

Relaxações Contínuas para o Problema da Amostra de Máxima Entropia Restrito e um Algoritmo de Trajetória Central de Passos Longos para Problemas de Programação Semidefinida

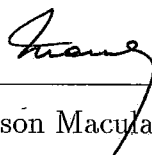
Marcia Helena Costa Fampa

Tese submetida ao corpo docente da Coordenação dos Programas de Pós-Graduação em Engenharia da Universidade Federal do Rio de Janeiro como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências em Engenharia de Sistemas e Computação.

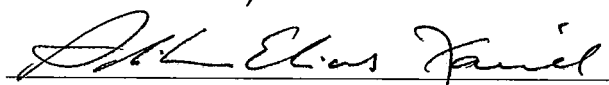
Aprovada por:



Prof. Clóvis C. Gonzaga, D.Sc. (Presidente)




Prof. Nelson Maculan Filho, D.Habil.



Prof. Adilson Elias Xavier, D.Sc.



Prof. Abilio P. de Lucena Filho, Ph.D.



Prof. Alfredo Noel Iussem, Ph.D.

Rio de Janeiro, RJ - Brasil
Novembro de 1996

FAMPA, MARCIA HELENA COSTA

Relaxações Contínuas para o Problema da Amostra de Máxima Entropia Restrito e um Algoritmo de Trajetória Central de Passos Longos para Problemas de Programação Semidefinida [Rio de Janeiro] 1996.

XIII, 113 p., (COPPE/UFRJ, D.Sc., Engenharia de Sistemas e Computação, 1996)

Tese - Universidade Federal do Rio de Janeiro, COPPE

A Marcus, Nathalie e Gabriel.
Aos meus pais Luiz e Cristina.

Agradecimentos

Ao Prof. Clóvis C. Gonzaga, pela orientação desta tese e pela contribuição prestada durante todo o seu desenvolvimento. Agradeço-lhe acima de tudo, pela confiança em mim depositada e pela oportunidade de concretizar o projeto de doutorado sanduiche no exterior. Sem a sua indicação, o trabalho na Universidade de Iowa, que tanto enriqueceu esta tese, provavelmente não teria se concretizado.

Ao Prof. Nelson Maculan Filho, pela co-orientação desta tese e pela participação na banca examinadora do exame de qualificação. Agradeço-lhe também o interesse e a ajuda prestada durante todo o desenvolvimento deste trabalho.

Ao Prof Adilson E. Xavier, pela participação na banca examinadora desta tese e do exame de qualificação. Agradeço-lhe também a colaboração prestada durante o período de defesa do exame de qualificação.

Ao Prof. Kenneth Kortanek, por assentir na minha estadia na Universidade de Iowa através do programa de doutorado sanduiche no exterior. Agradeço-lhe também por, junto a sua esposa Irene Kortanek, fazer-me sentir especialmente bem-vinda à Iowa City.

Ao Prof. Kurt Anstreicher, pela proposta do tema abordado nesta tese e pela excelente orientação do trabalho desenvolvido durante a minha estadia na Universidade de Iowa. Agradeço-lhe também pelo incentivo e interesse demonstrado na finalização desta tese, pela recepção e ajuda quando cheguei em Iowa City, e pela oportunidade de apresentar o trabalho por nós desenvolvido no congresso da SIAM em Victoria, Canadá em maio deste ano.

Ao Prof. Yinyu Ye, pelos conhecimentos transmitidos durante a minha estadia na Universidade de Iowa através do curso oferecido e através do esclarecimento de dúvidas em seu escritório. Agradeço-lhe também pela excelente acomodação no laboratório de otimização da Universidade e pela disponibilidade dos recursos oferecidos.

Ao Prof. Jon Lee, pela proposta de trabalho no tema principal desta dissertação e pelo fornecimento de dados dos problemas testes.

Aos Profs. Abilio Lucena e Alfredo Iusem, por assentirem em compor a banca examinadora.

Aos colegas de escritório Erling Andersen, Xiaojie Xu e Fernanda Raupp, pelo incentivo e companheirismo durante a minha estadia na Universidade de Iowa.

Ao colega Ricardo Duarte, pelas informações prestadas durante a fase final de desenvolvimento da tese.

Aos meus professores na COPPE/UFRJ, pelos conhecimentos transmitidos durante a minha passagem pelo programa de pós-graduação.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo suporte financeiro recebido.

Ao meu esposo e aos meus filhos pelo apoio e solidariedade durante todo o desenvolvimento deste trabalho. Agradeço-lhes também por me incentivarem a optar pelo doutorado sanduiche e por me acompanharem durante toda a minha estadia em Iowa City.

Aos meus pais pelo exemplo de perseverança, pelo incentivo, pela dedicação e pela grande ajuda prestada durante toda a minha pós-graduação, sem os quais a finalização deste trabalho se tornaria ao menos bem mais difícil.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para obtenção do grau de Doutor em Ciências (D.Sc.).

Relaxações Contínuas para o Problema da Amostra de Máxima Entropia Restrito e um Algoritmo de Trajetória Central de Passos Longos para Problemas de Programação Semidefinida

Marcia Helena Costa Fampa
Novembor, 1996

Orientador: Prof. Clóvis C. Gonzaga.

Programa: Engenharia de Sistemas e Computação.

Nós propomos uma nova relaxação não linear para o Problema da Amostra de Máxima Entropia Restrito – o problema de escolher uma submatriz principal $s \times s$, com determinante máximo, de uma matriz definida positiva $n \times n$, sujeito a restrições lineares. Implementamos o algoritmo de *branch-and-bound* para o problema, utilizando a nova relaxação. A performance nos problemas testes é bastante superior a uma implementação anterior que utiliza uma relaxação baseada em autovalores.

Consideramos também um algoritmo primal de trajetória central de passos longos para programação semidefinida. Nosso principal resultado nesta parte do trabalho é a extensão para programação semidefinida, da elegante análise de convergência quadrática de Roos e Vial para programação linear. Para problemas sujeitos a restrições nas quais matrizes $m \times m$ devem ser semidefinidas positivas, obtemos algoritmos com complexidade de $O(m \ln(t))$ ou $O(\sqrt{m} \ln(t))$ iterações para reduzir o *gap* primal–dual inicial por um fator t , dependendo de como o parâmetro da função barreira é reduzido.

Abstract of Thesis presented to COPPE/UFRJ as partial fulfillment of the requirement for the degree of Doctor of Science (D.Sc.).

Continuous Relaxations for Constrained Maximum-Entropy Sampling Problem and a Long-Step Path Following Algorithm for Semidefinite Programming Problems

Marcia Helena Costa Fampa
November, 1996

Thesis Supervisor: Clóvis C. Gonzaga
Department: Systems Engineering and Computer Science

We suggest a new nonlinear relaxation for the Constrained Maximum Entropy Sampling Problem – the problem of choosing the $s \times s$ principal submatrix with maximal determinant from a given $n \times n$ positive definite matrix, subject to linear constraints. We implement a branch-and-bound algorithm for the problem, using the new relaxation. The performance on test problems is far superior to a previous implementation using an eigenvalue-based relaxation.

We also consider a primal long-step path following algorithm for semidefinite programming. Our main result on this research, is that Roos and Vial's elegant analysis of quadratic convergence, for the linear programming case, extends in a very natural way to semidefinite programming. For problems with a semidefiniteness constraint on an $m \times m$ matrix we obtain algorithms with complexities of $O(m \ln(t))$ or $O(\sqrt{m} \ln(t))$ iterations to reduce the initial primal-dual gap by a factor of t , depending on how the barrier parameter is reduced.

Índice

1	Introdução	1
1.1	Motivação	1
1.2	Descrição dos Capítulos	4
1.3	Notação	5
2	O Problema da Amostra de Máxima Entropia	7
2.1	Introdução	7
2.2	Descrição do Problema	7
2.3	Complexidade	8
2.4	Abordagem Computacional	9
3	Obtenção de Limites Superiores para o Problema da Amostra de Máxima Entropia Restrito (CMESP)	11
3.1	Introdução	11
3.2	Uma Relaxação Baseada em Autovalores	12
3.3	Relaxações contínuas baseadas em Programação Não Linear	12
3.3.1	O Problema de Programação Não Linear (NLP)	13
3.3.2	Multiplicando a Matriz de Covariância C por uma Constante Positiva γ	18
3.3.3	Como Obter o Melhor Limite Possível Baseado na Relaxação NLP	18
3.3.4	Como Escolher D e γ	22
3.3.5	Uma Relaxação Baseada num Problema Complementar à NLP	23
3.3.6	Solução Dual para o Problema NLP e Critérios para fixar Variáveis em zero ou um na Solução de CMESP	24
3.4	Resolvendo a Relaxação	26

3.4.1	O Método de Barreira Logarítmica	26
3.4.2	A Trajetória Central	28
3.4.3	O Algoritmo	29
3.4.4	Aplicação do Algoritmo de Barreira Logarítmica ao Problema NLP	30
4	Obtenção de Limites Inferiores para o Problema da Amostra de Máxima Entropia	32
4.1	Introdução	32
4.2	O Problema Irrestrito	32
4.3	O Problema Restrito	33
5	Implementação do Método de Resolução para o CMESP	35
5.1	Introdução	35
5.2	O Algoritmo de <i>Branch-and-Bound</i>	35
5.3	Relação entre os Limites Obtidos pela Relaxação Não Linear para os Problemas Pais e Filhos na Árvore Enumerativa do <i>Branch-and-Bound</i>	39
5.3.1	A Opção “Identidade”	39
5.3.2	A Opção “Diagonal”	44
5.4	Aspectos da Implementação	48
5.5	Resultados Numéricos	67
6	Um Algoritmo de Trajetória Central de Passos Longos para o Problema de Programação Semidefinida	72
6.1	Introdução	72
6.2	O Problema de Programação Semidefinida	74
6.3	A Trajetória Central	76
6.4	Propriedades de Pontos Próximos da Trajetória Central	77
6.5	Complexidade de um Algoritmo de Passos Longos	84
6.6	Viabilidade e Solução Inicial para o Problema de Programação Semidefinida	88
6.6.1	Viabilidade	88
6.6.2	Solução Inicial	89
7	Implementação do Algoritmo de Trajetória Central de Passos Longos para	

o Problema de Programação Semidefinida	91
7.1 Introdução	91
7.2 Aspectos da Implementação	92
7.3 Resultados Numéricos	93
8 Conclusão	103
Referências Bibliográficas	106

Lista de Figuras

5.1	Diagrama do Algoritmo de <i>Branch-and-Bound</i>	38
5.2	Gaps entre soluções ótimas inteiras e limites gerados pelo método “Identidade” e pelo método dos autovalores para os problemas 52/16/s	56
5.3	Gaps entre soluções ótimas inteiras e limites gerados pelo método “Diagonal” e pelo método dos autovalores para os problemas 52/16/s	56
5.4	Gaps entre soluções ótimas inteiras e limites gerados pelo método “Traço Mínimo” e pelo método dos autovalores para os problemas 52/16/s	57
5.5	Gaps entre soluções ótimas inteiras e limites gerados pelo método “Identidade”, utilizando-se relaxações originais e complementares, e pelo método dos autovalores para os problemas 52/16/s	57
5.6	Gaps entre soluções ótimas inteiras e limites gerados pelo método “Diagonal”, utilizando-se relaxações originais e complementares, e pelo método dos autovalores para os problemas 52/16/s	58
5.7	Gaps entre soluções ótimas inteiras e limites gerados pelo método “Traço Mínimo”, utilizando-se relaxações originais e complementares, e pelo método dos autovalores para os problemas 52/16/s	58
5.8	Limites obtidos estabelecendo-se $\gamma = (1 - \alpha)1/d_{max} + (\alpha)1/d_{min}$ para os problemas 52/16/s	59
5.9	Limites obtidos estabelecendo-se $\gamma = (1 - \alpha)1/d_{max} + (\alpha)1/d_{min}$ para os problemas 63/27/s	59
5.10	Limites obtidos estabelecendo-se $\gamma = (1 - \alpha)1/d_{max} + (\alpha)1/d_{min}$ para os problemas 36/27/s	60
5.11	Gaps antes e depois de fixar-se variáveis nos problemas 52/16/s	67
7.1	Aplicação do algoritmo de passos longos ao problema de Traço Mínimo com $n = 36$ e $\epsilon = 10^{-6}$ para diferentes valores de σ	101

Lista de Tabelas

5.1	Limites obtidos para os problemas 52/16/s	50
5.2	Tempo em <i>wall seconds</i> para obtenção dos limites para os problemas 52/16/s	50
5.3	Limites obtidos para os problemas 63/27/s	51
5.4	Tempo em <i>wall seconds</i> para obtenção dos limites para os problemas 63/27/s	52
5.5	Limites obtidos para os problemas 36/27/s	53
5.6	Tempo em <i>wall seconds</i> para obtenção dos limites para os problemas 36/27/s	54
5.7	Tempo de execução em <i>wall seconds</i> com e sem busca linear (BL) para os problemas 52/16/s	63
5.8	Tempo de execução em <i>wall seconds</i> com e sem busca linear (BL) para os problemas 63/27/s	64
5.9	Tempo de execução em <i>wall seconds</i> com e sem busca linear (BL) para os problemas 36/27/s	65
5.10	Fixando variáveis nos problemas 52/16/s	66
5.11	Número de limites calculados na resolução de 50 problemas	68
5.12	Tempo em <i>wall seconds</i> para solucionar-se 50 problemas	69
5.13	Número de limites calculados - Fixando variáveis - na resolução de 50 problemas	71
5.14	Tempo em <i>wall seconds</i> - Fixando variáveis - para solucionar-se 50 problemas	71
7.1	Resultados numéricos para o problema de Traço Mínimo com $n = 36$ e $\epsilon = 10^{-6}$	95
7.2	Resultados numéricos para o problema de Traço Mínimo com $n = 36$ e $\epsilon = 10^{-8}$	96
7.3	Resultados numéricos para o problema de Traço Mínimo com $n = 52$ e $\epsilon = 10^{-6}$	97
7.4	Resultados numéricos para o problema de Traço Mínimo com $n = 52$ e $\epsilon = 10^{-8}$	98
7.5	Resultados numéricos para o problema de Traço Mínimo com $n = 63$ e $\epsilon = 10^{-6}$	99
7.6	Resultados numéricos para o problema de Traço Mínimo com $n = 63$ e $\epsilon = 10^{-8}$	100

7.7	Resultados finais para o problema de Traço Mínimo com $\epsilon = 10^{-8}$	101
7.8	Resultados finais para o problema de Máximo Autovalor com $\epsilon = 10^{-8}$	102

Capítulo 1

Introdução

1.1 Motivação

Um problema fundamental na área de modelagem empírica é o problema de selecionar um subconjunto mais informativo, com tamanho pré-especificado, de um conjunto finito de variáveis aleatórias. Este tipo de problema aparece, por exemplo, nas áreas de monitoria ambiental, meteorológica e geológica, onde variáveis aleatórias são distribuídas no tempo e no espaço (o clássico problema *vertex-packing* de otimização combinatória pode também ser modelado como um destes problemas; veja [48]). Uma vez que amostras podem ser muito dispendiosas, determina-se em geral, inferências com base em apenas algumas das variáveis aleatórias através da observação de um subconjunto delas. Assume-se que algum conhecimento a respeito da função distribuição de probabilidade do conjunto total de variáveis encontra-se disponível. Em geral, este conhecimento é adquirido através de dados históricos e/ou modelos matemáticos.

