



**Centro Federal de Educação
Tecnológica de Minas Gerais**

Departamento de Pesquisa e Pós-Graduação

Programa de Mestrado em Tecnologia

Um Estudo do Problema de Soluções Ótimas Inteiras 0-1 via Programação Semidefinida

Dissertação de Mestrado, submetida ao Programa de Mestrado em Tecnologia, como parte dos requisitos exigidos para a obtenção do título de Mestre em Tecnologia.

Dênis Emanuel da Costa Vargas

Licenciado em Matemática – UNI/BH

Orientador: Prof. Dr Sérgio Ricardo de Souza

Belo Horizonte, 27 de outubro de 2004.

V297e VARGAS, Dênis Emanuel da Costa
2004 Um estudo do problema de soluções ótimas
inteiras 0-1 via programação semidefinida . Belo
Horizonte, 2004.
68p.
Dissertação (Mestrado) Centro Federal de Educação
Tecnológica de Minas Gerais
CEFET/MG.
1. Programação semidefinida.
2. Programação linear inteira 0-1. 3. Otimização.
I. Título.

CDD: 001.642

“A coisa mais bonita que podemos experimentar é o misterioso.”

Albert Einstein

“A chave de todas as ciências é o ponto de interrogação.”

Honoré de Balzac

Agradecimento

A Deus.

A minha mãe, **Maria Rosália Vargas**, ao meu pai **Fernando Antônio Vargas** e a minha noiva, **Daniella Oliveira Prates** pelo amor, incentivo e compreensão inigualáveis.

Ao meu orientador e Coordenador do Curso de Mestrado em Tecnologia, **Professor Sérgio Ricardo de Souza**, pelo apoio no Mestrado.

Resumo

Nesta dissertação, estuda-se um problema de otimização de solução inteira 0-1, utilizando-se da teoria de programação semidefinida (PSD). Os problemas de otimização de solução inteira 0-1 possuem métodos computacionais eficientes até um certo número de variáveis. Cada vez que a complexidade do problema aumenta, a busca de novos algoritmos mais eficientes aumenta na mesma proporção. Dois métodos clássicos de solução dos problemas de otimização inteira 0-1 são os métodos do tipo *Branch* (*Branch-and-Bound* e *Branch-and-Cut*) e os métodos do tipo *Lift-and-Project* (BCC e Sherali-Adams). Esses dois métodos clássicos acima citados são baseados em programação linear (PL). Recentemente, descobriu-se que, quando esses mesmos métodos são baseados em programação semidefinida, se tornam muito mais eficientes, encontrando-se a solução com maior rapidez e com menor trabalho computacional. O objetivo deste trabalho é mostrar como funcionam esses dois métodos clássicos, baseados em programação linear e em programação semidefinida, e, ainda, concluir que este último tratamento é computacionalmente melhor que o primeiro, quando aplicado em problemas com um número de variáveis muito grande.

Abstract

In this dissertation pre-proposal, a problem of optimization of solution 0-1 using the theory of semidefinite programming is studied (PSD). The problems of optimization of solution 0-1 possess efficient computational methods until a certain number of variable. Each time that the complexity of the problem increases, the search of new more efficient algorithms increases in the same ratio. Two classic methods of solution of the problems of optimization 0-1 are the methods of the Branch type (*Branch-and-Bound* and *Branch-and-Cut*) and the methods of the *Lift-and-Project* type (BCC and Sherali-Adams). These two classic methods above cited are based on linear programming (PL). Recently, it was uncovered that, when these same methods are based on semidefined programming, them they become much more efficient, finding the solution much more fast and with lesser computational work. The objective of this work is show as these two established classic methods in linear programming and semidefined programming function these, and still, to conclude that this last treatment is computational better that the first one, when applied in problems with a very great number of variable.

LISTA DE DEFINIÇÕES

1. PSD := Programação Semidefinida
2. PL := Programação Linear
3. MPI := Método de Ponto Interior
4. $A \bullet B := tr(AB)$
5. $\|A\|_F := \sqrt{A \bullet A}$
6. $\mathbb{R}_+^{n \times n}$:= Espaço das matrizes semidefinidas positivas
7. $\mathbb{SR}^{n \times n}$:= Espaço das matrizes simétricas de ordem n
8. $A \succeq 0$:= Matriz Semidefinida Positiva
9. A_i := Matriz quadrada fornecida no problema PSD
10. $conv(S)$:= Cobertura convexa de S
11. $Diag(u)$:= Matriz diagonal com o vetor u em sua diagonal
12. $diag(A)$:= Vetor composto pela diagonal de A
13. $\lambda_i(A)$:= i -ésimo autovalor de A
14. $Tr(A)$:= traço da matriz A

Sumário

1	Introdução	3
1.1	Apresentação	3
1.2	Sinopse do Trabalho	6
2	Programação Semidefinida	7
2.1	Introdução	7
2.2	Alguns Aspectos de Álgebra Linear	7
2.3	Problema de Programação Semidefinida	12
2.4	Exemplos de Problemas em PSD	17
2.4.1	Minimizando o Máximo Autovalor	17
2.4.2	Programação Linear	18
2.4.3	O Problema do Corte Máximo	19
2.5	Método de Ponto Interior <i>Path Following</i> Primal Dual	22
2.6	Conclusão	26
3	Problema Linear 0-1 e os Métodos <i>Lift-and-Project</i>	29
3.1	Introdução	29
3.2	Otimização por Restrições Versus Otimização Sobre Cones	30
3.3	Método de Projeção	32
3.3.1	Algoritmo de Fourier - Motzkin	32
3.3.2	Algoritmo de Balas - Pulleyblank	33
3.4	Métodos <i>lift-and-project</i>	34
3.4.1	Método Balas-Ceria-Cornuéjols	37
3.4.2	Método Sherali-Adams	41
3.4.3	Método de Lovász-Schrijver	43
3.4.4	Método de Lasserre	46

3.5	Uma Comparação dos Métodos <i>lift-and-project</i>	49
3.6	Conclusão	52
4	Problema Quadrático 0-1 e o método <i>Branch and Cut</i>	53
4.1	Introdução	53
4.2	Um Simples Exemplo de <i>Branch and Bound</i>	53
4.3	O Problema Quadrático 0-1 e Formulações Equivalentes	56
4.4	Um Procedimento de <i>Branch</i> para PSD	58
4.5	Inequações Hipermétricas para Planos de Corte	60
4.6	Conclusão	61
5	Conclusão	62
	Bibliografia	65

Capítulo 1

Introdução

1.1 Apresentação

Programação Semidefinida (PSD) é um problema de otimização no qual a variável é uma matriz semidefinida positiva. Esse problema consiste na escolha de uma matriz simétrica, que otimize uma função linear sujeita às restrições lineares. Sua formulação é obtida a partir do problema de Programação Linear (PL), realizando-se a troca do vetor de variáveis por uma matriz de variáveis e das restrições de não-negatividade pela restrição da matriz semidefinida positiva. [36].

Dois fatores principais são suficientes para motivar o estudo da programação semidefinida. O primeiro é sua aplicação em diversas áreas, como Teoria de Controle e Otimização Combinatória. O segundo é o aspecto computacional, pois, em diversos problemas, mostrou-se que PSD possui um melhor condicionamento numérico do que os métodos tradicionais de resolução.

A programação semidefinida surgiu quando R. Bellman e K. Fan, em um artigo sobre sistemas de inequações lineares na matriz hermitiana, apresentado em 1963, resolveram efetuar, no problema de programação linear, a troca mencionada acima [3]. Eles propuseram, desse modo, um problema de difícil solução, porém, associado a ele apresentaram o problema dual, mais fácil de se tratar e, além disso, demonstraram a condição de dualidade forte.

A restrição de que essa matriz teria que ser semidefinida positiva veio quando essa proposição foi dada a um problema de estabilidade de sistemas de controle. Para a definição de estabilidade segundo o Método de Lyapunov, uma das restrições da serem satisfeitas é uma inequação matricial linear [4]. Em 1973, W. E. Donath

e A. J. Hoffman [9] e em 1975, W. E. Donath, J. Cullum e P. Wolfe [8] apontaram que alguns problemas de partição de grafos poderiam ser resolvidos considerando-se um problema de otimização de autovalores. Em 1996, A.S. Lewis e M.L. Overton mostraram como otimização de autovalores pode ser formulada como um problema de programação semidefinida [23].

Diversos trabalhos foram publicados visando apresentar a teoria de programação semidefinida e suas aplicações. Algumas das publicações mais importantes nesse sentido foram feitas por L. Vandenberghe e S. Boyd [39], C. Helmberg [15], M. J. Todd [36] e H. Wolkowicz, R. Saigal, L. Vandenberghe [40]. Essa última é conhecida como a bíblia da programação semidefinida, contendo 877 citações. Todas discutem a teoria de PSD e um número considerável de aplicações, dentre elas, as aplicações em Teoria de Controle e em vários problemas de otimização combinatória, como otimização em grafos, programação inteira, etc.

O segundo fator motivador se apresenta quando, em 1994, Yu. E. Nesterov e A.S. Nemirovski [32] e em 1995, F. Alizadeh [1] mostraram como os métodos de ponto interior, formulados inicialmente para problemas de programação linear, poderiam resolver problemas de programação semidefinida. Esse foi o grande passo que PSD precisava para ser uma formulação aceita na comunidade científica de otimização e tornar-se conhecida como a Programação Linear do século XXI. Assim, além de possuir um número considerável de aplicações, PSD também passava a possuir um método computacional muito eficiente.

Mesmo com os métodos de ponto interior (MPI) resolvendo a maior parte dos problemas de PSD, a busca de algoritmos que aproveitem a estrutura particular do problema especificado e diminua o número de operações aritméticas para problemas de grande porte continua. Muitos autores escreveram sobre algoritmos alternativos para PSD ou mesmo sobre aplicações de MPI. Dentre eles, estão C. Helmberg, F. Rendl, H. Wolkowicz e R. J. Vanderbei. K. Krishnan e J. E. Mitchell [29] estudaram um método alternativo de resolução de um PSD que tem um número de restrições muito grande.

Assim, PSD se mostra uma nova forma de modelagem de problemas de otimização, de bom desempenho computacional, aplicada em vários problemas clássicos de otimização. Entre essas aplicações, como já foi dito, está a Otimização Combinatória. A otimização combinatória é o problema de encontrar um subconjunto de uma família de subconjuntos de um conjunto dado que otimiza uma certa função de custo. Esse

tipo de problema requer uma enumeração de todos os subconjuntos factíveis.

As aplicações de PSD em otimização combinatória tiveram como precursores M. Grötschel, L. Lovász e A. Schrijver [14]. Eles pesquisaram aplicações em problemas de otimização combinatória, usando soluções aproximadas de relaxações lineares e relaxações semidefinidas.

Os métodos de resolução desses problemas visam reduzir o número de subconjuntos a serem enumerados. Uma dessas técnicas de reduzir os subconjuntos, descartando aqueles que forem detectados como impossíveis de serem soluções, é conhecida como Inserção de Planos de Corte. Essa técnica consiste em inserir algumas restrições, que são os Planos de Corte, de tal forma que o conjunto dos pontos factíveis do problema seja formado apenas por pontos que são candidatos a solução ótima. Geralmente, esse conjunto pode ser descrito por restrições lineares, o que faz com que esse tipo de problema esteja fortemente ligado a programação linear.

Assim, em 1995, M. X. Goemans e D. P. Williamson [12] mostraram que, ao invés de descrever esse conjunto por restrições lineares, descrevê-lo por restrições semidefinidas pode providenciar boa aproximação para o problema de max-cut.

O problema de max-cut é um bom exemplo de um problema de otimização combinatória que pode ser resolvido por PSD. Ele consiste em encontrar um corte em um grafo que seja o maior corte possível. Outros autores também escreveram sobre o uso de PSD no problema de max-cut, como M. Laurent [25], M. X. Goemans e D. P. Williamson [13] e Miguel F. Anjos e Henry Wolkowicz [2].

A resolução dos problemas de otimização combinatória utilizando-se de uma formulação de PSD se mostrou eficiente no aspecto computacional. O fato é que PSD constrói relaxações melhores do que as relaxações construídas normalmente pela programação linear.

Um outro problema clássico de otimização combinatória é o problema de otimização inteira do tipo 0-1. Em 1991, L. Lovász e A. Schrijver [24] mostraram que PSD pode providenciar melhores relaxações de problemas de otimização 0-1 do que as relaxações lineares providenciam. Posteriormente, J. Lasserre [20, 21] mostrou uma relaxação ainda melhor que a relaxação de L. Lovász e A. Schrijver. Todas essas formulações para problemas do tipo 0-1 eram lineares. Em 1996, C. Helmberg e F. Rendl [16] propuseram a solução de um problema quadrático do tipo 0-1, utilizando-se do conhecido procedimento de *Branch-and-Cut* baseado em PSD.

Problemas de otimização inteira 0-1 são aplicados em vários problemas, como

por exemplo, o problema de max-cut citado acima. Vale lembrar que esses métodos de solução inteira são considerados para um problema de um grande número de variáveis, uma vez que existem eficientes métodos de implementação mais simples para problemas pequenos, como *Branch-and-Cut* e *Branch-and-Bound*.

1.2 Sinopse do Trabalho

Este trabalho trata de uma abordagem por restrições semidefinidas para problemas de otimização inteira 0-1. O motivo de se usar essa abordagem é a vantagem computacional de programação semidefinida e o grande número de aplicações dos métodos tratados.

O capítulo 1 faz uma revisão bibliográfica e coloca à disposição do leitor o estado da arte em programação semidefinida. Cita também a evolução da programação semidefinida e os principais pesquisadores, suas aplicações e algoritmos, de modo especial, sua aplicação em programação inteira 0-1.

O capítulo 2 apresenta uma abordagem teórica de programação semidefinida, citando exemplos de aplicações, demonstrando a teoria de dualidade e desenvolvendo o algoritmo de solução primal dual *path-following*, que é um método da classe dos métodos de ponto interior.

O capítulo 3 mostra como se resolve um problema de programação linear 0-1 com os métodos *lift-and-project*. Mostra-se o tratamento com restrições lineares e o tratamento com restrições semidefinidas. Neste capítulo, conclui-se que as construções baseadas em programação semidefinida são melhores que as baseadas em programação linear.

O capítulo 4 mostra como se resolve um problema de programação quadrática 0-1 com os métodos *Brach-and-Bound* e *Brach-and-Cut*. Como feito no capítulo 3, também mostra-se o tratamento primeiramente com restrição linear e, depois, o tratamento com restrição semidefinida.

Conclusões desse trabalho são apresentadas no capítulo 5, ao mesmo tempo que são apresentadas perspectivas de trabalhos futuros.

Capítulo 2

Programação Semidefinida

2.1 Introdução

Neste capítulo, será apresentada a teoria de programação semidefinida. O capítulo é iniciado com uma revisão dos aspectos mais importantes de Álgebra Linear, relacionados ao desenvolvimento do presente trabalho de dissertação. Em seguida, são introduzidos os traços mais relevantes da teoria de programação semidefinida, em especial o desenvolvimento do problema dual e sua relação com programação linear. O capítulo é finalizado com a apresentação de métodos de pontos interiores que utilizam PSD como ferramenta.

2.2 Alguns Aspectos de Álgebra Linear

Considere o espaço vetorial das matrizes simétricas de ordem n como sendo o conjunto $\mathbb{S}\mathbb{R}^{n \times n}$. O produto interno desse espaço é definido como:

$$A \bullet B = \text{Tr}(AB)$$

onde $\text{Tr}(U)$ é o traço de U ou a soma dos elementos da diagonal principal de U . Pode-se demonstrar que $\text{Tr}(U)$ é a soma dos autovalores de U . O espaço $\mathbb{S}\mathbb{R}^{n \times n}$ é um espaço normado, tendo norma associada com o produto interno acima e definida como:

$$\|A\|_F = \sqrt{A \bullet A}.$$

Essa norma é conhecida como Norma de Frobenius.

Definição 2.1 Matriz Definida Positiva

Uma matriz $A \in \mathbb{R}^{n \times n}$ é definida positiva se e somente se

$$x^T A x > 0, \quad \forall x \in \mathbb{R}^n.$$

Se a desigualdade não é estrita, ou seja, se também é válida a igualdade, então a matriz é dita semidefinida positiva. ■

O espaço das matrizes semidefinidas positivas é denominado $\mathbb{R}_+^{n \times n}$. Pode-se demonstrar que, em uma matriz semidefinida positiva, todos os autovalores são não negativos. Para verificar isso, considere o seguinte teorema:

Teorema 2.2

Seja B uma matriz não singular, isto é, $\det(B) \neq 0$. Então, A é semidefinida positiva se e somente se $B^T A B$ também o é. ■

Demonstração:

Seja $x \in \mathbb{R}^n$. Considere $y = B^{-1}x$.

Assim,

$$x^T A x = x^T B^{-T} B^T A B B^{-1} x = (B^{-1}x)^T B^T A B B^{-1} x = y^T B^T A B y.$$

Como $x^T A x \geq 0$ para todo $x \in \mathbb{R}^n$, tem-se que $y^T B^T A B y \geq 0$ para todo $y \in \mathbb{R}^n$, fazendo com que $B^T A B \geq 0$. ■

Para finalizar a argumentação de que toda matriz semidefinida positiva tem seus auto-valores não negativos, observa-se o seguinte: se $A \succeq 0$, então existe uma matriz não singular P tal que $P^T A_0 P = A$ é decomposição em autovalores de A , isto é, A_0 é uma matriz diagonal com os autovalores de A na diagonal principal. Pelo teorema 2.2, ela é semidefinida positiva. Isso significa que

$$x^T A x \geq 0 \iff x^T P^T A_0 P x \geq 0.$$

Fazendo $y = P x$, tem-se $y^T A_0 y \geq 0$ para todo vetor $y \in \mathbb{R}^n$. Definindo os autovalores de A como sendo $\lambda_1, \dots, \lambda_n$, obtém-se:

$$y^T A_0 y = \lambda_1 y_1^2 + \dots + \lambda_n y_n^2 \geq 0.$$

Como essa relação é válida para todo $y \in \mathbb{R}^n$, tem-se que todos os auto-valores são não negativos, uma vez que, se algum $\lambda_i < 0$ para algum $i = 1, \dots, n$, fica fácil mostrar, como contra-exemplo, a existência de um vetor y com entradas y_j nulas para $j \neq i$ e com entrada positiva em $j = i$, tal que $y^T A_0 y < 0$. Assim, os autovalores de A são não negativos.

Dados $A, B \in \mathbb{R}_+^{n \times n}$, tem-se que $A \bullet B \geq 0$. No caso de $A \bullet B = 0$, tem-se $AB = 0$. Isso porque, se $A \in \mathbb{R}_+^{n \times n}$, então existe uma matriz não singular V tal que $A = V^T A_0 V$ seja a decomposição em autovalores de A . Assim,

$$A \bullet B = \text{Tr}(AB) = \text{Tr}(V^T A_0 V B) = \text{Tr}(A_0 V^T B V) = \sum_{i=1}^n \lambda_i(A_0) V_i^T B V_i$$

onde V_i é a i -ésima coluna de V e $\lambda_i(A_0)$ é o i -ésimo autovalor de A_0 . Logo, $\lambda_i(A_0) \geq 0$ para todo $i = 1, \dots, n$. Também tem-se $V_i^T B V_i$ não negativo, uma vez que $B \in \mathbb{R}_+^{n \times n}$. Portanto, $A \bullet B \geq 0$.

