanto $a^\top \chi^J = \alpha$, donde segue que $a_e = 0$.

Tendo provado (a) e (b), podemos concluir que F_A é uma faceta, o que completa a prova do teorema. \square

Aplicando o resultado anterior para matróides gráficos, temos o seguinte resultado.

Teorema 3.4.3 *Seja $G = (V, A)$ um grafo e $P_{Flo}(G)$ o poliedro das florestas de G , ou seja,*

$$P_{Flo}(G) := \text{conv}\{\chi^F \mid F \text{ é uma floresta de } G\}.$$

Então, $P_{Flo}(G)$ é descrito pelo seguinte sistema:

- (a) $x_e \geq 0$, para todo $e \in A$,
- (b) $x(A(S)) \leq |S| - 1$, para todo $S \subseteq V$ tal que $G[S]$ é não-separável.

□

Note que obtivemos uma caracterização completa do poliedro $P_{Mat}(M)$, sobre o qual sabemos que o problema $\max\{c^\top x \mid x \in P_{Mat}(M)\}$ pode ser resolvido em tempo polinomial.

Lembrando da equivalência entre problemas de otimização e separação (veja Capítulo 2), podemos concluir que o problema da separação relativo às inequações $x(A) \leq r(A)$, para todo $A \subseteq E$ pode ser resolvido em tempo polinomial. A existência de um tal algoritmo (via a equivalência acima citada), é um fato mais de interesse teórico do que prático. Um algoritmo combinatório que resolve esse problema, baseado em técnicas de caminhos aumentadores, foi obtido em 1984 por Cunningham [20].

No caso particular do matróide gráfico, isto é, do poliedro das florestas de um grafo G , um algoritmo combinatório para o problema da separação correspondente, foi obtido em 1984 por Padberg e Wolsey [93].

Vamos a seguir, considerar *intersecções* de matróides. Com isso, teremos uma linguagem padrão na qual vários problemas interessantes de otimização combinatória podem ser formulados.

Sejam $M_1 = (E, \mathcal{I}_1)$ e $M_2 = (E, \mathcal{I}_2)$ dois matróides definidos sobre o mesmo conjunto E . A **intersecção** desses dois matróides é o par $M_1 \cap M_2 = (E, \mathcal{I}_1 \cap \mathcal{I}_2)$. Em geral, essa intersecção não é um matróide, mas duas propriedades interessantes continuam valendo. Primeiro, o fato de que o problema de encontrar um conjunto independente de custo máximo em $\mathcal{I}_1 \cap \mathcal{I}_2$ é polinomial; segundo, o fato de que o poliedro associado a $M_1 \cap M_2$ tem uma descrição linear completa. O poliedro a que estamos nos referindo é o seguinte:

$$P_{Int}(M_1, M_2) := \text{conv}\{\chi^I \mid I \in \mathcal{I}_1 \cap \mathcal{I}_2\}.$$

O teorema enunciado abaixo foi provado por Edmonds [26]. O leitor pode consultar [53], para uma prova desse resultado.

Teorema 3.4.4 *Sejam M_1 e M_2 dois matróides definidos sobre um mesmo conjunto E , e sejam r_1 e r_2 as respectivas funções posto. Então*

$$P_{Int}(M_1, M_2) = P_{Mat}(M_1) \cap P_{Mat}(M_2),$$

ou seja, $P_{Int}(M_1, M_2)$ é descrito pelo seguinte sistema:

- (a) $x_e \geq 0$, para todo $e \in E$,
- (b) $x(A) \leq \min\{r_1(A), r_2(A)\}$, para todo $A \subseteq E$.

□

A seguinte fórmula interessante pode ser deduzida do teorema acima.

Teorema 3.4.5 *Sejam M_1 e M_2 dois matróides definidos sobre um mesmo conjunto E , e sejam r_1 e r_2 as respectivas funções posto. Então*

$$\max\{|I| \mid I \in \mathcal{I}_1 \cap \mathcal{I}_2\} = \min\{r_1(A) + r_2(E \setminus A) \mid A \subseteq E\}.$$

□

Como já mencionamos, alguns problemas combinatórios podem ser formulados como problemas de intersecção de dois matróides. É o caso, por exemplo, das r -arborescências de um grafo orientado, que pode ser expresso como a intersecção de um matróide gráfico e um matróide de partição (o leitor pode encontrar as definições desses conceitos em [53]). Só para encerrar, mencionamos que, fazendo uso desse fato e dos teoremas acima, podemos obter uma descrição completa do poliedro definido como o fecho convexo dos vetores de incidência das r -arborescências de um grafo orientado. Ademais, sabemos que é possível otimizar sobre esse poliedro em tempo polinomial, e que é conhecido um algoritmo polinomial combinatório para o problema da separação relativo a esse poliedro.

Lembramos também que, algoritmos polinomiais para achar um conjunto independente de custo máximo, comum a dois matróides, foram obtidos por Edmonds [28], Frank [36] e Lawler [78].

3.5 Poliedro do Problema do Caixeiro Viajante

O *Problema do Caixeiro Viajante* (PCV) pode ser definido da seguinte forma:

Dado um grafo $G = (V, A)$ e uma função custo $c : A \rightarrow \mathbb{R}$, encontrar um circuito hamiltoniano de custo mínimo.

O problema acima é geralmente referido como o *PCV simétrico*, O problema correspondente, para o caso em que G é um grafo orientado, é chamado *PCV assimétrico*. Nesta seção vamos só considerar o caso simétrico, que será simplesmente referido como PCV. Sabe-se que ambos os problemas são \mathcal{NP} -difíceis.

O primeiro uso do termo “Problema do Caixeiro Viajante” em círculos matemáticos parece ter ocorrido no início da década de 30, embora não seja muito claro quem foi que realmente introduziu o nome [80].

Apesar de sua aparente simplicidade, pelo menos para ser definido, o PCV tem resistido a todos os esforços no sentido de se encontrar “bons” algoritmos exatos ou mesmo algoritmos de aproximação com boas garantias de desempenho. Talvez exatamente por essa característica (fácil de ser formulado, mas difícil de ser resolvido), este problema tem merecido bastante atenção por parte dos pesquisadores, o que explica a vasta literatura a seu respeito. Há também outras razões, de ordem mais prática, que justificam tamanho interesse por esse problema, e também a sua importância. O fato é que muitos problemas reais relevantes podem ser formulados como instâncias do PCV: problemas de roteamento de veículos, problemas de seqüenciamento de tarefas, problemas de cortes de papel, projetos de *computer wiring*, etc.

Várias heurísticas têm sido propostas para o PCV, conforme pode ser comprovado no livro “The Traveling Salesman Problem: a Guided Tour of Combinatorial Optimization” [80], editado em 1985. Dentre essas, a abordagem poliédrica, que pode ser usada no tratamento de uma classe bem grande de problemas de otimização combinatória, tem sido usada com sucesso na resolução de instâncias reais do PCV, gerando soluções exatas para instâncias relativamente grandes. É

interessante observar que o primeiro problema onde esta abordagem foi utilizada foi exatamente o PCV [22].

Para estudar o PCV sob o ponto de vista poliédrico, vamos considerar o caso em que o grafo G é completo, e associar a este problema um poliedro (monótono) definido como o fecho convexo dos vetores de incidência dos conjuntos que são estendíveis a soluções viáveis do problema. Mais precisamente, denotando por $K_n = (V_n, A_n)$ o grafo completo com n vértices, e por \mathcal{T}^n o conjunto dos subconjuntos de arcos que ou são um circuito hamiltoniano ou são estendíveis a um circuito hamiltoniano em K_n , ou seja,

$$\mathcal{T}^n = \{T \subseteq A_n : \text{existe um circuito hamiltoniano } C \text{ em } K_n \text{ com } T \subseteq C\},$$

definimos tal politopo como sendo

$$P_{Ham}^n := \text{conv}\{\chi^T \in \mathbb{R}^{A_n} : T \in \mathcal{T}^n\}.$$

A razão de se considerar o politopo acima, em vez do politopo definido como o fecho convexo dos vetores de incidência dos circuitos hamiltonianos de G , é que dessa forma, obtemos um politopo de dimensão plena, tecnicamente, mais fácil de ser tratado (devido à unicidade da descrição de suas facetas). Note que não há perda de generalidade em supor que G é completo.

Lema 3.5.1 $\dim(P_{Ham}^n) = |A_n| = n(n-1)/2$, para $n \geq 3$.

Prova. Segue imediatamente do fato de que o vetor zero e os vetores unitários, que são afim-independentes, pertencem a P_{Ham}^n . \square

Teorema 3.5.2 Para toda aresta e em K_n , $n \geq 3$, a inequação $x_e \geq 0$ define uma faceta de P_{Ham}^n .

Prova. Basta notar que, para toda aresta $e \in A_n$, os vetores de incidência do conjunto vazio e dos conjuntos $\{f\}$, onde $f \in A_n \setminus \{e\}$, são afim-independentes, pertencem a P_{Ham}^n , e satisfazem a inequação $x_e = 0$. \square

Teorema 3.5.3 Para toda aresta e em K_n , $n \geq 3$, a inequação $x_e \leq 1$ define uma faceta de P_{Ham}^n .

Prova. Considere os $|A_n|$ conjuntos $\{e\}$ e $\{e, f\}$, onde $f \in A_n \setminus \{e\}$. Claramente, os vetores de incidência desses conjuntos são linearmente independentes e satisfazem a inequação $x_e \leq 1$ com igualdade. Como a face definida por essa inequação é uma face própria de P_{Ham}^n , segue que ela é uma faceta. \square

A prova do seguinte resultado é bem simples e será deixada como exercício. Ela pode ser feita usando a técnica apresentada na prova do Teorema 3.5.5.

Teorema 3.5.4 *Para todo vértice v em K_n , $n \geq 4$, a inequação $x(\delta(v)) \leq 2$ define uma faceta de P_{Ham}^n .* \square

Note que se $S \subseteq V$ é um conjunto de vértices com $2 \leq |S| \leq n-1$, então o conjunto das arestas $A_n(S)$ (arestas de K_n com ambos os extremos em S) intersecta cada circuito hamiltoniano em no máximo $|S| - 1$ arestas. Logo, o sistema de inequações

$$x(A_n(S)) \leq |S| - 1, \text{ para todo } S \subseteq V, \quad 2 \leq |S| \leq n - 1,$$

é satisfeito por todos os vetores em P_{Ham}^n .

As inequações acima são conhecidas como **restrições de eliminação de subcircuitos** (*subtour elimination constraints*). Elas foram introduzidas por Dantzig, Fulkerson e Johnson [54]. A prova de que elas definem facetas (tanto do poliedro que estamos considerando, quanto do poliedro definido como o fecho convexo dos vetores de incidência dos circuitos hamiltonianos de K_n) foi apresentada por Grötschel e Padberg [57].