Em situações reais, existem restrições que complicam a descrição do conjunto de soluções viáveis do problema. Tais restrições podem surgir de limitações no capital de investimento, metodologias científicas, ou podem estar relacionadas a considerações políticas e/ou geográficas.

Como medida da quantidade de informação disponível através de um conjunto de variáveis aleatórias, utilizamos a entropia. Se o conjunto de variáveis aleatórias $Y_n := \{Y_j | j \in N\}$, com $N := \{1, \dots, n\}$, tem função densidade de probabilidade $g_N(\cdot)$, então a entropia de N é dada por $h(N) := -E[\ln g_N(\cdot)]$. E de forma ainda mais específica, no caso em que N tem distribuição Gaussiana com matriz de covariância C , $h(N) := \alpha + \frac{1}{2} \ln(\det(C))$, para alguma constante α que depende de $|N|$ (veja [84]).

O conceito de entropia foi introduzido, em física, por Rudolph Clausius. Posteriormente, Boltzmann [16] formalizou a definição matematicamente, enquanto estabeleceu os fundamentos da mecânica estatística (veja também Jaynes [46]). Shannon [83] e Blackwell [15]

mais tarde desenvolveram e popularizaram o conceito de entropia nos contextos de teoria da informação e estatística, respectivamente. Neste último tópico, Lindley, [56, 57], também pode ser citado.

A aplicação da entropia à modelagem de redes de observação tem sido estudada por Caselton e Zidek [18], Shewry e Wynn [87], Fedorov et al. [30], Fedorov e Mueller [31], Caselton, Kan e Zidek [17], Wu e Zidek [94] e Guttorp et al. [42], entre outros.

Em 1994, Ko, Lee e Queyranne [48] desenvolveram um método computacional para resolver o problema da amostra de máxima entropia, quando nenhuma restrição é imposta. Este método constituiu o primeiro algoritmo para o problema irrestrito a buscar a sua solução ótima e comprovar a sua otimalidade. O método utiliza o clássico algoritmo de *branch-and-bound*, com limites superiores para os subproblemas, calculados com base nos autovalores da matriz de covariância condicional. Mais tarde, Lee [53], estendeu estes limites ao problema restrito utilizando uma abordagem Lagrangeana. Este método no entanto, mostrou-se limitado, devido ao aumento do número de iterações, executadas pelo algoritmo, necessárias à obtenção do limite, a medida que o número de restrições cresce.

Apresentamos neste trabalho uma nova relaxação não linear para o Problema da Amostra de Máxima Entropia Restrito. O algoritmo de *branch-and-bound* para obtenção da solução ótima do problema foi implementado com o uso de limites produzidos por esta relaxação. Os resultados numéricos obtidos através desta implementação mostram-se bem superiores aos obtidos com o uso dos limites baseados nos autovalores da matriz de covariância.

O artigo que apresenta esta nova relaxação e os resultados numéricos obtidos com a sua implementação [11], pode ser encontrado na revista *Lecture Notes in Computer Science*. Quanto a apresentação em congressos, o trabalho foi apresentado recentemente nos seguintes eventos:

- INFORMS Wash. Meeting, DC Spring 1996. Washington, D.C., U.S.A.. 5 a 9 de maio de 1996.
- Fifth SIAM Conference on Optimization. Victoria, British Columbia, Canada. 20 a 22 de maio de 1996.
- Fifth IPCO Conference – Integer Programming and Combinatorial Optimization. University of British Columbia, Vancouver, British Columbia, Canada. 3 a 5 de junho de 1996.
- VIII CLAIO – XXVIII SBPO. Rio de Janeiro, R.J., Brasil. 26 a 28 de agosto de 1996.

Como será visto adiante, uma das etapas de obtenção de limites superiores através da relaxação não linear introduzida, consiste na resolução de um problema de programação semidefinida.

No problema de programação semidefinida minimiza-se uma função linear sujeito a restrição de que uma combinação afim de matrizes simétricas seja semidefinida positiva. Esta

restrição, apesar de não linear e não suave, é convexa. Logo, problemas de programação semidefinida são na verdade uma classe de problemas de otimização convexa.

Apesar do problema de programação semidefinida ser bem mais geral do que o problema de programação linear, sua solução não é muito mais difícil. De fato, ele pode ser visto como uma extensão do problema de programação linear, onde as inequações vetoriais são substituídas por inequações matriciais, ou, de forma equivalente, o primeiro octante é substituído pelo cone de matrizes semidefinidas positivas. Consequentemente, não é inesperado que a teoria desenvolvida para programação linear tenha muitos pontos em comum com a teoria desenvolvida para programação semidefinida, ou que muitos algoritmos que resolvem a primeira possam ser generalizados para resolver a segunda. Existem, entretanto, algumas importantes diferenças a serem consideradas, como por exemplo nos resultados a respeito de dualidade, que são mais fracos no último caso. Uma outra diferença é a inexistência de um método semelhante ao simplex para programação semidefinida.

Existem boas razões para se estudar programação semidefinida. Uma delas é sua aplicação a diversos problemas reais e uma outra é o fato de que muitos problemas de otimização convexa, como por exemplo problemas de programação linear e de programação quadrática sujeitos a restrições quadráticas, podem ser considerados como problemas de programação semidefinida. Sendo assim, programação semidefinida constitui uma maneira unificada de estudar as propriedades e desenvolver algoritmos para a vasta classe de otimização convexa. A principal razão no entanto, é a forma eficiente com que o problema pode ser resolvido, tanto na teoria quanto na prática.

Consideramos neste trabalho, os métodos de pontos interiores para programação semidefinida. A razão do interesse por estes métodos, além da grande atenção que lhes tem sido ultimamente atribuída por pesquisadores, são as seguintes propriedades que lhes tornam particularmente atraentes:

- Eficiência na prática. É atualmente aceito que os métodos de pontos interiores para programação linear são competitivos com o método Simplex e até mesmo mais rápido para problemas com mais de 10.000 variáveis ou restrições (veja, por exemplo [58]). De forma análoga, resultados apresentados na literatura sugerem que métodos de pontos interiores para programação semidefinida são competitivos com outros métodos para problemas pequenos, e são mais rápidos para problemas médios e grandes.
- Eficiência na teoria. A análise de pior caso para métodos de pontos interiores para programação semidefinida mostra que o esforço necessário à resolução de um problema com uma dada precisão é limitado por um polinômio no tamanho do problema.
- Habilidade em explorar a estrutura do problema. O procedimento resolvido a cada iteração do método pode tirar vantagem da estrutura do problema, como por exemplo, da esparsidade.

As propriedades do problema de programação semidefinida já foram estudadas por Bell-

man e Fan em [14]. Outras referências que discutem as suas condições de otimalidade são Craven e Mond [20], ShapiroSha85, Fletcher [34], Allwright [4], Wolkowicz [93] e Kojima, Kojima e Hara [49].

Os métodos de pontos interiores para programação linear foram introduzidos por Karmarkar em 1984 [47], apesar da antecedência de muitos de seus princípios (veja por exemplo [32, 54, 25]). Estes métodos combinam um excelente comportamento na prática com uma complexidade de pior caso, polinomial. O algoritmo de Karmarkar produziu um grande impacto e muitos algoritmos de pontos interiores para programação linear foram desenvolvidos a partir de sua publicação, como pode ser visto por exemplo no *survey* de Gonzaga [41]. Em seguida, os algoritmos de pontos interiores foram estendidos a problemas de programação convexa quadrática e problemas de complementaridade linear (veja, por exemplo, Kojima, Megiddo, Noma e Yoshise [50]).

Um importante resultado foi obtido por Nesterov e Nemirovsky [64, 65, 66, 67]. Eles mostraram que os métodos de pontos interiores para programação linear podem, a princípio, ser generalizados para todos os problemas de otimização convexa. O elemento chave desta análise é o conhecimento de uma função barreira com uma certa propriedade: a auto-concordância. Para ser de utilidade prática a função barreira, assim como suas derivadas de primeira e segunda ordem, devem ser calculáveis.

Os problemas de programação semidefinida constituem uma importante classe de problemas de otimização convexa para os quais são conhecidas funções barreiras auto-concordantes. Logo, os métodos de pontos interiores se aplicam a estes problemas. Uma excelente referência sobre programação semidefinida é Vandenberghe e Boyd [91] e uma excelente referência sobre métodos de pontos interiores para problemas convexos em geral é Den Hertog [22].

Neste trabalho, consideramos um algoritmo primal de trajetória central de passos longos para programação semidefinida. Nosso principal resultado neste estudo, é a extensão de uma forma bastante natural, da elegante análise de convergência quadrática de Roos e Vial [81], do caso de programação linear para o caso de programação semidefinida. Para problemas com inequações matriciais em matrizes $m \times m$, obtemos algoritmos com complexidade de $O(m \ln(t))$ ou $O(\sqrt{m} \ln(t))$ iterações para reduzir o *gap* primal-dual inicial por um fator t , dependendo de como o parâmetro de penalização da função barreira é reduzido.

Os resultados obtidos para programação semidefinida [10] serão submetidos a revista "Optimization Methods and Software" e podem ser atualmente acessados via *anonymous ftp* nos arquivos longsemi.dvi.Z e longsemi.ps no diretório público /pub/papers/ da máquina col.biz.uiowa.edu.

1.2 Descrição dos Capítulos

O trabalho apresentado nesta tese divide-se em duas etapas. A primeira, abordada nos Capítulos 2,3,4 e 5, basicamente apresenta uma nova relaxação para o Problema da Amostra

de Máxima Entropia Restrito (CMESP), e utiliza os limites por ela gerados na obtenção da solução ótima do problema, através do algoritmo de *branch-and-bound*.

O Capítulo 2 apresenta a descrição do problema (CMESP), a análise de sua complexidade e a abordagem computacional que tem sido utilizada para resolvê-lo. No capítulo 3, são inicialmente apresentados os métodos de obtenção de limites superiores utilizados em [48] e em [53]. Em seguida é apresentada a nova relaxação não linear para o problema, que constitui o primeiro principal resultado desta tese. Ainda neste capítulo comentamos a metodologia utilizada para resolver a relaxação. O capítulo 4 contém o procedimento para produção de limites inferiores para o problema, que juntamente com os limites superiores são essenciais para a obtenção da sua solução ótima através do algoritmo de *branch-and-bound*. Por fim, o capítulo 5 descreve o algoritmo de *branch-and-bound*, e aborda detalhadamente os aspectos envolvidos na sua implementação. Os resultados numéricos apresentados ao final do capítulo demonstram a eficácia do método de resolução proposto e sua superioridade em relação ao outro método disponível na literatura.

Na segunda etapa do trabalho, descrita nos Capítulos 6 e 7, é apresentado um estudo sobre um algoritmo primal de trajetória central para o Problema de Programação Semidefinida. No capítulo 5 a análise de complexidade do algoritmo é desenvolvida e apresentada como o segundo principal resultado contido na tese. Finalmente, no Capítulo 7 são mostrados resultados numéricos obtidos com a implementação do algoritmo e os aspectos envolvidos com a implementação.

Para finalizar, o Capítulo 8 comenta os resultados obtidos nesta tese e os projetos de pesquisa que darão continuidade ao estudo realizado.

1.3 Notação

Dados:

- Dois vetores u e v , ambos em R^n , tal que $v \geq 0$;
- Duas matrizes quadradas A e B , $n \times n$;
- Duas matrizes simétricas W e Z , $n \times n$;

Usaremos a seguinte notação nos próximos capítulos:

$diag(u)$ Matriz diagonal com $diag(u)_{ii} := u_i$ para $i = 1, \dots, n$.

$Diag(A)$ Matriz diagonal cujos elementos da diagonal são os elementos da diagonal de A .

$A \circ B$ Produto de Hadamard de A e B : $(A \circ B)_{ij} := a_{ij} b_{ij}$, para $i = 1, \dots, n$, $j = 1, \dots, n$.

$B^{(2)}$ Representa $B \circ B$.

$W \preceq Z$ ou $Z \succeq W$ Representa que $Z - W$ é semidefinida positiva.

$W \prec Z$ ou $Z \succ W$ Representa que $Z - W$ é definida positiva.

V^u Matriz diagonal com $(V^u)_{ii} := v_i^{u_i}$, para $i = 1, \dots, n$.

$\lambda_i(A)$ i -ésimo maior autovalor da matriz A , para $i = 1, \dots, n$.

$\rho(U)$ Raio espectral de U , isto é, $\rho(U) := \max_i |\lambda_i(U)|$, $i = 1, \dots, n$.

I Matriz identidade cuja dimensão será dada pelo contexto.

$\|A\|$ Norma de Frobenius de A , ou seja,

$$\|A\| = \|A\|_{\mathcal{F}} = \left(\sum_{i=1}^n \sum_{j=1}^n A_{ij}^2 \right)^{\frac{1}{2}} = \left(\text{Tr}(A^T A) \right)^{\frac{1}{2}}.$$

Capítulo 2

O Problema da Amostra de Máxima Entropia

2.1 Introdução

O problema de selecionar o subconjunto mais informativo de tamanho pré-estabelecido, de um conjunto de variáveis aleatórias correlacionadas, é um problema fundamental na área de modelagem estatística. Ele aplica-se, por exemplo, a problemas de monitoria ambiental e meteorológica. Nestas aplicações, observações podem ser feitas em diferentes momentos e/ou locais. A informação associada a um subconjunto de variáveis aleatórias é medida pela sua entropia. No caso Gaussiano, o problema é formulado como o problema de maximizar o determinante da matriz de covariância do escolhido subconjunto.