Caso $A \bullet B = 0$, então os autovetores correspondentes aos autovalores positivos de A pertencem ao espaço nulo de B , obtendo-se, portanto, que $AB = 0$.

Definição 2.3 Transformação Linear

Seja $T : \mathbb{V} \longrightarrow \mathbb{W}$ uma função que vai do espaço vetorial \mathbb{V} para o espaço vetorial \mathbb{W} . Então T é chamada de Transformação Linear se, para todo vetor u, v de \mathbb{V} e para todos escalares λ, β , tem-se que:

$$T(\lambda u + \beta v) = \lambda T(u) + \beta T(v).$$

■

Definição 2.4 Isomorfismo

Diz-se que uma transformação linear $T : \mathbb{V} \longrightarrow \mathbb{W}$ é um isomorfismo se é injetora (isto é, T leva vetores distintos de \mathbb{V} em vetores distintos de \mathbb{W}) e sobrejetora (isto

é, todo vetor de \mathbb{W} é imagem de algum vetor de \mathbb{V} por T). Diz-se que os espaços vetoriais \mathbb{V} e \mathbb{W} são isomorfos se existe um isomorfismo de \mathbb{V} sobre \mathbb{W} . ■

Definição 2.5 A Transformação Linear A e sua adjunta A^*

A transformação linear A é a aplicação $A : \mathbb{S}\mathbb{R}^{n \times n} \longrightarrow \mathbb{R}^m$ definida por

$$AX = (A_i \bullet X)_{i=1}^m$$

onde A_i é uma matriz quadrada simétrica de ordem n dada, para todo i entre 1 e m .

Observe que, se $X \in \mathbb{S}\mathbb{R}^{n \times n}$ e $v \in \mathbb{R}^m$, então:

$$(AX)^T v = \sum_{i=1}^m (A_i \bullet X) v_i = \sum_{i=1}^m (v_i A_i) \bullet X.$$

Assim, a adjunta de A é dada por:

$$A^* v = \sum_{i=1}^m v_i A_i$$

e também é uma transformação linear $A^* : \mathbb{R}^m \longrightarrow \mathbb{S}\mathbb{R}^{n \times n}$. ■

Definição 2.6 Poliedro

Um poliedro $\mathcal{P} \subseteq \mathbb{R}^n$ é o conjunto dos vetores de \mathbb{R}^n que satisfazem um número finito de inequações lineares. Assim,

$$\mathcal{P} = \{x \in \mathbb{R}^n : Ax \leq b\}.$$

Definição 2.7 Politopo

Um poliedro é limitado quando existe um escalar α tal que, para todo elemento de \mathcal{P} , tem-se $-\alpha \leq x_i \leq \alpha$, para $i = 1, \dots, n$. Um poliedro limitado é chamado de politopo. ■

Definição 2.8 Conjunto Convexo

Um conjunto $\mathcal{T} \subseteq \mathbb{R}^n$ é convexo quando a combinação convexa de dois elementos do conjunto também pertencer ao conjunto. Matematicamente, $\mathcal{T} \subseteq \mathbb{R}^n$ é convexo se, e somente se,

$$\forall x_1, x_2 \in \mathcal{T} \Rightarrow \lambda x_1 + (1 - \lambda)x_2 \in \mathcal{T}, \quad 0 \leq \lambda \leq 1$$

■

Definição 2.9 Cone

Um conjunto $\mathcal{C} \subseteq \mathbb{R}^n$ é um cone se, e somente se,

$$\forall x \in \mathcal{C} \Rightarrow \lambda x \in \mathcal{C}, \quad \forall \lambda \geq 0$$

■

O espaço $\mathbb{R}_+^{n \times n}$ é isomorfo a um cone em \mathbb{R}^p , onde $p = \binom{n+1}{2}$.

A cobertura convexa de um conjunto \mathcal{S} é o conjunto de todos os pontos que são combinações convexas dos elementos de \mathcal{S} e é denotado por $\text{conv}(\mathcal{S})$. Se \mathcal{S} é convexo, então $\text{conv}(\mathcal{S}) = \mathcal{S}$. Caso contrário, $\mathcal{S} \subseteq \text{conv}(\mathcal{S})$ e não existe nenhum outro conjunto convexo \mathcal{V} tal que $\mathcal{S} \subseteq \mathcal{V} \subseteq \text{conv}(\mathcal{S})$.

Definição 2.10 Raios e Raios Extremos de um Poliedro

Seja $\mathcal{P} = \{x \in \mathbb{R}^n : Ax \leq b\}$ um poliedro não-vazio. Então, todo $r \in \mathbb{R}^n$ tal que $Ar \leq 0$ é chamado de raio de \mathcal{P} .

Um raio r de \mathcal{P} é chamado de raio extremo se não existem raios r_1 e r_2 de \mathcal{P} , com $r_1 \neq \lambda r_2$, para qualquer escalar positivo λ , tal que

$$r = \frac{1}{2}r_1 + \frac{1}{2}r_2.$$

■

2.3 Problema de Programação Semidefinida

Nesta seção será apresentado o problema de programação semidefinida e suas consequências. Leituras mais avançadas poderão ser encontradas em [15, 36, 39, 40].

Definição 3.1 Problema de Programação Semidefinida (PSD)

O problema de programação semidefinida na forma primal é o problema:

$$\begin{aligned} \min_x \quad & C \bullet X \\ \text{s. a} \quad & A_i \bullet X = b_i \quad i = 1, \dots, m. \\ & X \succeq 0 \end{aligned} \tag{2.1}$$

O dual de (2.1) é

$$\begin{aligned} \max_{y, S} \quad & b^T y \\ \text{s. a} \quad & \sum_{i=1}^m y_i A_i + S = C. \\ & S \succeq 0 \end{aligned} \tag{2.2}$$

Em (2.1), $A_i, C \in \mathbb{S}\mathbb{R}^{n \times n}$ e $b \in \mathbb{R}^m$ são os dados do problema e $X \in \mathbb{S}\mathbb{R}^{n \times n}$ é a variável. Já em (2.2), as variáveis passam a ser $S \in \mathbb{S}\mathbb{R}^{n \times n}$ e $y \in \mathbb{R}^m$, onde S é chamada de matriz de folga. ■

O problema (2.2) pode ser formulado alternativamente como:

$$\begin{aligned} \max_y \quad & b^T y \\ \text{s. a} \quad & \sum_{i=1}^m y_i A_i \preceq C \end{aligned} \tag{2.3}$$

ou, então, como:

$$\begin{aligned}
& \max_y \quad b^T y \\
& \text{s . a } \quad C - \sum_{i=1}^m y_i A_i \succeq 0.
\end{aligned} \tag{2.4}$$

Teorema 3.2 Teorema da Dualidade Fraca

Se X é factível em (2.1) e (y, S) é factível em (2.2) então, tem-se que:

$$C \bullet X - b^T y = X \bullet S \geq 0.$$

■

Demonstração:

$$C \bullet X - b^T y = \left(\sum_{i=1}^m y_i A_i + S \right) \bullet X - b^T y = \sum_{i=1}^m y_i A_i \bullet X + S \bullet X - b^T y$$

Mas:

$$\sum_{i=1}^m y_i A_i \bullet X + S \bullet X - b^T y = \sum_{i=1}^m (A_i \bullet X) y_i + S \bullet X - b^T y.$$

Como $A_i \bullet X = b_i$, então:

$$\sum_{i=1}^m (A_i \bullet X) y_i + S \bullet X - b^T y = \sum_{i=1}^m b_i y_i + S \bullet X - b^T y$$

Porém,

$$b^T y = \sum_{i=1}^m b_i y_i$$

Então:

$$\begin{aligned}
\sum_{i=1}^m b_i y_i + S \bullet X - b^T y &= b^T y + S \bullet X - b^T y = S \bullet X \\
&= \text{Tr}(SX) = \text{Tr}(XS) = X \bullet S.
\end{aligned}$$

Falta provar que $X \bullet S \geq 0$. Para isso, considere o seguinte fato. Como $X \succeq 0$, então X admite raiz quadrada, isto é, existe $X^{\frac{1}{2}}$. Logo:

$$X = X^{\frac{1}{2}}X^{\frac{1}{2}} \implies X \bullet S = \text{Tr}(XS) = \text{Tr}(X^{\frac{1}{2}}X^{\frac{1}{2}}S) = \text{Tr}(X^{\frac{1}{2}}SX^{\frac{1}{2}}).$$

Mas $S \succeq 0 \implies X^{\frac{1}{2}}SX^{\frac{1}{2}} \succeq 0$ para toda matriz $X \succeq 0$. Se $X \succeq 0$, então $\text{Tr}(X) \geq 0$. Logo:

$$X \bullet S = \text{Tr}(X^{\frac{1}{2}}SX^{\frac{1}{2}}) \geq 0$$

o que leva a $X \bullet S \geq 0$. ■

A diferença entre o ótimo primal e o ótimo dual é denominada *gap* de Dualidade e é sempre não negativo.

Pode-se escrever as forma primal e dual usando a transformação linear A e sua adjunta A^* . Assim, tem-se (2.1) como o problema:

$$\begin{aligned} \min_x \quad & C \bullet X \\ \text{s. a} \quad & AX = b \end{aligned} \tag{2.5}$$

$$X \succeq 0 \tag{2.6}$$

e (2.2) como:

$$\begin{aligned} \max_{y,S} \quad & b^T y \\ \text{s. a} \quad & A^*y + S = C \end{aligned} \tag{2.7}$$

$$S \succeq 0 \tag{2.8}$$

A demonstração do teorema da dualidade fraca pode ser escrita, assim, da seguinte maneira:

Demonstração:

$$C \bullet X - b^T y = (A^*y + S) \bullet X - b^T y = A^*y \bullet X + S \bullet X - b^T y.$$

Mas

$$\begin{aligned}
 A^*y \bullet X &= \left(\sum_{i=1}^m y_i A_i \right) \bullet X \\
 &= y_1 A_1 \bullet X + y_2 A_2 \bullet X + \dots + y_m A_m \bullet X \\
 &= [(A_i \bullet X)_{i=1}^m]^T y \\
 &= (AX)^T y.
 \end{aligned}$$

Portanto:

$$\begin{aligned}
 A^*y \bullet X + S \bullet X - b^T y &= (AX)^T y + S \bullet X - b^T y \\
 &= b^T y + S \bullet X - b^T y \\
 &= X \bullet S.
 \end{aligned}$$

■

Lema 3.3 Lema de Farkas para PSD

Seja $\{BX : X \succeq 0\}$ um conjunto fechado e $b \in \mathbb{R}^m$. Então um, e apenas um, dos dois sistemas abaixo tem solução:

$$(S1) \begin{cases} BX = b \\ X \succeq 0 \end{cases} \quad \text{ou} \quad (S2) \begin{cases} B^*y \succeq 0 \\ b^T y < 0 \end{cases}$$

■

Demonstração:

Suponha que exista solução para o sistema (S1). Assim, existe uma matriz $X \succeq 0$ com $BX = b$. Suponha também que exista y tal que $b^T y < 0$. Assim:

$$b^T y = (BX)^T y = X \bullet B^*y < 0.$$

Como $X \succeq 0$, tem-se que B^*y não é semidefinida positiva, fazendo com que o sistema (S2) não tenha solução.

Agora suponha que não exista solução para o sistema (S1). Isso significa que existe um y tal que $b^T y < 0$ e o produto interno entre y e BX seja não negativo. Mas esse produto interno é o mesmo que $B^*y \bullet X$, fazendo com que $B^*y \succeq 0$, uma vez que $X \succeq 0$. ■

Teorema 3.4 Teorema da Dualidade Forte

Suponha que (y, S) é uma solução estritamente factível, isto é, $S \succ 0$ em (??). Seja p^* e d^* as soluções ótimas dos problemas (2.1) e (2.2), respectivamente. Então $p^* = d^*$ e se p^* é finito, ele é encontrado para algum X factível em (2.1). ■

Demonstração:

Descarta-se a hipótese de $p^* = -\infty$ pelo teorema da dualidade fraca. O gap de dualidade é sempre não negativo e nesse caso a função objetiva de (2.2) sera menor que $-\infty$, o que gera um absurdo.

Suponha, por absurdo, que $p^* > d^*$, onde d^* é finito.

Como p^* é o mínimo do problema e não existe nenhum outro X factível em (2.1) que faz com que $C \bullet X$ seja menor que p^* , o sistema

$$C \bullet X = d^*$$

$$AX = b$$

$$X \succeq 0$$

se torna infactível.

Pelo Lema de Farkas, tem-se que existe um vetor $\begin{pmatrix} y_0 \\ y_1 \end{pmatrix}$ tal que

$$d^* y_0 + b^T y_1 < 0 \quad e \quad y_0 C + A^* y_1 \succeq 0.$$

Se $y_0 = 0$ então

$$b^T y_1 < 0 \quad e \quad A^* y_1 \succeq 0.$$

Mas

$$b^T y_1 = (AX)^T y_1 = (A^* y_1) \bullet X < 0$$

o que é um absurdo, uma vez que o produto interno entre duas matrizes semidefinidas é sempre não negativo.

Se $y_0 > 0$ então, dividindo por y_0 ,

$$d^* + b^T y_1 / y_0 < 0 \quad e \quad C + A^* y_1 / y_0 \succeq 0.$$

Escolhe-se $y = -y_1/y_0$

$$d^* - b^T y < 0 \quad e \quad C - A^* y \succeq 0.$$

Isso significa que existe uma solução dual factível y com valor da função objetiva melhor que d^* , já que $d^* - b^T y < 0$ implica que $d^* < b^T y$. Mas isso é um absurdo, já que d^* é o melhor.

Se $y_0 < 0$ então, dividindo por $-y_0$,

$$-d^* - b^T y_1/y_0 < -a \quad e \quad -C - A^* y_1/y_0 \succeq 0$$

onde $a > 0$. Toma-se uma solução factível y com $d^* - b^T y < e$, onde $e < a$ e $e > 0$ e $C - A^* y \succeq 0$.

Somando as inequações $-d^* - b^T y_1/y_0 < -a$ e $d^* - b^T y < e$, tem-se $-b^T y_1/y_0 - b^T y < e - a < 0$. Essa inequação pode ser escrita como $b^T y_2 < 0$, onde $y_2 = -y_1/y_0 - y$. Da mesma maneira, somando as inequações $C - A^* y \succeq 0$ e $-C - A^* y_1/y_0 \succeq 0$, tem-se $-A^* y_1/y_0 - A^* y \succeq 0$, que também pode ser reescrita como $A^* y_2 \succeq 0$. Como já dito anteriormente, isso é um absurdo.

Como não existe possibilidade para $p^* > d^*$, pode-se concluir que $p^* = d^*$. ■

Conclui-se que se (2.1) é estritamente factível com p^* finito, então $p^* = d^*$ e encontra-se esse valor para (2.2). Também conclui-se que, se (2.2) é estritamente factível com d^* finito, então $p^* = d^*$ e encontra-se esse valor para (2.1). Analogamente, quando tanto (2.1) como (2.2) são estritamente factíveis, encontra-se o valor para ambos.

2.4 Exemplos de Problemas em PSD

2.4.1 Minimizando o Máximo Autovalor

Suponha que se queira minimizar o maior autovalor de uma matriz. Seja $M(z) \in \mathbb{SR}^{n \times n}$ e $z \in \mathbb{R}^n$ onde $M(z)$ depende linearmente de z . Deve-se escolher z de tal

maneira que z minimize o autovalor máximo de $M(z)$. Seja d um escalar. Sabe-se que

$$\begin{aligned}\lambda_{\max}(M(z)) \leq d &\iff \lambda_{\max}(M(z) - dI) \leq 0 \iff \\ \lambda_{\min}(dI - M(z)) &\geq 0 \iff dI - M(z) \succeq 0.\end{aligned}$$

Logo, o problema de otimização de autovalores pode ser escrito como um problema de programação semidefinida da seguinte forma :

$$\begin{aligned}\max_{d,z} \quad & -d \\ \text{s. a} \quad & dI - M(z) \succeq 0.\end{aligned}$$

Leituras mais avançadas sobre o problema de minimizar o máximo autovalor poderão ser encontradas em [8, 23].

2.4.2 Programação Linear

Outro problema que pode ser modelado como um problema de programação semidefinida é o problema de programação linear. Um problema de programação linear é um problema que pode ser escrito na forma

$$\begin{aligned}\max \quad & b^T y \\ \text{s. a} \quad & A^T y \leq c.\end{aligned}$$

ou

$$\begin{aligned}\max \quad & b^T y \\ \text{s. a} \quad & c - A^T y \geq 0.\end{aligned}$$

Mas

$$c - A^T y \geq 0 \iff \text{Diag}(c - A^T y) \succeq 0$$

onde $\text{Diag}(v)$ é a matriz diagonal com o vetor v em sua diagonal principal. Assim, o problema de programação linear pode ser escrito como um problema de programação semidefinida do seguinte modo:

$$\begin{aligned} \max \quad & b^T y \\ \text{s. a} \quad & \text{Diag}(c - A^T y) \succeq 0. \end{aligned}$$

Pode-se concluir que programação linear é um caso particular de programação semidefinida. Isso é um fator importante, que motiva a busca de algoritmos eficientes para programação semidefinida, tanto do ponto de vista teórico como do ponto de vista computacional, pois a importância de programação linear é inquestionável.

Leituras mais avançadas sobre a relação entre PL e PSD poderão ser encontradas em [18].

2.4.3 O Problema do Corte Máximo

O problema do corte máximo ou *max-cut* é um dos problemas de otimização combinatória mais pesquisados no que diz respeito às aplicações de programação semidefinida. A maioria dos autores que trabalham com o uso da programação semidefinida em otimização combinatória trabalham também com o problema do *max-cut*.

Para entender esse problema, considere $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ um grafo, no qual $\mathcal{V} = \{1, \dots, n\}$ é o conjunto de nós e o par ordenado $(i, j) \in \mathcal{E}$ representa uma borda com extremidades i e j . Seja um subconjunto de \mathcal{V} denominado \mathcal{S} , ou seja, $\mathcal{S} \subseteq \mathcal{V}$. Um corte é o conjunto $\delta(\mathcal{S})$ de bordas tal que uma extremidade esteja em \mathcal{S} e outra em $\mathcal{V} - \mathcal{S}$.