Teorema 3.5.5 *Para todo $n \geq 4$, a inequação*

$$x(A_n(S)) \leq |S| - 1, \text{ para todo } S \subseteq V, \quad 2 \leq |S| \leq n - 1$$

define uma faceta de P_{Ham}^n ,

Prova. Seja $S \subseteq V$, $3 \leq |S| \leq n-1$. Note que se $|S| = 2$, o resultado segue do Teorema 3.5.3. Seja $a^\top x \leq \alpha$ uma inequação que define uma faceta de P_{Ham}^n e suponha que

$$F_S := \{x \in \mathbb{R}^{A_n} \mid x(A_n(S)) = |S| - 1\} \subseteq F_a := \{x \in \mathbb{R}^{A_n} \mid a^\top x = \alpha\}.$$

Vamos mostrar que a inequação $a^\top x \leq \alpha$ é um múltiplo escalar da inequação $x(A_n(S)) \leq |S| - 1$.

Seja C um circuito hamiltoniano em $K_n[S]$. Para cada aresta e em C , defina $T_e := C - e$. Claramente, $\chi_e^T \in F_S \subseteq F_a$. Logo, $a_e := \gamma_C$ para cada aresta e em C . Como isso vale para todo circuito hamiltoniano em $K_n[S]$, segue imediatamente que,

$$a_e := \gamma \text{ para toda aresta } e \in A_n(S). \quad (1)$$

Para toda aresta $f \in A_n \setminus A_n(S)$, existe um caminho hamiltoniano T em $K_n[S]$ tal que o grafo gerado por $T \cup \{f\}$ não tem vértices de grau 3. Então, o vetor de incidência de tal conjunto está em $F_S \subseteq F_a$, e portanto,

$$a_f = 0 \text{ para toda aresta } f \in A_n \setminus A_n(S). \quad (2)$$

Como $F_S \subseteq F_a$, de (1) e (2) segue o resultado que queríamos provar. \square

Cabe aqui ressaltar que o problema da separação relativo às restrições de eliminação de subcircuitos pode ser resolvido em tempo polinomial (veja Crowder e Padberg [18]).

Diversas outras classes de facetas do poliedro P_{Ham}^n foram descobertas (veja [3] para uma resenha a respeito). Não se conhecem, porém, algoritmos polinomiais para os correspondentes problemas de separação. Considerações a respeito da qualidade das diversas classes de facetas de P_{Ham}^n , podem ser encontradas em Goemans [41].

Nesta seção focalizamos apenas o caso do PCV simétrico. Resultados análogos foram obtidos para o caso assimétrico. Para esse caso, algumas classes de inequações que definem facetas, um tanto intratáveis (definidas por grafos hipo-hamiltonianos) foram obtidas por Grötschel e Wakabayashi [59, 60]. A literatura sobre resultados poliédricos relativos ao PCV é bastante extensa, incluindo estudo de diâmetro e adjacência de vértices nos respectivos poliedros. Esses resultados, bem como os aspectos computacionais relativos à abordagem poliédrica são amplamente discutidos em Grötschel e Padberg [57, 89].

3.6 Poliedro do Problema de Steiner em Grafos

O *Problema de Steiner em Grafos* pode ser descrito da seguinte forma:

*Dado um grafo $G = (V, A)$, um conjunto de vértices $Z \subseteq V$ chamados **terminais** e uma função custo $c : A \rightarrow \mathbb{R}$, encontrar $S \subseteq A$ tal que $G[S]$ conecta todos os terminais e $c(S)$ é mínimo.*

Um conjunto S tal que $G[S]$ conecta todos os terminais é chamado uma **solução de Steiner**. O **Poliedro do Problema de Steiner em Grafos** é definido como

$$P_{Ste}(G, Z) := \text{conv}\{\chi^S \mid S \text{ é uma solução de Steiner}\}.$$

Vamos inicialmente determinar a dimensão desse politopo. Para isso, precisamos da seguinte definição. Uma aresta em G é chamada uma **ponte de Steiner** se é uma ponte que separa dois vértices terminais de G .

Lema 3.6.1 *Seja $G = (V, A)$ um grafo conexo, Z um conjunto de terminais e $B(G)$ o conjunto das pontes de Steiner de G . Então,*

$$\dim(P_{Ste}(G, Z)) = |A| - |B(G)|.$$

Prova. Se uma aresta $e \in B(G)$, então toda solução de Steiner S deve necessariamente incluir e . Assim, $P_{Ste}(G, Z) \subseteq \{x_e = 1\}$. Com isso, $\dim(P_{Ste}(G, Z)) \leq |A| - |B(G)|$.

Por outro lado, como G é conexo, $\chi^A \in P_{Ste}(G, Z)$ e $\chi^{A-e} \in P_{Ste}(G, Z)$ para toda aresta $e \notin B(G)$, o que mostra que $\dim(P_{Ste}(G, Z)) \geq |A| - |B(G)|$. \square

Caso o grafo G contenha pontes de Steiner, podemos dividir o problema em problemas menores sem pontes de Steiner. Assim, sem perda de generalidade, podemos supor que G é conexo e não tem pontes de Steiner (consequentemente o poliedro tem dimensão plena).

Uma formulação inteira para o problema de Steiner em grafos foi sugerida por Aneja [2]. Considere as variáveis x_e indexadas pelas arestas do grafo, com a seguinte interpretação: $x_e = 1$ se e somente se a aresta $e \in S$. Considere agora o seguinte poliedro:

$$P(G, Z) := \left\{ x \in \mathbb{R}^A \mid \begin{array}{l} x(\delta(W)) \geq 1, \quad \text{para todo } W \subseteq V, \emptyset \neq W \cap Z \neq Z \\ 0 \leq x_e \leq 1, \quad \text{para toda aresta } e \in E \end{array} \right\}.$$

É fácil ver que $P_{Ste}(G, Z) = P(G, Z)_I$. Esta formulação foi estudada por diversos autores no final dos anos oitenta ([55], [13], [30]). Outras formulações para o problema de Steiner em grafos podem ser encontradas em [40], [83], [8] e [84]. Nesta seção trabalhamos com a formulação de Aneja, que, ao nosso ver, é mais natural para o problema.

Antes de discutirmos a estrutura do poliedro $P_{Ste}(G, Z)$, vejamos uma definição. Chamamos de **corte de Steiner** um conjunto $\delta(W)$ onde $W \subseteq V$ satisfaz $\emptyset \neq W \cap Z \neq Z$. A correspondente inequação $x(\delta(W)) \geq 1$, é chamada de **inequação de corte de Steiner**.

É fácil ver que as inequações de corte de Steiner não são suficientes para descrever completamente o poliedro $P_{Ste}(G, Z)$. Na Figura 3.1, onde os terminais são representados por círculos cheios e os vértices de Steiner por círculos vazados, mostramos uma solução fracionária que respeita todas as inequações de corte de Steiner, e é vértice do poliedro $P(G, Z)$.

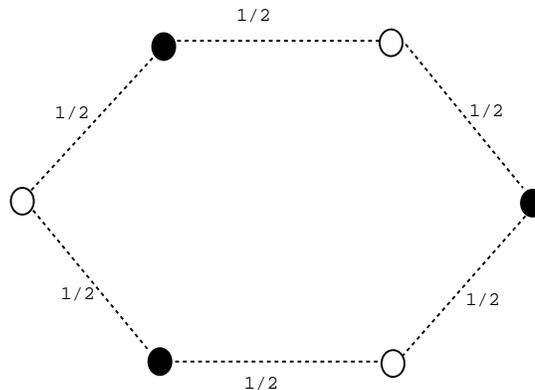


Figura 3.1: Vértice fracionário de $P(G, Z)$

De fato, seria pouco provável que apenas as inequações de corte de Steiner fornecessem uma descrição completa de $P_{Ste}(G, Z)$. Afinal, o problema é \mathcal{NP} -difícil (veja o Capítulo 2 para mais detalhes).

No lema a seguir, mostramos uma classe de facetas de $P_{Ste}(G, Z)$. Lembramos que estamos supondo que G é conexo e não tem pontes de Steiner.

Lema 3.6.2 *A inequação $x_e \geq 0$ define uma faceta de $P_{Ste}(G, Z)$ se e somente se $G - e$ não tem pontes de Steiner.*

Prova. Se $G - e$ não tem pontes de Steiner, então, para toda aresta $f \in E$, χ^{A-e} e $\chi^{A \setminus \{e,f\}}$ estão em $P_{Ste}(G, Z)$, satisfazem a inequação com igualdade e são afim-independentes. Logo, essa inequação define uma faceta de $P_{Ste}(G, Z)$.

Suponha agora que $f \in A$ é uma ponte de Steiner de $G - e$. Então, a inequação $x_e + x_f \geq 1$ é válida para todos os pontos de $P_{Ste}(G, Z)$. Então, $x_e \geq 0$ pode ser obtida somando-se a inequação acima e a inequação $x_f \leq 1$, e portanto, não define uma faceta de $P_{Ste}(G, Z)$. \square

Exercício 3.6.3 *Mostre que a inequação $x_e \leq 1$ define uma faceta de $P_{Ste}(G, Z)$ para todo $e \in A$.*

Exercício 3.6.4 *Mostre que para todo corte de Steiner $\delta(W)$, a inequação $x(\delta(W)) \geq 1$ define uma faceta de $P_{Ste}(G, Z)$.*

Exercício 3.6.5 *Mostre que o problema da separação para as inequações de corte de Steiner pode ser resolvido em tempo polinomial.*

Grötschel e Monma [55] obtiveram uma classe de facetas de $P_{Ste}(G, Z)$ que generaliza as facetas induzidas pelos cortes de Steiner, chamada partição de Steiner.

Uma partição $\mathcal{P} = (V_1, \dots, V_k)$ de V tal que $V_i \cap Z \neq \emptyset$ para $i = 1, \dots, k$ é chamada de **partição de Steiner**. Seja $\Delta(\mathcal{P}) := \{uv \in A \mid u \in V_i, v \in V_j, i \neq j\}$.

Teorema 3.6.6 *Seja G conexo e sem pontes de Steiner e tome uma partição de Steiner $\mathcal{P} = (V_1, \dots, V_k)$ de V . Então, a inequação*

$$x(\Delta(\mathcal{P})) \geq k - 1$$

define uma faceta de $P_{Ste}(G, Z)$ se e somente se

- (a) $G[V_i]$ é conexo;
- (b) $G[V_i]$ não contém pontes de Steiner com relação a $Z_i := Z \cap V_i$ e
- (c) \hat{G} obtido de G contraindo-se cada um dos conjuntos V_i a um vértice é 2-conexo.

Prova. Sem perda de generalidade, suponhamos V_1 desconexo. Seja (V'_1, V''_1) uma bipartição de V_1 tal que $V'_1 \cap Z \neq \emptyset$, $V''_1 \neq \emptyset$ e $[V'_1 : V''_1] = \emptyset$. Como G é conexo, podemos supor — sem perda de generalidade — que existe uma aresta ligando V''_1 a V_2 . Considere então a partição $\mathcal{P}' := (V'_1, V_2 \cup V''_1, V_3, \dots, V_k)$. A inequação $x(\Delta(\mathcal{P}')) \geq k - 1$ é válida. Somando essa inequação com a inequação válida $x([V''_1 : V_2]) \geq 0$ obtemos a inequação da partição de Steiner \mathcal{P} . Portanto, esta última não define uma faceta de $P_{Ste}(G, Z)$.