Na prática, restrições podem ser impostas à região viável do problema. Tais restrições podem resultar de limitações no capital de investimento, metodologias científicas, ou podem estar relacionadas a considerações geográficas ou políticas. Desta forma originam-se os problemas da amostra de máxima entropia restritos.

Neste capítulo, o problema será descrito e formulado como um problema de programação não linear inteira. Em seguida veremos que este é um problema NP-difícil e comentaremos a abordagem computacional utilizada na sua resolução.

2.2 Descrição do Problema

Seja n um inteiro positivo. Para $N := \{1, \dots, n\}$, considere $Y_N := \{Y_j \mid j \in N\}$ um conjunto de n variáveis aleatórias, com função densidade de probabilidade $g_N(\cdot)$. Seja s um inteiro que satisfaz $0 < s \leq n$. Para $S \subset N$, $|S| = s$, considere $Y_S := \{Y_j \mid j \in S\}$, e defina $g_S(\cdot)$

como a função densidade de probabilidade marginal de Y_S . A *entropia* de S é definida por

$$h(S) := -E[\ln g_S(Y_S)].$$

Seja m um inteiro não negativo, e seja $M := \{1, 2, \dots, m\}$. O *Problema da Amostra de Máxima Entropia Restrito* é então o problema de escolher um subconjunto S , com $|S| = s$, que maximize a entropia de S , enquanto satisfaz as restrições lineares

$$\sum_{j \in S} a_{ij} \leq b_i, \quad i \in M.$$

Entropia é um conceito atraente porque provê uma medida robusta da informação disponível em um conjunto de variáveis aleatórias. Além disso, para o importante caso em que Y_N tem função distribuição de probabilidade Gaussiana com matrix de covariância C , $h(S)$ tem uma forma facilmente calculável, dada por:

$$h(S) = k_s + \alpha \text{ldet } C[S, S],$$

onde $\text{ldet} := \ln \det$, α e k_s são constantes, e $C[S, T]$ é a submatriz de C com linhas indexadas por S e colunas indexadas por T . Portanto, $C[S, S]$ é a submatriz principal de C indexada por S .

Assim sendo, no caso Gaussiano, podemos formular o problema como:

$$\begin{aligned} \text{CMESP:} \quad & \max \quad \text{ldet } C[\underline{x}, \underline{x}] \\ & \text{s.t.} \quad Ax \leq b \\ & \quad \quad e^T x = s \\ & \quad \quad x_j \in \{0, 1\}, \quad j \in N, \end{aligned}$$

onde $e \in R^n$ é o vetor cujas componentes são todas iguais a um, $b \in R^m$ é o vetor cuja i -ésima componente é dada por b_i , A é a matriz $m \times n$ cujo elemento na linha i e coluna j é dado por a_{ij} , e $\underline{x} \subset N$ denota o suporte de x ; isto é, $\underline{x} = \{j \in N \mid x_j = 1\}$. Assumiremos que a matriz de covariância C é uma matriz definida positiva, isto é, $x^T C x > 0$ para todo $x \in R^n$.

2.3 Complexidade

Ko, Lee e Queeyranne demonstraram em [48] que o *Problema de Máxima Entropia Irrestrito-MESP* ($M = \emptyset$) é um problema NP-difícil até mesmo quando as entradas da matriz de covariância C estão contidas em $\{0, 1, 3n\}$. Para tal, eles consideraram que para um dado grafo não direcionado \mathcal{G} com n nós, os elementos de C fora da diagonal são dados pelos elementos da matriz de incidência de \mathcal{G} ; e os elementos da diagonal de C são iguais a $3n$. Finalmente eles demonstraram que \mathcal{G} contém um conjunto estável se e somente se o valor

ótimo da função objetivo de MESP é igual a $s \ln(3n)$.

Mais tarde, Lee (veja [53]) demonstrou que o *Problema de Máxima Entropia Restrito* é NP-difícil até mesmo quando as variáveis aleatórias são independentes, têm mesma variância, e apenas duas restrições são impostas. A seguir apresentamos a demonstração desenvolvida por Lee.

Teorema 2.3.1 *CMESP é NP-difícil, até mesmo se $C = I$ e $m = 2$.*

Demonstração: Consideramos o problema NP-difícil de particionamento de um conjunto de $n = 2k$ inteiros positivos $\{a_1, a_2, \dots, a_n\}$ em dois subconjuntos com k elementos tendo a mesma soma (veja o *Problema de Partição* em [38], página 223). Podemos reduzir polinomialmente este problema de partição a CMESP, estabelecendo $C = I$, $m = 2$, $a_{1j} = a_j$, $a_{2j} = -a_j$, $b_1 = \sum_{j \in N} a_j/2$, e $b_2 = -b_1$. ■

É interessante notar que quando as variáveis aleatórias são independentes, ou seja, C é uma matriz diagonal, CMESP pode ser visto como o seguinte problema de programação linear inteira:

$$\begin{aligned} \text{ILP:} \quad & \max \quad \sum_{j \in N} \ln(c_{jj}) \cdot x_j \\ & \text{s.t.} \quad Ax \leq b \\ & \quad \quad e^T x = s \\ & \quad \quad x_j \in \{0, 1\}, \quad j \in N, \end{aligned}$$

2.4 Abordagem Computacional

Abordagens computacionais para o *Problema da Amostra de Máxima Entropia Restrito* têm sido concentradas no caso Gaussiano. Abordagens mais antigas se detiveram no caso ir-restrito ($M = \emptyset$), utilizando heurísticas baseadas em permutações e heurísticas do tipo guloso. Ko, Lee e Queyranne desenvolveram o primeiro algoritmo para encontrar soluções ótimas para os problemas irrestritos e provaram sua otimalidade. O método é do tipo *branch-and-bound* (veja, por exemplo [63], para uma apresentação do algoritmo de *branch-and-bound*).

Com a aplicação do algoritmo de *branch-and-bound* ao *Problema da Amostra de Máxima Entropia*, a divisão de um problema em subproblemas é dada ao se fixar índices dentro ou fora de S . Se f elementos F estão fixos em S , e u elementos U estão fixos fora de S , então é necessário escolher $s' := s - f$ elementos S' dentre os $n' := n - f - u$ elementos $N' := N \setminus (F \cup U)$, que maximizem a entropia *condicional* de S' .

Neste ponto vale observar que fixar um elemento fora de S é equivalente a remover as correspondentes linha e coluna da matriz de covariância C .

Por outro lado, ao fixarmos um elemento dentro de S , podemos utilizar o fato de que se $P \subset N, T \subset N$ e $P \cap T = \emptyset$, então

$$\det \begin{pmatrix} C[P, P] & C[P, T] \\ C[T, P] & C[T, T] \end{pmatrix} = \det(C[P, P])\det(\bar{C}),$$

onde \bar{C} é o complemento de Schur dado por $\bar{C} = C[T, T] - C[T, P]C[P, P]^{-1}C[P, T]$.

Portanto, no caso em que fixamos f elementos F em S , a entropia *condicional* é também facilmente tratável, uma vez que a função distribuição de probabilidade condicional resultante para $Y_{N'}$ é Gaussiana, e sua matriz de covariância é

$$C_{N'|F} := C[N', N'] - C[N', F]C[F, F]^{-1}C[F, N']. \quad (2.1)$$

Em outras palavras, a matriz de covariância condicional $C_{N'|F}$ é simplesmente o complemento de Schur de $C[F, F]$ em $C[N \setminus U, N \setminus U]$ e no problema reduzido a ser considerado então, queremos maximizar $\text{ldet}(C[F, F]) + \text{ldet}(C_{N'|F})$.

Capítulo 3

Obtenção de Limites Superiores para o Problema da Amostra de Máxima Entropia Restrito (CMESP)

3.1 Introdução

Para a obtenção de uma solução ótima para o *Problema da Amostra de Máxima Entropia* através da aplicação do algoritmo de *branch-and-bound*, é imprescindível a disponibilidade de limites superiores razoáveis para os subproblemas gerados.

No método de resolução do problema irrestrito proposto por Ko, Lee e Queyranne em [48], limites superiores para os subproblemas gerados pelo algoritmo de *branch-and-bound*, foram calculados com base nos autovalores das matrizes de covariância condicional. Mais tarde, Lee estendeu estes limites ao problema restrito utilizando uma abordagem Lagrangeana.

O propósito deste capítulo é abordar os limites superiores apresentados nestes trabalhos e propor uma nova maneira de calcular limites para os subproblemas através de relaxações de programação não linear. Analisaremos os problemas de programação não linear de forma a finalmente apresentar quais as melhores escolhas dentre as possíveis relaxações para a obtenção dos melhores limites possíveis, ou seja, dos limites mais próximos das soluções ótimas dos problemas de programação inteira associados. Veremos também como é possível fixar variáveis no problema original CMESP nos valores zero ou um, com base em soluções duais. Este procedimento pode agilizar bastante a obtenção da solução ótima do problema, uma vez que o número de nós da árvore enumerativa do *branch-and-bound* pode ser muito reduzido. Finalmente apresentaremos o método de pontos interiores utilizado na resolução da relaxação de programação não linear e comentaremos sobre as vantagens da utilização deste método.

3.2 Uma Relaxação Baseada em Autovalores

Ko, Lee e Queyranne [48] observaram que um limite superior para a função objetivo em CMESP, para dados conjuntos de elementos F e U fixos dentro e fora de S respectivamente, pode ser dado por

$$\text{ldet } C[F, F] + \sum_{i=1}^{s'} \ln \lambda_i (C_{N \setminus F}), \quad (3.1)$$

onde $C_{N \setminus F}$ é definida em (2.1).

No caso irrestrito, os limites gerados por (3.1) são suficientemente bons para permitir a obtenção de uma solução ótima para problemas reais de tamanho moderado. Infelizmente, entretanto, não existe uma maneira de incorporar informações a respeito de restrições usando os limites dados por (3.1), e este método portanto, não é muito eficiente quando $M \neq \emptyset$.

Lee [53] estendeu os limites baseados em (3.1) para os problemas restritos usando uma abordagem Lagrangeana. Minimizar o Lagrangeano requer a aplicação de um algoritmo de programação não linear. Cada iteração do método de minimização requer a resolução de um sistema de autovalores, para que se obtenha o valor da função objetivo e do gradiente do Lagrangeano. (O valor da função objetivo é dado pela aplicação do limite baseado em autovalores para problemas irrestritos, a uma matriz de covariância condicional simetricamente escalonada, onde o escalonamento é baseado nos multiplicadores de Lagrange para as restrições. Um gradiente, ou subgradiente, é então obtido dos autovetores). A maior limitação deste método é dada pelo crescimento no número de iterações do algoritmo de programação não linear necessárias à minimização aproximada do Lagrangeano, quando o número de restrições aumenta.

3.3 Relaxações contínuas baseadas em Programação Não Linear

Nesta seção apresentamos uma nova forma de obtenção de um limite superior para o *Problema da Amostra de Máxima Entropia Restrito* baseado numa relaxação não linear.

A seguir definiremos o problema de programação não linear utilizado como relaxação. Em seguida serão impostas condições a respeito dos parâmetros contidos na formulação da relaxação de forma a torná-la um problema tratável, e mostraremos como escolher tais parâmetros de forma a obter um bom limite. Definiremos ainda uma segunda relaxação que pode ser vista como um problema “complementar” à primeira. Finalmente apresentaremos critérios para fixar variáveis em 0 ou 1 e discutiremos o procedimento utilizado na resolução da relaxação.

3.3.1 O Problema de Programação Não Linear (NLP)

Considere as constantes $d_j > 0$ e $p_j \geq 1$ para $j \in N$. Seja $D := \text{diag}(d)$, $P := \text{diag}(p)$. Para $x \in \mathbb{R}^n$ não negativo, seja $X := \text{diag}(x)$. Defina

$$f(x) := \text{ldet } M(x), \quad \text{onde } M(x) := X^{p/2}(C - D)X^{p/2} + D^x. \quad (3.2)$$

Note que:

- $x_i = 1$ significa que a linha i e a coluna i de $C - D$ são multiplicadas por um, e o termo da diagonal de $M(x)$ é $(c_{ii} - d_i) + d_i = c_{ii}$. Logo, os elementos da i -ésima linha e coluna de $M(x)$ são exatamente os elementos da i -ésima linha e coluna de C .
- $x_i = 0$ significa que a linha i e a coluna i de $C - D$ são multiplicadas por zero, e o termo da diagonal é $d_i^0 = 1$. Logo, os elementos da i -ésima linha e coluna de $M(x)$ são todos iguais a zero, com exceção do elemento da diagonal que é igual a um.

Portanto, podemos concluir dos dois comentários acima que, se $x \in \{0, 1\}$, então

$$f(x) = \text{ldet } C[\underline{x}, \underline{x}].$$

Sendo assim, consideramos a seguinte relaxação:

$$\begin{aligned} \text{NLP:} \quad & \max f(x) \\ & \text{s.a. } Ax \leq b \\ & e^T x = s \\ & 0 \leq x \leq e. \end{aligned}$$

A relaxação NLP é motivada pela observação de que se C é diagonal (caso em que as variáveis aleatórias são independentes), então a atribuição $D = C$ leva a seguinte função objetivo linear:

$$\sum_{j \in N} \ln(c_{jj}) \cdot x_j,$$

a qual é a mesma função objetivo do problema ILP definido na Seção 2.3. No caso irrestrito ($M = \emptyset$), NLP será então um problema trivial de programação linear que tem como solução ótima um vetor 0/1 que é também solução ótima do problema original da amostra de máxima entropia.

Para que NLP seja útil do ponto de vista computacional, ele deve ser um problema tratável. Em particular, $f(\cdot)$ deve ser côncava na região viável de NLP.

3.3.1.1 Gradiente e Hessiana de $f(\cdot)$

Nesta seção iremos fornecer as expressões para o gradiente e para a Hessiana da função $f(\cdot)$ num determinado ponto x .

Primeiramente, no entanto, apresentaremos dois lemas conhecidos, a respeito das funções $ldet(V)$ e V_{ii}^{-1} . Os resultados apresentados nestes lemas são utilizados não somente no desenvolvimento das expressões para o gradiente e a Hessiana de $f(\cdot)$, como também em várias outras demonstrações apresentadas na tese.