Assim, o problema do corte máximo é o problema de, dado um subconjunto $\mathcal{S} \subseteq \mathcal{V}$, encontrar o corte $\delta(\mathcal{S})$ tal que a soma dos pesos seja máxima. Isto é,

$$\begin{aligned} \max_{\mathcal{S} \subseteq \mathcal{V}} \quad & \sum a_{ij}. \\ \text{s. a} \quad & (i, j) \in \delta(\mathcal{S}) \end{aligned} \tag{2.9}$$

Para encontrar uma formulação em programação semidefinida para (2.9), considere o problema abaixo, que é uma formulação diferente para o mesmo problema:

$$\begin{aligned} \max \quad & \frac{1}{2} \sum_{i < j} a_{ij} \cdot (1 - x_i x_j) \\ \text{s. a} \quad & x_i, x_j \in \{-1, 1\} \end{aligned} \quad (2.10)$$

Em (2.10), tem-se que $x_i = 1$ se $i \in \mathcal{S}$ e $x_i = -1$ se $i \in \mathcal{V} - \mathcal{S}$. Portanto, se $x_i = x_j$, então:

$$\frac{1 - x_i x_j}{2} = 0$$

o que significa que i, j estão no mesmo conjunto \mathcal{S} ou $\mathcal{V} - \mathcal{S}$. Por outro lado, se $x_i \neq x_j$, então:

$$\frac{1 - x_i x_j}{2} = 1$$

e isso faz com que x_i, x_j estejam cada um em um conjunto \mathcal{S} e $\mathcal{V} - \mathcal{S}$.

A restrição $i < j$ garante a existência de um único produto $x_i x_j$. Se i, j fossem quaisquer, dois produtos $x_i x_j$ seriam formados. Para não modificar o problema, divide-se por dois novamente. Assim, o problema (2.10) ficaria dividido por 4 e escrito assim:

$$\begin{aligned} \max \quad & \frac{1}{4} \sum_{i, j} a_{ij} \cdot (1 - x_i x_j) \\ \text{s. a} \quad & x_i, x_j \in \{-1, 1\} \end{aligned} \quad (2.11)$$

Separando os somatórios do problema (2.11), obtém-se a seguinte formulação:

$$\begin{aligned} \max \quad & \frac{1}{4} \sum_{i=1}^n \left(\sum_{j=1}^n a_{ij} x_i^2 - \sum_{j=1}^n a_{ij} x_i x_j \right) \\ \text{s. a} \quad & x_i, x_j \in \{-1, 1\} \end{aligned} \quad (2.12)$$

O problema (2.12) também pode ser formulado como segue:

$$\begin{aligned} \max \quad & \frac{1}{4} x^T (\text{Diag}(Ae) - A) x \\ \text{s. a} \quad & x \in \{-1, 1\}^n \end{aligned} \quad (2.13)$$

onde $A = \{a_{ij}\}$ é chamada de matriz de peso do grafo e $\text{Diag}(Ae)$ é a matriz diagonal com o vetor Ae em sua diagonal principal. O vetor e é um vetor com entradas unitárias.

Seja a matriz $L(\mathcal{G}) = \text{Diag}(Ae) - A$. Essa matriz recebe o nome de matriz de Laplace do grafo \mathcal{G} . O problema (2.13) torna-se, então:

$$\begin{aligned} \max \quad & \frac{1}{4}x^T L(\mathcal{G})x \\ \text{s. a} \quad & x \in \{-1, 1\}^n \end{aligned} \tag{2.14}$$

Considerando-se $C = \frac{1}{4}L(\mathcal{G})$, tem-se o problema:

$$\begin{aligned} \max \quad & x^T C x \\ \text{s. a} \quad & x \in \{-1, 1\}^n \end{aligned} \tag{2.15}$$

Assim, o problema de corte máximo (2.9) pode ser formulado como (2.15). Resta, no entanto, escrever (2.15) em uma formulação de programação semidefinida. Para isso, observe que $x^T C x = C \bullet X$, onde $X = x x^T$. Tem-se ainda que $\text{diag}(X) = e$, onde $\text{diag}(X)$ é a diagonal principal da matriz X . Também pode-se dizer que $\text{posto}(X) = 1$, pois toda matriz construída como o produto de um vetor com ele mesmo tem essa propriedade. Além do mais, toda matriz com essa forma é semidefinida positiva.

Logo, o problema (2.15) pode ser formulado como um problema de programação semidefinida:

$$\begin{aligned} \max \quad & C \bullet X \\ \text{s. a} \quad & X \succeq 0 \\ & \text{diag}(X) = e \\ & \text{posto}(X) = 1 \end{aligned} \tag{2.16}$$

Leituras mais avançadas sobre o problema do corte máximo poderão ser encontradas em [2, 9, 13, 25, 28].

2.5 Método de Ponto Interior *Path Following* Primal Dual

Suponha que exista solução ótima X^* do problema primal e (y^*, S^*) do problema dual de programação semidefinida, de modo que a trinca (X^*, y^*, S^*) satisfaz as duas condições abaixo:

$$A \bullet X^* = b, \quad X^* \succeq 0 \quad \Rightarrow \quad \text{Condição de Factibilidade Primal}$$

$$A^* y^* + S^* = C, \quad S^* \succeq 0 \quad \Rightarrow \quad \text{Condição de Factibilidade Dual}$$

Teorema 5.1 Teorema da Complementariedade

Se a trinca (X^*, y^*, S^*) é a solução ótima do problema de programação semidefinida, então $S^* X^* = 0$, onde 0 é a matriz nula. ■

Demonstração:

Se a trinca (X^*, y^*, S^*) é a solução ótima, pelo teorema da dualidade forte, o *gap* de dualidade é zero. Assim:

$$\begin{aligned} 0 &= C \bullet X - b^T y \\ &= C \bullet X - (A \bullet X)^T y \\ &= C \bullet X - (A^* y) \bullet X \\ &= (C - A^* y) \bullet X = S \bullet X. \end{aligned}$$

Assim, como X e S são matrizes semidefinidas positivas, tem-se

$$X \bullet S = 0 \iff SX = 0$$

Uma prova desse fato é que, como X e S são semidefinidas positivas, podem ser decompostas na forma $X = uu^T$ e $S = vv^T$, para algum u e v de tamanhos apropriados. Logo:

$$S \bullet X = (vv^T) \bullet (uu^T) = \|v^T u\|^2$$

Portanto, $v^T u = 0$. Então, $SX = vv^T uu^T = 0$ ■

Para uma trinca ser a solução ótima do problema de programação semidefinida, tem que satisfazer as condições de factibilidade primal e dual e também satisfazer o Teorema da Complementariedade.

Os métodos de ponto interior começam a busca pelo ótimo dentro do cone das matrizes semidefinidas. Para não buscar soluções fora do cone, introduz-se o chamado problema de barreira.

No problema de barreira, acrescenta-se, na função de custo, o parâmetro de barreira $\mu > 0$ e a função de barreira $-\log \det(X)$. O termo de barreira fica $-\mu \log \det(X)$ e o problema de barreira é formulado como sendo

$$\begin{aligned} \min \quad & C \bullet X - \mu \log \det(X). \\ \text{S.a} \quad & A \bullet X = b \\ & X \succeq 0 \end{aligned} \tag{2.17}$$

A idéia principal agora é transformar o problema de barreira em um problema irrestrito. Para isso, introduz-se o multiplicador lagrangeano $y \in \mathbb{R}^m$. Desse modo, o problema (2.17) se transforma em:

$$\min \ell_\mu(X, y) = C \bullet X - \mu \log \det(X) + y(b - A \bullet X). \tag{2.18}$$

O gradiente do multiplicador lagrangeano fica:

$$\begin{aligned} \nabla_X \ell_\mu(X, y) &= C - \mu X^{-1} - A^* y = 0 \\ \nabla_y \ell_\mu(X, y) &= b - A \bullet X = 0. \end{aligned} \tag{2.19}$$

Considerando-se $S = \mu X^{-1}$, obtêm-se as condições de Karush-Kuhn-Tucker para esse problema, dadas por:

$$\begin{aligned} A \bullet X &= b \\ A^* y + S &= C \\ SX &= \mu I. \end{aligned} \tag{2.20}$$

As duas primeiras expressões definem as condições de factibilidade primal e dual, respectivamente. A única condição que falta satisfazer é a condição de complementariedade. Como citado acima, a condição de complementariedade é $SX = 0$.

Logo, o problema posto agora é encontrar uma trinca (X, S, y) que satisfaça as três equações em (2.20), substituindo a condição de complementariedade por $SX = \mu I$.

Deve-se notar, no entanto, que a condição $SX = 0$ deve ser determinada a partir da condição $SX = \mu I$. Para cada valor de $\mu > 0$, existe uma única solução para (2.20). Seja (X_μ, y_μ, S_μ) essa solução. O conjunto

$$\{(X_\mu, y_\mu, S_\mu) \mid \mu > 0\}$$

é chamado de conjunto do caminho central ou *Central Path*.

Esse método, que é chamado de *Método de Ponto Interior Primal Dual Seguidor de Caminho Central*, consiste em aplicar o Método de Newton à curva do caminho central, que nada mais é do que uma curva parametrizada por μ , para $\mu \rightarrow 0$. C. Helmberg [15] prova que, quando $\mu \rightarrow 0$, os valores de X^* e S^* são encontrados.

Entretanto, para aplicar o Método de Newton para resolver as equações, necessita-se que essas matrizes sejam simétricas. Nesse fato, esbarra-se em um problema que deve ser inicialmente: a matriz $SX - \mu I$ não necessariamente é simétrica, mesmo quando S e X o são. Assim, deve-se torná-la simétrica antes de aplicar o Método de Newton.

Para isso, considere uma matriz P inversível e seja a aplicação:

$$H_P(M) = \frac{1}{2} \left[PMP^{-1} + (PMP^{-1})^T \right].$$

Em [41] mostra-se que, para $X, S \succ 0$, tem-se:

$$H_P(SX) = \mu I \Leftrightarrow SX = \mu I.$$

Dessa maneira, troca-se a equação $SX - \mu I = 0$ por $H_P(SX) - \mu I = 0$. Mas essa equação depende da escolha de P , isto é, diferentes escolhas de P produzem equações diferentes.

Algumas escolhas melhoram a facilidade de resolução da equação. Em [35, 36], M. J. Todd escreveu sobre 20 diferentes variantes que devem ser observadas na escolha de P para que se produza uma equação mais fácil de ser resolvida.

Aplicando o Método de Newton no sistema (2.20), obtem-se um novo sistema:

$$\begin{aligned} A^T \Delta y + \Delta S &= 0 \\ A \Delta X &= 0 \\ \Delta SX + S \Delta X &= 0. \end{aligned} \tag{2.21}$$

Na última equação de (2.20), tem-se $SX = \mu I$, o que implica que $\mu I - SX = 0$. Substituindo-se na última equação de (2.21), obtem-se:

$$\begin{aligned} A^T \Delta y + \Delta S &= 0 \\ A \Delta X &= 0 \\ \Delta SX + S \Delta X &= \mu I - SX. \end{aligned} \tag{2.22}$$

Substituindo-se a equação linearizada $H_P(\Delta SX + S \Delta X + SX) = \mu I$ na última equação de (2.22), pode-se computar os valores de $(\Delta X, \Delta y, \Delta S)$.

Considere $\delta(X, S, \mu)$ como a distância entre (X, y, S) e (X_μ, y_μ, S_μ) na curva de caminho central, definida por $\|S - \mu X^{-1}\|_X^*$, onde:

$$\|V\|_X^* = [(f''(X))^{-1} V \bullet V]^{1/2}$$

e $f(X) = -\ln \det(X)$. Pode-se resumir o método da seguinte maneira:

1. Tome uma solução inicial factível (X_0, y_0, S_0) . Seja $r > 0$ um raio de proximidade.
2. Tome $\mu = 1$ e faça o primeiro teste:
Se $\delta(X_0, S_0, \mu)$ for maior que r , então calcule $(\Delta X_\mu, \Delta y_\mu, \Delta S_\mu)$. Escolha $\alpha \in (0, 1]$ tal que as $X_\mu + \alpha \Delta X_\mu \succ 0$ e $S_\mu + \alpha \Delta S_\mu \succ 0$. Para essas novas matrizes, tem-se que $\delta(X_0, S_0, \mu)$ é reduzida.
3. Faça $(X_\mu + \alpha \Delta X_\mu) \bullet (S_\mu + \alpha \Delta S_\mu)$. Se o resultado for menor que um certo critério de parada $\epsilon > 0$, então pare. A solução é $(X_\mu + \alpha \Delta X_\mu, y_\mu, S_\mu + \alpha \Delta S_\mu)$. Senão, calcule $\delta(X_0, S_0, \mu)$. Caso seja maior que r , faça o mesmo do item anterior para reduzi-lo, tomando $X_\mu := X_\mu + \alpha \Delta X_\mu$ e $S_\mu := S_\mu + \alpha \Delta S_\mu$. Mas se $\delta(X_0, S_0, \mu) < r$, reduza o parâmetro μ e faça o algoritmo todo de novo, considerando a solução factível inicial como (X_μ, y_μ, S_μ) .

Leituras mais avançadas sobre Métodos de Ponto Interior para Programação Semidefinida poderão ser encontradas em [1, 14, 32, 35, 41, 42].

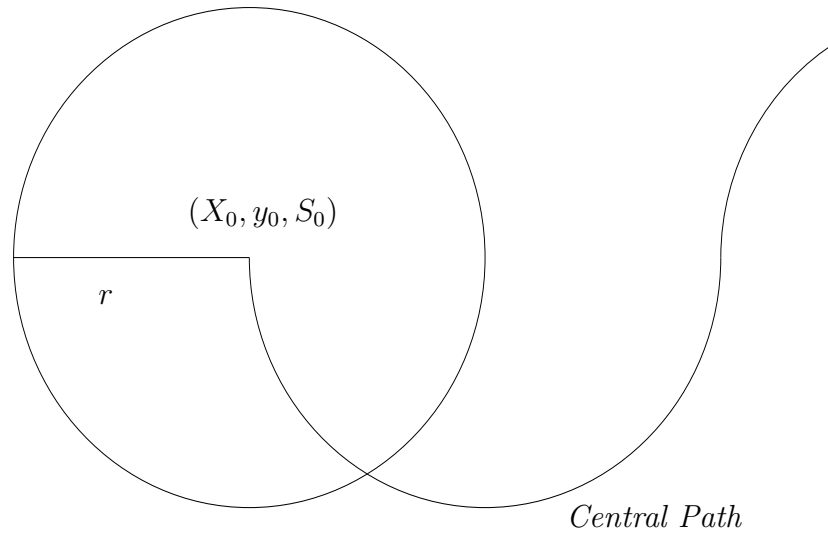


Figura 2.1: A solução inicial factível (X_0, y_0, S_0) e um raio de proximidade $r > 0$.

2.6 Conclusão

Nesse capítulo mostrou-se a teoria de PSD, acompanhada de uma revisão sucinta de álgebra linear. Também foram mostrados exemplos de problemas diversos de grande importância (como o problema do autovalor), modelados como problemas de Programação Semidefinida, além da discussão dos métodos de pontos interiores aplicados à solução de problemas de Programação Semidefinida.

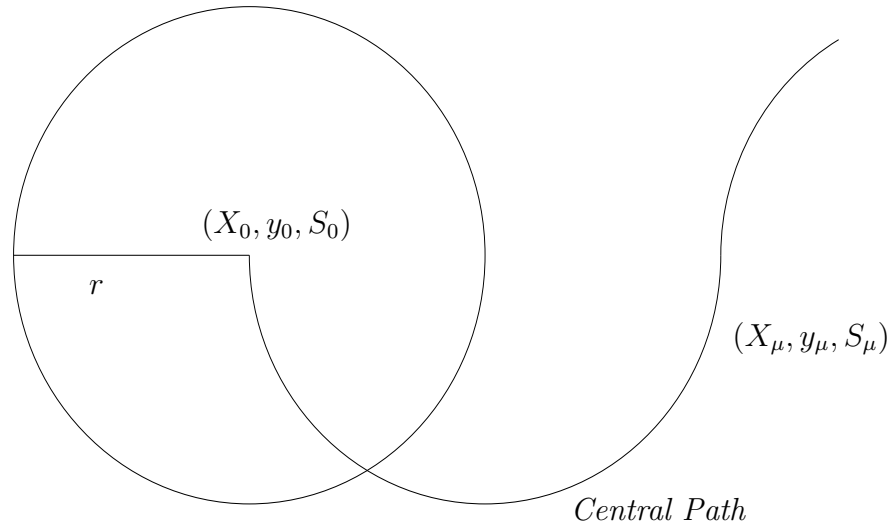


Figura 2.2: Calcula-se (X_μ, y_μ, S_μ) . Se o primeiro teste detectar $\delta(X_0, S_0, \mu) > r$, aplica-se o redutor $(\Delta X_\mu, \Delta y_\mu, \Delta S_\mu)$ e tem-se um novo ponto $(X_\mu, y_\mu, S_\mu) := (X_\mu + \alpha \Delta X_\mu, y_\mu + \alpha \Delta y_\mu, S_\mu + \alpha \Delta S_\mu)$. Esse novo ponto ainda pode estar fora do raio, aplicando-se novamente o redutor até que ele fique a uma distância menor que r do ponto inicial (X_0, y_0, S_0) .

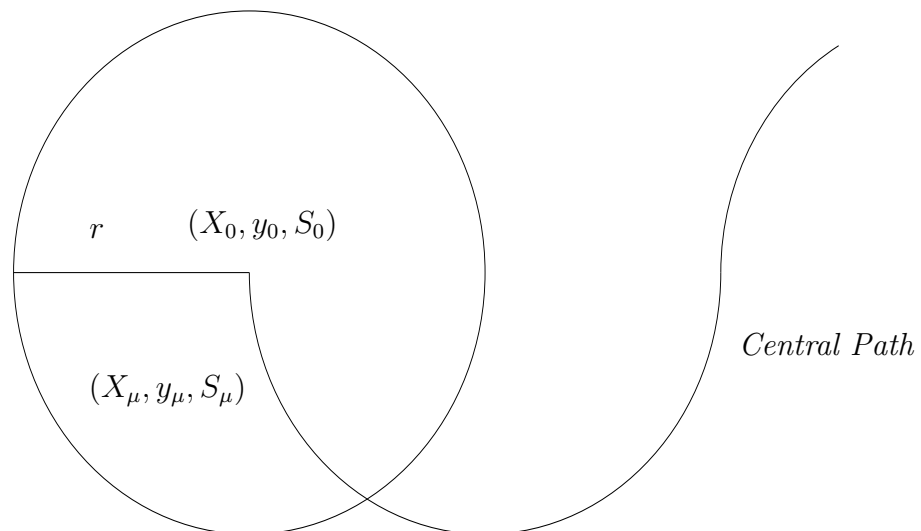


Figura 2.3: Chegando-se em (X_μ, y_μ, S_μ) , tal que $\delta(X_0, S_0, \mu) < r$, verifica-se o critério de parada $X_\mu \bullet S_\mu < \epsilon$. Caso sim, a solução do problema é (X_μ, y_μ, S_μ) .