Com isso, podemos supor que $G[V_i]$ é conexo para todo $i = 1, \dots, k$. Suponha agora que $G[V_i]$ contém alguma ponte de Steiner e com respeito a $Z_i = Z \cap V_i$. Vamos mostrar que toda

solução de Steiner S que satisfaz a equação $x(\Delta(\mathcal{P})) = k - 1$ contém a aresta e . Suponha que isso não ocorra, e que exista uma solução de Steiner S com $\chi^S(\Delta(\mathcal{P})) = k - 1$ e $e \notin S$. Como e é ponte de Steiner de $G[V_i]$, a solução S deve conter algum caminho ligando dois vértices de Z em componentes diferentes de $G[V_i] - e$. Tome $S' := S \cup \{e\}$. O conjunto S' deve conter um circuito passando por uma aresta $f \in \Delta(\mathcal{P})$. Mas, então, $S'' := S \cup \{e\} \setminus \{f\}$ é solução de Steiner e $\chi^{S''}(\Delta(\mathcal{P})) = k - 2$, uma contradição.

Finalmente, suponhamos que \hat{G} não seja 2-conexo. Sem perda de generalidade, suponhamos que o vértice \hat{v}_1 de \hat{G} correspondente a V_1 é um vértice de corte de \hat{G} e os vértices correspondentes a V_2, \dots, V_c estão na mesma componente de $\hat{G} - \hat{v}_1$. Tome as partições $\mathcal{P}_1 := (V_1 \cup V_{c+1} \cup \dots \cup V_k, V_2, \dots, V_c)$ e $\mathcal{P}_2 := (V_1 \cup V_2 \cup \dots \cup V_c, V_{c+1}, \dots, V_k)$. Note que ambas são partições de Steiner, e além disso, é fácil observar que a inequação $x(\Delta(\mathcal{P})) \geq k - 1$ pode ser obtida somando as inequações $x(\Delta(\mathcal{P}_1)) \geq c - 1$ e $x(\Delta(\mathcal{P}_2)) \geq k - c$.

Resta mostrar que se \mathcal{P} é uma partição de Steiner que satisfaz (a), (b) e (c), então a inequação $x(\Delta(\mathcal{P}_1)) \geq k - 1$ define uma faceta de $P_{Ste}(G, Z)$. Para isso, tome uma inequação $a^\top x \leq \alpha$ que define uma faceta F de $P_{Ste}(G, Z)$ tal que $\{x \in P_{Ste}(G, Z) \mid x(\Delta(\mathcal{P})) = k - 1\} \subseteq F$. Vamos mostrar que $a^\top x \leq \alpha$ é um múltiplo escalar da inequação da partição de Steiner.

Inicialmente observe que toda árvore geradora \hat{T} de \hat{G} corresponde a uma floresta T de G tal que

$$\chi^{T \cup D}(\Delta(\mathcal{P})) = k - 1, \quad \text{para todo } D \subseteq \cup_{i=1}^k A(V_i) =: B.$$

Em particular, como $G[V_i]$ é conexo, $\chi^{T \cup B}$ é solução de Steiner e satisfaz a inequação com igualdade. Como nenhum $G[V_i]$ contém pontes de Steiner com relação a Z_i , segue que $\chi^{T \cup B - e}$ também é uma solução de Steiner para toda aresta $e \in G[V_i]$. Assim,

$$a^\top \chi^{T \cup B} = a^\top \chi^{T \cup B - e} = \alpha,$$

e portanto, $a_e = 0$ para todo $e \in G[V_i]$, $i = 1, \dots, k$.

Considere agora $e, f \in \Delta(\mathcal{P})$. Como \hat{G} é 2 conexo, existe em \hat{G} um circuito C que contém as arestas e e f . Tome uma árvore geradora T' de \hat{G} que contém $C - f$. Então, $T'' := T' - e + f$ é uma árvore geradora de \hat{G} . Assim,

$$a^\top \chi^{T' \cup B} = a^\top \chi^{T'' \cup B},$$

e portanto $a_e = a_f$. Repetindo o mesmo argumento para todo par de arestas em $\Delta(\mathcal{P})$, concluímos que $a^\top x \geq \alpha$ é um múltiplo escalar da inequação da partição de Steiner. Logo, esta define uma faceta de $P_{Ste}(G, Z)$. Isso completa a prova do teorema. \square

Em 1991, Grötschel, Monma e Stoer [56] mostraram que o problema da separação das inequações de partição de Steiner é \mathcal{NP} -difícil.

Outras inequações válidas e facetas do poliedro $P_{Ste}(G, Z)$ foram encontradas por Chopra e Rao [13]. Uma classe interessante é constituída pelas *odd hole inequalities*. Na Figura 3.2 mostramos um 3-*odd hole*. É fácil ver que se o grafo G tem um 3-*odd hole* H , como subgrafo

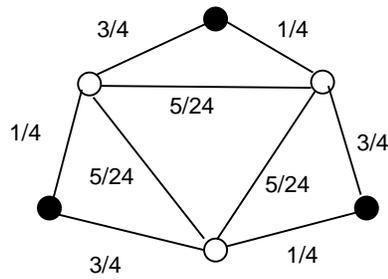


Figura 3.2: 3-odd hole

induzido, então uma solução de Steiner precisa conter no mínimo 4 arestas de H . Note que a solução indicada na figura tem custo $3 + \frac{5}{8}$, e satisfaz todas as inequações de partição de Steiner.

Vale ressaltar que, mesmo quando consideramos instâncias particulares do problema de Steiner, poucas descrições completas do poliedro $P_{Ste}(G, Z)$ são conhecidas. As inequações de corte de Steiner são suficientes para caracterizar o poliedro apenas no caso em que $|Z| = 2$. Conjectura-se que as partições de Steiner são suficientes para caracterizar o poliedro, caso $|Z| \geq |V| - 2$. Goemans [40] mostra uma descrição completa do poliedro para grafos série-paralelos usando uma formulação diferente.

Capítulo 4

Um Algoritmo Baseado em Cortes Faciais

Neste capítulo apresentamos uma implementação de um algoritmo *branch and cut*. O nosso objetivo é mostrar todos os passos da construção do algoritmo através de um exemplo: o problema de Steiner em grafos. O problema, já estudado no capítulo anterior, pode ser formulado da seguinte forma:

Dado um grafo $G = (V, A)$, um conjunto $Z \subseteq V$ e uma função $c : A \rightarrow \mathbb{R}_+$, achar um conjunto de arestas $S \subseteq A$, tal que o subgrafo $G[S]$ contém um caminho entre cada par de vértices de Z e $c(S)$ é a menor possível.

Para simplificar, este problema será chamado simplesmente de **problema de Steiner**. Chamamos as soluções viáveis deste problema, ou seja, os conjuntos $S \subseteq A$ tais que $G[S]$ contém um caminho entre cada par de vértices de Z uma **solução de Steiner**.

Na Seção 3.6 mostramos a formulação de Aneja para este problema, assim como algumas inequações válidas e facetas do poliedro $P_{Ste}(G, Z)$ correspondente. Defina as variáveis $x \in \{0, 1\}^A$ com a seguinte interpretação:

$$x_e = \begin{cases} 1, & \text{se a aresta } e \in S; \\ 0, & \text{caso contrário.} \end{cases}$$

Assim, o poliedro $P_{Ste}(G, Z)$ pode ser definido como

$$P_{Ste}(G, Z) := \text{conv}\{\chi^S \in \mathbb{R}^A \mid S \subseteq A \text{ uma solução de Steiner}\}.$$

Com as variáveis acima definidas, o problema de Steiner tem a seguinte formulação inteira:

$$(P) \quad \begin{array}{ll} \min & \sum_{e \in A} c_e x_e \\ & (1) \quad x(\delta(W)) \geq 1 \quad \text{para todo } W \subseteq V, \emptyset \neq W \cap Z \neq Z; \\ & (2) \quad x_e \in \{0, 1\} \quad \text{para todo } e \in A. \end{array}$$

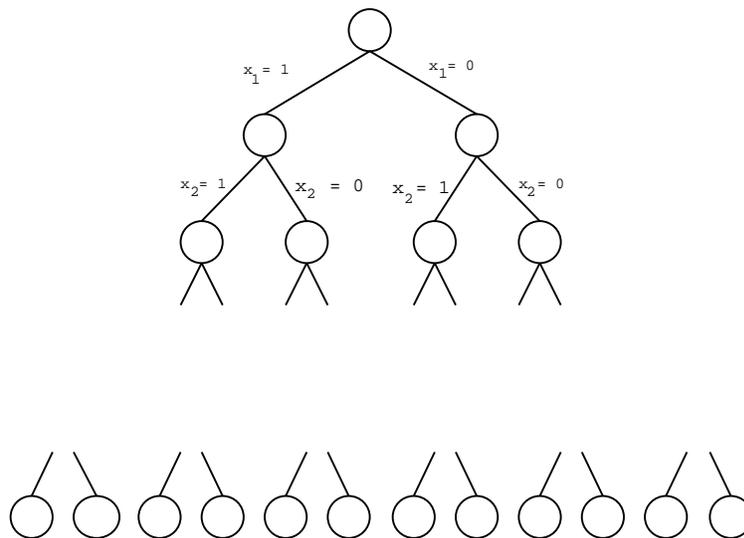


Figura 4.1: Método *branch and bound*

É fácil ver que o sistema acima é de fato uma formulação inteira para o problema, ou seja, $P_{Ste}(G, Z) = P_I$. Podemos, então, utilizar esta formulação na construção de um algoritmo *branch and cut*.

4.1 O Método *Branch and Cut*

No Capítulo 2 discutimos os métodos de planos-de-corte, e, em especial, os cortes faciais. O método *branch and cut* combina as idéias de tais métodos com o método *branch and bound* para resolver, ou obter bons limitantes inferiores e superiores para problemas difíceis de otimização combinatória.

A idéia do método *branch and bound* é enumerar todas as soluções viáveis de um problema em uma estrutura de árvore de decisão. No caso do problema de Steiner, para cada aresta $e \in A$, podemos considerar dois subproblemas: aquele em que a aresta está na solução ($x_e = 1$) e aquele em que a aresta não está ($x_e = 0$). Assim, poderíamos enumerar todas as possíveis soluções através de uma árvore de decisão binária com $2^{|A|}$ folhas (veja Figura 4.1).

Se conhecemos boas soluções heurísticas (limitantes superiores) e boas relaxações (limitantes inferiores) para o problema, a busca na árvore de decisão pode ser muito mais eficiente. Se em um dado nó dessa árvore o valor do limitante inferior para este nó (e portanto para todos os seus descendentes) for superior ao valor de uma solução heurística conhecida, então a solução ótima do problema não poderá estar em um descendente desse nó, e podemos considerar toda a subárvore a partir desse nó como visitada. Com isso, eliminamos vários “ramos” da árvore de decisão, e a busca se torna mais eficiente.

A idéia do método *branch and cut* é utilizar em cada um dos nós da árvore de decisão o método dos planos-de-corte, a fim de gerar bons limitantes inferiores. Nas próximas seções

discutimos detalhadamente diversos aspectos da implementação de um algoritmo deste tipo.