Lema 3.3.1 *Seja V uma matriz $n \times n$ simétrica e não singular. Então*

$$\frac{\partial}{\partial v_{ij}} ldet(V) = (V^{-1})_{ij}. \quad (3.3)$$

Demonstração: Por definição

$$\frac{\partial}{\partial v_{ij}} ldet(V) = \lim_{\lambda \rightarrow 0} \frac{ldet(V + \lambda e_i e_j^T) - ldet(V)}{\lambda}, \quad (3.4)$$

onde e_i e e_j denotam os vetores unitários com componentes i e j iguais a um respectivamente. No entanto, $V + \lambda e_i e_j^T = V(I + \lambda V^{-1} e_i e_j^T)$, logo

$$\begin{aligned} ldet(V + \lambda e_i e_j^T) &= ldet(V) + ldet(I + \lambda V^{-1} e_i e_j^T) \\ &= ldet(V) + \ln(1 + \lambda V_{ij}^{-1}), \end{aligned} \quad (3.5)$$

onde a última igualdade usa o fato conhecido de que $\det(I + uv^T) = 1 + u^T v$. Substituindo (3.5) em (3.4),

$$\frac{\partial}{\partial v_{ij}} ldet(V) = \lim_{\lambda \rightarrow 0} \frac{\ln(1 + \lambda V_{ij}^{-1})}{\lambda} = \left. \frac{d}{d\lambda} \right|_{\lambda=0} \ln(1 + \lambda V_{ij}^{-1}) = V_{ij}^{-1}.$$

■

Lema 3.3.2 *Seja V uma matriz $n \times n$ simétrica e não singular. Então*

$$\frac{\partial}{\partial v_{jj}} V_{ii}^{-1} = - (V_{ij}^{-1})^2.$$

Demonstração: Por definição,

$$\frac{\partial}{\partial v_{jj}} V_{ii}^{-1} = \lim_{\lambda \rightarrow 0} \frac{(V + \lambda e_j e_j^T)_{ii}^{-1} - V_{ii}^{-1}}{\lambda}. \quad (3.6)$$

Usando a expressão de Sherman–Morrison–Woodbury, temos

$$(V + \lambda e_j e_j^T)^{-1} = V^{-1} - \frac{\lambda V^{-1} e_j e_j^T V^{-1}}{1 + \lambda V_{jj}^{-1}}. \quad (3.7)$$

Substituído (3.7) em (3.6),

$$\frac{\partial}{\partial v_{jj}} V_{ii}^{-1} = \lim_{\lambda \rightarrow 0} - \frac{e_i^T V^{-1} e_j e_j^T V^{-1} e_i}{1 + \lambda V_{jj}^{-1}} = - (V_{ij}^{-1})^2.$$

■

A seguir, os dois próximos lemas caracterizam então o gradiente e a Hessiana de $f(x) = \text{ldet } M(x)$, onde $M(x) = X^{p/2}(C - D)X^{p/2} + D^x$, C é definida positiva, $d > 0$, e $D = \text{diag}(d)$.

Lema 3.3.3 *Seja $g(\cdot) = \nabla f(\cdot)^T$ o gradiente de $f(\cdot)$. Então para $x > 0$,*

$$g_i(x) = \frac{p_i}{x_i} + \frac{d_i^{x_i} [\ln(d_i^{x_i}) - p_i]}{x_i} M_{ii}^{-1}(x).$$

Demonstração: Temos:

$$f(x) = \text{ldet } M(x) = \text{ldet } X^{p/2}(C - D + X^{-p} D^x) X^{p/2} = \sum_{j=1}^n p_j \ln(x_j) + \text{ldet}(C - D + X^{-p} D^x),$$

portanto,

$$g_i(x) = \frac{p_i}{x_i} + \frac{\partial}{\partial x_i} \text{ldet}(C - D + X^{-p} D^x). \quad (3.8)$$

Usando a regra da cadeia e o Lema 3.3.1 ,

$$\frac{\partial}{\partial x_i} \text{ldet}(C - D + X^{-p} D^x) = (C - D + X^{-p} D^x)_{ii}^{-1} \frac{\partial}{\partial x_i} \frac{d_i^{x_i}}{x_i^{p_i}}, \quad (3.9)$$

Diferenciando então obtemos

$$\frac{\partial}{\partial x_i} \frac{d_i^{x_i}}{x_i^{p_i}} = \frac{d_i^{x_i} [\ln(d_i^{x_i}) - p_i]}{x_i^{p_i+1}}. \quad (3.10)$$

Substituído (3.9) e (3.10) em (3.8),

$$\begin{aligned} g_i(x) &= \frac{p_i}{x_i} + (C - D + X^{-p} D^x)_{ii}^{-1} \frac{d_i^{x_i} [\ln(d_i^{x_i}) - p_i]}{x_i^{p_i+1}} \\ &= \frac{p_i}{x_i} + \frac{d_i^{x_i} [\ln(d_i^{x_i}) - p_i]}{x_i} M_{ii}^{-1}(x). \end{aligned} \quad (3.11)$$

■

Lema 3.3.4 Defina $l(x) \in \mathbb{R}^n$ por $l_i(x) = p_i - \ln(d_i^{x_i})$. Seja $L(x) = \text{diag}(l(x))$, $P = \text{diag}(p)$. Logo para $x > 0$ a Hessiana de $f(\cdot)$ é:

$$\begin{aligned} \nabla^2 f(x) &= X^{-1}P \left[D^x \text{Diag}(M^{-1}(x)) - I \right] X^{-1} \\ &+ X^{-1}L(x) \left[D^x \text{Diag}(M^{-1}(x)) - D^x(M^{-1}(x))^{(2)}D^x \right] L(x)X^{-1}. \end{aligned} \quad (3.12)$$

Demonstração: Usando (3.11), a regra da cadeia, e Lema 3.3.2, para $j \neq i$ obtemos

$$\begin{aligned} \frac{\partial g_i(x)}{\partial x_j} &= -\frac{d_i^{x_i} [\ln(d_i^{x_i}) - p_i]}{x_i^{p_i+1}} \left[(C - D + X^{-p}D^x)_{ij}^{-1} \right]^2 \frac{\partial}{\partial x_j} \frac{d_j^{x_j}}{x_j^{p_j}} \\ &= -\frac{d_i^{x_i} [\ln(d_i^{x_i}) - p_i]}{x_i^{p_i+1}} \frac{d_j^{x_j} [\ln(d_j^{x_j}) - p_j]}{x_j^{p_j+1}} \left[(C - D + X^{-p}D^x)_{ij}^{-1} \right]^2 \\ &= -\frac{d_i^{x_i} d_j^{x_j} l_i(x) l_j(x)}{x_i x_j} \left[M_{ij}^{-1}(x) \right]^2, \end{aligned} \quad (3.13)$$

onde a segunda igualdade usa (3.10). De forma equivalente, para $j = i$ obtemos

$$\frac{\partial g_i(x)}{\partial x_i} = -\frac{p_i}{x_i^2} - \left(\frac{d_i^{x_i} l_i(x)}{x_i} \right)^2 \left[M_{ii}^{-1}(x) \right]^2 + (C - D + X^{-p}D^x)_{ii}^{-1} \frac{\partial}{\partial x_i} \frac{d_i^{x_i} [\ln(d_i^{x_i}) - p_i]}{x_i^{p_i+1}} \quad (3.14)$$

Diferenciando então obtemos

$$\frac{\partial}{\partial x_i} \frac{d_i^{x_i} [\ln(d_i^{x_i}) - p_i]}{x_i^{p_i+1}} = \frac{d_i^{x_i}}{x_i^{p_i+2}} \left(p_i + [\ln(d_i^{x_i}) - p_i]^2 \right). \quad (3.15)$$

Substitutindo (3.15) em (3.14), e simplificando, temos

$$\frac{\partial g_i(x)}{\partial x_i} = -\frac{p_i}{x_i^2} - \left(\frac{d_i^{x_i} l_i(x)}{x_i} \right)^2 \left[M_{ii}^{-1}(x) \right]^2 + \frac{d_i^{x_i}}{x_i^2} \left(p_i + [l_i(x)]^2 \right) M_{ii}^{-1}(x). \quad (3.16)$$

Finalmente o lema resulta de (3.13) e (3.16). ■

3.3.1.2 Condições para a concavidade de $f(\cdot)$

No início desta seção comentamos sobre a necessidade de se ter $f(\cdot)$ uma função côncava na região viável do problema NLP para que este seja um problema tratável. A seguir, mostraremos que este é o caso quando D e p são escolhidos apropriadamente.

Lema 3.3.5 Considere $p_i \geq 1$ e $0 < d_i \leq \exp(p_i)$ para $i \in N$. Então $M(x)$ é definida positiva para $0 \leq x \leq e$.

Demonstração: $M(0) = I$ é definida positiva, e portanto $M(x)$ é definida positiva para todo x em alguma vizinhança de $x = 0$. Para $x > 0$, $X^{p/2}CX^{p/2}$ é certamente definida positiva. Agora observe que se $0 < d_i \leq \exp(p_i)$ e $0 < x_i \leq 1$, temos:

$$\begin{aligned} \ln(x_i) &\leq x_i - 1 \\ \Rightarrow p_i \ln(x_i) &\leq p_i(x_i - 1) \end{aligned} \quad (3.17)$$

E,

$$\begin{aligned} \ln(d_i) &\leq p_i \\ \Rightarrow \ln(d_i)(x_i - 1) &\geq p_i(x_i - 1) \end{aligned} \quad (3.18)$$

Logo, por (3.17) e (3.18), temos:

$$\begin{aligned} p_i \ln(x_i) &\leq \ln(d_i)(x_i - 1) \\ \Rightarrow \ln(x_i^{p_i}) &\leq x_i \ln(d_i) - \ln(d_i) \\ \Rightarrow \ln(x_i^{p_i} d_i) &\leq \ln(d_i^{x_i}) \\ \Rightarrow d_i^{x_i} &\geq d_i x_i^{p_i} \\ \Rightarrow D^x &\succeq X^{p/2} D X^{p/2} \end{aligned}$$

Segue-se que $M(x) \succeq X^{p/2} C X^{p/2}$ para $x > 0$, e portanto $M(x)$ é definida positiva para $0 \leq x \leq e$. \blacksquare

Lema 3.3.5 implica em $M(x)$ ser não singular para $0 \leq x \leq e$. Baseados neste fato, mostraremos agora que para determinados valores de D e p , $f(\cdot)$ é côncava para $0 < x \leq e$. Para tanto, considere $l(x)$ definida na seção anterior e $L(x) = \text{diag}(l(x))$. Note que as condições do Lema 3.3.5 implicam em $l(x) > 0$ para $0 \leq x \leq e$.

Teorema 3.3.6 *Assuma que $D \succeq C$, $p_j \geq 1$ e $0 < d_j \leq \exp(p_j - \sqrt{p_j})$, $j \in N$. Então $f(\cdot)$ é côncava para $0 < x \leq e$.*

Demonstração: Temos por construção que $M(x) \preceq D^x$ para $0 \leq x \leq e$, o qual implica em (veja [45, Seção 7.7])

$$M^{-1}(x) \succeq D^{-x}. \quad (3.19)$$

Segue de (3.19) que (veja [45, Observação 7.7.2])

$$D^x M^{-1}(x) D^x \succeq D^x. \quad (3.20)$$

Portanto, (3.19) e (3.20) juntos implicam em (veja [45, Seção 7.7])

$$D^x M^{-1}(x) D^x \circ (M^{-1}(x) - D^{-x}) \succeq D^x \circ (M^{-1}(x) - D^{-x}),$$

o qual pode ser reescrito como

$$D^x (M^{-1}(x))^{(2)} D^x - D^x \text{Diag}(M^{-1}(x)) \succeq D^x \text{Diag}(M^{-1}(x)) - I. \quad (3.21)$$

Entretanto, (3.19) também implica em

$$D^x \text{Diag}(M^{-1}(x)) = D^x \circ M^{-1}(x) \succeq D^x \circ D^{-x} = I, \quad (3.22)$$

logo $[D^x \text{Diag}(M^{-1}(x)) - I]$ é uma matriz diagonal semidefinida positiva. A seguir, note que as condições $d_i \leq \exp(p_i - \sqrt{p_i})$ e $p_i \geq 1$ juntas com $0 \leq x_i \leq 1$ implicam em:

- Se $\ln(d_i) > 0 \Rightarrow x_i \ln(d_i) \leq \ln(d_i) \leq p_i - \sqrt{p_i}$. Portanto, $p_i - \ln(d_i^{x_i}) \geq \sqrt{p_i}$, ou seja, $l_i(x) \geq \sqrt{p_i}$.
- Se $\ln(d_i) \leq 0 \Rightarrow l_i(x) = p_i - x_i \ln(d_i) \geq p_i \geq \sqrt{p_i}$.

Segue que $l_i(x) \geq \sqrt{p_i}$ para $0 \leq x_i \leq 1$, e portanto

$$L(x) [D^x \text{Diag}(M^{-1}(x)) - I] L(x) \succeq P [D^x \text{Diag}(M^{-1}(x)) - I]. \quad (3.23)$$

Sendo assim, (3.21) e (3.23) juntas implicam em

$$L(x)[D^x(M^{-1}(x))^{(2)}D^x - D^x \text{Diag}(M^{-1}(x))]L(x) \succeq P [D^x \text{Diag}(M^{-1}(x)) - I]$$

e o teorema segue imediatamente de (3.12), uma vez que $\nabla^2 f(x) \preceq 0$. ■

3.3.2 Multiplicando a Matriz de Covariância C por uma Constante Positiva γ

Podemos observar que multiplicando a matriz de covariância C por uma constante positiva γ no problema original CMESP, simplesmente adicionamos $s \ln(\gamma)$ a função objetivo, para qualquer x viável. Esta multiplicação ou mudança de escala pode também ser aplicada a NLP. Particularmente, definiremos

$$\begin{aligned} f_\gamma(x) &= \text{ldet } M_\gamma(x) - s \ln(\gamma), \\ \text{onde } M_\gamma(x) &= \gamma X^{p/2}(C - D)X^{p/2} + (\gamma D)^x. \end{aligned} \quad (3.24)$$

Sendo assim, $f(x) = f_\gamma(x)$ para todo x que satisfaz $e^T x = s$, $x_j \in \{0, 1\}$, $j \in N$. Podemos portanto, utilizar $f_\gamma(x)$ ao invés de $f(x)$ na relaxação não linear. Esta mudança de escala é na verdade bastante relevante uma vez que, em geral, o limite obtido de NLP depende do fator γ .