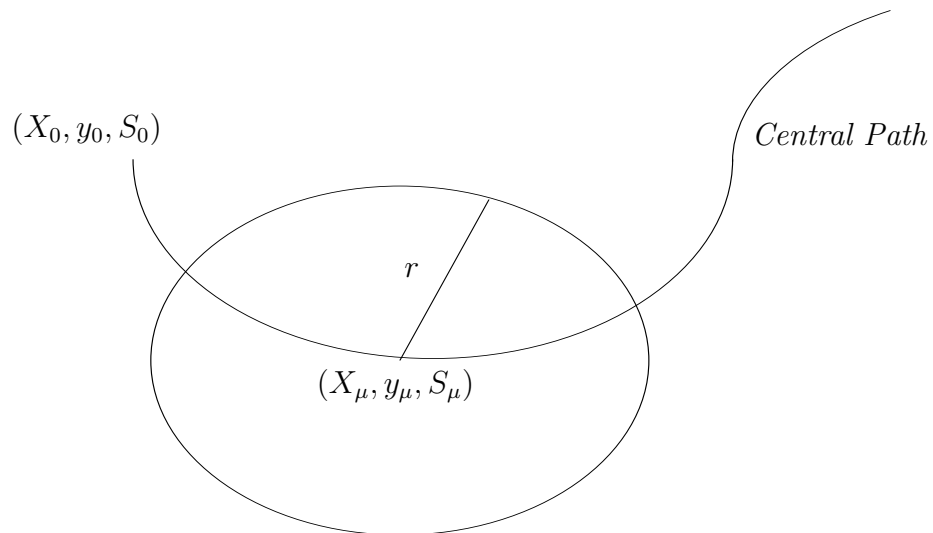


Figura 2.4: Caso (X_μ, y_μ, S_μ) não atenda o critério de parada, ele vira a solução factível inicial, reduz-se o parâmetro de barreira μ e começa a repetição toda, até que para algum μ o critério de parada seja satisfeito, tornando (X_μ, y_μ, S_μ) para esse μ a solução do problema.

Capítulo 3

Problema Linear 0-1 e os Métodos *Lift-and-Project*

3.1 Introdução

O objetivo desse capítulo é estudar os métodos *Lift-and-Project* para conseguir soluções inteiras do tipo 0-1 para algum tipo de problema de otimização linear, utilizando-se de programação semidefinida. A vantagem desses métodos é computacional, já que programação semidefinida pode ajudar a encontrar soluções inteiras de uma maneira muito mais fácil para os problemas de grande complexidade, devido ao bom desempenho computacional que fornece. Esse desempenho computacional da programação semidefinida está ligado às relaxações do problema baseado em restrições semidefinidas, que são relaxações que encontram o conjunto factível muito mais rapidamente que as relaxações tradicionais, baseadas em restrições lineares.

Primeiramente, mostra-se a relação entre otimização por restrições e otimização sobre cones. Em seguida, é introduzido o método de projeção e algumas de suas mais significativas variações. Após, é apresentado o método de *Lift-and-Project* e diversas de suas formulações. O capítulo é finalizado com uma comparação entre as formulações do método *Lift-and-Project*.

3.2 Otimização por Restrições Versus Otimização Sobre Cones

Definição 2.1 Otimização por Restrições

Um problema de otimização por restrição é um problema do tipo

$$\min\{f(x) : x \in \mathcal{Q} \subseteq \mathbb{R}^n\}$$

no qual \mathcal{Q} é o conjunto definido pelos vetores de \mathbb{R}^n que satisfazem todas as restrições do problema. ■

Assim, esse conjunto pode ser definido como:

$$\mathcal{Q} = \{x : h(x) = 0, g(x) \leq 0, x \in \mathbb{R}^n\}.$$

As principais dificuldades desse problema envolvem as restrições do tipo inequação.

Definição 2.2 Otimização sobre Cones

Um problema de Otimização sobre Cones é um problema do tipo

$$\min\{\langle c, x \rangle : Ax = b, x \in \mathcal{K}\}$$

onde \mathcal{K} é um cone, isto é, é um conjunto tal que, se $x \in \mathcal{K}$, então $\lambda x \in \mathcal{K}$ para todo $\lambda \geq 0$. ■

No caso de Programação Linear, um cone natural a ser escolhido é

$$\mathcal{K} = \{x \in \mathbb{R}^n : x \geq 0\}.$$

Já no caso de Programação Semidefinida, pode-se escolher um cone da seguinte forma:

$$\mathcal{K} = \{X \in \mathbb{R}^{n \times n} : X = X^T \succeq 0\}.$$

Os dois problemas são equivalentes quando se trata de uma função objetivo linear. Neste caso, tem-se que $f(x) = \langle c, x \rangle$.

Será apresentado, em seguida, como relacionar o cone \mathcal{K} com o conjunto de restrições \mathcal{Q} .

Sabe-se que $\mathcal{Q} \subset \mathbb{R}^n$. Pode-se fazer um isomorfismo de espaços e tem-se um subespaço em \mathbb{R}^{n+1} , que é isomorfo a \mathcal{Q} . Assim, resolver o problema nesse subespaço é resolver o problema em \mathcal{Q} . Considere o subespaço de \mathbb{R}^{n+1} como sendo

$$\mathcal{Q}' = \{(x, 1) : x \in \mathcal{Q}\}.$$

Portanto,

$$\mathcal{Q}' = \{(x, 1) : x \in \mathcal{Q}\} = \{\alpha(x, 1) : x \in \mathcal{Q}, \alpha = 1\}.$$

Considere o cone \mathcal{K} em \mathbb{R}^{n+1} da seguinte maneira :

$$\mathcal{K} = \{(\alpha x, \alpha) : x \in \mathcal{Q}, \alpha \geq 0\}.$$

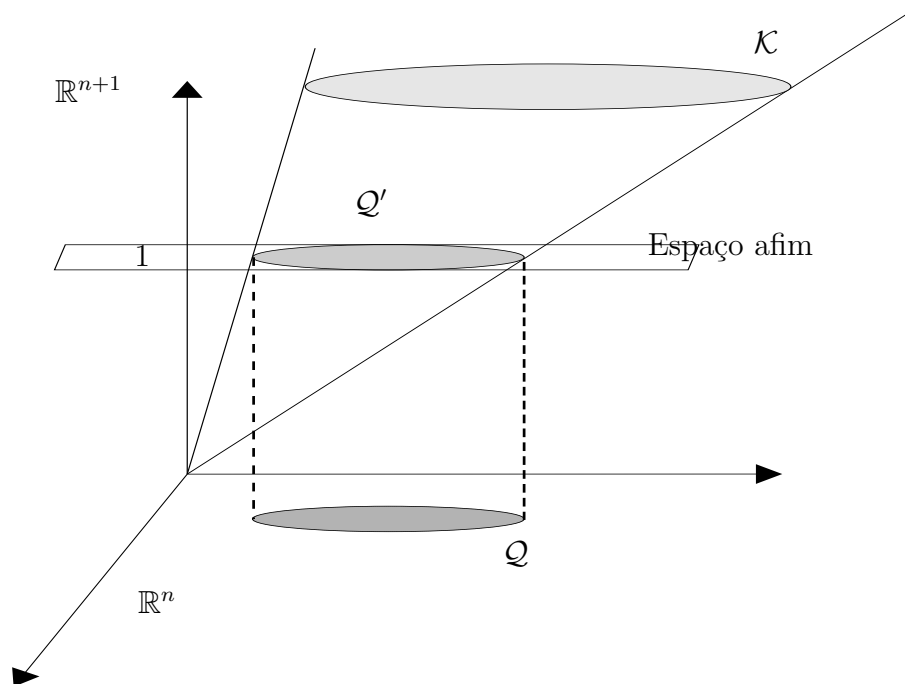


Figura 3.1: $\mathcal{Q}' = \text{espaço afim} \cap \mathcal{K}$

Assim,

$$\{(\alpha x, \alpha) : x \in \mathcal{Q}, \alpha \geq 0\} \cap \{\alpha(x, 1) : x \in \mathcal{Q}, \alpha = 1\} = \mathcal{Q}'.$$

Seja o conjunto $\{\alpha(x, 1) : x \in \mathcal{Q}, \alpha = 1\}$ o espaço afim e ele é isomorfo a \mathbb{R}^n . Pode-se dizer então que:

$$\mathcal{Q}' = \text{espaço afim} \cap \mathcal{K}.$$

Leituras mais avançadas sobre a relação entre otimização por restrições e otimização sobre cones poderão ser encontradas em [41].

3.3 Método de Projeção

Definição 3.1 Projeção de um poliedro em um espaço de menor dimensão

Seja um poliedro $\mathcal{P} = \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^p : Ax + Gy \leq b\}$. A projeção de \mathcal{P} em \mathbb{R}^n é o conjunto dado por:

$$\text{Proj}_x(\mathcal{P}) = \{x \in \mathbb{R}^n \mid \exists y \in \mathbb{R}^p : (x, y) \in \mathcal{P}\}.$$

■

Para se determinar essa projeção, serão apresentados, em seguida, dois algoritmos desenvolvidos para esse fim.

3.3.1 Algoritmo de Fourier - Motzkin

Fourier [10] já havia descoberto um algoritmo, no início do século XX, para eliminar uma variável em um espaço vetorial e escrever os vetores como vetores do espaço anterior. Esse algoritmo foi melhorado por Motzkin [30].

O núcleo dos dois artigos é a prova do teorema abaixo, que aqui será apenas enunciado, ficando a prova fora dos objetivos desse texto.

Teorema 3.2 Teorema de Fourier-Motzkin

Dado um poliedro \mathcal{P} em \mathbb{R}^{n+1} definido pelas inequações:

$$\begin{cases} a^k x + y \leq a_0^k, & k = 1, \dots, m_k \\ b^l x - y \leq b_0^l, & l = 1, \dots, m_l \\ c^h x \leq c_0^h, & h = 1, \dots, m_h \end{cases}$$

a projeção de \mathcal{P} sobre \mathbb{R}^n é dada pelos vetores de \mathbb{R}^n que satisfazem o sistema:

$$\begin{cases} b^l x - b_0^l \leq a_0^k - a^k x, & l = 1, \dots, m_l, \quad k = 1, \dots, m_k \\ c^h x \leq c_0^h, & h = 1, \dots, m_h. \end{cases}$$

■

Observe que $\text{Proj}_x(\mathcal{P})$ também é um poliedro, já que é um conjunto de vetores que satisfazem um número finito de inequações lineares. Observe também que esse método elimina uma variável a cada iteração.

3.3.2 Algoritmo de Balas - Pulleyblank

Outro algoritmo para encontrar a projeção $\text{Proj}_x(\mathcal{P})$ é o Algoritmo de eliminação de variáveis de Balas - Pulleyblank [6]. Analogamente ao algoritmo de Fourier-Motzkin, um dos objetivos centrais desse artigo é a prova do teorema abaixo:

Teorema 3.3 Teorema de Balas - Pulleyblank

A projeção de $\mathcal{P} = \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^p : Ax + Gy \leq b\}$ é dada por:

$$\text{Proj}_x(\mathcal{P}) = \{x \in \mathbb{R}^n : v_t(b - Ax) \geq 0\}$$

para $t = 1, \dots, k$, onde $\{v_t\}$ são os raios extremos de

$$\mathcal{Q} = \{v \in \mathbb{R}_+^m : vG = 0\}.$$

■

O conjunto \mathcal{Q} é chamado de cone de projeção de \mathcal{P} . Esse método, diferentemente do algoritmo de Fourier-Motzkin, permite a eliminação de várias variáveis na mesma iteração.

3.4 Métodos *lift-and-project*

Os métodos *Lift-and-Project* constróem relaxações através do mecanismo de *lift*, que significa **Elevação**, e os planos de corte para essas relaxações com o mecanismo de *Project*, que significa **Projeção** [27].

Para que se possa compreender o funcionamento do método *lift-and-project* para programação linear inteira do tipo 0/1, suponha que se queira resolver o problema abaixo:

$$\begin{aligned} \max \quad & c^T x \\ \text{s. a} \quad & Ax \leq b \\ & x \in \{0, 1\}^n. \end{aligned} \tag{3.1}$$

Pode-se reformular o problema (3.1) da seguinte maneira:

$$\begin{aligned} \max \quad & c^T x \\ \text{s. a} \quad & x \in \mathcal{P}. \end{aligned} \tag{3.2}$$

onde

$$\mathcal{P} = \text{conv}(\{x \in \{0, 1\}^n : Ax \leq b\}).$$

Dado o conjunto $\mathcal{Q} \subseteq \{0, 1\}^n$ das restrições do problema (3.1), deseja-se encontrar uma descrição linear para o politopo $\mathcal{P} = \text{conv}(\mathcal{Q})$. Isto é, deseja-se encontrar um sistema linear $Ax \leq b$, que construa o politopo $\mathcal{K} = \{x \in \mathbb{R}^n : Ax \leq b\}$. Esse politopo \mathcal{K} terá que satisfazer à condição

$$\mathcal{K} \cap \{0, 1\}^n = \mathcal{Q}.$$

Se todos os vértices de \mathcal{K} são inteiros, então $\mathcal{P} = \mathcal{K}$. Caso contrário, deseja-se encontrar planos de corte que permitam apertar a relaxação \mathcal{K} , isto é, retirar do poliedro \mathcal{K} os vértices que não são 0 – 1. Assim, encontra-se \mathcal{P} após um número finito de iterações.

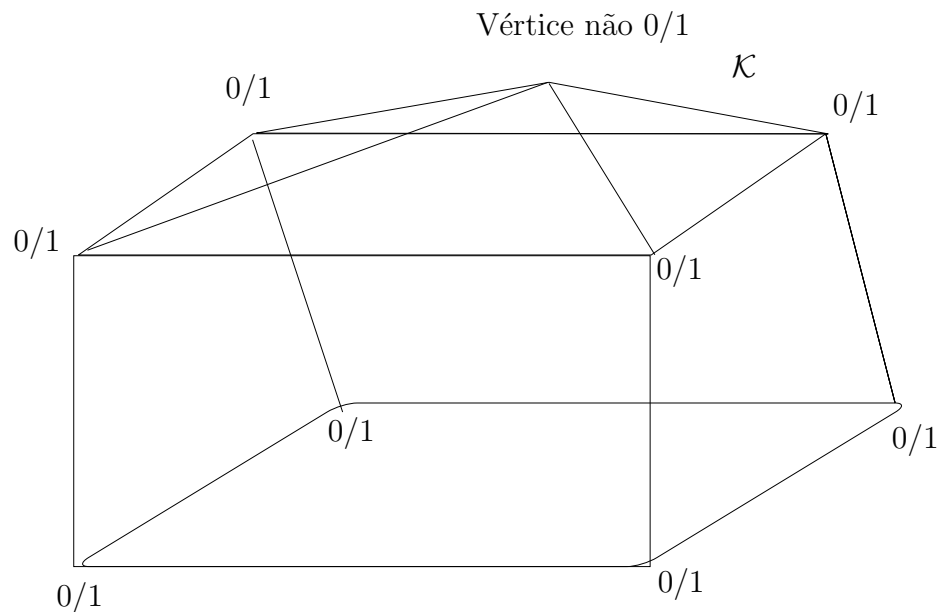


Figura 3.2: A relaxação \mathcal{K} e um vértice não 0/1

Voltando ao problema (3.2), uma relaxação inicial para \mathcal{P} é o conjunto $\mathcal{K} = \{x \in \mathbb{R}^n : Ax \leq b\}$.

Deve-se verificar se o conjunto $\mathcal{K} = \mathcal{P}$. Caso $\mathcal{K} \neq \mathcal{P}$, deseja-se encontrar planos de corte que permitam retirar, da relaxação \mathcal{K} , vértices não-inteiros.

Os métodos lift-and-project consistem em representar \mathcal{P} como a projeção de um politopo \mathcal{R} encontrado em um espaço de maior dimensão que a dimensão do politopo \mathcal{P} . Os principais métodos para construção dessa projeção para encontrar politopos do tipo 0/1 são:

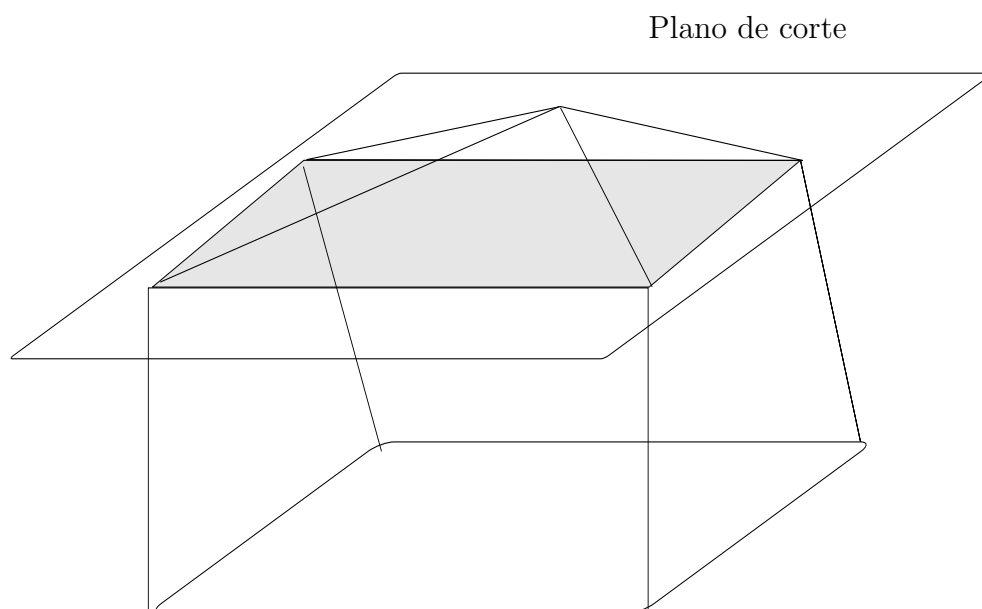


Figura 3.3: O plano de corte aperta a relaxação \mathcal{K} e retira o vértice não 0/1

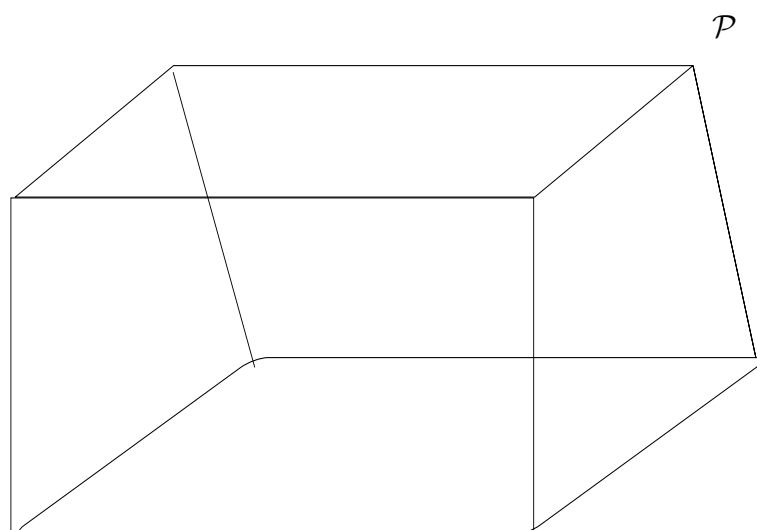


Figura 3.4: Encontrado \mathcal{P} , cujos vértices são 0/1

- Método BCC, conhecido como *lift-and-project* para BCC, desenvolvido por Balas, Ceria e Cornuéjols [5].
- Método SA, conhecido como Técnica de Reformulação e Linearização para SA, desenvolvido por Sherali e Adams [33, 34].
- Método LS, desenvolvido por Lovász e Schrijver e conhecido como Cortes de Matrizes para LS [24].
- Método de Lasserre, desenvolvido por Jean Lasserre [20, 21, 22].