Exercício 4.1.1 *Implemente um algoritmo branch and bound para o Problema de Steiner utilizando como limitante inferior para cada nó o valor das variáveis fixadas no caminho desde a raiz até aquele nó. Considere para isso, o seguinte esquema.*

Algoritmo BRANCH AND BOUND PARA O PROBLEMA DE STEINER

Variáveis: LB , (** limitante inferior **)
 UB , (** limitante superior **)
 $Ativos$; (** lista de nós ativos da árvore **)

Lê os dados do problema;
 $LB := 0$;
 $UB := \infty$;
 $Ativos := raiz$;

Enquanto $L < U$ **faça**
 Se não existe nenhum nó ativo na lista
 então a busca está encerrada
 senão
 $i :=$ Nó ativo da lista $Ativos$;
 Acerte variáveis fixadas nesse nó;
 Calcule limitante inferior LB_i para o nó;
 Se $LB_i > LB$
 então atualiza LB ;
 Calcule limitante superior UB_i para o nó;
 Se $UB_i < UB$
 então atualiza UB ;
 Se $LB_i > UB$ (** nó pode ser eliminado **)
 então Elimina i da lista $Ativos$
 senão
 Escolha variável a ser fixada;
 Retire i da lista $Ativos$;
 Crie dois novos nós, fixando a variável
 em 0 e 1 e os inclua na lista $Ativos$;

Observação: Pode-se implementar diversas formas de calcular limitantes inferiores e superiores, testar várias escolhas diferentes para a variável a ser fixada, assim como testar várias escolhas para percorrer a árvore de decisão.

4.2 Limitante Inferior Inicial

A escolha de uma boa relaxação inicial pode ser decisiva para a solução de problemas práticos através do algoritmo *branch and cut*. O caso ideal é quando dispomos de uma formulação inteira para o problema que não seja muito grande. Nesse caso, essa formulação é um candidato natural para PL inicial.

Nem sempre isso é possível. Para o problema de Steiner, a formulação inteira (P), mostrada anteriormente tem tamanho exponencial. Usamos, então, como relaxação inicial, um subconjunto das inequações dos cortes de Steiner (inequações (1)). Em geral, quanto maior esse subconjunto, melhor será o valor do limitante inferior obtido. Vamos, no entanto, utilizar apenas as inequações dos cortes $\delta(v)$ para $v \in Z$.

4.3 Estratégias de Separação

O sucesso de uma implementação de um algoritmo *branch and cut* está ligado à geração de bons limitantes inferiores em cada nó da árvore de decisão. Para isso, precisamos de inequações válidas de boa qualidade, preferencialmente, as que definem facetas do poliedro em estudo. Além disso, é necessário que rotinas eficientes de separação sejam construídas. Nesta seção discutimos alguns dos aspectos relacionados à estratégia de separação.

Lembramos novamente o que foi discutido no Capítulo 2. Sabemos que resolver o problema da separação tem a mesma complexidade computacional que resolver o problema de otimização. Assim, quando tratamos de problemas \mathcal{NP} -difíceis, não podemos esperar (a menos que se prove que $\mathcal{P} = \mathcal{NP}$) que conheçamos uma descrição completa dos poliedros correspondentes, e muito menos que sejamos capazes de projetar algoritmos eficientes de separação para essas classes de inequações.

4.3.1 Algoritmos polinomiais para separação

Ao estudar um poliedro, muitas vezes conseguimos caracterizar classes de facetas que podem ser separadas em tempo polinomial. No caso do exemplo do poliedro $P_{Ste}(G, Z)$, mostramos no último capítulo, sob que condições as inequações

$$x_e \geq 0, \text{ para } e \in A$$

definem facetas. Claramente, por simples inspeção podemos verificar eficientemente se, dado $x' \in \mathbb{R}^A$, existe alguma inequação deste tipo violada.

Muitas vezes, conhecemos outras classes de inequações que podem ser separadas eficientemente, além dos casos triviais, como mostrado acima. Vamos analisar o caso do $P_{Ste}(G, Z)$. No início deste capítulo, mostramos uma formulação inteira (P) para o problema de Steiner. Uma desvantagem dessa formulação é que o número de inequações do tipo (1) é exponencial. Chamamos tais inequações de **inequações de cortes de Steiner**. Note que o número

de inequações desse tipo é exponencial, já que em um grafo com $|V| = n$ vértices e $|Z| = z$ terminais teremos $z2^{n-1}$ cortes $\delta(W)$ que incluem algum elemento de Z , mas não o incluem completamente. Lembramos que no Capítulo 3 vimos que as inequações dos cortes de Steiner definem facetas do poliedro $P_{Ste}(G, Z)$.

Apesar de o número de tais cortes de Steiner ser exponencial no tamanho do grafo G , podemos separar as inequações correspondentes em tempo polinomial. Seja $x' \in [0, 1]^A$, e considere o grafo $G' = (V, A)$, com uma função de capacidade c associada às arestas tal que $c(e) = x'_e$ para todo $e \in A$. Agora calculamos em G' para cada par de vértices $\{u, v\}$ de Z um fluxo máximo de valor f_{uv} entre u e v . Se $f_{uv} < 1$, então, pelo Teorema de Ford e Fulkerson [14], existe um corte $\delta(W)$ que separa u e v cuja capacidade é igual a f_{uv} . Mas, nesse caso, $\delta(W)$ é um corte de Steiner, e

$$x'(\delta(W)) = c(\delta(W)) = f_{uv} < 1.$$

Assim, encontramos uma inequação do tipo (1) que está violada por x' . Se, para todo par $\{u, v\}$ de terminais, $f_{uv} \geq 1$, então não existe inequação de corte de Steiner violada por x' .

Lembramos que podemos achar um fluxo máximo em um grafo em tempo $O(n^3)$ [14] (na verdade em tempo até melhor, veja [42] ou [23] para um estudo de algoritmos de fluxo máximo em grafos). Como precisamos fazer isso para todo par de terminais ($O(|Z|^2)$ vezes), o procedimento descrito acima pode ser executado em tempo polinomial. Note que isso implica que a relaxação linear de (P), ou seja, o problema de programação linear que obtemos ao trocar as restrições de integralidade (2) por $0 \leq x_e \leq 1$ para todo $e \in A$ pode ser resolvido em tempo polinomial, mesmo que o número de cortes de Steiner seja exponencial. Esse fato é uma consequência direta do resultado de Grötschel, Lovász e Schrijver [53] discutido no Capítulo 2, a respeito da equivalência entre separação e otimização.

Na verdade, existem algoritmos mais eficientes para resolver o problema da separação dos cortes de Steiner. Gomory e Hu mostram em [44] um algoritmo de complexidade $O(n^4)$ para resolver o seguinte problema, chamado **problema do corte de capacidade mínima em um grafo**: dado um grafo $G = (V, A)$ e uma função $c : A \rightarrow \mathbb{R}^+$, encontre um conjunto $W \subset V$, $W \neq \emptyset$, tal que $c(\delta(W))$ seja o menor possível. O algoritmo de Gomory e Hu constrói uma árvore geradora *fluxo-equivalente* do grafo G . Nessa árvore, cada aresta α tem capacidade dada pelo valor da capacidade do menor corte em G que separa os vértices dessa árvore que são separados pela aresta α . Assim, caso o corte de capacidade mínima do grafo não seja um corte de Steiner, precisamos apenas percorrer a árvore gerada pelo algoritmo para achar o corte de Steiner de capacidade mínima. Recentemente, Nagamochi, Ono e Ibaraki [85] mostraram uma implementação mais eficiente do algoritmo de Gomory e Hu, com complexidade $O(mn + n^2 \log n)$, onde $m = |A|$.

Já mostramos no Capítulo 3 diversos algoritmos exatos e eficientes para a separação de classes de inequações que definem facetas de poliedros relacionados com problemas de otimização combinatória. Encontrar tais algoritmos pode ser decisivo para o desempenho de um algoritmo *branch and cut*.

Exercício 4.3.1 Implementar um algoritmo exato para separação dos cortes de Steiner.

Exercício 4.3.2 Modifique o algoritmo branch and bound do Exercício 4.1.1 para utilizar o valor da relaxação linear do problema como limitante inferior.

Exercício 4.3.3 Modifique o algoritmo do Exercício 4.3.2, incorporando o algoritmo de separação do Exercício 4.3.1 (esta é a primeira versão de um algoritmo branch and cut para o problema).

4.3.2 Heurísticas de separação

Na maioria das vezes, nos encontramos na situação em que conhecemos uma classe de inequações que definem facetas de um poliedro, mas não somos capazes de construir algoritmos eficientes para a separação dessas inequações. Para muitas classes, inclusive, podemos mostrar que o problema da separação é \mathcal{NP} -difícil, o que nos deixa com poucas esperanças de encontrar algoritmos eficientes para resolvê-lo. Como nas outras seções, exemplificaremos a abordagem usando para isso o problema de Steiner.

No Capítulo 3 mostramos uma classe de inequações que generaliza a classe das facetas induzidas pelos cortes de Steiner, e que definem facetas do $P_{Ste}(G, Z)$: as **partições de Steiner** [55]. Dado um grafo $G = (V, A)$ e um conjunto Z de terminais, chamamos uma partição $\mathcal{P} = (V_1, \dots, V_k)$ de V de partição de Steiner se $V_i \cap Z \neq \emptyset$ para todo $i = 1, \dots, k$. Na Figura 4.2 mostramos um exemplo de partição de Steiner, onde as arestas do corte da partição estão tracejadas.

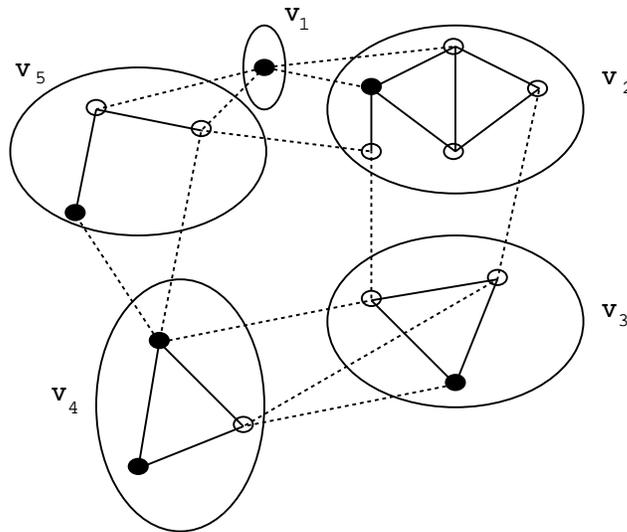


Figura 4.2: Partição de Steiner

A inequação da partição de Steiner é dada por

$$x(\Delta(\mathcal{P})) \geq k - 1,$$

onde $\Delta(\mathcal{P}) := \{uv \in A \mid u \in V_i, v \in V_j \text{ e } i \neq j\}$. Grötschel e Monma mostram em [55] condições necessárias e suficientes para essa inequação definir uma faceta do poliedro $P_{Ste}(G, Z)$.

Em outro artigo, Grötschel, Monma e Stoer [56] mostram que o problema da separação para a classe das partições de Steiner é \mathcal{NP} -completo.

Isto justifica o estudo de heurísticas de separação para essas inequações.

O problema que gostaríamos de resolver é o seguinte: dado $x' \in [0, 1]^A$, encontre, se existir, uma partição de Steiner $\mathcal{P} = (V_1, \dots, V_k)$, tal que a inequação correspondente esteja violada por x' . Mostramos a seguir dois procedimentos heurísticos que podem ser usados na separação destas inequações.