3.3.3 Como Obter o Melhor Limite Possível Baseado na Relaxação NLP

A partir do Teorema 3.3.6, concluímos que qualquer D , p e d_j tal que $D \succeq C$, $p_j > 1$ e $0 < d_j \leq \exp(p_j - \sqrt{p_j})$, $j \in N$; podem ser usados para se obter $f(\cdot)$ côncava na região viável

de NLP. A seguir mostraremos que limitando mais a escolha de D , p e γ , podemos melhorar a qualidade do limite obtido. Primeiramente veremos que para D fixo, tal que $D \succeq C$, o melhor limite baseado em NLP será obtido usando-se o menor p que satisfaça as condições do Teorema 3.3.6.

Lema 3.3.7 *Suponha que p e d satisfazem as condições do Teorema 3.3.6, e $p' \geq p$. Seja $f'(\cdot)$ definida como em (3.2), substituindo-se p por p' . Então, $f'(x) \geq f(x)$ para todo $0 < x \leq e$.*

Demonstração: Consideraremos $f(x)$ uma função de p , e mostraremos que para qualquer $0 < x \leq e$, e para todo i ,

$$\frac{\partial}{\partial p_i} f(x) \geq 0. \quad (3.25)$$

Note que podemos escrever

$$f(x) = \sum_{j=1}^n p_j \ln(x_j) + \text{ldet}(C - D + X^{-p} D^x),$$

e portanto

$$\frac{\partial}{\partial p_i} f(x) = \ln(x_i) + \frac{\partial}{\partial p_i} \text{ldet}(C - D + X^{-p} D^x). \quad (3.26)$$

Usando a regra da cadeia e (3.3),

$$\begin{aligned} \frac{\partial}{\partial p_i} \text{ldet}(C - D + X^{-p} D^x) &= (C - D + X^{-p} D^x)_{ii}^{-1} \frac{d}{dp_i} d_i^{x_i} x_i^{-p_i} \\ &= -[\ln(x_i)] d_i^{x_i} x_i^{-p_i} (C - D + X^{-p} D^x)_{ii}^{-1} \\ &= -[\ln(x_i)] d_i^{x_i} M_{ii}^{-1}(x). \end{aligned} \quad (3.27)$$

Substituindo (3.27) em (3.26) temos então

$$\frac{\partial}{\partial p_i} f(x) = \ln(x_i) \left(1 - d_i^{x_i} M_{ii}^{-1}(x)\right). \quad (3.28)$$

No entanto, (3.22) implica em $d_i^{x_i} M_{ii}^{-1}(x) \geq 1$ para todo i , e consequentemente (3.28) implica em (3.25). \blacksquare

Considerando agora também o problema escalonado por γ , podemos concluir do Lema 3.3.7, que para qualquer $D \succeq C$, e constante γ , o melhor limite é obtido usando-se

$$p_j(\gamma) = \begin{cases} 1 & \text{if } \gamma d_j \leq 1, \\ \left(1 + \sqrt{1 + 4 \log(\gamma d_j)}\right)^2 / 4 & \text{if } \gamma d_j > 1. \end{cases} \quad (3.29)$$

Particularmente, se $\gamma D \preceq I$, então $p = e$ é a escolha que leva ao melhor limite possível.

Nos dois próximos lemas mostraremos ainda que para obter o melhor limite possível de NLP, não devemos ter $\gamma D \prec I$, nem $\gamma D \succ I$.

Lema 3.3.8 *Assuma que $C \preceq D \preceq I$, e $p = e$. Então $f_\gamma(x) \geq f(x)$ para todo $0 \leq x \leq e$, $e^T x = s$, e $0 < \gamma \leq 1$.*

Demonstração: É suficiente mostrarmos que

$$\frac{d}{d\gamma} f_\gamma(x) \leq 0,$$

para todo $0 < \gamma \leq 1$. Usando (3.24) temos

$$\frac{d}{d\gamma} f_\gamma(x) = \frac{d}{d\gamma} \text{ldet } M_\gamma(x) - \frac{s}{\gamma}. \quad (3.30)$$

Ao invés de trabalharmos diretamente com $M_\gamma(\cdot)$, é conveniente considerarmos na análise, o produto de γ por cada linha e coluna de C e D separadamente. Conseqüentemente, considere $w > 0$ como um vetor em R^n , $W := \text{diag}(w)$, e defina

$$\begin{aligned} M_w(x) &:= X^{e/2} W^{e/2} (C - D) W^{e/2} X^{e/2} + (WD)^x \\ &= W^{e/2} \left(X^{e/2} (C - D) X^{e/2} + W^{x-e} D^x \right) W^{e/2}. \end{aligned}$$

Logo,

$$\text{ldet } M_w(x) = \sum_{j=1}^n \ln(w_j) + \text{ldet}(X^{e/2} (C - D) X^{e/2} + W^{x-e} D^x).$$

Usando a regra da cadeia e (3.3), obtemos

$$\begin{aligned} \frac{\partial}{\partial w_i} \text{ldet } M_w(x) &= \frac{1}{w_i} + (X^{e/2} (C - D) X^{e/2} + W^{x-e} D^x)^{-1}_{ii} d_i^{x_i} (x_i - 1) w_i^{x_i - 2} \\ &= \frac{1}{w_i} (1 + (x_i - 1) [M_w^{-1}(x)]_{ii} (w_i d_i)^{x_i}). \end{aligned}$$

A seguir, o análogo a (3.22) para $M_w(x)$ implica em:

$$M_x(x) \preceq (WD)^x \Rightarrow M_w^{-1}(x) \succeq (WD)^{-x} \Rightarrow (WD)^x \text{Diag}(M_w^{-1}(x)) \succeq I \Rightarrow [M_w^{-1}(x)]_{ii} (w_i d_i)^{x_i} \geq 1$$

Portanto,

$$\begin{aligned} (x_i - 1) [M_w^{-1}(x)]_{ii} (w_i d_i)^{x_i} &\leq x_i - 1 \\ \frac{\partial}{\partial w_i} \text{ldet } M_w(x) &\leq \frac{1}{w_i} (1 + x_i - 1) \\ &= \frac{x_i}{w_i} \end{aligned}$$

Finalmente, $M_\gamma(x) = M_w(x)$ para $w = \gamma e$, logo

$$\frac{d}{d\gamma} \text{ldet } M_\gamma(x) = \sum_{i=1}^n \frac{\partial}{\partial w_i} \text{ldet } M_w(x) |_{w=\gamma e} \leq \frac{s}{\gamma}, \quad (3.31)$$

onde usamos o fato de que $e^T x = s$. A prova se completa com a combinação de (3.30) e de (3.31). ■

Lema 3.3.9 *Suponha que $C \preceq D$, e $D \succeq I$. Então $f_\gamma(x) \geq f(x)$ para qualquer $0 < x \leq e$, $e^T x = s$, e $\gamma \geq 1$, onde $p = p(\gamma)$ é dado por (3.29) para todo γ .*

Demonstração: É suficiente mostrar que

$$\frac{d}{d\gamma} f_\gamma(x) \geq 0,$$

for $\gamma \geq 1$. Como na prova do Lema 3.3.8 consideramos o vetor w , onde agora $w \geq e$, e consideramos

$$\begin{aligned} M_w(x) &:= X^{p(w)/2} W^{e/2} (C - D) W^{e/2} X^{p(w)/2} + (WD)^x \\ &= X^{p(w)/2} W^{e/2} (C - D + X^{-p(w)} W^{x-e} D^x) W^{e/2} X^{p(w)/2}, \end{aligned}$$

onde $p(w)$ é definido por

$$p_i(w) := \left(1 + \sqrt{1 + 4 \ln(w_i d_i)}\right)^2 / 4, \quad i = 1, \dots, n. \quad (3.32)$$

Segue que

$$\text{ldet } M_w(x) = \sum_{j=1}^n \ln(w_j) + \sum_{j=1}^n p_j(w) \ln(x_j) + \text{ldet}(C - D + X^{-p(w)} W^{x-e} D^x).$$

Usando a regra da cadeia e (3.3), obtemos

$$\begin{aligned} \frac{\partial}{\partial w_i} \text{ldet } M_w(x) &= \frac{1}{w_i} + \ln(x_i) \frac{\partial p_i(w)}{\partial w_i} \\ &\quad + (C - D + X^{-p(w)} W^{x-e} D^x)_{ii}^{-1} d_i^{x_i} \frac{\partial}{\partial w_i} \left(x_i^{-p_i(w)} w_i^{x_i-1} \right). \end{aligned} \quad (3.33)$$

A seguir, diferenciando em relação a w_i obtemos

$$\frac{\partial}{\partial w_i} x_i^{-p_i(w)} w_i^{x_i-1} = x_i^{-p_i(w)} w_i^{x_i-1} \left(\frac{x_i - 1}{w_i} - \ln(x_i) \frac{\partial p_i(w)}{\partial w_i} \right). \quad (3.34)$$

Substituindo (3.34) em (3.33), e simplificando, temos então

$$\begin{aligned} \frac{\partial}{\partial w_i} \text{ldet } M_w(x) &= \frac{1}{w_i} + \ln(x_i) \frac{\partial p_i(w)}{\partial w_i} \\ &\quad + [M_w^{-1}(x)]_{ii} (w_i d_i)^{x_i} \left(\frac{x_i - 1}{w_i} - \ln(x_i) \frac{\partial p_i(w)}{\partial w_i} \right). \end{aligned} \quad (3.35)$$

A seguir, para $p(w)$ dado como em (3.32), temos

$$\begin{aligned} \frac{\partial p_i(w)}{\partial w_i} &= \frac{2 \left(1 + \sqrt{1 + 4 \ln(w_i d_i)}\right)}{4} \cdot \frac{1}{2\sqrt{1 + 4 \ln(w_i d_i)}} \cdot \frac{4}{w_i d_i} \cdot d_i \\ &= \frac{2\sqrt{p_i(w)}}{w_i(2\sqrt{p_i(w)} - 1)} \end{aligned} \quad (3.36)$$

Usando (3.36), obtemos

$$\begin{aligned}
\frac{x_i - 1}{w_i} - \ln(x_i) \frac{\partial p_i(w)}{\partial w_i} &= \frac{1}{w_i} \left(x_i - 1 - \ln(x_i) \frac{2\sqrt{p_i(w)}}{2\sqrt{p_i(w)} - 1} \right) \\
&= \frac{1}{w_i(2\sqrt{p_i(w)} - 1)} (2\sqrt{p_i(w)}[x_i - 1 - \ln(x_i)] + 1 - x_i) \\
&\geq 0,
\end{aligned} \tag{3.37}$$

onde a inequação segue de $x_i \leq 1$ e de $\ln(x_i) \leq x_i - 1$ para $x_i > 0$. Finalmente, o análogo de (3.22) para $M_w(x)$ implica em $[M_w^{-1}(x)]_{ii} (w_i d_i)^{x_i} \geq 1$, e este fato combinado com (3.35) e (3.37) nos dá

$$\frac{\partial}{\partial w_i} \text{ldet } M_w(x) \geq \frac{1}{w_i} + \ln(x_i) \frac{\partial p_i(w)}{\partial w_i} + \frac{x_i - 1}{w_i} - \ln(x_i) \frac{\partial p_i(w)}{\partial w_i} = \frac{x_i}{w_i}.$$

O restante da prova é análogo à prova do Lema 3.3.8. ■

3.3.4 Como Escolher D e γ

Os resultados da seção anterior sugerem algumas estratégias razoáveis para a escolha de D e γ , de forma a obter um bom limite da relaxação NLP. Por exemplo:

1. Seja ρ o maior autovalor de C , $D = \rho I$, $\gamma = 1/\rho$.
2. Seja $D_0 = \text{Diag}(C)$. Seja $\rho > 0$ o menor número tal que $\rho D_0 - C$ é semidefinida positiva. (Logo, ρ é o maior autovalor de $D_0^{-1/2} C D_0^{-1/2}$). Seja $D = \rho D_0$, e escolha $\gamma \in [1/d_{\max}, 1/d_{\min}]$.
3. Escolha D que minimize alguma função dos autovalores de $D - C$. Por exemplo, encontre D que minimize aproximadamente o traço de $D - C$, ou o maior autovalor de $D - C$, sujeito a $D - C$ sendo semidefinida positiva. Escolha então $\gamma \in [1/d_{\max}, 1/d_{\min}]$.

A vantagem da segunda estratégia em relação a primeira é que o mesmo trabalho é requerido em ambas (o cálculo do maior autovalor de uma matriz), mas no caso em que C é diagonal e $M = \emptyset$, usando a segunda estratégia, a solução de NLP é também solução do problema original discreto. Os cálculos de D pela terceira estratégia são mais complexos. No entanto, como os problemas envolvidos em tais cálculos são exemplos de problemas de *programação semidefinida*, eles podem ser eficientemente resolvidos com o uso dos recentes avanços na teoria de pontos interiores.

3.3.5 Uma Relaxação Baseada num Problema Complementar à NLP

Limites baseados em relaxações não lineares também podem ser obtidos a partir de um “problema complementar” se observarmos que para qualquer S ,

$$ldet C[S, S] = ldet C + ldet C^{-1}[\bar{S}, \bar{S}], \quad (3.38)$$

onde $\bar{S} = N \setminus S$. Consequentemente, podemos considerar C^{-1} como a matriz de covariância e calcular limites para o problema no qual queremos escolher $\bar{s} = n - s$ índices. A relaxação baseada no “problema complementar” é então formulada da seguinte forma:

$$\begin{aligned} \overline{\text{NLP}} : \quad & \max \bar{f}(\bar{x}) \\ & \text{s.a. } \bar{A}\bar{x} \leq \bar{b} \\ & e^T \bar{x} = \bar{s} \\ & 0 \leq \bar{x} \leq e. \end{aligned}$$

Onde, sendo A e b a matriz e o vetor que definem as restrições de NLP, temos:

- $\bar{f}(\bar{x}) := ldet \bar{M}(\bar{x})$, onde $\bar{M}(\bar{x}) := \bar{X}^{p/2}(C^{-1} - D)\bar{X}^{p/2} + D\bar{x}$
- $\bar{A} = -A$
- $\bar{b} = -Ae + b$

Podemos facilmente observar que se \bar{x} é solução viável de $\overline{\text{NLP}}$, então $e - \bar{x}$ é solução viável de NLP.

Logo, baseados em (3.38) e na observação anterior, concluímos que se \bar{x}^* é uma solução ótima de $\overline{\text{NLP}}$ então $\bar{f}(\bar{x}^*) + ldet C$ é um limite superior para CMESP e $x := e - \bar{x}^*$ é uma solução relaxada de CMESP.