Uma característica comum desses métodos é a construção da hierarquia $\mathcal{P} \subseteq \mathcal{K}_n \subseteq \dots \subseteq \mathcal{K}_1 \subseteq \mathcal{K}$, onde \mathcal{K} é a relaxação inicial de \mathcal{P} . Encontram-se, por meio desses métodos, os conjuntos $\mathcal{K}_1, \mathcal{K}_2, \dots, \mathcal{K}_n$, de tal forma que o conjunto \mathcal{P} seja encontrado em um número finito de iterações. Será mostrado que o método de Lasserre encontra o conjunto \mathcal{P} de forma mais rápida e com menor trabalho computacional, quando é comparada a sua velocidade de convergência com a velocidade de convergência dos outros métodos. Esse é o grande motivo de se usar relaxação semidefinida para resolver problemas de soluções inteiras [26].

Será visto agora como funciona a construção de cada uma das relaxações citadas.

3.4.1 Método Balas-Ceria-Cornuéjols

Sabe-se que $\mathcal{K} = \{x \in \mathbb{R}^n : Ax \leq b\}$ e $\mathcal{P} = \text{conv}(\mathcal{K} \cap \{0, 1\}^n)$ é o polítopo 0/1 cuja descrição linear é o que se procura. Assume-se que as restrições de contorno $0 \leq x_i \leq 1$, com $i = 1, \dots, n$, estão presentes. Reescreve-se os sistemas $Ax \leq b$ e $0 \leq x_i \leq 1$ com $i = 1, \dots, n$ como o sistema $\tilde{A}x \leq \tilde{b}$ e denomina-se o número de linhas da matriz A de m .

Será visto agora como funciona a construção do Método Balas-Ceria-Cornuéjols, também conhecido como BCC. Primeiramente, fixa-se um valor de $j \in \{1, \dots, n\}$ e logo depois multiplica-se o sistema $\tilde{A}x \leq \tilde{b}$ por x_j e $1 - x_j$. Assim, tem-se um sistema não-linear, a saber,

$$x_j (\tilde{A}x - \tilde{b}) \leq 0 \quad \text{e} \quad (1 - x_j) (\tilde{A}x - \tilde{b}) \leq 0. \quad (3.3)$$

Lineariza-se esse sistema, trocando x_j^2 por x_j e introduzindo novas variáveis $y_i = x_i x_j$, com $i = 1, \dots, n$ e $i \neq j$. Percebe-se que $y_j = x_j$.

O novo sistema linear é

$$\left(\tilde{A}y - \tilde{b}x_j \right) \leq 0 \quad \text{e} \quad \tilde{A}(x - y) - \tilde{b}(1 - x_j) \leq 0. \quad (3.4)$$

Esse novo sistema tem $2(m + 2n)$ inequações. Isso porque a matriz A tem m inequações e as restrições de contorno tem $2n$ inequações. Logo, cada um dos sistemas acima tem $m + 2n$ inequações. Sendo dois sistemas, tem-se $2(m + 2n)$ inequações.

O conjunto dos vetores (x, y) factíveis para essas $2(m + 2n)$ inequações definem um politopo no espaço (x, y) de dimensão maior que n . Chama-se esse conjunto no espaço (x, y) de $\mathcal{M}_j(\mathcal{K})$. Fazendo a projeção de $\mathcal{M}_j(\mathcal{K})$ no espaço \mathbb{R}^n , ou seja, no espaço original x , a projeção encontrada é $\mathcal{P}_j(\mathcal{K})$.

Tem-se que $\mathcal{P}_j(\mathcal{K}) \subseteq \mathcal{K}$, pois se $x \in \mathcal{P}_j(\mathcal{K})$, então existe y , de dimensão apropriada, tal que $(x, y) \in \mathcal{M}_j(\mathcal{K})$. Mas isso significa que x satisfaz as inequações (3.3). Como $1 - x_j \geq 0$ e $x_j \geq 0$ são condições de contorno para esse problema, tem-se que $\tilde{A}x \leq \tilde{b}$, o que implica $x \in \mathcal{K}$.

Teorema 4.1

Se \mathcal{K} , \mathcal{P} e $\mathcal{M}_j(\mathcal{K})$ são definidos como acima, então:

$$\mathcal{P}_j(\mathcal{K}) = \text{conv}(\mathcal{K} \cap \{x \in \mathbb{R}^n : x_j \in \{0, 1\}\}).$$

■

Demonstração:

Supõe-se que $x \in \text{conv}(\mathcal{K} \cap \{x : x_j = 0, 1\})$. Quer se provar que $x \in \mathcal{P}_j(\mathcal{K})$. Para isso, pode-se escrever x como sendo

$$x = \alpha u + (1 - \alpha)v$$

para algum $0 \leq \alpha \leq 1$, com $u, v \in \mathcal{K}$ sendo que $u_j = 1$ e $v_j = 0$.

Chama-se $y = \alpha u$ e consideramos $\alpha = x_j$.

Lembre-se que (3.4) é o sistema que gera $\mathcal{M}_j(\mathcal{K})$. Os valores de x e y definidos acima satisfazem esse sistema. O que implica que $(x, y) \in \mathcal{M}_j(\mathcal{K})$, fazendo com o que $x \in \mathcal{P}_j(\mathcal{K})$, uma vez que $\mathcal{P}_j(\mathcal{K})$ é a projeção de $\mathcal{M}_j(\mathcal{K})$ no espaço da variável x .

Supõe-se agora que $x \in \mathcal{P}_j(\mathcal{K})$. Quer se provar que $x \in \text{conv}(\mathcal{K} \cap \{x : x_j = 0, 1\})$. Separa-se essa prova em duas partes:

Em primeiro lugar, suponha que $\mathcal{K} \cap \{x : x_j = 0\} = \emptyset$. Isto é, os vetores de \mathcal{K} não têm entradas x_j nulas. Assim, $0 < x_j \leq 1$. Portanto $1 - x_j \geq 0$ e $x_j - \varepsilon \geq 0$, para algum $\varepsilon > 0$. Logo, tem-se:

$$(1 - x_j)(x_j - \varepsilon) \geq 0 \iff x_j - \varepsilon - x_j^2 + x_j\varepsilon \geq 0.$$

Fazendo $x_j^2 = x_j$, obtém-se:

$$\varepsilon(x_j - 1) \geq 0 \Rightarrow (x_j - 1) \geq 0 \Rightarrow x_j \geq 1.$$

Mas as condições de contorno dizem que $x_j \leq 1$, o que implica que $x_j = 1$. Portanto, se não existem vetores em \mathcal{K} cuja entrada $x_j = 0$, conclui-se que essa entrada obrigatoriamente será $x_j = 1$. Essas inequações utilizadas acima são válidas para $\mathcal{M}_j(\mathcal{K})$ e $\mathcal{P}_j(\mathcal{K})$, implicando que $x \in \mathcal{P}_j(\mathcal{K})$. Mas já foi provado que $\mathcal{P}_j(\mathcal{K}) \subseteq \mathcal{K}$. Mais ainda, $x \in \mathcal{K}$ e $x \in \{x : x_j \in \{0, 1\}\}$. Logo, $x \in \text{conv}(\mathcal{K} \cap \{x : x_j = 0, 1\})$.

A conclusão obtida acima é que, se nenhum $x \in \mathcal{K}$ tem entrada x_j nula, essa entrada obrigatoriamente será $x_j = 1$. Pode-se concluir também o contrário, isto é, se nenhum $x \in \mathcal{K}$ tem entrada $x_j = 1$, essa entrada obrigatoriamente será nula. Isso porque, suponha $\mathcal{K} \cap \{x : x_j = 1\} = \emptyset$. Assim, existe um $\varepsilon > 0$ tal que $x_j + \varepsilon \leq 1$. Multiplicando a inequação acima por x_j , consegue-se $x_j^2 + \varepsilon x_j \leq x_j$. Fazendo $x_j^2 = x_j$, obtém-se $\varepsilon x_j \leq 0$. Como $\varepsilon > 0$, conclui-se que $x_j \leq 0$. Mas, pelas condições de contorno, $x_j \geq 0$, concluindo que $x_j = 0$. De maneira análoga, essas inequações foram utilizadas acima pois suponha-se que $x \in \mathcal{P}_j(\mathcal{K})$ e, a partir desse fato, conclui-se que $x \in \text{conv}(\mathcal{K} \cap \{x : x_j = 0, 1\})$.

Em segundo e último lugar, suponha que, se $x \in \mathcal{K}$ pode ter entrada $x_j = 1$ ou nula, então, se $x \in \mathcal{P}_j(\mathcal{K})$, tem-se $x \in \text{conv}(\mathcal{K} \cap \{x : x_j = 0, 1\})$. Essa prova é feita mostrando que toda inequação $\alpha x \leq \beta$ válida para $\mathcal{P}_j(\mathcal{K})$ é válida para $\text{conv}(\mathcal{K} \cap \{x : x_j = 0, 1\})$.

Assim, $\alpha x \leq \beta$ é válida para $\text{conv}(\mathcal{K} \cap \{x : x_j = 0, 1\})$ se e somente se $\alpha x \leq \beta$ é válida para $\text{conv}(\mathcal{K} \cap \{x : x_j = 0\})$ ou $\alpha x \leq \beta$ é válida para $\text{conv}(\mathcal{K} \cap \{x : x_j = 1\})$.

Separam-se esses casos, na forma:

- $\alpha x \leq \beta$ é válida para $\text{conv}(\mathcal{K} \cap \{x : x_j = 0\})$ se e somente se $\exists \lambda \geq 0$ tal que $\alpha x + \lambda x_j \leq \beta$ é válida para \mathcal{K} .

- $\alpha x \leq \beta$ é válida para $\text{conv}(\mathcal{K} \cap \{x : x_j = 1\})$ se e somente se $\exists \mu \geq 0$ tal que $\alpha x + \mu(1 - x_j) \leq \beta$ é válida para \mathcal{K} .

Multiplica-se as inequações $\alpha x + \lambda x_j - \beta \leq 0$ e $\alpha x + \mu(1 - x_j) - \beta \leq 0$ por $1 - x_j$ e x_j , respectivamente, obtendo-se:

$$(1 - x_j)(\alpha x + \lambda x_j - \beta) \leq 0 \quad \text{e} \quad x_j[\alpha x + \mu(1 - x_j) - \beta] \leq 0.$$

Somam-se essas inequações e obtém-se $\alpha x + (\lambda + \mu)(x_j - x_j^2) - \beta \geq 0$. Como supõe-se que $x \in \mathcal{P}_j(\mathcal{K})$, tem-se que $x_j^2 = x_j$. Logo, $\alpha x \leq \beta$ também em $\mathcal{P}_j(\mathcal{K})$. ■

Portanto, o conjunto $\mathcal{P}_j(\mathcal{K})$ satisfaz à condição

$$\mathcal{P} \subseteq \mathcal{P}_j(\mathcal{K}) \subseteq \mathcal{K}.$$

Isso porque todo elemento de $\mathcal{P}_j(\mathcal{K})$ é elemento de \mathcal{K} , já provado acima, e, ao mesmo tempo, todo elemento de \mathcal{P} é elemento de $\mathcal{P}_j(\mathcal{K})$, já que $\mathcal{P}_j(\mathcal{K}) = \text{conv}(\mathcal{K} \cap \{x \in \mathbb{R}^n : x_j \in \{0, 1\}\})$, pelo teorema acima.

A iteração é definida por:

$$\mathcal{P}_{j_1, \dots, j_t}(\mathcal{K}) = \mathcal{P}_{j_t}(\mathcal{P}_{j_{t-1}} \dots (\mathcal{P}_{j_1}(\mathcal{K}))).$$

Prova-se também que

$$\mathcal{P}_{j_1, \dots, j_t}(\mathcal{K}) = \text{conv}(\mathcal{K} \cap \{x : x_{j_i} \in \{0, 1\}, i = \{1, \dots, t\}\})$$

fazendo indução sobre j_i .

A seguinte relação de inclusão é obtida:

$$\mathcal{P} \subseteq \mathcal{P}_{j_1, \dots, j_n}(\mathcal{K}) \subseteq \mathcal{P}_{j_1, \dots, j_{n-1}}(\mathcal{K}) \subseteq \dots \subseteq \mathcal{P}_{j_1}(\mathcal{K}) \subseteq (\mathcal{K})$$

e $\mathcal{P} = \mathcal{P}_{j_1, \dots, j_n}(\mathcal{K})$.

Assim, esse método fornece o polítopo \mathcal{P} de todos os pontos factíveis do problema inicial de soluções inteiras do tipo 0/1 em n iterações. Agora, basta resolver o problema (3.2) sobre esse polítopo.

3.4.2 Método Sherali-Adams

Como anteriormente, sabe-se que $\mathcal{K} = \{x \in \mathbb{R}^n : Ax \leq b\}$ e o politopo 0/1 cuja descrição linear se procura é $\mathcal{P} = \text{conv}(\mathcal{K} \cap \{0, 1\}^n)$. Assume-se que as restrições de contorno $0 \leq x_i \leq 1$, com $i = 1, \dots, n$, estão presentes em \mathcal{P} , mas não em \mathcal{K} . Assim, tem-se:

$$\mathcal{P} = \left\{ x \in \mathbb{R}^n : \begin{array}{l} \sum_{j=1}^n \alpha_{rj} x_j \geq \beta_r, \quad r = 1, \dots, R_1, \\ \sum_{j=1}^n a_{rj} x_j = b_r, \quad r = 1, \dots, R_2, \quad 0 \leq x \leq e, \quad x \in \mathbb{Z}^n \end{array} \right\}$$

e

$$\mathcal{K} = \left\{ x \in \mathbb{R}^n : \sum_{j=1}^n \alpha_{rj} x_j \geq \beta_r, \quad r = 1, \dots, R_1, \quad \sum_{j=1}^n a_{rj} x_j = b_r, \quad r = 1, \dots, R_2 \right\}$$

onde $e \in \{1\}^n$.

Será mostrado agora como construir as relaxações de \mathcal{P} usando vários fatores de multiplicação, que são polinômios de grau d . Caso $d = 0$, assume-se que a relaxação de \mathcal{P} seja \mathcal{K} . Para $d = 1, \dots, n$, define-se o fator polinomial de multiplicação como sendo:

$$F_d(\mathcal{J}_1, \mathcal{J}_2) = \prod_{j \in \mathcal{J}_1} x_j \cdot \prod_{j \in \mathcal{J}_2} (1 - x_j),$$

para todo par de conjuntos $(\mathcal{J}_1, \mathcal{J}_2)$, onde \mathcal{J}_1 e $\mathcal{J}_2 \subseteq \mathcal{V} = \{1, \dots, n\}$, com interseção vazia, ou seja, $\mathcal{J}_1 \cap \mathcal{J}_2 = \emptyset$ e que a união tenha cardinalidade, isto é, o número de elementos de $\mathcal{J}_1 \cup \mathcal{J}_2 = d$. Se os conjuntos \mathcal{J}_1 e \mathcal{J}_2 satisfazem essas condições, então usa-se a terminologia $(\mathcal{J}_1, \mathcal{J}_2)$ tem ordem d . Logo, $F_d(\mathcal{J}_1, \mathcal{J}_2) \geq 0$ para todo $d = 1, \dots, n$ e $x \in \{0, 1\}^n$.

Será visto agora como encontra-se a relaxação de \mathcal{P} na iteração onde se quer restringir a entrada x_d do vetor x para o conjunto $\{0, 1\}$.

Multiplica-se o sistema $Ax \leq b$ por cada fator $F_d(\mathcal{J}_1, \mathcal{J}_2)$, para todo par de conjuntos $(\mathcal{J}_1, \mathcal{J}_2)$ de ordem d . Obtém-se, assim, um sistema não-linear. O primeiro passo para linearizar esse sistema é fazer, primeiramente, $x_j^2 = x_j$, obtendo-se a

relação $x_j(1 - x_j) = 0$. Até aqui, tem-se um sistema de inequações ainda não-lineares de grau menor ou igual a d . Acrescenta-se ainda os polinômios de grau $D = \min\{d + 1, n\}$, o qual chama-se $F_D(\mathcal{J}_1, \mathcal{J}_2) \geq 0$.

Para finalmente linearizar esse sistema, troca-se a variável $\prod_{j \in \mathcal{J}} x_j$ por y_j , para todo conjunto \mathcal{J} com cardinalidade $|\mathcal{J}| \geq 2$. Obtem-se um poliedro $\mathcal{R}_d(\mathcal{K})$, definido no espaço (x, w) como sendo:

$$\mathcal{R}_d(\mathcal{K}) = \left\{ (x, y) : \sum_{j \in \mathcal{J}_1} (\alpha_{rj} - \beta_r) f_d(\mathcal{J}_1, \mathcal{J}_2) + \sum_{j \in V - (\mathcal{J}_1 \cup \mathcal{J}_2)} \alpha_{rj} f_{d+1}(\mathcal{J}_1 + j, \mathcal{J}_2) \geq 0, \quad r = 1, \dots, R_1, \right. \\ \left. \sum_{j \in \mathcal{J}_1} (a_{rj} - b_r) f_d(\mathcal{J}_1, \mathcal{J}_2) + \sum_{j \in V - (\mathcal{J}_1 \cup \mathcal{J}_2)} a_{rj} f_{d+1}(\mathcal{J}_1 + j, \mathcal{J}_2) = 0, \right. \\ \left. r = 1, \dots, R_2, f_D(\mathcal{J}_1, \mathcal{J}_2) \geq 0 \right\} \quad (3.5)$$

onde $f_d(\mathcal{J}_1, \mathcal{J}_2)$ é a expressão linear obtida fazendo-se as linearizações em (x, y) quando multiplicada por um fator $F_d(\mathcal{J}_1, \mathcal{J}_2)$. Essa multiplicação acima é para cada $(\mathcal{J}_1, \mathcal{J}_2)$ de ordem d .

Fazendo-se a projeção de $\mathcal{R}_d(\mathcal{K})$ sobre o espaço x original, obtem-se $\mathcal{S}_d(\mathcal{K})$. O teorema abaixo prova que $\mathcal{S}_d(\mathcal{K})$ é uma relaxação de \mathcal{P} . Fazendo d variar entre 1 e n , percorre-se todas as entradas do vetor x , fazendo com que, depois de construir todas as relaxações $\mathcal{R}_d(\mathcal{K})$, o poliedro \mathcal{P} seja encontrado após n construções de $\mathcal{R}_d(\mathcal{K})$.