Heurística gulosa

A heurística é iterativa. Em cada iteração temos uma partição $\mathcal{P} = (V_0, V_1, \dots, V_k)$ do grafo G , tal que V_0 contém apenas vértices de Steiner e $V_i \cap Z \neq \emptyset$ para $i = 1, \dots, k$. Na primeira iteração, a partição é formada por $V \setminus Z$ e cada vértice de Z separado.

Em cada iteração, dada a partição \mathcal{P} , calcule $x'(\Delta(\mathcal{P}))$.

- Se $x'(\Delta(\mathcal{P})) < k - 1$, então, para cada vértice v de V_0 , encontre um conjunto V_i de \mathcal{P} que maximiza $x'([v : V_i])$, e acrescente-o a esse conjunto, ou seja, faça $V_i \leftarrow V_i \cup \{v\}$. A partição obtida no fim deste processo é uma partição de Steiner, que pode ou não estar violada pela solução da relaxação linear corrente x' .
- Se $x'(\Delta(\mathcal{P})) \geq k - 1$, encontre uma aresta $e \in \Delta(\mathcal{P})$ que maximiza x'_e . Se e liga duas componentes V_i e V_j , para $i, j \geq 1$, então, tome a partição \mathcal{P}' formada a partir de \mathcal{P} removendo-se os conjuntos V_i e V_j e incluindo-se o conjunto $V_i \cup V_j$. Comece nova iteração com \mathcal{P}' no lugar de \mathcal{P} . Caso a aresta e ligue um vértice u de V_0 a um conjunto V_i , tome a partição $\mathcal{P}' := (V_0 - u, V_1, \dots, V_i + u, \dots, V_k)$, e comece nova iteração com \mathcal{P}' no lugar de \mathcal{P} .

Heurística de cortes mínimos

Esta heurística também é iterativa. Em cada iteração temos uma partição de Steiner $\mathcal{P} = (V_1, \dots, V_k)$. Na primeira iteração temos $\mathcal{P} = (V)$.

Em cada iteração do algoritmo calcule $x'(\Delta(\mathcal{P}))$.

- No caso em que $x'(\mathcal{P}) < k - 1$, então esta partição de Steiner induz uma inequação violada. Pare e retorne a inequação encontrada.
- Se $x'(\mathcal{P}) \geq k - 1$, então escolha $i \in \{1, \dots, k\}$ tal que $|V_i \cap Z| \geq 2$ e calcule no grafo $G[V_i]$ um corte de Steiner de capacidade mínima, usando como peso das arestas o valor da solução atual x' . Isso pode ser feito usando o algoritmo de Gomory e Hu mencionado na Seção 4.3.1. Seja $[V_i^1 : V_i^2]$ o corte encontrado. Tome $\mathcal{P}' := (V_1, \dots, V_i^1, V_i^2, \dots, V_k)$ no lugar de \mathcal{P} e comece nova iteração.

A segunda heurística apresenta, em geral, melhores resultados que a primeira, mas o custo computacional é muito maior. Uma alternativa para reduzir este custo é utilizar, ao invés de um algoritmo exato para achar o corte de capacidade mínima, alguma heurística eficiente que acha boas aproximações para este corte (veja, por exemplo, [92]).

Exercício 4.3.4 *Implemente uma heurística de separação para as inequações de partições de Steiner.*

4.3.3 Seleção de inequações encontradas

Na implementação de algoritmos *branch and cut* verificamos que nem sempre é bom adicionar ao PL todas as inequações encontradas pelos algoritmos de separação. Muitas vezes, algumas inequações são encontradas várias vezes, e devemos evitar introduzir cópias de uma mesma inequação, o que pode ocasionar instabilidade numérica.

Sabemos também que, quanto maior for o problema de programação linear a ser resolvido, mais tempo levaremos para resolvê-lo. Assim, se por um lado, adicionar mais inequações melhora a descrição do poliedro, por outro lado, torna também a solução da relaxação linear mais lenta. A questão que surge é *quantas e quais inequações devemos adicionar ao PL em cada iteração?*

Para esta pergunta não podemos dar uma resposta exata. Cada problema tem uma estratégia ótima diferente para a escolha das inequações. A seguir apresentamos algumas recomendações gerais, decorrentes de experiências que tivemos com os problemas que tratamos.

- *Escolha inequações de classes diferentes.* Parece ser verdade que a escolha de inequações de diversas classes é mais efetiva do que concentrar a atenção em apenas uma classe;
- *Selecione as inequações “mais violadas”.* Para cada inequação $a^\top x \leq \alpha$ encontrada, calcule $\alpha - a^\top x'$, onde x' é a solução da relaxação atual. Quanto maior for a violação da inequação, maiores parecem ser as chances de haver crescimento do limitante inferior;
- *Selecione inequações que cobrem todo o espaço das variáveis.* Um fenômeno que ocorre ao introduzirmos inequações ao sistema linear é que a solução “se adapta” às novas inequações, tentando mudar o mínimo possível. Quando introduzimos inequações mais “globais” a chance disso ocorrer é menor.

Outro aspecto bastante discutido em trabalhos mais recentes é que, algumas vezes, mesmo introduzindo inequações violadas para o problema, o ganho no valor da função objetivo é muito pequeno. Esse fenômeno é conhecido como *tailing off*. Neste caso, alguns autores sugerem que não se insista na separação para este nó, e se parta direto para a fase de *branching*. Veja [4] ou [67] para maiores detalhes a respeito.

4.3.4 Manutenção do *pool* de inequações

Como já mencionamos, às vezes encontrar inequações violadas é uma tarefa árdua que requer tempo de processamento muito elevado. Assim, as inequações encontradas devem ser mantidas para busca futura, já que percorrer uma lista de inequações candidatas é, em geral, muito mais rápido do que executar algoritmos exatos ou mesmo heurísticas de separação.

A idéia de manter um *pool* (depósito) de inequações encontradas está ligada à eliminação de inequações da relaxação linear, discutida na Seção 4.6.2. Assim, todas as inequações encontradas pelos algoritmos de separação são armazenadas para busca futura, o que pode ocasionar problemas com a quantidade de espaço disponível. Uma forma de resolver esse problema é adotar a idéia de “envelhecimento” das inequações do *pool*.

Cada vez que o *pool* é percorrido em busca de inequações violadas pela solução da relaxação linear x' , as inequações que são satisfeitas por x' envelhecem em um ano. De tempos em tempos, um processo de coleta de lixo é executado, procurando inequações muito velhas (ou seja, que há muito tempo não estão mais ativas na relaxação linear). Estas inequações são, então, eliminadas do *pool*¹.

Outra utilização do *pool* de inequações está ligada à fase *branch and bound* do algoritmo. Quando selecionamos um nó da árvore de decisão para execução, podemos querer restabelecer o problema de programação linear que este nó dispunha na hora que foi criado. Para isso, devemos armazenar para cada nó da árvore de decisão, quais inequações estão ativas para aquele nó. Para fazê-lo, basta armazenar no nó uma lista de inequações ativas com os índices dessas inequações no *pool*. Com isso, reconstituímos o estado do PL na hora da criação deste nó.

Exercício 4.3.5 *Implemente as rotinas de gerenciamento e manipulação do pool de inequações.*

4.4 Heurísticas Primais

Mencionamos nesta seção, como utilizar a informação dada pela relaxação linear para construir heurísticas para o problema. O algoritmo *branch and cut*, por si só, é um bom instrumento de teste de heurísticas, já que provê um limitante inferior para o valor da solução ótima.

Além das heurísticas existentes para o problema, podemos aproveitar a informação fornecida pela solução da relaxação linear para obter soluções viáveis. É o que mostraremos a seguir usando o exemplo do problema de Steiner. Para uma resenha a respeito de heurísticas para o problema, veja [68] ou [30].

¹A vida em um algoritmo *branch and cut* é duríssima.

Heurística de árvore geradora máxima

Seja x' a solução da relaxação linear atual, e considere uma árvore geradora de custo máximo de $G = (V, A)$ onde os custos das arestas são dados pelos valores de x' . Enquanto houver na árvore folhas que não são terminais, remova essas folhas.

A idéia da heurística é que, quanto maior é o valor da relaxação linear em uma aresta e , tanto maior é a chance desta aresta aparecer na solução ótima. Algumas vezes vale a pena, em caso de empate do valor de x' nas arestas, tomar uma decisão aleatória, a fim de impedir escolhas viciadas (como, por exemplo, sempre escolher as arestas de menores índices).

Heurística de subdivisão em componentes

Seja x' a solução da relaxação linear atual, e considere um valor limite $0 < l < 1$. Construa, então, um subgrafo de G apenas com as arestas em que $x'_e \geq l$. Este grafo pode ter várias componentes. Resolvemos o problema de Steiner de forma ótima para cada uma das componentes, e combinamos as soluções obtidas a fim de produzir uma solução viável para o problema.

4.4.1 Fixação de variáveis por custo reduzido

Quando encontramos limitantes superiores para o valor da solução ótima (soluções viáveis), podemos fixar o valor de algumas variáveis, levando em conta o ganho que tal alteração acarretaria no valor objetivo da relaxação linear. A seguir descrevemos a idéia mais precisamente.

Seja z' o valor da solução ótima da relaxação linear atual $\min\{c^\top x \mid Ax \leq b\}$. Sabemos que x' está relacionado com uma base A_B de A . Com isso, seja d_e o valor dos custos reduzidos das variáveis obtido na solução da relaxação linear, ou seja, $d := c - c_B^\top A_B^{-1} A_N$, onde N é o conjunto das variáveis fora da base. O valor do custo reduzido da variável indica o ganho no valor da solução caso esta variável seja incluída na base. Seja também \bar{z} um limitante superior para o valor da solução ótima.

Para cada variável não básica e podemos ter dois casos:

- $x'_e = 0$ e $\bar{z} - z' < d_e$. Neste caso, se a variável for incluída na base, o valor da solução da relaxação linear ultrapassa o valor do limitante superior, e portanto podemos fixar a variável x_e em zero;
- $x'_e = 1$ e $\bar{z} - z' < -d_e$. De forma análoga, podemos fixar o valor da variável em 1.

Em algumas aplicações, quando ativamos a fixação de variáveis, o tamanho dos PL a serem resolvidos diminuem drasticamente, o que ocasiona um ganho bastante significativo no desempenho do algoritmo.

4.4.2 Fixação de variáveis por implicações lógicas

Ao se fixar o valor de certas variáveis, podemos ter como consequência que outras variáveis devem ser fixadas. Mostramos nesta seção alguns exemplos de como isso pode ocorrer no problema de Steiner.

Seja $G' = (V, E')$ o grafo obtido a partir de G eliminando as arestas fixadas em 0 e contraindo as arestas fixadas em 1 (caso a aresta seja incidente a pelo menos um terminal, o novo vértice será também um terminal).

Através de observações simples podemos fixar o valor de algumas arestas.

- caso exista $v \in Z$ com grau 1 em G' , então a aresta correspondente pode ser fixada em 1;
- caso exista $v \in V \setminus Z$ com grau 1 em G' , então a aresta correspondente pode ser fixada em zero;
- caso exista $v \in V \setminus Z$ com grau 2 em G' , podemos fixar as arestas incidentes a v em zero, acrescentando uma aresta ligando os vizinhos de v em G' , com custo igual à soma das arestas fixadas em zero;
- caso exista ponte de Steiner $e \in E$, esta pode ser fixada em um.