É importante notar que podemos também trabalhar com o problema complementar escalonado em analogia ao que propomos na Seção 3.3.2. Neste caso, definimos:

$$\begin{aligned} \bar{f}_\gamma(\bar{x}) &= ldet \bar{M}_\gamma(\bar{x}) - (n - s) \ln(\gamma), \\ \text{onde } \bar{M}_\gamma(\bar{x}) &= \gamma \bar{X}^{p/2}(C^{-1} - D)\bar{X}^{p/2} + (\gamma D)\bar{x}. \end{aligned}$$

Sendo assim, $\bar{f}(\bar{x}) = \bar{f}_\gamma(\bar{x})$ para todo \bar{x} que é viável à $\overline{\text{NLP}}$, tal que $\bar{x}_j \in \{0, 1\}$, $j \in N$. Podemos portanto, utilizar $\bar{f}_\gamma(\bar{x})$ ao invés de $\bar{f}(\bar{x})$ na relaxação não linear.

Os limites obtidos a partir do problema original e do complementar podem diferir bastante um do outro. Por este motivo, para garantir a obtenção do melhor limite, é aconselhável calcular ambos e então utilizar o melhor dos dois no algoritmo de “*branch-and-bound*”.

3.3.6 Solução Dual para o Problema NLP e Critérios para fixar Variáveis em zero ou um na Solução de CMESP

Consideramos inicialmente o problema de programação não linear

$$\begin{aligned} NLP : \quad \max \quad & f(x) \\ & Ex = h \\ & Ax \leq b \\ & 0 \leq x \leq e, \end{aligned}$$

onde $f(\cdot)$ é côncava, E é uma matriz $k \times n$, e A é uma matriz $m \times n$. Supomos que NLP tem um ponto interior relativo, isto é, existe x tal que $Ex = h$, $Ax < b$, $0 < x < e$. As condições de otimalidade de Karush-Kuhn-Tucker para a solução ótima de NLP são necessárias e suficientes, e podem ser escritas na seguinte forma:

$$u - \bar{u} - A^T \lambda + E^T \nu = -\nabla f(x)^T, \quad (3.39)$$

$$Ex = h, \quad Ax \leq b, \quad 0 \leq x \leq e, \quad (3.40)$$

$$u \geq 0, \quad \bar{u} \geq 0, \quad \lambda \geq 0 \quad (3.41)$$

$$x^T u = 0, \quad \bar{x}^T \bar{u} = 0, \quad \lambda^T (b - Ax) = 0, \quad (3.42)$$

onde $\bar{x} = e - x$. De resultados conhecidos com o uso do dual, se $(x, u, \bar{u}, \lambda, \nu)$ satisfaz (3.39)-(3.41), então

$$f(x) \leq v_{NLP} \leq f(x) + x^T u + \bar{x}^T \bar{u} + \lambda^T (b - Ax). \quad (3.43)$$

Para $\mu > 0$ considere a função barreira logarítmica, definida no interior da região viável de NLP,

$$F(x, \mu) := -f(x) - \mu \sum_{j=1}^n \ln(x_j) - \mu \sum_{j=1}^n \ln(1 - x_j) - \mu \sum_{i=1}^m \ln(b_i - a_i^T x),$$

onde a_i^T é a i -ésima linha de A . Logo, se $x(\mu)$ minimiza $F(x, \mu)$ em $\{x \mid Ex = h, 0 < x < e, Ax < b\}$, existe consequentemente $\nu(\mu) \in R^k$ tal que

$$-\nabla f(x(\mu))^T - \mu x(\mu)^{-1} + \mu \bar{x}(\mu)^{-1} + \mu A^T (b - Ax(\mu))^{-1} - E^T \nu(\mu) = 0, \quad (3.44)$$

onde para qualquer $x > 0$, x^{-1} representa o vetor cujas componentes são as recíprocas das componentes de x . Definindo $u(\mu) := \mu x(\mu)^{-1}$, $\bar{u}(\mu) := \mu \bar{x}(\mu)^{-1}$, e $\lambda(\mu) := \mu (b - Ax(\mu))^{-1}$, concluímos a partir de (3.44) que $(x(\mu), u(\mu), \bar{u}(\mu), \lambda(\mu), \nu(\mu))$ satisfaz a (3.39)-(3.41), portanto (3.43) implica em

$$f(x(\mu)) \leq v_{NLP} \leq f(x(\mu)) + (2n + m)\mu. \quad (3.45)$$

A partir de (3.45), o mínimo exato de $F(\cdot, \mu)$, para um dado μ , provê um limite superior \bar{v} tal que $\bar{v} - v_{NLP}$ é arbitrariamente pequeno. Na prática, entretanto, o mínimo exato $x(\mu)$ não pode ser obtido, portanto precisamos considerar uma metodologia para obter um limite superior a partir de um mínimo aproximado de $F(\cdot, \mu)$. A abordagem utilizada leva em consideração a estrutura particular de NLP para simplificar a construção de uma solução dual. Para um dado $\mu > 0$, e x tal que $Ex = h$, $Ax < b$, e $0 < x < e$, considere o problema

$$\begin{aligned} \text{minimizar } & \|Xu - \mu e\|^2 + \|\bar{X}\bar{u} - \mu e\|^2 \\ & u - \bar{u} + E^T \nu = -\nabla f(x)^T + \mu A^T(b - Ax)^{-1}. \end{aligned} \quad (3.46)$$

As condições de otimalidade de Karush-Kuhn-Tucker para (3.46) podem ser escritas por:

$$\begin{aligned} X^2 u - \mu x - y &= 0, \\ \bar{X}^2 \bar{u} - \mu \bar{x} + y &= 0, \\ u - \bar{u} + E^T \nu &= -\nabla f(x)^T + \mu A^T(b - Ax)^{-1}, \\ Ey &= 0. \end{aligned} \quad (3.47)$$

Utilizando os dois primeiros conjuntos de equações em (3.47) para eliminar u e \bar{u} , chegamos ao seguinte sistema reduzido:

$$(X^{-2} + \bar{X}^{-2})y + E^T \nu = \nabla_x F(x, \mu)^T \quad (3.48)$$

$$Ey = 0. \quad (3.49)$$

Definindo $T := \text{diag}(t)$, onde $t_i := x_i \bar{x}_i / \sqrt{x_i^2 + \bar{x}_i^2}$, (3.48) implica em $T^{-1}y$ ser a projeção ortogonal de $T \nabla_x F(x, \mu)^T$ no espaço nulo de ET . Dado este y , temos $E^T \nu = \nabla_x F(x, \mu)^T - T^{-2}y$, e $u \geq 0$ e $\bar{u} \geq 0$ que satisfazem a terceira equação de (3.47) são dados por

$$u := [-\nabla f(x)^T + \mu A^T(b - Ax)^{-1} - E^T \nu]^+ = [\mu x^{-1} - \mu \bar{x}^{-1} + T^{-2}y]^+, \quad (3.50)$$

$$\bar{u} := [-\nabla f(x)^T + \mu A^T(b - Ax)^{-1} - E^T \nu]^- = [\mu x^{-1} - \mu \bar{x}^{-1} + T^{-2}y]^-. \quad (3.51)$$

Usando $\lambda := \mu(b - Ax)^{-1}$, concluímos que $(x, u, \bar{u}, \lambda, \nu)$ satisfaz a (3.39)-(3.41), e portanto, a partir de (3.43),

$$v_{NLP} \leq f(x) + u^T x + \bar{u}^T \bar{x} + m\mu. \quad (3.52)$$

Além de gerar o limite superior (3.52), as soluções duais u e \bar{u} podem ser úteis para fixar variáveis nos valores zero ou um em CMESP, o problema original discreto no qual as restrições $0 \leq x \leq e$ de NLP são substituídas por $x_i \in \{0, 1\}$ para cada i . Em particular, assumimos que v é o valor da função objetivo para alguma solução viável inteira de NLP. Seja então x uma solução aproximada de $F(\cdot, \mu)$, e seja $(u, \bar{u}, \lambda, \nu)$ gerado como acima colocado. Seja x^* qualquer solução ótima de CMESP. Usando a concavidade de $f(\cdot)$ e (3.39), temos então

$$\begin{aligned} v \leq f(x^*) &\leq f(x) + \nabla f(x)^T(x^* - x) \\ &= f(x) + (x^* - x)^T(\bar{u} - u + A^T \lambda) \\ &\leq f(x) + (x^* - x)^T(\bar{u} - u) + \lambda^T(b - Ax), \end{aligned} \quad (3.53)$$

onde a igualdade usa o fato de que $E(x^* - x) = 0$, e a última inequação usa $Ax^* \leq b$. Defina $I_0 := \{i \mid u_i > 0\}$, $I_1 := \{i \mid \bar{u}_i > 0\}$, e observe que por (3.51), $I_0 \cap I_1 = \emptyset$. Usando (3.53), para cada i podemos escrever

$$\begin{aligned} (x_i - x_i^*)(\bar{u}_i - u_i) &\leq f(x) + m\mu - v + \sum_{j \neq i} (x_j^* - x_j)(\bar{u}_j - u_j) \\ &\leq f(x) + m\mu - v + \sum_{j \in I_0, j \neq i} x_j u_j + \sum_{j \in I_1, j \neq i} \bar{x}_j \bar{u}_j. \end{aligned} \quad (3.54)$$

Seja $\bar{v} := f(x) + x^T u + \bar{x}^T \bar{u} + m\mu$ o limite para v_{NLP} , como em (3.52). Para $i \in I_0$, (3.54) implica em

$$x_i^* \leq x_i + \frac{f(x) + m\mu - v + (u^T x + \bar{u}^T \bar{x} - u_i x_i)}{u_i} = \frac{\bar{v} - v}{u_i}, \quad (3.55)$$

enquanto para $i \in I_1$ obtemos

$$x_i^* \geq x_i - \frac{f(x) + m\mu - v + (u^T x + \bar{u}^T \bar{x} - \bar{u}_i \bar{x}_i)}{\bar{u}_i} = 1 - \frac{\bar{v} - v}{\bar{u}_i}. \quad (3.56)$$

Devemos observar que se o valor no lado direito de (3.55) for estritamente menor que um, então (3.55) nos dá uma prova de que $x_i^* = 0$ em qualquer solução discreta ótima de CMESP. De forma equivalente, se o valor no lado direito de (3.56) for estritamente positivo, então (3.56) nos dá uma prova de que $x_i^* = 1$.

Os resultados obtidos na prática com a implementação deste procedimento que fixa variáveis, foi muito bom como será visto na Seção 5.4.

3.4 Resolvendo a Relaxação

Para resolver os problemas NLP e $\overline{\text{NLP}}$ aproximadamente, nós utilizamos o método de barreira logarítmica. Nesta seção apresentaremos o algoritmo utilizado e comentaremos sobre a vantagem de sua escolha. Den Hertog [22] analisou este algoritmo de forma unificada para três importantes classes de problemas: Programação Linear, Programação Quadrática Convexa e Programação Convexa onde a função objetivo satisfaz a condição de auto concordância introduzida por Nesterov e Nemirovsky [65].

3.4.1 O Método de Barreira Logarítmica

Consideramos a seguinte formulação para o problema de programação convexa:

$$\begin{aligned} \text{CP:} \quad &\text{maximizar} && f_0(y) \\ &\text{sujeito à} && f_i(y) \leq 0, \quad i = 1, \dots, n \\ &&& y \in R^n \end{aligned}$$

Denotamos por \mathcal{F} a região viável de CP e por \mathcal{F}^0 o interior desta região. Assumiremos as seguintes hipóteses:

1. As funções $-f_0(y)$ e $f_i(y)$, $1 \leq i \leq n$ são funções convexas com derivadas de primeira e segunda ordem contínuas em \mathcal{F}^0 .
2. \mathcal{F}^0 não é vazio.
3. \mathcal{F}^0 é limitado.

O problema dual associado à CP é modelado como:

$$\begin{aligned} \text{CD:} \quad & \text{minimizar} \quad f_0(y) - \sum_{i=1}^n x_i f_i(y) \\ & \text{sujeito à} \quad \sum_{i=1}^n x_i \nabla f_i(y) = \nabla f_0(y) \\ & \quad \quad \quad x_i \geq 0 \end{aligned}$$

Devido às hipóteses feitas, CD tem uma solução ótima e o valor da solução ótima de CD é igual ao valor da solução ótima de CP.

A função barreira logarítmica associada à CP é definida como

$$\phi_B(y, \mu) = -\frac{f_0(y)}{\mu} - \sum_{i=1}^n \ln(-f_i(y)), \quad (3.57)$$

onde $\mu > 0$ é um parâmetro de penalização.

A função barreira evita que o ponto resultante de cada iteração torne-se inviável. Desta forma, o método da função de barreira logarítmica é denominado um *Método de Pontos Interiores*.

Para descrever o método de barreira logarítmica necessitamos conhecer as derivadas de primeira e segunda ordem de $\phi_B(y, \mu)$, as quais denotaremos por $g_B(y, \mu)$ e $H_B(y, \mu)$ respectivamente. Logo,

$$g_B(y, \mu) = \nabla \phi_B(y, \mu) = -\frac{\nabla f_0(y)}{\mu} + \sum_{i=1}^n \frac{\nabla f_i(y)}{-f_i(y)} \quad (3.58)$$

$$H_B(y, \mu) = \nabla^2 \phi_B(y, \mu) = -\frac{\nabla^2 f_0(y)}{\mu} + \sum_{i=1}^n \left[\frac{\nabla^2 f_i(y)}{-f_i(y)} + \frac{\nabla f_i(y) \nabla f_i(y)^T}{f_i(y)^2} \right] \quad (3.59)$$

Escreveremos g_b e H_B ao invés de $g_B(y, \mu)$ e $H_B(y, \mu)$ sempre que a clareza do texto não fique comprometida. É fácil verificar que $H_B(y, \mu)$ é semidefinida positiva, mas não podemos garantir que em geral $H_B(y, \mu)$ é definida positiva. No entanto pode-se provar (ver [22]) que para uma vasta classe de problemas (programação linear, programação quadrática e programação convexa que satisfaça à condição de auto-concordância introduzida por Nesterov e

Nemirovsky [65]) $H_B(y, \mu)$ é definida positiva. Mais tarde veremos que nos casos específicos dos problemas NLP e $\overline{\text{NLP}}$, $H_B(y, \mu)$ é também definida positiva.

Utilizaremos a medida $\|\cdot\|_H$ para medir distância entre dois pontos e definiremos esta medida como

$$\|z\|_H := \sqrt{z^T H_B z}$$

Como H_B é definida positiva, $\|\cdot\|_H$ define uma norma, que no caso, depende do ponto y , já que $H_B = H_B(y, \mu)$.