Teorema 4.2

Para $\mathcal{S}_d(\mathcal{K})$, \mathcal{P} e \mathcal{K} definidos como acima, a relação de inclusão

$$\mathcal{P} \subseteq \mathcal{S}_{1, \dots, n}(\mathcal{K}) \subseteq \mathcal{S}_{1, \dots, n-1}(\mathcal{K}) \subseteq \dots \subseteq \mathcal{S}_1(\mathcal{K}) \subseteq \mathcal{K},$$

onde $\mathcal{S}_{1, \dots, d}(\mathcal{K}) = \mathcal{S}_d(\mathcal{S}_{1, \dots, d-1}(\mathcal{K}))$ para $d = 1, \dots, n$ é válida. ■

Demonstração:

Separa-se essa demonstração em três partes. A primeira é a demonstração de que, para cada iteração, um conjunto menor é encontrado. Ou seja, quer se mostrar que $\mathcal{S}_{1, \dots, d}(\mathcal{K}) \subseteq \mathcal{S}_{1, \dots, d-1}(\mathcal{K})$. Para isso, basta mostrar que todo $x \in \mathcal{S}_{1, \dots, d}(\mathcal{K})$ também é um elemento de $\mathcal{S}_{1, \dots, d-1}(\mathcal{K})$. Sendo assim, basta mostrar que o par (x, y) satisfaz as restrições de $\mathcal{R}_{d-1}(\mathcal{S}_{1, \dots, d-2}(\mathcal{K}))$ sempre que $(x, y) \in \mathcal{R}_d(\mathcal{S}_{1, \dots, d-1}(\mathcal{K}))$.

Para isso, considere uma inequação $\sum_{j=1}^n \alpha_{rj} x_j \geq \beta_r, r = 1, \dots, R_1$. Considere também $(\mathcal{J}_1, \mathcal{J}_2)$ de ordem $d - 1$. Seja $\mathcal{J}_3 = V - (\mathcal{J}_1 \cup \mathcal{J}_2)$. Então, para qualquer $k \in \mathcal{J}_3$, tem-se:

$$\sum_{j \in \mathcal{J}_1} (\alpha_{rj} - \beta_r) f_d(\mathcal{J}_1, \mathcal{J}_2 + k) + \sum_{j \in (\mathcal{J}_3 - k)} \alpha_{rj} f_{d+1}(\mathcal{J}_1 + j, \mathcal{J}_2 + k) \geq 0$$

e

$$\sum_{j \in \mathcal{J}_1} (\alpha_{rj} - \beta_r) f_d(\mathcal{J}_1 + k, \mathcal{J}_2) + \alpha_{r,k} f_d(\mathcal{J}_1 + k, \mathcal{J}_2) + \sum_{j \in \mathcal{J}_3 - k} \alpha_{rj} f_{d+1}(\mathcal{J}_1 + k + j, \mathcal{J}_2) \geq 0.$$

Essas duas inequações são inequações de $\mathcal{R}_d(\mathcal{S}_{1, \dots, d-1}(\mathcal{K}))$. Portanto, (x, y) satisfazem essas inequações.

Somando essas duas inequações, obtem-se

$$\left(\sum_{j \in \mathcal{J}_1} \alpha_{rj} - \beta_r \right) f_{d-1}(\mathcal{J}_1, \mathcal{J}_2) + \sum_{j \in \mathcal{J}_3} \alpha_{rj} f_d(\mathcal{J}_1 + j, \mathcal{J}_2) \geq 0$$

que é uma restrição de $\mathcal{R}_{d-1}(\mathcal{S}_{1, \dots, d-2}(\mathcal{K}))$. Logo (x, y) satisfazem essas inequações. Analogamente a prova é válida para as outras restrições.

A segunda parte da demonstração é provar que $\mathcal{S}_d(\mathcal{K}) \subseteq \mathcal{K}$ para todo d . Isso é mais fácil, pois se $x \in \mathcal{S}_d(\mathcal{K})$ significa que existe y tal que $(x, y) \in \mathcal{R}_d(\mathcal{K})$. Desfazendo todas as linearizações e multiplicações feitas sobre \mathcal{K} para se encontrar $\mathcal{R}_d(\mathcal{K})$, conclui-se que $x \in \mathcal{K}$.

A última parte da demonstração é provar que $\mathcal{P} \subseteq \mathcal{S}_{1, \dots, n}(\mathcal{K})$. Seja $x \in \mathcal{P}$. Defina $y_{\mathcal{J}} = \prod_{j \in \mathcal{J}} x_j$, com $|\mathcal{J}| = n$. Por construção, $(x, y) \in \mathcal{R}_{1, \dots, n}(\mathcal{K})$, logo $x \in \mathcal{S}_{1, \dots, n}(\mathcal{K})$. ■

A prova que $\mathcal{P} = \mathcal{S}_{1, \dots, n}(\mathcal{K})$ necessita de mais lemas e teoremas, ficando portanto, fora dos objetivos desse trabalho, podendo ser encontrada em [33].

3.4.3 Método de Lovász-Schrijver

O método de Lovász-Schrijver introduz uma matriz variável definida por

$$Y = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 & x^T \end{pmatrix},$$

onde $x \in \{0, 1\}^n$.

Antes de se mostrar o método, é interessante mostrar algumas propriedades da matriz Y :

- Y é simétrica
- $y_{jj} = y_{0j}$ para $j = 1, \dots, n$. É equivalente à $\text{diag}(Y) = Y e_0$, onde e_i é o vetor da base canônica de \mathbb{R}^{n+1} , com entrada i sendo 1, para $0 \leq i \leq n$.

Essa última propriedade significa que a diagonal da matriz Y é igual à primeira coluna de Y .

Como anteriormente, tem-se que \mathcal{P} como o conjunto de soluções factíveis inteiras do tipo 0/1 do sistema linear $Ax \leq b$ e \mathcal{K} o conjunto de soluções factíveis reais do mesmo sistema $Ax \leq b$, com condições de contorno $0 \leq x \leq e$.

O primeiro passo para a construção de relaxações de \mathcal{P} é multiplicar o sistema $Ax \leq b$ por algum x_i , resultando em $bx_i - Axx_i \geq 0$. Essa inequação é não linear, e pode ser formulada em termos da matriz variável Y como sendo

$$BYe_i \geq 0$$

onde $B = (b \mid A)$ e $0 \leq i \leq n$.

Analogamente, pode-se multiplicar o sistema $Ax \leq b$ por algum $1 - x_i$, e formulando essa nova restrição em termos da matriz variável Y , obtem-se

$$BY(e_0 - e_i) \geq 0$$

onde $B = (b \mid A)$ e $0 \leq i \leq n$.

Define-se o conjunto $\tilde{\mathcal{K}} = \{(1, x) : x \in \mathcal{K}\}$. Esse conjunto $\tilde{\mathcal{K}}$ é chamado de homogenização de \mathcal{K} e é um cone no espaço \mathbb{R}^{n+1} . Tem-se, então, uma nova propriedade a respeito de Y , a saber, $Ye_i \in \tilde{\mathcal{K}}$ e $Y(e_0 - e_i) \in \tilde{\mathcal{K}}$.

Define-se também os conjuntos abaixo:

$$\mathcal{M}(\mathcal{K}) = \left\{ Y : \text{diag}(Y) = Y e_0, Ye_i \in \tilde{\mathcal{K}}, Y(e_0 - e_i) \in \tilde{\mathcal{K}} \right\}$$

e

$$\mathcal{M}_+(\mathcal{K}) = \{Y \in \mathcal{M}(\mathcal{K}) : Y \succeq 0\}.$$

Define-se a projeção de $\mathcal{M}(\mathcal{K})$ e $\mathcal{M}_+(\mathcal{K})$ no espaço \mathbb{R}^n , respectivamente, por:

$$\mathcal{N}(\mathcal{K}) = \left\{ x \in \mathbb{R}^n : \begin{pmatrix} 1 \\ x \end{pmatrix} = Y e_0, \text{ para algum } Y \in \mathcal{M}(\mathcal{K}) \right\}$$

e

$$\mathcal{N}_+(\mathcal{K}) = \left\{ x \in \mathbb{R}^n : \begin{pmatrix} 1 \\ x \end{pmatrix} = Y e_0 \text{ para algum } Y \in \mathcal{M}_+(\mathcal{K}) \right\}.$$

É fácil ver que $\mathcal{N}_+(\mathcal{K}) \subseteq \mathcal{N}(\mathcal{K})$, uma vez que $\mathcal{M}_+(\mathcal{K}) \subseteq \mathcal{M}(\mathcal{K})$. Define-se a iteração:

$$\mathcal{N}^t(\mathcal{K}) = \mathcal{N}(\mathcal{N}^{t-1}(\mathcal{K})).$$

Teorema 4.3

$\mathcal{N}(\mathcal{K}) \subseteq \text{conv}(\mathcal{K} \cap \{x : x_i = 0, 1\})$. ■

Demonstração:

Seja $x \in \mathcal{N}(\mathcal{K})$. Então, existe $Y \in \mathcal{M}(\mathcal{K})$ tal que

$$Y e_0 = \text{diag}(Y) = \begin{pmatrix} 1 \\ x \end{pmatrix}$$

Como $Y \in \mathcal{M}(\mathcal{K})$, tem-se que $Y e_i \in \mathcal{K}$ e $Y(e_0 - e_i) \in \mathcal{K}$. Assim, $Y e_i$ nada mais é que a i -ésima coluna de Y , a qual chama-se y_i . Pela definição de Y , a saber, $y_{jj} = y_{0j}$, então $y_i = y_0$. Isto é, a i -ésima coluna de Y é igual à primeira coluna de Y . Logo $x_i = x_0 = 1$. Assim, $y_i \in \text{conv}(\mathcal{K} \cap \{x : x_i = 1\})$.

Por outro lado, como $Y(e_0 - e_i) \in \mathcal{K}$ significa que a diferença entre a 1ª coluna de Y e a i -ésima coluna de Y é um vetor, no qual $x_i = x_0 - x_i = 1 - 1 = 0$. Portanto, $y_0 - y_i \in \text{conv}(\mathcal{K} \cap \{x : x_i = 0\})$.

Logo, $\begin{pmatrix} 1 \\ x \end{pmatrix} = Y e_0 = y_0 = y_0 - y_i + y_i$.

Assim, $Y e_0$ pertence à homogeneização de $\text{conv}(\mathcal{K} \cap \{x : x_i = 0, 1\})$, fazendo com que $x \in \text{conv}(\mathcal{K} \cap \{x : x_i = 0, 1\})$. ■

Corolário 4.4

$\mathcal{N}_+(\mathcal{K}) \subseteq \text{conv}(\mathcal{K} \cap \{x : x_i = 0, 1\})$. ■

Demonstração:

$\mathcal{N}_+(\mathcal{K}) \subseteq \mathcal{N}(\mathcal{K})$. ■

Corolário 4.5

$\mathcal{N}(\mathcal{K}) \subseteq \mathcal{K}$. ■

Demonstração:

$\mathcal{N}(\mathcal{K}) \subseteq \text{conv}(\mathcal{K} \cap \{x : x_i = 0, 1\}) \subseteq \mathcal{K}$. ■

Fazendo a iteração definida acima, tem-se $\mathcal{N}^n(\mathcal{K}) = \mathcal{P}$. Tem-se, ainda, a relação de inclusão

$$\mathcal{P} = \mathcal{N}^n(\mathcal{K}) \subseteq \dots \subseteq \mathcal{N}^{t+1}(\mathcal{K}) \subseteq \mathcal{N}^t(\mathcal{K}) \subseteq \dots \subseteq \mathcal{N}(\mathcal{K}) \subseteq \mathcal{K}$$

e

$$\mathcal{P} = \mathcal{N}_+^n(\mathcal{K}) \subseteq \dots \subseteq \mathcal{N}_+^{t+1}(\mathcal{K}) \subseteq \mathcal{N}_+^t(\mathcal{K}) \subseteq \dots \subseteq \mathcal{N}_+(\mathcal{K}) \subseteq \mathcal{K}.$$

O método de Lovász-Schrijver é o primeiro método *lift-and-project* que utiliza restrições semidefinidas. Foi mostrado, então, que, para esse método, as restrições semidefinidas encontram melhores planos de corte que as relaxações lineares.

3.4.4 Método de Lasserre

Seja \mathcal{K} o conjunto:

$$\mathcal{K} = \{x \in [0, 1]^n : g_l(x) \geq 0\}$$

para $l = 1, \dots, m$.

O conjunto \mathcal{K} descrito acima apresenta o sistema linear $Ax \leq b$ como formado por m inequações lineares, a saber, $g_l(x) \geq 0$, para $l = 1, \dots, m$.

Como antes, $\mathcal{V} = \{1, \dots, n\}$. Seja $\mathcal{P}(\mathcal{V})$ o conjunto de todos os subconjuntos de \mathcal{V} e $\mathcal{P}_t(\mathcal{V})$ o conjunto de todos os subconjuntos de \mathcal{V} com cardinalidade menor ou igual a t . Seja o espaço $\mathbb{R}^{\mathcal{P}(\mathcal{V})}$, cujos componentes são denotadas por y_I , onde $I \in \mathcal{P}(\mathcal{V})$. O espaço $\mathbb{R}^{\mathcal{P}(\mathcal{V})}$ é um espaço isomorfo a \mathbb{R}^{2^n} . As entradas dos vetores em ambos espaços são números reais. A única diferença é a notação do indexador das entradas. Em \mathbb{R}^{2^n} , as entradas variam de 1 a 2^n , isto é, as entradas são x_1 até x_{2^n} . No espaço $\mathbb{R}^{\mathcal{P}(\mathcal{V})}$, as entradas variam de \emptyset a \mathcal{V} , ou seja, y_\emptyset até $y_\mathcal{V}$.

Seja Z a matriz quadrada de entradas $\{0, 1\}$, indexadas por $\mathcal{P}(\mathcal{V})$, definida como $Z(\mathcal{I}, \mathcal{J}) = 1$ se $\mathcal{I} \subseteq \mathcal{J}$ e $Z(\mathcal{I}, \mathcal{J}) = 0$, no caso contrário. A matriz Z é chamada de Matriz Zeta de $\mathcal{P}(\mathcal{V})$. A matriz Zeta tem a propriedade de sua diagonal principal ser composta de entradas 1. Sendo assim, ela é não-singular, o que implica a existência de sua inversa, conhecida como Matriz de Möbius e definida por

$$Z^{-1}(\mathcal{I}, \mathcal{J}) = (-1)^{|\mathcal{J}-\mathcal{I}|}$$

se $\mathcal{I} \subseteq \mathcal{J}$ e $Z^{-1}(\mathcal{I}, \mathcal{J}) = 0$, caso contrário.

Se $\mathcal{J} \subseteq \mathcal{V}$, então $Z^{\mathcal{J}}$ é a coluna indexada por \mathcal{J} da matriz Zeta. Dado um subconjunto $\mathcal{H} \subseteq \mathcal{P}(\mathcal{V})$, considere $\mathcal{C}_{\mathcal{H}}$ como o cone de $\mathbb{R}^{\mathcal{P}(\mathcal{V})}$ gerado pelas colunas $Z^{\mathcal{J}}$, com $\mathcal{J} \subseteq \mathcal{H}$.

Pode-se definir também $\mathcal{C}_{\mathcal{H}}$ como:

$$\mathcal{C}_{\mathcal{H}} = \{y \in \mathbb{R}^{\mathcal{P}(\mathcal{V})} : Z^{-1}y \geq 0, Z^{-1}y(\mathcal{J}) = 0 \Leftrightarrow \mathcal{J} \in \mathcal{P}(\mathcal{V}) - \mathcal{H}\}.$$

Monique Laurent[27] prova que a projeção de $\mathcal{C}_{\mathcal{H}} \cap \{y \in \mathbb{R}^{\mathcal{P}(\mathcal{V})} : y_{\emptyset} = 1\}$ em \mathbb{R}^n é o politopo 0/1, denominado $\mathcal{P}_{\mathcal{H}}$. Esse fato prova que qualquer politopo 0/1 em \mathbb{R}^n pode ser escrito como a projeção de um politopo em \mathbb{R}^{2^n-1} .

Define-se as matrizes $M_t(y)$ e $M_{\mathcal{U}}(y)$ como:

$$M_t(y) = (y(\mathcal{I} \cup \mathcal{J}))_{|\mathcal{I}|, |\mathcal{J}| \leq t}$$

e

$$M_{\mathcal{U}}(y) = (y(\mathcal{I} \cup \mathcal{J}))_{\mathcal{I}, \mathcal{J} \subseteq \mathcal{U}}$$

onde $|\mathcal{I}|$ é a cardinalidade de \mathcal{I} , $1 \leq t \leq n$ e $\mathcal{U} \subseteq \mathcal{V}$.

Tem-se que $M_t(y)$ é uma matriz simétrica, uma vez que $\mathcal{I} \cup \mathcal{J} = \mathcal{J} \cup \mathcal{I}$, com entradas $y(\mathcal{I} \cup \mathcal{J})$, onde as cardinalidades de \mathcal{I} e \mathcal{J} são, no máximo, o valor de t , para algum t inteiro entre 1 e n .

Da mesma forma, $M_{\mathcal{U}}(y)$ é uma matriz simétrica com entradas $y(\mathcal{I} \cup \mathcal{J})$, onde os conjuntos \mathcal{I} e \mathcal{J} são subconjuntos de \mathcal{U} , para algum \mathcal{U} subconjunto de \mathcal{V} .

Dessa maneira, fica fácil ver que $M_n(y) = M_{\mathcal{V}}(y)$. Isso porque a entrada da matriz $M_n(y)$ é $y(\mathcal{I} \cup \mathcal{J})$, com $|\mathcal{I}|, |\mathcal{J}| \leq n$. Mas isso acontece para todo $\mathcal{I}, \mathcal{J} \subseteq \mathcal{V}$. Portanto, são as mesmas entradas da matriz $M_{\mathcal{V}}(y)$. Essa matriz $M_n(y) = M_{\mathcal{V}}(y)$ recebe o nome de Matriz Momento de y .

Pode-se provar que $M_{\mathcal{V}}(y) = Z \text{Diag}(Z^{-1}y)Z^T$ [27]. Isso significa que:

$$y \in \mathcal{C}_{\mathcal{P}(\mathcal{V})} \iff Z^{-1}y \geq 0 \iff M_{\mathcal{V}}(y) \succeq 0.$$

Estendendo o vetor g_l de \mathbb{R}^n para $\mathbb{R}^{\mathcal{P}(\mathcal{V})}$ e adicionando coordenadas nulas, define-se o produto $g_l * y \in \mathbb{R}^{\mathcal{P}(\mathcal{V})}$ como sendo

$$g_l * y = M_{\mathcal{V}}(y)g_l.$$

Assim, o teorema abaixo é provado [27].

Teorema 4.6

$$y \in \mathcal{C}_{\mathcal{H}} \iff M_{\mathcal{V}}(y) \succeq 0 \quad \vee \quad M_{\mathcal{V}}(g_l * y) \succeq 0$$

para $l = 1, \dots, m$. ■

Agora, dado t entre 1 e n , seja $\mathcal{P}_t(\mathcal{K})$ o politopo definido pelas condições:

$$M_{t+1}(y) \succeq 0$$

e

$$M_t(g_l * y) \succeq 0$$

para $l = 1, \dots, m$.