Exercício 4.4.1 *Implemente uma heurística primal para o problema de Steiner, e incorpore esta heurística na geração de limitantes superiores do algoritmo branch and cut.*

Exercício 4.4.2 *Implemente os procedimentos de fixação de variáveis no seu algoritmo.*

4.5 A Fase *Branch and Bound*

Cada aspecto de um grande sistema deve ser estudado e planejado com cuidado a fim de obtermos bons resultados computacionais. Isso não é diferente no projeto de um algoritmo *branch and cut*. O mesmo ocorre com a fase de enumeração do algoritmo. Nesta seção tratamos dos dois principais aspectos desta fase: a escolha da forma de fazer o *branching* e como percorrer a árvore de decisão.

4.5.1 Como efetuar o *branching*

Quando decidimos que devemos parar a fase de inclusão de planos-de-corte em um determinado nó da árvore de decisão e devemos gerar dois subproblemas, surge uma primeira questão, que é de que forma estes subproblemas devem ser criados.

***Branching* em variáveis**

Neste caso, devemos escolher uma variável livre (que não foi fixada no caminho entre a raiz da árvore de decisão e o nó corrente) e a fixamos em 0 ou 1. Aqui também diversas estratégias de escolhas de variáveis são adotadas.

- Escolha a variável i com $|x'_i - 0.5|$ mínimo. De certa forma, essa é a variável “mais fracionária” da solução atual x' . Tal estratégia tem sido usada pela maior parte dos pesquisadores, e mostra-se eficiente ao diminuir a altura da árvore de decisão quando uma solução viável de boa qualidade é conhecida. Antes de obtermos limitantes superiores devemos usar outras estratégia de escolha de variáveis.
- Escolha a variável fracionária i que maximiza $|x'_i - 0.5|$. Neste caso, estamos selecionando a variável fracionária “mais próxima” de ser inteira na atual solução. Tal estratégia auxilia na busca de soluções viáveis para o problema.

***Branching* em inequações**

Nos últimos anos alguns pesquisadores vêm relatando boas experiências computacionais com o uso de inequações na geração de subproblemas. Este método tem a desvantagem de exigir um gerenciamento mais complicado da árvore de decisão, já que, quando fazemos *branching em variáveis*, a árvore de decisão é binária, e podemos guardar apenas a informação da variável utilizada. A seguir exemplificamos a estratégia para o problema de Steiner.

Neste caso, para gerar um subproblema, poderíamos — ao invés de selecionar uma aresta para ocorrer ou não na solução —, escolher um vértice de Steiner v para estar na árvore de Steiner encontrada. Neste caso, a árvore de decisão teria em um ramo a inequação extra de que $x(\delta(v)) \geq 1$ e, no outro ramo as variáveis das arestas incidentes a este vértice poderiam ser fixadas em zero.

Outro exemplo de inequação que poderíamos utilizar é a seguinte. Dado um vértice v de Z , podemos gerar dois subproblemas. Em um deles, fixamos o grau de v na solução ótima em 1 (inserindo a equação $x(\delta(v)) = 1$ na descrição do poliedro). No outro, a inequação $x(\delta(v)) \geq 2$ é introduzido.

É intuitivo que tal estratégia torna o gerenciamento da árvore muito mais trabalhoso, mas traz um ganho no valor dos limitantes inferiores dos subproblemas.

Exercício 4.5.1 *Teste várias estratégias de branching em seu algoritmo branch and cut.*

4.5.2 Buscas na árvore de decisão

Além da forma escolhida para construir os subproblemas da árvore de decisão, diversas estratégias podem ser utilizadas na hora de percorrer a árvore. Algumas das estratégias mais usadas são descritas a seguir.

- *Busca em profundidade.* Em geral só vale a pena utilizar tal estratégia quando ainda não dispomos de uma solução viável para o problema, já que se enveredamos por um ramo ruim da árvore, podemos investir muito tempo sem grande retorno.
- *Escolha do nó ativo de menor limitante inferior.* A idéia da estratégia é trabalhar nos nós da árvore em que o limitante inferior ainda é ruim, a fim de melhorar a diferença entre os valores do melhor limitante inferior e da melhor solução heurística de que dispomos.

4.6 O Programa de Programação Linear

Na maioria das implementações de algoritmos *branch and cut*, verificamos que grande parte do tempo (em torno de 60%) é gasto na solução do problema de programação linear. É importante, portanto, mantermos este problema o menor possível, e utilizarmos pacotes rápidos e eficientes para a solução de programação linear. A comunidade de otimização combinatória tem se decidido nos últimos anos pelo uso do CPLEX [17], um pacote eficiente e robusto que vem demonstrando ótimos resultados. Uma grande vantagem do CPLEX é a utilização de sua biblioteca de funções (Cplex Callable Library), projetada sob medida para a utilização no contexto de algoritmos *branch and cut*.

4.6.1 Eliminação de variáveis

Para problemas de grande porte, pode ser que trabalhar com todas as variáveis do problema já é intratável. Por exemplo, para o problema de Steiner, se o grafo for completo com 1000 vértices, o número de arestas já é da ordem de 10^6 .

Para tratar este problema, podemos selecionar algumas arestas do grafo com as quais trabalhamos. Por exemplo, podemos tomar para cada vértice do grafo, as 10 arestas de menor custo incidentes a este vértice, e tentamos resolver o problema para este grafo esparso. No final, de posse da solução ótima, devemos percorrer todo o espaço de variáveis, verificando se a inserção de uma variável pode trazer ganho para o valor da solução. Caso isso ocorra, estas variáveis são adicionadas ao PL e o procura-se resolver o novo problema.

Observe que não trabalhar com todas as variáveis pode trazer problemas no gerenciamento do problema de programação linear. Por exemplo, as inequações que são acrescentadas ao problema reduzido devem ser “traduzidas” para o problema com mais variáveis. Isso nem sempre é simples, e pode trazer um custo elevado.

4.6.2 Eliminação de inequações

A cada novo problema de programação linear resolvido, verificamos se cada inequação $a^\top x \leq \alpha$ do problema é satisfeita com igualdade pela solução atual x' . Caso isso não ocorra, é um sinal de que tal inequação não está sendo necessária no PL, e pode ser removida sem danos. Note

que a inequação continua no *pool*, e pode retornar ao PL caso fique violada no decorrer do algoritmo.

4.7 Comentários Finais

Neste capítulo descrevemos de forma detalhada a implementação de um algoritmo *branch and cut*, discutindo vários dos aspectos aos quais devemos ficar atentos a fim de obter uma implementação eficiente. Certamente não pudemos cobrir todos os pontos importantes, mas consideramos que este capítulo pode ser usado de guia para os alunos de uma disciplina de pós-graduação que tenha tal implementação como um projeto de curso. Os exercícios distribuídos neste capítulo tem a intenção de ajudar o aluno nas várias fases deste projeto.

Vale também lembrar que muitos detalhes que não são descritos neste texto (como problemas de instabilidade numérica, que estariam além do escopo pretendido), podem levar a muitas horas de busca de erros no programa, que, enfim, descobrimos estar em algum destes detalhes.

Apesar de considerarmos que este projeto é do nível de um curso intermediário de pós-graduação, fazer implementações eficientes de algoritmos do tipo descrito aqui é projeto de pesquisa de ponta. Basta citar que pesquisadores como Applegate, Bixby, Chvátal e Cook [3] investiram horas de processamento de uma rede de estações de trabalho em paralelo para resolver uma instância de 4461 vértices do problema do caixeiro viajante. Instâncias ainda menores já estão “além do estado-da-arte” em problemas como o problema de Steiner ou o problema do roteamento de veículos.

Algoritmos *branch and cut* têm sido usados com muito sucesso na solução de diversos problemas de otimização combinatória. O aluno interessado em conhecer outros algoritmos baseados nesta estratégia pode consultar, entre outros, [89], [50], [58], [54], [33], [32], [83], [67] e [77]. Na tese de doutorado de Thienel [101] foi construído um *framework* (o *ABACUS*) para desenvolvimento de algoritmos *branch and cut*.

Na figura a seguir mostramos um fluxograma simplificado de um algoritmo *branch and cut*, incluindo vários dos blocos sugeridos neste capítulo.

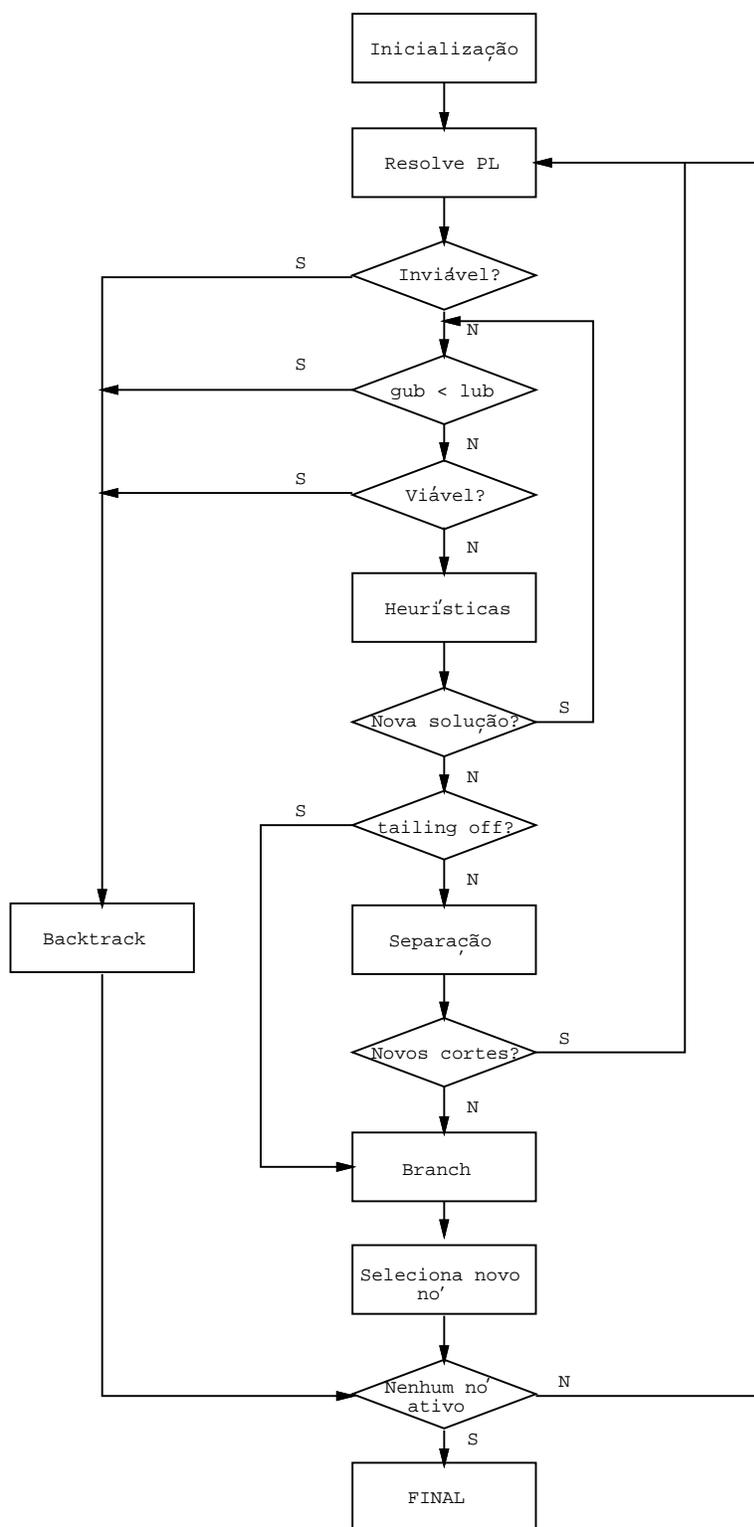


Figura 4.3: Fluxograma simplificado de um algoritmo *branch and cut*.