Como $\phi_B(\cdot, \mu)$ é estritamente convexa no seu domínio limitado \mathcal{F}^0 e assume valores infinitos na fronteira de \mathcal{F} , ela assume um valor mínimo no seu domínio em um único ponto (para um valor fixo de μ), o qual denotaremos por $y(\mu)$. Este ponto é denominado ponto μ -central. As condições necessárias e suficientes de Karush-Kuhn-Tucker para $y(\mu)$ são:

$$\begin{aligned} f_i(y) &\geq 0, & 1 \leq i \leq n \\ \sum_{i=1}^n x_i \nabla f_i(y) &= \nabla f_0(y), & x \geq 0 \\ -f_i(y)x_i &= \mu, & 1 \leq i \leq n \end{aligned}$$

3.4.2 A Trajetória Central

A trajetória central primal(dual) é definida pelo conjunto de pontos $y(\mu)$ ($x(\mu, y(\mu))$), onde μ varia de ∞ a 0. É importante notar que não somente $y(\mu)$ é primal viável como também ($x(\mu, y(\mu))$) é dual viável. Além disso, o gap primal dual satisfaz a:

$$f_0(y(\mu)) - \sum_{i=1}^n x_i(\mu) f_i(y(\mu)) - f_0(y(\mu)) = n\mu \quad (3.60)$$

Fiacco e McCormick [32] provaram que $x(\mu)$ e $y(\mu)$ são continuamente diferenciáveis. É sabido também que $y(\mu)$ e $x(\mu)$ convergem para a solução ótima de CP e CD, se $\mu \rightarrow 0$, o que significa que a trajetória central termina numa solução ótima do problema. Observamos ainda que se a região viável é limitada, a trajetória central começa num ponto único onde $\sum_{i=1}^n \ln(-f_i(y))$ é maximizada, uma vez que:

$$\lim_{\mu \rightarrow \infty} \phi_B(y, \mu) = - \sum_{i=1}^n \ln(-f_i(y)).$$

Este ponto é denominado centro analítico da região viável.

A trajetória central foi estudada por Fiacco e McCormick [32] e McLinden [59] para o caso geral convexo, e por Megiddo [60], Bayer e Lagarias [12] e Sonnevend [88] para o caso linear.

A função de barreira logarítmica foi introduzida por Frisch [36, 37] e posteriormente estudada por Fiacco e McCormick [32]. Neste método o problema original restrito CP é substituído por uma seqüência de problemas de minimização irrestrita, nos quais minimizamos

$\phi_B(y, \mu)$ sucessivamente para uma seqüência de valores positivos e decrescentes de μ . De uma forma geral, pode-se dizer que este método segue aproximadamente a trajetória central até o ponto ótimo.

Os principais pontos a serem determinados no método são:

- O método utilizado para minimizar aproximadamente $\phi_B(y, \mu)$;
- O critério de parada para a minimização aproximada;
- Como atualizar o parâmetro de penalização μ .

Fiacco e McCormick [32] propuseram um algoritmo onde buscas lineares são realizadas na direção de Newton, a cada iteração μ é reduzido por uma constante menor do que um e $\phi_B(y, \mu)$ é minimizada de forma exata. Esta técnica de minimização irrestrita seqüencial (*Sequential Unconstrained Minimization Technique* - SUMT) foi implementada nos códigos SUMT-3 e SUMT-4, descritos por Mylander et al. [62]. Anstreicher [8], baseado na análise contida em [9], provou que este algoritmo resolve problemas de programação linear e quadrática em $O(\sqrt{n}L \ln L)$ iterações, com devida inicialização e escolha de parâmetros.

O método de barreira logarítmica para programação convexa que utilizamos para minimizar aproximadamente NLP, foi analisado por Den Hertog et al. em [23].

3.4.3 O Algoritmo

A seguir apresentaremos o algoritmo para encontrar uma solução tal que diferença entre o valor da função objetivo neste ponto e o valor ótimo do problema é no máximo igual a ϵ .

No algoritmo apresentado, é realizada uma busca linear na direção de Newton definida como $p = -H_B^{-1}g_B$. O critério de parada da minimização aproximada de $\phi_B(y, \mu)$ é $\|p\|_{H_B} \leq \tau < 1$. A medida de proximidade é portanto, a norma Hessiana do passo de Newton. Observamos neste ponto que $\|p\|_{H_B} = 0$ se e somente se $y = y(\mu)$.

Algoritmo de Barreira Logarítmica

Dados:

ϵ é o parâmetro de precisão;

τ é o parâmetro de proximidade;

θ é o parâmetro de redução;

μ_0 é o valor inicial do parâmetro de penalização;

y^0 é um dado ponto interior viável tal que $\|p(y^0, \mu_0)\|_{H_B(y^0, \mu_0)} \leq \tau$;

Início

$y := y_0$;

$\mu := \mu_0$;

Enquanto $\mu > \frac{\epsilon}{4n}$ **faça**

Início (passos externos)

$\mu := (1 - \theta)\mu;$

Enquanto $\|p\|_{H_B} \geq \tau$ **faça**

Início (passos internos)

$\tilde{\alpha} := \arg \min_{\alpha > 0} \{\phi_B(y + \alpha p, \mu) : y + \alpha p \in \mathcal{F}^0\}$

$y := y + \tilde{\alpha}p$

Fim (passos internos)

Fim (passos externos)

Fim

É importante notar que a busca linear nos passos internos não precisa ser exata, nem mesmo na análise da complexidade (ver [22]).

O algoritmo acima é denominado:

- algoritmo de passos longos, se θ é uma constante ($0 < \theta < 1$), independente de n e de ϵ ;
- algoritmo de passos médios, se $\theta = \frac{\nu}{\sqrt{n}}$, onde $\nu > 0$ é uma constante arbitrária, possivelmente grande e independente de n e de ϵ .
- algoritmo de passos curtos, se $\theta = \frac{\nu}{\sqrt{\mu}}$ e ν é tão pequeno (por exemplo $\frac{1}{9}$), que depois de uma redução de μ , apenas um único passo de Newton é suficiente para alcançar a vizinhança do novo ponto μ -central.

Apesar da análise teórica nos mostrar melhores resultados para os algoritmos de passos curtos e médios (ver [22]), na prática os algoritmos de passos longos são os mais eficientes.

3.4.4 Aplicação do Algoritmo de Barreira Logarítmica ao Problema NLP

Para que o algoritmo de barreira logarítmica seja aplicado à relaxação NLP, é necessário que as hipóteses 1-3 descritas na Seção 3.4.1 sejam satisfeitas. Assim sendo, assumiremos sem perda de generalidade, que NLP tem uma solução interior, ou seja, existe x tal que $e^T x = s$, $0 < x < e$, $Ax < b$. (Se tal solução não existe então algumas das restrições de NLP são ativas para todas as soluções viáveis. Neste caso, tais restrições são identificadas através da resolução de uma “Fase I” de programação linear. A seguir são transformadas de inequações em equações em NLP).

Para a constante $\mu > 0$, definimos então a função barreira logarítmica

$$F(x, \mu) := -f(x) - \mu \sum_{j=1}^n \log(x_j) - \mu \sum_{j=1}^n \log(1 - x_j) - \mu \sum_{i=1}^m \log(b_i - a_i^T x), \quad (3.61)$$

onde a_i^T é a i -ésima linha de A . Para μ fixo, nós então minimizamos aproximadamente $F(\cdot, \mu)$ em $\{x \mid e^T x = s\}$ usando passos de Newton, e então reduzimos μ e repetimos o processo. Este método de resolução nos é bastante conveniente por uma série de razões. Primeiramente, é sabido que se $x^*(\mu)$ minimiza $F(\cdot, \mu)$ em $\{x \mid e^T x = s\}$, então

$$f(x^*(\mu)) \leq v_{NLP} \leq f(x^*(\mu)) + \epsilon, \quad (3.62)$$

onde v_{NLP} denota o valor da solução de NLP, e $\epsilon = (2n + m)\mu$. Obviamente a solução exata $x^*(\mu)$ não pode ser obtida. No entanto, para qualquer $\epsilon > (2n + m)\mu$, uma minimização suficientemente precisa de $F(\cdot, \mu)$ produz uma “solução dual” que provê um limite superior como em (3.62). Este limite é essencial a nossa aplicação, uma vez que usamos NLP como uma relaxação de CMESP, e precisamos portanto de um limite *superior* para a solução ótima de NLP. Em particular, estabelecemos a precisão desejada ϵ , decrescemos μ até que $\mu \leq \epsilon/(4n + 2m)$, e então continuamos minimizando $F(\cdot, \mu)$ para este valor final de μ , até que a solução dual que faça valer (3.62) seja gerada. Aproximações muito precisas da solução de NLP *não* são necessária em nossa aplicação. Por exemplo, utilizando-se $\epsilon = .001$, a imprecisão na solução de NLP será menor do que a diferença entre v_{NLP} e o valor ótimo da função objetivo do problema original CMESP. Utilizando ϵ relativamente grande, economizamos substancialmente o tempo necessário à obtenção do limite superior a partir de NLP. Uma outra vantagem deste método é que quando um limite inferior para CMESP é conhecido, a solução dual que gera o limite superior em NLP pode ser utilizada para *fixar* variáveis em NLP em 0 ou 1. No contexto do *branch-and-bound*, onde NLP é finalmente aplicado, esta é uma característica importante, uma vez que a relaxação que está perto atender as condições para podar a árvore de enumeração de subproblemas, pode ser capaz de fixar muitas variáveis, e fixando-as, melhorará os limites de subproblemas seguintes. Uma outra observação interessante é que sempre que $f(x)$ exceder um limite inferior para CMESP, podemos interromper os cálculos em NLP, uma vez que a poda da árvore neste nó será impossível.

Além de resolver aproximadamente NLP, precisamos também considerar o problema de encontrar D quando a terceiro método sugerido na Seção 3.3.4 é empregado. Por exemplo, podemos considerar o problema de obter $D \succeq C$ que minimiza aproximadamente o traço de $D - C$. Neste caso, também utilizamos um algoritmo de trajetória central de passos longos, o qual está detalhadamente descrito e analisado nos capítulos 6 e 7.

Capítulo 4

Obtenção de Limites Inferiores para o Problema da Amostra de Máxima Entropia

4.1 Introdução

No capítulo anterior comentamos sobre a necessidade do conhecimento de um bom limite superior para o *Problema da Amostra de Máxima Entropia*, para a obtenção de sua solução ótima através da aplicação do algoritmo de *branch-and-bound*.

Na verdade, além do conhecimento de um bom limite superior é necessária também a obtenção de uma solução inicial viável capaz de fornecer um bom limite inferior para o problema.

O número de subproblemas podados da árvore enumerativa do algoritmo de *branch-and-bound* será evidentemente proporcional à qualidade do nosso limite inferior. Sendo assim, faz-se importante a elaboração de boas heurísticas para a obtenção de uma solução viável para o problema.

Neste capítulo, apresentaremos as heurísticas utilizadas por Ko, Lee e Queyranne em [48] para o problema irrestrito e em seguida veremos qual o procedimento utilizado para a obtenção de uma boa solução inicial no caso restrito.

4.2 O Problema Irrestrito

O método mais natural de se obter uma solução inicial viável para o problema da amostra de máxima entropia irrestrito é através de uma heurística baseada ao mesmo tempo em permutação e no procedimento “guloso”. A seguir descreveremos o método apresentado e

utilizado por Ko, Lee e Queyranne em [48].

Dados N , o conjunto total de índices; s , o tamanho do subconjunto de N que pretendemos escolher e F , o conjunto de índices fixos na solução, iniciamos a heurística com o conjunto $S = F$ ($|S| < s$). Em seguida substituímos repetidamente S por $S \cup i$ e o conjunto de índices elegíveis E por $E \setminus i$, desde que $|S| < s$ e

$$\det(C[S \cup i, S \cup i]) \geq \det(C[S \cup j, S \cup j]) \quad (4.1)$$

para todo $j \in (N \setminus S)$.

Em seguida, substituímos S por $S \setminus i \cup j$ e E por $E \setminus j \cup i$, se $\det(C[S \setminus i \cup j, S \setminus i \cup j]) > \det(C[S, S])$ para algum $i \in S \setminus F$ e $j \in E$. Repetimos estas permutações até que não exista mais i ou j que satisfaça tais condições. Os cálculos a serem realizados são simplificados se observarmos que

$$\det(C[S \setminus T \cup j, S \setminus T \cup j]) = \det(C[S \setminus T, S \setminus T])(C[j, j] - C[j, S \setminus T]C[S \setminus T, S \setminus T]^{-1}C[S \setminus T, j]), \quad (4.2)$$

o qual, para cada T fixo, é um simples cálculo para todo j , que utiliza a fatorização de Cholesky de $C[S \setminus T, S \setminus T]$. Na fase inicial gulosa, tomamos $T = \emptyset$, e na fase de permutações tomamos $T = \{i\}$.

Um procedimento alternativo à inicialização gulosa considera uma abordagem gulosa dual, na qual começamos com $S = N$, e enquanto $|S| > s$, substituímos repetidamente S por $S \setminus i$, desde que $i \in S \setminus F$ satisfaça

$$\det(C[S \setminus i, S \setminus i]) \geq \det(C[S \setminus j, S \setminus j]) \quad (4.3)$$

para todo $j \in (S \setminus F)$.

4.3 O Problema Restrito

Quando restrições são impostas ao *Problema da Amostra de Máxima Entropia*, as heurísticas apresentadas na Seção anterior não mais se aplicam. Neste caso, assim como Lee (veja [53]), propomos um método para gerar uma solução viável inicial para o problema, que utiliza a solução ótima contínua da relaxação NLP.

Seja y a solução ótima da relaxação não linear NLP. Caso pelo menos uma das componentes de y seja não inteira, ou seja, diferente de zero e de um, o método propõe a resolução do seguinte problema de programação linear inteira, no qual usamos y como um vetor de coeficientes numa linearização do problema:

$$\max y^T x$$

$$\begin{aligned} \text{s.a. } Ax &\leq b \\ e^T x &= s \\ x_i &\in \{0, 1\} \quad i = 1, \dots, n \end{aligned}$$

Como veremos no próximo capítulo, esta mesma linearização pode também ser aplicada a subproblemas gerados na árvore enumerativa do *branch-and-bound* com o intuito de se melhorar o limite inferior obtido até o momento.