O método de Lasserre consiste em se fazer a projeção de $\mathcal{P}_t(\mathcal{K})$ em \mathbb{R}^n , gerando assim o politopo $\mathcal{Q}_t(\mathcal{K})$.

Pretende-se, a partir de agora, provar a relação de inclusão abaixo:

$$\mathcal{P} \subseteq \mathcal{Q}_n(\mathcal{K}) \subseteq \mathcal{Q}_{n-1}(\mathcal{K}) \subseteq \dots \subseteq \mathcal{Q}_1(\mathcal{K}) \subseteq \mathcal{K}.$$

Teorema 4.7

$$\mathcal{Q}_t(\mathcal{K}) \subseteq \mathcal{K}$$

para $t = 1, \dots, n$. ■

Demonstração:

Observe que $\mathcal{Q}_t(\mathcal{K})$ é a projeção de $\mathcal{P}_t(\mathcal{K})$ em \mathbb{R}^n . Mas $\mathcal{P}_t(\mathcal{K})$ é o politopo definido pelas condições $M_{t+1}(y) \succeq 0$ e $M_t(g_l * y) \succeq 0$, para $l = 1, \dots, m$. Pelo teorema acima,

$y \in \mathcal{C}_{\mathcal{H}}$ para algum $\mathcal{H} \subseteq \mathcal{P}(\mathcal{V})$ e, portanto, $\mathcal{P}_t(\mathcal{K}) = \mathcal{C}_{\mathcal{H}} \cap \{y : y_0 = 1\}$ para algum $\mathcal{H} \subseteq \mathcal{P}(\mathcal{V})$. Portanto, sua projeção é um polítopo 0/1 na variável x_t e satisfaz as inequações $g_l(x) \geq 0$, concluindo a prova. ■

Corolário 4.8

$\mathcal{Q}_t(\mathcal{K}) \subseteq \mathcal{Q}_{t-1}(\mathcal{K})$ e $\mathcal{P} \subseteq \mathcal{Q}_n(\mathcal{K})$ ■

Demonstração:

Prova análoga ao teorema anterior. ■

3.5 Uma Comparação dos Métodos *lift-and-project*

As relaxações de \mathcal{P} encontradas até agora foram

- $\mathcal{P}_{j_1, \dots, j_t}(\mathcal{K})$ do método BCC;
- $\mathcal{S}_{1, \dots, t}(\mathcal{K})$ do método SA;
- $N^t(\mathcal{K})$ do método LS com restrição linear;
- $N_+^t(\mathcal{K})$ do método LS com restrição semidefinida
- $Q_t(\mathcal{K})$ do método de Lasserre.

Suponha que todas essas relaxações estão no mesmo passo t , onde t varia de 1 a n . Surge a seguinte pergunta: qual é a hierarquia entre essas relaxações, isto é, qual está incluída na outra? Essa pergunta é o que a partir de agora o texto pretende mostrar.

O teorema a seguir mostra que a relaxação de Sherali-Adams refina a relaxação de Lovász-Schrijver com restrições lineares.

Teorema 5.1

$\mathcal{S}_{1, \dots, t}(\mathcal{K}) \subseteq N^t(\mathcal{S}_{1, \dots, t-1}(\mathcal{K}))$ ■

Demonstração:

Basta provar que todo elemento $(y_1, \dots, y_n)^T \in \mathcal{S}_{1, \dots, t}(\mathcal{K})$ também é elemento de

$\mathcal{N}^t(\mathcal{S}_{1,\dots,t-1}(\mathcal{K}))$. Como $(y_1, \dots, y_n)^T \in \mathcal{S}_{1,\dots,t}(\mathcal{K})$, então existe um $y \in \mathcal{R}_{1,\dots,t}(\mathcal{K})$ cuja projeção em \mathbb{R}^n é $(y_1, \dots, y_n)^T$. A matriz $Y = M_1(y) \in \mathcal{M}(\mathcal{S}_{1,\dots,t-1}(\mathcal{K}))$. Logo, sua projeção pertence a $\mathcal{N}^t(\mathcal{S}_{1,\dots,t-1}(\mathcal{K}))$. Para concluir a prova, observe que a projeção de Y é o vetor $(y_1, \dots, y_n)^T$, pois $(1, y_1, \dots, y_n)^T = Y e_0$. ■

Corolário 5.2

$$\mathcal{S}_{1,\dots,t}(\mathcal{K}) \subseteq \mathcal{N}^t(\mathcal{K}) \quad \blacksquare$$

Demonstração:

Considere $\mathcal{S}_0(\mathcal{K}) = \mathcal{K}$. Será feito agora uma indução sobre t .

A inclusão inicia em $t = 1$. De fato:

$$\mathcal{S}_1(\mathcal{K}) \subseteq \mathcal{N}(\mathcal{S}_0(\mathcal{K})) = \mathcal{N}(\mathcal{K}) = \mathcal{N}^1(\mathcal{K}).$$

Supõe-se que a relação de inclusão a ser provada ocorre para um certo k . Isto é, $\mathcal{S}_{1,\dots,k}(\mathcal{K}) \subseteq \mathcal{N}^k(\mathcal{K})$. Prova-se agora a relação para $k + 1$, ou seja, $\mathcal{S}_{1,\dots,k+1}(\mathcal{K}) \subseteq \mathcal{N}^{k+1}(\mathcal{K})$.

Isso ocorre pois, se $\mathcal{S}_{1,\dots,k}(\mathcal{K}) \subseteq \mathcal{N}^k(\mathcal{K})$, então:

$$\mathcal{N}(\mathcal{S}_{1,\dots,k}(\mathcal{K})) \subseteq \mathcal{N}(\mathcal{N}^k(\mathcal{K})).$$

Mas, pela hierarquia do método LS, $\mathcal{N}(\mathcal{N}^k(\mathcal{K})) = \mathcal{N}^{k+1}(\mathcal{K})$. Portanto,

$$\mathcal{S}_{1,\dots,k+1}(\mathcal{K}) \subseteq \mathcal{N}(\mathcal{S}_{1,\dots,k}(\mathcal{K})) \subseteq \mathcal{N}(\mathcal{N}^k(\mathcal{K})) = \mathcal{N}^{k+1}(\mathcal{K}). \quad \blacksquare$$

O teorema a seguir mostra que a relaxação de Lasserre refina a relaxação de Sherali-Adams.

Teorema 5.3

$$\mathcal{Q}_t(\mathcal{K}) \subseteq \mathcal{S}_{1,\dots,t}(\mathcal{K}) \quad \blacksquare$$

Demonstração:

Suponha $y \in \mathcal{P}_t(\mathcal{K})$, cuja projeção pertence a $\mathcal{Q}_t(\mathcal{K})$. Assim, $M_{t+1}(y)$ e $M_t(g_l * y)$ são matrizes semidefinidas positivas. A restrição de y até \mathcal{P}_t faz com que $y \in \mathcal{R}_{1,\dots,t}(\mathcal{K})$. Portanto, sua projeção é elemento de $\mathcal{S}_{1,\dots,t}(\mathcal{K})$. ■

O teorema a seguir mostra que a relaxação de Lasserre refina a relaxação de Lovász-Schrijver com restrições semidefinidas.

Teorema 5.4

$$\mathcal{Q}_t(\mathcal{K}) \subseteq \mathcal{N}_+(\mathcal{Q}_{t-1}(\mathcal{K})) \quad \blacksquare$$

Demonstração:

Essa prova é análoga ao teorema anterior. \blacksquare

Corolário 5.5

$$\mathcal{Q}_t(\mathcal{K}) \subseteq \mathcal{N}_+^t(\mathcal{K}) \quad \blacksquare$$

Demonstração:

Para $t = 1$, $\mathcal{Q}_1(\mathcal{K}) \subseteq \mathcal{N}_+(\mathcal{Q}_0(\mathcal{K}))$. Considere $\mathcal{Q}_0(\mathcal{K}) = \mathcal{K}$ e $\mathcal{N}_+(\mathcal{K}) = \mathcal{N}_+^1(\mathcal{K})$. Portanto, $\mathcal{Q}_1(\mathcal{K}) \subseteq \mathcal{N}_+^1(\mathcal{K})$.

Suponha que a inclusão é válida para algum valor de k , isto é,

$$\mathcal{Q}_k(\mathcal{K}) \subseteq \mathcal{N}_+^k(\mathcal{K}).$$

Prova-se que $\mathcal{Q}_{k+1}(\mathcal{K}) \subseteq \mathcal{N}_+^{k+1}(\mathcal{K})$. Isso porque já sabe-se que $\mathcal{Q}_{k+1}(\mathcal{K}) \subseteq \mathcal{N}_+(\mathcal{Q}_k(\mathcal{K}))$, pelo teorema. Além disso, também sabe-se que $\mathcal{Q}_k(\mathcal{K}) \subseteq \mathcal{N}_+^k(\mathcal{K})$. Fazendo a relaxação LS com restrição semidefinida dos dois conjuntos, obtém-se:

$$\mathcal{N}_+(\mathcal{Q}_k(\mathcal{K})) \subseteq \mathcal{N}_+(\mathcal{N}_+^k(\mathcal{K})) = \mathcal{N}_+^{k+1}(\mathcal{K}).$$

Portanto,

$$\mathcal{Q}_{k+1}(\mathcal{K}) \subseteq \mathcal{N}_+(\mathcal{Q}_k(\mathcal{K})) \subseteq \mathcal{N}_+(\mathcal{N}_+^k(\mathcal{K})) = \mathcal{N}_+^{k+1}(\mathcal{K}). \quad \blacksquare$$

Teorema 5.6

$$\mathcal{S}_{1,\dots,t}(\mathcal{K}) \subseteq \mathcal{P}_{j_1,\dots,j_t}(\mathcal{K}) \quad \blacksquare$$

Demonstração:

Já foi provado que $\mathcal{N}_t(\mathcal{K}) \subseteq \{x : x_s = 0, 1\}$, com s entre 1 e t . Mas $\mathcal{P}_{j_1,\dots,j_t}(\mathcal{K}) = \{x : x_s = 0, 1\}$, com s entre 1 e t . Como $\mathcal{S}_{1,\dots,t}(\mathcal{K}) \subseteq \mathcal{N}_t(\mathcal{K})$, tem-se a prova completa. \blacksquare

Assim, a conclusão que se chega é que o método de Lasserre produz melhores relaxações, quando comparado com os outros métodos tradicionais BCC, SA e LS.

Portanto, as relações de inclusão mostradas foram:

- $\mathcal{Q}_t(\mathcal{K}) \subseteq \mathcal{N}_+^t(\mathcal{K})$.
- $\mathcal{S}_{1,\dots,t}(\mathcal{K}) \subseteq \mathcal{N}_t(\mathcal{K}) \subseteq \mathcal{P}_{j_1,\dots,j_t}(\mathcal{K})$.
- $\mathcal{Q}_t(\mathcal{K}) \subseteq \mathcal{S}_{1,\dots,t}(\mathcal{K})$.

Leituras mais avançadas sobre a relação entre os métodos *lift-and-project* podem ser encontradas em [26].

3.6 Conclusão

Este capítulo mostrou os métodos *Lift-and-Project* e suas construções baseadas em Programação Linear (BCC e SA) e aquelas baseadas em Programação Semidefinida (LS, SA e Lasserre). Mostrou-se a equivalência das formulações de um problema por restrições e um problema por cones. Foram apresentados também dois métodos de projeção.

Capítulo 4

Problema Quadrático 0-1 e o método *Branch and Cut*

4.1 Introdução

Uma outra maneira de se resolver um problema de soluções inteiras utilizando a programação semidefinida é o uso do *Branch and Cut*. Será abordado neste capítulo como o uso do *Branch and Cut* pode encontrar soluções inteiras em Programação Semidefinida (PSD).

4.2 Um Simples Exemplo de *Branch and Bound*

O *Branch and Cut* é um algoritmo que combina a vantagem dos planos de corte e do método *Branch and Bound*. Este último gera uma árvore pela partição de um problema original em dois subproblemas, onde uma das variáveis é dividida em duas novas restrições. A partição é chamada *Branch* e cada uma das soluções dos subproblemas partidos é limitada pela solução do problema inicial. Esse limite é chamado *Bound*.

Um pequeno exemplo de um problema linear inteiro, baseado em relaxações lineares, descreve o algoritmo.

Exemplo 2.1 Suponha que se queira resolver um problema linear inteiro com o método *Branch-and-Bound*. Define-se o problema inicial (P_0) como

$$\begin{array}{ll} \max & 5x_1 + 8x_2 \\ \text{s. a} & x_1 + x_2 \leq 6 \\ & 5x_1 + 9x_2 \leq 45 \\ & x_1, x_2 \in \mathbb{Z}_+ \end{array}$$

Resolve-se o problema relaxado de \mathbb{Z}_+ para \mathbb{R} , encontrando, como solução ótima:

$$x_1 = \frac{9}{4}; \quad x_2 = \frac{15}{4}; \quad z = \frac{165}{4}.$$

É realizada a partição do problema em relação à variável x_2 . Sabe-se que, para $x_2 \in \mathbb{Z}_+$:

$$x_2 \geq \lfloor \frac{15}{4} \rfloor + 1 \quad \text{ou} \quad x_2 \leq \lfloor \frac{15}{4} \rfloor$$

onde

$$\lfloor x \rfloor = \max\{y \in \mathbb{Z} \mid y \leq x\}.$$

Assim, tem-se que $x_2 \geq 4$ ou $x_2 \leq 3$. Logo, o problema foi separado em dois problemas, a saber:

$$(P_1) \quad \begin{array}{ll} \max & 5x_1 + 8x_2 \\ \text{s. a} & x_1 + x_2 \leq 6 \\ & 5x_1 + 9x_2 \leq 45 \\ & x_2 \leq 3 \\ & x_1, x_2 \in \mathbb{Z}_+ \end{array}$$

e

$$(P_2) \quad \begin{array}{ll} \max & 5x_1 + 8x_2 \\ \text{s. a} & x_1 + x_2 \leq 6 \\ & 5x_1 + 9x_2 \leq 45 \\ & x_2 \geq 4 \\ & x_1, x_2 \in \mathbb{Z}_+. \end{array}$$

Solucionando-se (P_1) e (P_2) relaxados da mesma forma que (P_0), obtém-se, em (P_1), os valores de $x_1 = 3$ e $x_2 = 3$, sendo o valor ótimo da função objetivo dado por 39. Obtém-se, em (P_2), os valores de $x_1 = 1,8$ e $x_2 = 4$, para $z = 41$. Da mesma

forma que o ótimo de (P_0) é um limitante superior dos ótimos de (P_1) e (P_2) , tem-se a seguinte análise:

- Não é necessário realizar a partição de (P_1) , já que suas variáveis ótimas são inteiras e seu valor ótimo também o é. Caso fosse feita a partição, o valor ótimo da função objetivo não seria maximizado e, então, seria realizado um procedimento sem qualquer progresso no valor da função.
- O problema (P_2) tem valor ótimo $z = 41$. Caso tivesse variáveis inteiras, a busca teria terminado e a solução ótima de P_2 seria a solução ótima do problema. Como as variáveis não são inteiras, tem-se, então, que, caso se realizasse a partição de (P_2) , seria produzido um valor da função objetivo menor ou, no máximo, igual a 41, mas, no entanto, com variáveis inteiras. Isso é interessante, pois, em qualquer caso, poderia ser maior que a aproximação primária da solução, igual a 39.

Fazendo-se a partição de (P_2) , tem-se dois novos problemas, a saber, (P_3) e (P_4) . Solucionando-os também na forma relaxados, tem-se (P_3) infactível e, em (P_4) , encontra-se que $x_1 = 1$, $x_2 = 4,25$ e $z = 40,4$. Pelo mesmo motivo acima, ou seja, encontrar variáveis ótimas inteiras que forneça valor da função objetivo maior que 39, pode-se fazer a partição de (P_4) , já que seu valor ótimo é 40,4, o que poderia fornecer, na partição, como ótimo, $z = 40$.

Observe que, se o valor ótimo de (P_4) fosse menor que 40, a solução do problema P_0 já estaria obtida. Nesse caso, a partição daria valores menores ou iguais a 39, o que não satisfaz a busca de valores maiores que 39.

Na partição de (P_4) , encontra-se (P_5) , tendo variáveis $x_1 = 0$, $x_2 = 5$ e valor ótimo da função objetivo $z = 40$. Desse modo, o problema está resolvido e sua solução é a solução de (P_5) . \square

A árvore de subproblemas é conhecida como *Árvore de Branch and Bound*, cuja raiz é o problema inicial. A diferença entre os dois métodos é que no método *Branch and Cut* insere-se planos de corte nos subproblemas da *Árvore de Branch and Bound*, até que nenhum progresso seja feito. Neste caso, faz-se uma nova divisão.

Leituras mais avançadas sobre os métodos do tipo *Branch* poderão ser encontradas em [11, 31].

4.3 O Problema Quadrático 0-1 e Formulações Equivalentes

Muitos problemas de otimização combinatória podem ser escritos como o problema quadrático 0 – 1 abaixo:

$$\begin{aligned} \min \quad & y^T C y \\ & y \in \{0, 1\}^n. \end{aligned} \tag{4.1}$$

Em (4.1), C é uma matriz simétrica de ordem n . É muito comum, na maior parte dos problemas de otimização combinatória, que C seja uma matriz esparsa, isto é, C tenha um grande número de entradas nulas. Nos últimos anos, tem-se estudado relaxações para problemas inteiros do tipo $-1/1$. Será mostrado agora que o problema (4.1) é equivalente a um problema do tipo $-1/1$ [16, 29].

Para isso, primeiramente, introduz-se uma variável x tal que:

$$y = \frac{x + e}{2}$$

onde e é o vetor de entradas unitárias. Assim,

$$\begin{aligned} y^T C y &= \frac{(x + e)^T}{2} C \frac{(x + e)}{2} \\ &= \frac{x^T C x}{4} + \frac{e^T C x}{2} + \frac{e^T C e}{4}. \end{aligned}$$

Fazendo-se:

$$\begin{aligned} Q &= \frac{C}{4} \\ c &= \frac{-C e}{4} \\ d &= \frac{e^T C e}{4} \end{aligned}$$

obtem-se:

$$y^T C y = x^T Q x - 2c^T x + d.$$

Define-se, então, a matriz D :

$$D = \begin{pmatrix} Q & -c \\ -c^T & d \end{pmatrix}.$$

Logo, o problema (4.1) de variáveis do tipo 0/1 é equivalente ao problema de variáveis $-1/1$, a saber:

$$\begin{aligned} \max \quad & z^T D z \\ \text{suj. a} \quad & z \in \{-1, 1\}^{n+1}. \end{aligned} \tag{4.2}$$

Com o objetivo de trabalhar com as variáveis convencionais, considere o problema (4.2) reescrito como:

$$\begin{aligned} \max \quad & x^T C x \\ \text{suj. a} \quad & x \in \{-1, 1\}^n. \end{aligned} \tag{4.3}$$

Será visto agora como trabalhar o método *Branch and Cut* no problema (4.3). Primeiramente, será usado o modelo de PSD para (4.3), já mostrado anteriormente, a saber:

$$\begin{aligned} \max \quad & C \bullet X \\ \text{suj. a} \quad & X \succeq 0 \\ & \text{diag}(X) = e \\ & \text{posto}(X) = 1. \end{aligned} \tag{4.4}$$

Seja $F = \{-1, 1\}^n$ a região factível do problema (4.3). Sabe-se que x_{ij} é a entrada i, j da matriz X e que qualquer matriz xx^T tem posto 1, para todo $x \in \mathbb{R}^n$. A primeira relaxação semidefinida que se faz é obtida, retirando-se a restrição de X ser uma matriz de posto 1. Isso porque essa restrição é não convexa, o que dificulta o problema. Assim, o problema relaxado torna-se:

$$\begin{aligned} \max \quad & C \bullet X \\ \text{suj. a} \quad & X \succeq 0 \\ & \text{diag}(X) = e. \end{aligned} \tag{4.5}$$

O problema dual é dado por:

$$\begin{aligned} \min \quad & e^T y \\ \text{suj. a} \quad & \text{Diag}(y) - S = C \\ & S \succeq 0. \end{aligned} \tag{4.6}$$

Leituras mais avançadas sobre o problema quadrático 0/1 e suas formulações PL e PSD poderão ser encontradas em [7, 16, 19, 29].