Referências Bibliográficas

- [1] A.V. Aho, J.E. Hopcroft, J.D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison–Wesley, Reading (1974).
- [2] Y.P. Aneja, “An integer linear programming approach to the Steiner problem in graphs”, *Networks* 10, 167–178 (1980).
- [3] D. Applegate, R.E. Bixby, V. Chvátal, W. Cook, “Finding cuts in the TSP”, *Technical Report* (1994).
- [4] E. Ballas, S. Ceria, G. Cornuejols, “A lift-and-project cutting plane algorithm for mixed 0-1 programs”, *Mathematical Programming* 58, 295–324 (1993).
- [5] M.L. Balinski, “Establishing the matching polytope”, *Journal of Combinatorial Theory* (B), 1–13, (1972).
- [6] F. Barahona, E. Maccioni, “On the exact ground states of the three-dimensional Ising spin glasses”, *Journal of Physics A: Math. Gen.* 15, 1611–1615 (1982).
- [7] F. Barahona, A.J. Majoub, “On the cut polytope”, *Mathematical Programming* 36, 17–173 (1986).
- [8] J.E. Beasley, “An algorithm for the Steiner problem in graphs”, *Networks* 14, 147–159 (1984).
- [9] C. Berge, *Graphs and Hypergraphs*, Elsevier, New York (1976).
- [10] G. Birkoff, “Tres observaciones sobre el algebra lineal”, *Revista Facultad de Ciencias Exactas, Puras y Aplicadas*, Universidad Nacional de Tucuman 5, 147–151 (1946).
- [11] B. Bollobás, *Graph Theory: An Introductory Course*, Graduate Texts in Mathematics 63, Springer–Verlag, Berlin (1979).
- [12] J.A. Bondy, U.S.R. Murty, *Graph Theory with Applications*, Macmillan, London (1976).
- [13] S. Chopra, M.R. Rao, “The Steiner problem I: formulations, compositions and extensions of facets”, *Mathematical Programming* 64, 209–229 (1994).
- [14] V. Chvátal, *Linear Programming*, Freeman, New York (1983).

- [15] W. Cook, C. Coullard, G. Turán, “On the complexity of cutting plane proofs”, *Discrete Applied Mathematics* 18, 25–38 (1987).
- [16] W. Cook, A.M.H. Gerards, A. Schrijver, E. Tardos, “Sensitivity theorems in integer linear programming”, *Mathematical Programming* 34, 251–264 (1986).
- [17] *Using the CPLEX Callable Library and CPLEX Mixed Integer Library*, CPLEX Optimization (1995).
- [18] H. Crowder, M.W. Padberg, “Solving large-scale symmetric travelling salesman problems to optimality”, *Management Science* 26, 495–509 (1980).
- [19] W.H. Cunningham, J. Edmonds, “A combinatorial decomposition theory”, *Canadian Journal of Mathematics* 32, 734–765 (1980).
- [20] W.H. Cunningham, “Testing membership in matroid polyhedra”, *Journal of Combinatorial Theory (B)*, 36, 161–188 (1984).
- [21] G.B. Dantzig, “Application of the simplex method to a transportation problem”, in: *Activity Analysis of Production and Allocation* (Tj. C. Koopmans, ed.), Wiley, New York, 359–373 (1951).
- [22] G.B. Dantzig, R. Fulkerson, S.M. Johnson, “Solution of a large scale traveling salesman problem”, *Operations Research* 2, 393–410 (1954).
- [23] R. Duó, *Fluxos Máximos em Redes: Algoritmos e Estruturas de Dados*, *Dissertação de Mestrado*, Instituto de Matemática e Estatística da USP (1992).
- [24] J. Edmonds, “Maximum matching and a polyhedron with 0,1-vertices”, *Journal of Research of the National Bureau of Standards (B)* 69, 125–130 (1965).
- [25] J. Edmonds, “Paths, trees and flowers”, *Canadian Journal of Mathematics* 17, 449–467 (1965).
- [26] J. Edmonds, “Submodular functions, matroids and certain polyhedra”, in: *Combinatorial Structures and Their Applications*, Gordon and Breach, New York, 69–87 (1970).
- [27] J. Edmonds, “Matroids and the greedy algorithm”, *Mathematical Programming* 1, 127–136 (1971).
- [28] J. Edmonds, “Matroid intersection”, in: *Discrete Optimization I*, (P.L. Hammer, E.L. Johnson. B.H. Korte, eds.), *Annals of Discrete Mathematics* 4, 39–49 (1979).
- [29] E. Egerváry, “Bemerkungen zum Transportproblem”, *MTW Mitteilungen* 5, 278–284 (1958).
- [30] C.E. Ferreira, *O Problema de Steiner em Grafos: Uma Abordagem Poliédrica*, *Dissertação de Mestrado*, Instituto de Matemática e Estatística da USP (1989).

- [31] C.E. Ferreira, *On Combinatorial Optimization Problems Arising in Computer System Design*, Tese de doutorado, Technische Universität Berlin (1994).
- [32] C.E. Ferreira, A. Martin, C.C. de Souza, R. Weismantel, L.A. Wolsey, “The node capacitated graph partitioning problem: a computational study”, aceito para publicação no *Mathematical Programming* (1996).
- [33] C.E. Ferreira, A. Martin, R. Weismantel, “Solving multiple knapsack problems by cutting planes”, *SIAM Journal on Optimization* 6, (1996).
- [34] L.R. Ford Jr., D.R. Fulkerson, “Maximal flow through a network”, *Canadian Journal of Mathematics* 8, 399–404 (1956).
- [35] L.R. Ford Jr, D.R. Fulkerson, “A simple algorithm for finding maximal network flows and an application to the Hitchcock problem”, *Canadian Journal of Mathematics* 9, 210–218 (1957).
- [36] A. Frank, “A weighted matroid intersection algorithm”, *Journal of Algorithms* 2, 328–336 (1981).
- [37] D.R. Fulkerson, “Blocking and anti-blocking pairs of polyhedra”, *Mathematical Programming* 1, 168–194 (1971).
- [38] M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco (1979).
- [39] A.M.H. Gerards, “Matching”, in: *Handbook in Operations Research and Management Science* (M.O. Ball, T.L. Magnanti, C.L. Monma and G.L. Nemhauser, eds.), Volume 7, North-Holland, Amsterdam (1995).
- [40] M.X. Goemans, “The Steiner tree polytope and related polyhedra”, *Mathematical Programming* 63, 157–182 (1994).
- [41] M.X. Goemans, “Worst–case comparison of valid inequalities for the TSP”, *Mathematical Programming* 69, 335–349 (1995).
- [42] A.V. Goldberg, R.E. Tarjan, “A new approach to the maximum flow problem”, *Proceedings 18th Annual ACM Symposium on the Theory of Computing*, 136–146 (1986).
- [43] R.E. Gomory, “Outline of an algorithm for integer solutions to linear programs”, *Bulletin of the American Mathematical Society* 64, 275–278 (1958).
- [44] R.E. Gomory, T.C. Hu, “Multi-terminal network flows”, *Journal of the Society for Industrial and Applied Mathematics* 9, 551–570 (1961).
- [45] M. Grötschel, *Polyedrische Charakterisierungen kombinatorischer Optimierungsprobleme*, Verlag A. Hain, Meisenheim, 1977.

- [46] M. Grötschel, *Optimierungsmethoden I*, notas de aula do curso de Métodos de Otimização I, Universität Augsburg (1985).
- [47] M. Grötschel, *Optimierungsmethoden II*, notas de aula do curso de Métodos de Otimização II, Universität Augsburg (1986).
- [48] M. Grötschel, O. Holland, “Solving matching problems with linear programming”, *Mathematical Programming* 33, 243–259 (1985).
- [49] M. Grötschel, M. Jünger, G. Reinelt, “Facets of the linear ordering polytope”, *Mathematical Programming* 33 (1985) 43–60.
- [50] M. Grötschel, M. Jünger, G. Reinelt, “A cutting plane algorithm for the linear ordering problem”, *Operations Research* 32, 1195–1220 (1985).
- [51] M. Grötschel, L. Lovász, A. Schrijver, “The ellipsoid method and its consequences in combinatorial optimization”, *Combinatorica* 1, 169–197 (1981).
- [52] M. Grötschel, L. Lovász, A. Schrijver, “Corrigendum to our paper “The ellipsoid method and its consequences in combinatorial optimization””, *Combinatorica* 4, 291–295 (1984).
- [53] M. Grötschel, L. Lovász, A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer, Berlin (1988).
- [54] M. Grötschel, A. Martin, R. Weismantel, “Packing steiner trees: a cutting plane algorithm and computational results”, *Preprint 92-9*, Konrad-Zuse-Zentrum für Informationstechnik Berlin (1992).
- [55] M. Grötschel, C.L. Monma, “Integer polyhedra associated with certain network design problems with connectivity constraints”, *SIAM Journal on Discrete Mathematics* 3, 502–523 (1990).
- [56] M. Grötschel, C.L. Monma, M. Stoer, “Facets for polyhedra arising in the design of communication networks with low connectivity constraints, *SIAM Journal on Optimization* 2, (1992).
- [57] M. Grötschel, M. Padberg, “Polyhedral aspects of the traveling salesman problem I: theory”, in: *The Traveling Salesman Problem* (E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys, eds.), Wiley, New York, 1985.
- [58] M. Grötschel, Y. Wakabayashi, “A cutting plane algorithm for a clustering problem”, *Mathematical Programming* 45, 59–96 (1989).
- [59] M. Grötschel, Y. Wakabayashi, “On the Structure of the Monotone Asymmetric Travelling Salesman Polytope I: Hypohamiltonian Facets”, *Discrete Mathematics* (34), 43–59 (1981).
- [60] M. Grötschel, Y. Wakabayashi, “On the Structure of the Monotone Asymmetric Travelling Salesman Polytope II: Hypotractable Facets, *Mathematical Programming Study* (14) 77–97 (1981).

- [61] M. Grötschel, Y. Wakabayashi, “Facets for the Clique Partitioning Polytope”, *Mathematical Programming* 47, 367–388 (1990).
- [62] B. Grünbaum, *Convex Polytopes*, Interscience, London (1967); edição revisada (V. Klee e P. Kleinschmidt, eds.), Graduate Texts in Mathematics, Springer–Verlag, em preparação.
- [63] A. Hoffman, “Linear programming and combinatorics”, in: *Relations Between Combinatorics and Other Parts of Mathematics – Proceedings of Symposia in Pure Mathematics Vol. XXXIV*, (D.K. Ray-Chaudhuri, ed.) American Mathematical Society, Providence, R.I., 245–253 (1979).
- [64] A. Hoffman, “The role of unimodularity in applying linear inequalities to combinatorial theorems”, *Annals of Discrete Mathematics* 4, 73–84 (1979).
- [65] A. Hoffman, J.B. Kruskal, “Integral boundary points of convex polyhedra”, in: *Linear Inequalities and Related Systems* (H.W. Kuhn, A.W. Tucker, eds.), Princeton University Press, 223–246 (1956).
- [66] A. Hoffman, H.W. Kuhn, “Systems of distinct representatives and linear programming”, *The American Mathematical Monthly* 63, 455–460 (1956).
- [67] K. Hoffman, M.W. Padberg, “Solving airline crew scheduling problems by branch-and-cut”, *Management Science* 39, 657–682 (1993).
- [68] F.K. Hwang, D.S. Richards, P. Winter, *The Steiner Tree Problem*, Annals of Discrete Mathematics 53, Amsterdam (1992).
- [69] IBM, *MPSX/370 Mathematical Programming System Extended/370: Program Reference Manual*, SH 19-1095-3, IBM (1979).
- [70] M. Jünger, *Polyhedral Combinatorics and the Acyclic Subdigraph Problem*, Research and Exposition in Mathematics 7, Heldermann–Verlag, Berlin (1985).
- [71] N. Karmarkar, “A new polynomial-time algorithm for linear programming”, *Combinatorica* 4, 373–395 (1984).
- [72] R.M. Karp, C.H. Papadimitriou, “On linear characterizations of combinatorial optimization problems”, *SIAM Journal on Computing* 11, 620–632 (1982)
- [73] L.G. Khachiyan, “A polynomial algorithm in linear programming”, *Soviet Mathematics Doklady* 20, 191–194 (1979).
- [74] Tj.C. Koopmans, M. Beckmann, “Assignment problems and the location of economic activities”, *Econometrica* 25, 53–76 (1957).
- [75] H.W. Kuhn, “The Hungarian method for the assignment problem”, *Naval Research Logistics Quarterly* 2, 83–97 (1955).

- [76] H.W. Kuhn, “On certain convex polyhedra” (abstract), *Bulletin of the American Mathematical Society* 61, 557–558 (1955).
- [77] M. Jünger, G. Reinelt, S. Thienel, “Practical problem solving with cutting plane algorithms in combinatorial optimization”, in: *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* (W. Cook, L. Lovász, P. Seymour, eds.) Vol. 20, 111–152 (1995).
- [78] E.L. Lawler, “Matroid intersection algorithms”, *Mathematical Programming* 9, 31–56 (1975).
- [79] E.L. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, New York, 1976.
- [80] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys (eds.), *The Traveling Salesman Problem*, Wiley, New York, 1985.
- [81] L. Lovász, “Graph theory and integer programming”, *Annals of Discrete Mathematics* 4, 141–158 (1979).
- [82] L. Lovász, M.D. Plummer, *Matching Theory*, Annals of Discrete Mathematics 29, North-Holland, Amsterdam (1986).
- [83] A. Lucena, “Steiner problem in graphs; lagrangean relaxation and strong valid inequalities”, *Technical Report*, Imperial College (1993).
- [84] N. Maculan, “The Steiner problem in graphs”, *Annals of Discrete Mathematics* 31, 185–222 (1987).
- [85] H. Nagamochi, T. Ono, T. Ibaraki, “Implementing an efficient minimum capacity cut algorithm”, *Mathematical Programming* 67, 325–341 (1994).
- [86] G.L. Nemhauser, L.E. Trotter Jr., “Properties of vertex packing and independent system polyhedra”, *Mathematical Programming* 6, 48–61 (1974).
- [87] J. von Neumann, “A certain zero sum two persons game equivalent to the optimal assignment problem”, in: *Contributions to the Theory of Games II* (H.W. Kuhn, A.W. Tucker eds.) Annals of Mathematics Studies 28, 5–12 (1953).
- [88] M. O’heigeartaigh, J.K. Lenstra, A.H.G. Rinnooy Kan (eds.), *Combinatorial Optimization: Annotated Bibliographies*, Wiley, Chichester, 1985.
- [89] M.W. Padberg, M. Grötschel, “Polyhedral aspects of the travelling salesman problem II: Computations”, in: *The Travelling Salesman Problem* (E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys, eds.), Wiley, New York (1985).
- [90] M.W. Padberg, M.R. Rao, “Odd minimum cut-sets and b -matchings”, *Mathematics of Operations Research* 7, 67–80 (1982).

- [91] M.W. Padberg, G. Rinaldi, “Optimization of a 532-city symmetric traveling salesman problem by branch and cut”, *Operations Research Letters* 6, 1–7 (1987).
- [92] M.W. Padberg, G. Rinaldi, “An efficient algorithm for the minimum capacity cut problem”, *Mathematical Programming* 47, 19–36 (1990).
- [93] M.W. Padberg, L.A. Wolsey, ‘Fractional covers for forests and matchings”, *Mathematical Programming* 29, 1–14 (1984).
- [94] C.H. Papadimitriou, M. Yannakakis, “The complexity of facets (and some facets of complexity), *Journal of Computer and System Science* 28, 244–259 (1984).
- [95] C.H. Papadimitriou, K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice–Hall, Englewood Cliffs, New Jersey (1982).
- [96] R. Rado, “Theorems on linear combinatorial topology and general measure”, *Annals of Mathematics* 44, 228–270 (1943).
- [97] A. Schrijver, “Short proofs on the matching polytope”, *Journal of Combinatorial Theory* (B), 34 104–108 (1983)
- [98] A. Schrijver, “Min-max results in combinatorial optimization”, in: *Mathematical Programming, Bonn 1982 — The State of the Art*, Springer–Verlag, Berlin, 439–500 (1983).
- [99] A. Schrijver, *Theory of Linear and Integer Programming*, Wiley, Chichester (1986).
- [100] P.D. Seymour, “Decomposition of regular matroids”, *Journal of Combinatorial Theory* (B) 28, 305–359 (1980).
- [101] S. Thienel, *ABACUS — a Branch-and-cut System*, Tese de Doutorado, Universität zu Köln (1995).
- [102] K. Truemper, “A decomposition theory for matroids V. Testing of matrix total unimodularity”, *Journal of Combinatorial Theory* (B) 49, 241–281 (1990).
- [103] B.L. van der Waerden, *Moderne Algebra I*, Springer–Verlag, Berlin (1937).
- [104] H. Whitney, “On the abstract properties of linear dependence”, *American Journal of Mathematics* 57, 509–533 (1935).
- [105] G.M. Ziegler, *Lectures on Polytopes*, Springer–Verlag, New York (1995).

Índice Remissivo

A

afim

- dependente, 11
- independente, 11

algoritmo

- dual fracionário, 64

arcos, 8

aresta, 7, 44

árvore, 9

B

base

- de um matróide, 88

blossom, 79

branching

- em inequações, 112
- em variáveis, 112

C

caminho, 9

- (v_1, v_k) -caminho, 9
- alternante, 8
- comprimento, 9

circuito, 9

- hamiltoniano, 9

cobertura

- de arestas, 82

combinação

- afim, 11
- cônica, 11
- convexa, 11
- linear, 11

complemento ortogonal, 31

complexidade computacional, 7

componente

conexa, 9

comprimento

de prova de planos-de-corte, 58, 76

cone, 11, 14

polar, 31

poliédrico, 14

recessional, 46

conjunto

- convexo, 11
- dos índices-faceta, 41
- estável, 82
- fechado, 90
- igualdade, 36
- independente, 88
- não-separável, 90
- soma, 10

corte, 8, 85

(s, t) , 8

ímpar, 83

de Gomory, 63

de Steiner, 96, 104

facial, 65

D

diferença simétrica, 10

dimensão, 11, 38

plena, 11

distância, 9

E

eliminação de Fourier–Motzkin, 22

emparelhamento, 8

perfeito, 8

equação

implícita, 40

F

face, 34
 definida, 34
 induzida, 34
 induzida por I , 36
 faceta, 40
 fecho
 afim, 11
 cônico, 11
 convexo, 11
 inteiro, 49
 linear, 11
 fixação de variáveis
 por custo reduzido, 110
 floresta, 9
 fluxo, 86

G

grafo, 7
 2-conexo, 9
 acíclico, 9
 bipartido, 8
 completo, 8
 conexo, 9
 hipo-emparelhável, 8, 79
 orientado, 8
 grau
 de um vértice, 7

H

hiperplano, 14
 suporte, 34

I

inequação
 blossom, 79
 de eliminação de subcircuitos, 94
 do *odd hole*, 99
 válida, 34
 intersecção de matróides, 92

L

lema
 de Farkas, 25
 lemas
 alternativos, 25
 linearmente
 dependente, 11
 independente, 11

M

método
 branch and bound, 102
 branch and cut, 102, 114
 de planos-de-corte, 56
 matróide, 88
 gráfico, 88
 linear, 88
 matriz
 incidência, 8, 9
 totalmente unimodular, 50
 unimodular, 51

P

partição
 de Steiner, 98, 106
 PCV
 assimétrico, 93
 simétrico, 93
 planos-de-corte, 56
 polaridade
 de cones, 31
 poliedro, 14
 de um matróide, 89
 do problema de Steiner, 96
 dos caminhos, 84
 dos circuitos hamiltonianos, 93
 dos cortes, 85
 dos emparelhamentos, 75
 inteiro, 50
 pontudo, 46
 politopo, 14
 ponte de Steiner, 96

ponto interior, 38
pool de inequações, 109
 posto, 11
 afim, 11
 coluna completo, 11
 de Chvátal, 60, 76
 de um matróide, 88
 linha completo, 11
 problema
 da árvore geradora mínima, 70
 da separação
 da inequação de eliminação de subcircuitos, 95
 das inequações *blossom*, 83
 de otimização combinatória linear, 70
 de Steiner, 101
 em grafos, 70, 96
 do (s, t) -corte de capacidade mínima, 70, 85
 do caixeiro viajante, 70, 93
 do caminho mínimo, 70
 do corte de capacidade mínima, 105
 do emparelhamento máximo, 70
 do fluxo máximo, 86
 dual, 28
 primal, 28
 projeção, 13, 17
 ortogonal, 17
 prova de planos-de-corte, 58, 76

R

raio extremal, 44
 representação
 externa, 13, 30
 interna, 13, 30
 reta extremal, 44

S

(s, t) -corte, 9, 85
 (s, t) -fluxo, 86
 semi-espaço, 14
 sistema

irredundante, 40
 redundante, 40
 sistema TDI, 54
 minimal, 55
 solução
 ótima, 27
 viável, 27
 solução de Steiner, 96, 101
 subespaço
 afim, 11
 linear, 11
 subgrafo, 8
 gerado, 8
 gerador, 8
 induzido, 8

T

T -corte, 83
tailing off, 108
 teorema
 de folgas complementares, 29
 de Hoffman e Kruskal, 52
 de König, 82
 de Menger, 87
 de Minkowski, 33
 de Weyl, 30
 do fluxo máximo–corte mínimo, 87
 forte de dualidade, 28
 fraco de dualidade, 28

V

vértice, 7, 40, 44
 adjacente, 7
 vetor
 característico, 10
 de incidência, 10