4.4 Um Procedimento de *Branch* para PSD

Será mostrado aqui como fazer o procedimento de separação do problema original (4.5) em dois subproblemas de dimensões menores, o chamado *Branch*.

Para isso, a matriz Laplaciana L do problema (2.14) ser escrita como:

$$L = \begin{pmatrix} \bar{L} & p_1 & p_2 \\ p_1^T & \alpha & \beta \\ p_2^T & \beta & \gamma \end{pmatrix}$$

Na matriz acima, tem-se que $\bar{L} \in \mathbb{S}\mathbb{R}^{(n-2) \times (n-2)}$. Também tem-se $p_1, p_2 \in \mathbb{R}^{n-2}$ e $\alpha, \beta, \gamma \in \mathbb{R}$.

Sabe-se que $C = \frac{1}{4}L$, onde C é a matriz do problema (4.5). O *Branch* do problema (4.5) o divide em dois subproblemas, definidos abaixo como:

$$\begin{aligned} \max \quad & \frac{1}{4} \begin{pmatrix} \bar{L} & p_1 + p_2 \\ p_1^T + p_2^T & \alpha + 2\beta + \gamma \end{pmatrix} \bullet X \\ \text{suj. a} \quad & X \succeq 0 \\ & \text{diag}(X) = e \end{aligned} \tag{4.7}$$

cujo dual é dado por:

$$\begin{aligned} & \min && e^T y \\ \text{suj. a} & \text{Diag}(y) - S = \frac{1}{4} \begin{pmatrix} \bar{L} & p_1 + p_2 \\ p_1^T + p_2^T & \alpha + 2\beta + \gamma \end{pmatrix} \\ & && S \succeq 0 \end{aligned} \quad (4.8)$$

e

$$\begin{aligned} & \max && \frac{1}{4} \begin{pmatrix} \bar{L} & p_1 - p_2 \\ p_1^T - p_2^T & \alpha - 2\beta + \gamma \end{pmatrix} \bullet X \\ \text{suj. a} & && X \succeq 0 \\ & && \text{diag}(X) = e \end{aligned} \quad (4.9)$$

cujo dual é:

$$\begin{aligned} & \min && e^T y \\ \text{suj. a} & \text{Diag}(y) - S = \frac{1}{4} \begin{pmatrix} \bar{L} & p_1 - p_2 \\ p_1^T - p_2^T & \alpha - 2\beta + \gamma \end{pmatrix} \\ & && S \succeq 0. \end{aligned} \quad (4.10)$$

Caso existam restrições no problema do tipo $A_i \bullet X = b_i$, basta considerar:

$$A_i = \begin{pmatrix} \bar{A}_i & q_{i1} & q_{i2} \\ q_{i1}^T & \alpha_i & \beta_i \\ q_{i2}^T & \beta_i & \gamma_i \end{pmatrix}.$$

As novas restrições dos subproblemas ficam:

$$\begin{pmatrix} \bar{A}_i & q_{i1} + q_{i2} \\ q_{i1}^T + q_{i2}^T & \alpha_i + 2\beta_i + \gamma_i \end{pmatrix} \bullet X + a_i^T x = b_i$$

e

$$\begin{pmatrix} \bar{A}_i & q_{i1} - q_{i2} \\ q_{i1}^T - q_{i2}^T & \alpha_i - 2\beta_i + \gamma_i \end{pmatrix} \bullet X + a_i^T x = b_i$$

respectivamente.

Ao se fazer o *Branch*, a matriz variável X , que no problema raiz era simétrica de ordem n , agora passa a ter ordem $n - 1$. Assim, a vantagem computacional do *Branch* é diminuir a ordem da matriz variável, o que poderá possibilitar melhor desempenho computacional e maior facilidade de resolução [7, 16, 29].

4.5 Inequações Hipermétricas para Planos de Corte

Será visto agora como pode-se melhorar um *branch* inserindo-se planos de corte. Para a definição desses planos de corte, utiliza-se as inequações hipermétricas [16].

Seja $b \in \mathbb{Z}^n$ tal que $|b^T x| \geq 1$ para todo $x \in \{-1, 1\}^n$. Uma escolha de b que satisfaça a essas condições é fazer b_1 até b_{n-1} inteiros arbitrariamente e, em seguida, fazer b_n de tal forma que a soma das entradas do vetor b seja ímpar. Seja:

$$P_c = \text{conv}(xx^T; x \in \{-1, 1\}^n)$$

Assim, $|b^T x| \geq 1$ implica que $x^T b b^T x \geq 1$. Seja $B = b b^T$. Logo, tem-se que $x^T B x \geq 1$, o que é a mesma coisa que $B \bullet X \geq 1$, com $X \in P_c$.

Portanto, as inequações hipermétricas são inequações do tipo $B \bullet X \geq 1$, onde $B = b b^T$ e b é um vetor de entradas inteiras, com a soma dessas entradas sendo um número ímpar.

As inequações triangulares são as inequações hipermétricas do tipo mais simples. Elas são definidas como:

$$X_{ij} + X_{ik} + X_{jk} \geq -1$$

e

$$X_{ij} - X_{ik} - X_{jk} \geq -1$$

onde $i, j, k \in \mathcal{V} = \{1, \dots, n\}$. Uma inequação hipermétrica é dita violada quando $b^T X b < 1$.

Assim, o algoritmo para inserir planos de corte fica sendo:

Passo 1: Faça a relaxação do PSD.

Passo 2: Resolva essa relaxação. Isso fornece um limite superior.

Passo 3: Escreva todas as inequações triangulares.

Passo 4: Insira, nas restrições do problema, as inequações triangulares violadas. Caso não exista nenhuma inequação violada, verifique o critério de parada.

Passo 5: Resolva o novo problema, com as inequações triangulares violadas.

Passo 6: O critério de parada é a diferença entre a solução da relaxação e a solução do problema com as inequações triangulares violadas. Caso seja menor que $\epsilon > 0$, então pare. Senão, faça um *Branch* e continue a busca de inequações triangulares violadas.

4.6 Conclusão

Este capítulo se concentrou em apresentar como PSD pode ser usada para resolver um problema quadrático do tipo 0/1 pelos métodos de *branch and bound* e *branch and cut*. Experimentos numéricos feitos em [7, 16, 28, 29] demonstram a eficiência computacional para esse tratamento de problemas com grande número de variáveis.

Capítulo 5

Conclusão

Nesse trabalho, mostrou-se que Programação Semidefinida (PSD) é um problema de otimização no qual a variável é uma matriz semidefinida positiva. Dois fatores motivam o estudo da programação semidefinida, sendo o primeiro sua aplicação em diversas áreas e o segundo, o aspecto computacional, que tem um bom condicionamento numérico. Mostrou-se também como foi o desenvolvimento da teoria de programação semidefinida e seus precursores, bem como o estado da arte de suas aplicações e os métodos usados atualmente.

Assim, o interesse deste trabalho foi a aplicação de PSD para resolver problemas de otimização do tipo 0/1. Para atingir esse objetivo, o trabalho iniciou com a apresentação da fundamenção de álgebra linear necessária ao entendimento do problema a ser tratado. Depois disso, foi apresentada a programação semidefinida, seus conceitos, bem como a teoria de dualidade e a sua relação com o problema primal. Também foi apresentado nesse trabalho exemplos de problemas importantes do campo acadêmico que podem ser modelados como um problema de PSD. Os problemas apresentados aqui foram o problema de minimizar o maior autovalor de uma matriz, o problema de programação linear e o problema do *Max-Cut*.

O problema de programação linear, em especial, permitiu mostrar a importância de programação semidefinida, pois ele retrata que programação linear é apenas um caso particular de programação semidefinida. Mas a importância de programação linear, que dispensa comentários, coloca a programação semidefinida com uma im-

portância equivalente. Já o problema do *Max-Cut*, aplicado em diversos outros problemas, tais como o problema do caixeiro viajante, mostra que programação semidefinida tem importância fundamental no aspecto computacional, já que o número de enumerações a ser feito é muito grande. Assim, utilizar métodos que reduzem essas enumerações é fundamental.

O método clássico de resolução de programação semidefinida é o método de ponto interior primal dual path-following, apresentado nesse trabalho. Esse método é, do ponto de vista computacional, muito eficiente e tem a habilidade de trabalhar com uma certa rapidez com problemas com um número de variáveis elevado, cujos métodos tradicionais certamente teriam lentidão, que não pode ser um empecilho para o campo da ciência.

Dentre essas aplicações de programação semidefinida, foram abordados nesse trabalho dois problemas típicos de programação inteira 0/1. O primeiro é o problema de programação linear 0/1, que tem como algoritmo principal os métodos da classe de métodos *Lift-and-Project*. Foram mostrados os métodos baseados em programação linear (BCC e SA) e os métodos baseados em programação semidefinida (SA,LS e Lasserre). Deve ser ressaltado que o método *Lift-and-Project* SA pode ser formulado tanto com restrições lineares quanto com restrições semidefinidas, sendo obtido o mesmo resultado. A conclusão que se chegou foi que os métodos baseados em restrições semidefinidas cortam o poliedro mais profundamente que os métodos construídos por restrições lineares.

O segundo problema foi o problema quadrático 0/1. Foram mostrados diversas formulações desse problema, inclusive a formulação semidefinida. Também neste trabalho foi apresentado um simples exemplo do método *Branch-and-Bound* para um problema linear inteiro, com o objetivo de recordar o método. Foi feita uma abordagem semidefinida para os métodos *Branch-and-Bound* e *Branch-and-Cut* e, apesar de não demonstrar sua melhor eficiência computacional teoricamente, são citados nesse trabalho diversos trabalhos que fizeram experimentos computacionais com essa abordagem.

Como perspectivas para trabalhos futuros, deve ser investigada a aplicação de Programação Semidefinida em outras classes de problemas, buscando melhor condicionamento numérico e menor custo computacional. Além disso, no próprio campo de programação inteira, dada a complexidade dos problemas envolvidos, é importante a continuidade do desenvolvimento de métodos que permitam o tratamento

de problemas mais próximos da realidade, tendo maior maior número de variáveis.

Referências Bibliográficas

- [1] Alizadeh, F., "Interior point methods in semidefinite programming with applications to combinatorial optimization", *SIAM Journal on Optimization*, vol.5(1),pág.13-51,1995.
- [2] Anjos, Miguel F. e Wolkowicz, Henry., "Geometry of Semidefinite Max-Cut Relaxations via Ranks", *Journal of Combinatorial Optimization*, vol.6, pág.237-270,2002.
- [3] Bellman, R. e Fan, K., "On systems of linear inequalities in Hermitian matrix variables", *Proceeding of Symposia in Pure Mathematics of American Mathematical Society*, vol.7,1963.
- [4] Boyd, S., Ghaoui, L., Feron, E. e Balakrishnan, V., "Linear Matrix Inequalities in System and Control Theory", *SIAM*,1994.
- [5] Balas, E., Ceria, S. e Cornuéjols, G., "A lift-and-project cutting plane algorithm for mixed 0-1 programs", *Mathematical Programming*, vol.58,pág. 295-324,1993.
- [6] Balas,E. e Pulleyblank,W.,"The Perfectly Matchable Subgraph Polytope of an Bipartite Graph", *Networks*, vol.13,pág.486-516,1983.
- [7] Barahona,F., Jünger,M. e Reinelt,G.,"Experiments in Quadratic (0,1) Programming", <http://www.optimization-online.org>
- [8] Cullum,J., Donath,W. e Wolfe,P.,"The Minimization of Certain Nondifferentiable sums of Eigenvalues of Symmetric Matrices", *Mathematical Programming Study*, vol.3,pág.35-55,1975.
- [9] Donath,W. e Hoffman,A.,"Lower Bounds for the Partitioning of Graphs", *IBM Journal of Research and Development*, vol.17,pág.420-425,1973.

-
- [10] Fourier, J.B., "Analyse des Travaux l'Academie Royale des Sciences Historiel de 'Academie Royale des Sciences de l'Institute de France", vol.7,1927.
- [11] Goldberg, M. C. e Luna, H. P. L., *Otimização Combinatória e Programação Linear: Modelos e Algoritmos*, Editora Campus, Rio de Janeiro, RJ, 2000.
- [12] Goemans, Michel X. e Williamson, David P., "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming", *Journal of the ACM (JACM)*, vol.42, pág.1115 - 1145, 1995.
- [13] Goemans, Michel X. e Williamson, David P., "Approximation Algorithms for MAX-3-CUT and Other Problems via Complex Semidefinite Programming", *ACM Symposium on Theory of Computing*, 2001.
- [14] Grötschel, M., Lovász, L. e Schrijver, A., *Geometric algorithms and combinatorial optimization*, Springer, 1988.
- [15] Helmberg, C., "Semidefinite Programming in Combinatorial Optimization", *ZIP-Report*, pág.00-34, 2000.
- [16] Helmberg, C., Rendl, F., "Solving Quadratic (0,1)-problems by Semidefinite Programs and Cutting Planes", *Waterloo: University of Waterloo, Department of Combinatorics and Optimization*, 1996.
- [17] Helmberg, C., Rendl, F., Poljak, S., Wolkwicz, H., "Combining Semidefinite and Polyhedral Relaxations for Integer Programs", *Waterloo: University of Waterloo, Department of Combinatorics and Optimization*, 1997.
- [18] Krishnan, K. , Mitchell, J. E. "A Linear Programming Approach to Semidefinite Programming Problems" *Dept. of Mathematical Sciences, RPI, Troy, NY, April 2001 (submitted to SIAM Journal on Optimisation 81)*.
- [19] Krishnan, K., Terlaky, T. "Interior Point and Semidefinite Approaches in Combinatorial Optimization", *AdvOl-Report No 2/2004*.
- [20] Lasserre, J.B., "An explicit exact SDP relaxation for nonlinear 0-1 programs", *IPCO. Lecture Notes in Computer Science*, vol. 2081, pág. 293-303, 2001.

-
- [21] Lasserre, J.B. "Global optimization with polynomials and the problem of moments", *SIAM Journal on Optimization*, vol.11, págs.796-817, 2001.
- [22] Lasserre, J.B. "Optimality conditions and LMI relaxations for 0-1 programs", *Technical Report. LAAS*, vol. 99, 2000.
- [23] Lewis, A.S., Overton, M.L., "Eigenvalue optimization", *Acta No. 5*, págs.149-190, 1996.
- [24] Lovász, L., Schrijver, A., "Cones of matrices and set-functions and 0-1 optimization", *SIAM Journal on Optimization*, vol.1, págs.166-190, 1991.
- [25] Laurent, M., "Semidefinite Relaxations for Max-Cut Problem", *Centrum voor Wiskunde en Informatica*, 2000.
- [26] Laurent, M., Rendl, F., "A comparison of the Sherali-Adams, Lovász-Schrijver and Lasserre relaxations for 0-1 programming", *Probability, Networks and Algorithms - Centrum voor Wiskunde en Informatica*, 2001.
- [27] Laurent, M., Rendl, F., "Semidefinite Programming and integer programming", *Probability, Networks and Algorithms - Centrum voor Wiskunde en Informatica*, 2002.
- [28] Mitchell, J., "Restarting after branching in the SDP approach to MAX-CUT and similar combinatorial optimization problems", *Rensselaer Polytechnic Institute*, 1999.
- [29] Mitchell, J., "Solving Semidefinite Programming Problems Using Branch-and-Bound", *Rensselaer Polytechnic Institute. IPM 2000 and EURO 2000.*, Budapest, 2000.
- [30] Motzkin, T.S., "Contributions to the Theory of linear Inequalities", *RAND Corporation Translation 22*, Santa Monica, 1952.
- [31] Nemhauser, G.L., Wolsey, L.A., "Integer and Combinatorial Optimization", *Wiley Interscience series in discrete mathematics and optimization*, 1988.
- [32] Nesterov, Y., Nemirovskii, A., "Interior-Point Polynomial Algorithms in Convex Programming", *SIAM Publications*, vol. 37, Philadelphia, 1994.

-
- [33] Sherali,H., Adams,W., "A hierarchy of relaxations and convex hull representations for mixed integer zero-one programming problems", *Discrete Applied Mathematics*, vol.52, pág.83-106, 1994.
- [34] Sherali,H., Adams,W., "A hierarchy of relaxations between continuous and convex hull representations for zero-one programming problems", *SIAM Journal on Discrete Mathematics*, vol. 3 ,pág.411-430, 1990.
- [35] Todd,M.J., "A Study of Search Directions in Primal-Dual Interior-Point Methods for Semidefinite Programming", *Optimization Methods and Software*, 1998.
- [36] Todd,M.J., "Semidefinite Optimization", 2001, <http://www.user.tu-chemnitz.de/helmberg/semidef.html>
- [37] Vargas. D., Souza. S.R., "Sobre Teoria de Programação Semidefinida: Aplicações e o Método de Ponto Interior", *V Seminário de Progresso - CEFET-MG*, Belo Horizonte, 2003.
- [38] Vargas. D., Souza. S.R., "O Uso de Programação Semidefinida para Melhores Planos de Corte em Programação 0-1: O Caso dos Métodos Lift-and-Project", *XXVII Congresso Nacional de Matemática Aplicada e Computacional - CNMAC*, Porto Alegre, 2004.
- [39] Vandenberghe,L., Boyd,S. , "Semidefinite Programming", *SIAM Review*, 1996.
- [40] Wolkowicz,H. Saigal, R. Vandenberghe,L., "Handbook on Semidefinite Programming", *Kluwer Academic Publishers*, 2000.
- [41] Zhang,Y., "Interior-Point Methods and Semidefinite Programming", *SIAM - Annual Meeting*, Puerto Rico, 2000.
- [42] Zhang,Y., "On extending some primal-dual interior-point algorithms from linear programming to semidefinite programming", *SIAM J. Optim*, pág. 365-386, 1998.