Capítulo 5

Implementação do Método de Resolução para o CMESP

5.1 Introdução

O método de resolução proposto para o *Problema da Amostra de Máxima Entropia*, consiste na aplicação do algoritmo de *branch-and-bound* ao problema. Como já é sabido, o sucesso na busca da solução ótima depende da obtenção de bons limites para os subproblemas gerados na árvore enumerativa do *branch-and-bound*.

Neste capítulo, comentaremos os detalhes relacionados a implementação do método. Na seção seguinte será apresentado o procedimento de criação dos subproblemas e outros aspectos relacionados ao algoritmo de *branch-and-bound*. Em seguida, demonstraremos a relação existente entre os limites gerados pela relaxação não linear para subproblemas “pais” e “filhos”. Finalmente, serão expostos resultados numéricos obtidos na resolução da relaxação seguidos dos resultados finais obtidos com a resolução do problema CMESP. Através destes resultados, comprovamos a qualidade dos limites produzidos pela relaxação proposta e mostramos que na grande maioria dos exemplos resolvidos eles são melhores que os limites baseados nos autovalores da matriz de covariância, utilizados em [48].

5.2 O Algoritmo de *Branch-and-Bound*

O algoritmo completo que resolve o problema CMESP é do tipo *branch-and-bound*. Subproblemas são descritos pelo subconjunto de elementos $F \subset N$ fixos na solução e pelo subconjunto de elementos $E \subset (N \setminus F)$ que permanecem elegíveis. Mantemos um conjunto de subproblemas ativos, para os quais a relaxação já foi resolvida. O subproblema descrito

pelos subconjuntos F e E pode ser formulado pelo seguinte modelo matemático:

$$\begin{aligned} \text{CMESP}(F, E): \quad & \max \quad \text{ldet } C[\underline{x}, \underline{x}] \\ & \text{s.t.} \quad Ax \leq b \\ & \quad \quad e^T x = s \\ & \quad \quad x_j = 1, \quad j \in F, \\ & \quad \quad x_j = 0, \quad j \in N \setminus (E \cup F), \\ & \quad \quad x_j \in \{0, 1\}, \quad j \in E, \end{aligned}$$

onde $|F| \leq s \leq |F| + |E|$.

Sendo $C_{N \setminus F}$ o complemento de Schur de $C[F, F]$ em $C[F \cup E, F \cup E]$ como definido na Seção 2.4, podemos reformular o problema $\text{CMESP}(F, E)$ como:

$$\begin{aligned} \text{CMESP}(F, E): \quad & \max \quad \text{ldet } C[F, F] + \text{ldet } C_{N \setminus F}[\underline{x}, \underline{x}] \\ & \text{s.t.} \quad \sum_{j \in E} a_{ij} x_j \leq b_i - \sum_{j \in F} a_{ij} \quad i \in M \\ & \quad \quad \sum_{j \in E} x_j = s - |F| \\ & \quad \quad x_j \in \{0, 1\}, \quad j \in E, \end{aligned}$$

Como este subproblema tem a mesma forma que o problema original CMESP , podemos aplicar a mesma metodologia descrita no capítulo anterior para a obtenção de limites superiores para o mesmo.

Inicialmente o único subproblema ativo é $\text{CMESP}(\emptyset, N)$. A cada passo do algoritmo então, um subproblema é removido do conjunto ativo, e seu limite é comparado ao limite inferior válido no momento, para uma possível poda. Caso não seja possível podar, escolhemos um índice elegível j e criamos dois subproblemas : um no qual j é fixo na solução e um outro no qual j é fixo fora da solução. Para cada um destes dois subproblemas, o seguinte procedimento é aplicado:

1. Se existir apenas uma única solução 0/1 para a restrição de cardinalidade $e^T x = s$, checamos a viabilidade da solução. Se a solução for inviável, podamos a árvore por inviabilidade e descartamos o subproblema. Se a solução for viável, checamos o valor da função objetivo na solução, se possível atualizamos o limite inferior, e descartamos o subproblema.
2. Caso contrário, resolvemos um problema de programação linear para determinar uma solução interior à região viável da relaxação contínua. Identificamos assim, as restrições que só podem ser satisfeitas como equações, ou seja, as restrições ativas para qualquer solução viável.
3. Se a dimensão do interior relativo é -1 , nós podamos a árvore por inviabilidade, descartando o subproblema.
4. Se a dimensão do interior relativo é 0 , observamos se a solução é inteira. Caso ela seja inteira, checamos o valor da função objetivo para possível atualização do limite

inferior. Caso contrário, podamos por inviabilidade. Em ambos os casos, descartamos o subproblema.

5. Se a dimensão do interior relativo é positiva, determinamos uma base para o espaço nulo da matriz de restrições ativas, e usamos no cálculo de um limite superior e da solução contínua a ele associada. Em seguida podamos a árvore por comparação entre limites superior e inferior sempre que possível, e neste caso, descartamos o subproblema.
6. Se o subproblema não foi descartado pelos limites, aplicamos o procedimento descrito na Seção 3.3.6 para possivelmente fixar variáveis nos valores zero ou um.
7. Caso pelo menos uma variável tenha sido fixada, reformulamos o subproblema. Calculamos o limite superior para o subproblema reformulado e a solução contínua a ele associada. Neste ponto podamos a árvore por comparação entre limites superior e inferior e descartamos o subproblema, caso seja possível. Caso contrário, voltamos ao passo anterior.
8. Após fixar todas as possíveis variáveis em zero ou um, se o subproblema não foi descartado pelos limites e pelo menos uma componente da solução contínua a ele associada não pertence ao conjunto $\{0, 1\}$, usamos a solução contínua como um vetor de coeficientes na seguinte linearização do problema:

$$\begin{aligned}
 \max \quad & y^T x \\
 \text{s.a.} \quad & Ax \leq b \\
 & e^T x = s \\
 & 0 \leq x_i \leq 1 \quad i = 1, \dots, n
 \end{aligned}$$

Resolvemos a linearização por programação linear ou por programação linear inteira. No segundo caso, a última expressão é substituída por $x_i \in \{0, 1\}, i = 1, \dots, n$. Se encontrarmos uma solução inteira para a linearização, checamos o valor da função objetivo de CMESP nesta solução e atualizamos o limite inferior se possível. Na prática utilizamos programação linear inteira na linearização do primeiro subproblema, para assegurar a obtenção de um limite inferior razoável. Para os demais subproblemas, usamos programação linear.

O algoritmo de *branch-and-bound* utilizado para resolver CMESP é representado na Figura 1.

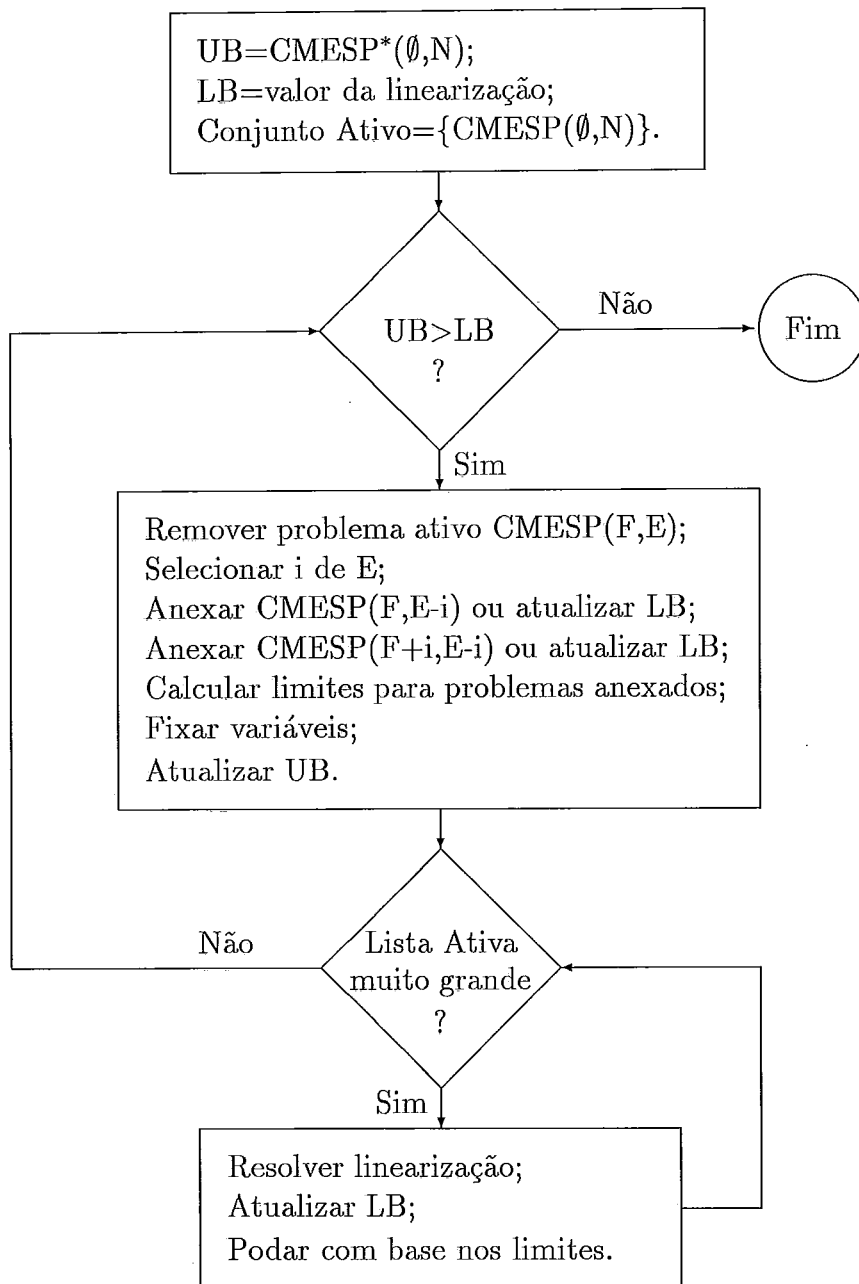


Fig 5.1: Diagrama do Algoritmo de *Branch-and-Bound*

5.3 Relação entre os Limites Obtidos pela Relaxação Não Linear para os Problemas Pais e Filhos na Árvore Enumerativa do *Branch-and-Bound*

No algoritmo de *branch-and-bound*, a divisão de um problema em subproblemas é realizada ao fixarmos índices dentro e fora de S .

Veremos nesta seção que o limite superior fornecido pela relaxação não linear NLP para um problema “pai” na árvore enumerativa é maior ou igual ao limite fornecido para o mais restrito problema “filho”.

Este resultado será verificado para algumas escolhas de D e de γ , não somente quando o limite é gerado pelo problema original NLP, como também quando ele é gerado pelo problema complementar definido na Seção 3.3.5.

Neste capítulo \bar{x}^* e \bar{x} representam soluções ótimas para os problemas “pai” e “filho” respectivamente.

5.3.1 A Opção “Identidade”

Consideraremos inicialmente a primeira escolha de D e γ sugerida na Seção 3.3.4, na qual $\gamma D = I$. Neste caso teremos ainda a opção de calcular os limites baseados no problema original ou no complementar.

Limites Baseados no Problema Original

Fixando o índice i fora de S

Consideraremos o caso em que o problema filho é gerado ao fixarmos o índice i fora de S no problema pai.

Neste caso, as funções objetivos dos problemas pai e filho na árvore enumerativa são escritas respectivamente como:

$$\bar{f}_{\gamma_1}(\bar{x}) = \text{ldet}(\bar{X}^{e/2}(\gamma_1 \bar{C} - I)\bar{X}^{e/2} + I) - s \ln(\gamma_1) \quad (5.1)$$

e

$$f_{\gamma_2}(x) = \text{ldet}(X^{e/2}(\gamma_2 C - I)X^{e/2} + I) - s \ln(\gamma_2) \quad (5.2)$$

onde:

- C é obtida de \bar{C} removendo-se suas i -ésimas linha e coluna.
- $\gamma_1 = 1/\rho(\bar{C})$ e $\gamma_2 = 1/\rho(C)$

Teorema 5.3.1 *Sejam $\bar{f}_{\gamma_1}(\cdot)$ e $f_{\gamma_2}(\cdot)$ definidas por (5.1) e (5.2). Então*

$$\bar{f}_{\gamma_1}(\bar{x}^*) \geq f_{\gamma_2}(x^*)$$

Demonstração: Seja

$$\hat{x}_j = \begin{cases} x_j^*, & \text{if } j < i \\ 0, & \text{if } j = i \\ x_{j-1}^*, & \text{if } j > i \end{cases}$$

Como \hat{x} é uma solução viável para o problema pai, é suficiente mostrar que $\bar{f}_{\gamma_1}(\hat{x}) \geq f_{\gamma_2}(x^*)$. Temos:

$$\begin{aligned} \bar{f}_{\gamma_1}(\hat{x}) &= \text{ldet}(\hat{X}^{e/2}(\gamma_1 \bar{C} - I)\hat{X}^{e/2} + I) - s \ln(\gamma_1) \\ f_{\gamma_2}(x^*) &= \text{ldet}(X^{*e/2}(\gamma_2 C - I)X^{*e/2} + I) - s \ln(\gamma_2) \end{aligned}$$

$\hat{x}_i = 0$ significa que a linha i e a coluna i de $\gamma_1 \bar{C} - I$ são multiplicadas por zero. Sendo assim, os elementos da linha i e da coluna i de $\hat{X}^{e/2}(\gamma_1 \bar{C} - I)\hat{X}^{e/2} + I$ são nulos, com exceção dos elementos da diagonal, os quais são um. Segue-se então que:

$$\bar{f}_{\gamma_1}(\hat{x}) = \text{ldet}(X^{*e/2}(\gamma_1 C - I)X^{*e/2} + I) - s \ln(\gamma_1)$$

Como $0 < \gamma_1 \leq \gamma_2$, é suficiente mostrar que:

$$\frac{d}{d\gamma} f_{\gamma}(x) \leq 0$$

para $\gamma > 0$.

Temos:

$$\frac{d}{d\gamma} f_{\gamma}(x) = \frac{d}{d\gamma} \text{ldet}(X^{e/2}(\gamma C - I)X^{e/2} + I) - \frac{s}{\gamma}. \quad (5.3)$$

O restante da demonstração é análogo ao da prova do Lema 3.3.8. Consideramos o vetor $w > 0$, e definimos

$$\begin{aligned} M_w(x) &:= X^{e/2}(W^{e/2} C W^{e/2} - I)X^{e/2} + I \\ &= W^{e/2}(X^{e/2}(C - W^{-e})X^{e/2} + W^{-e})W^{e/2}. \end{aligned}$$

Então

$$\text{ldet } M_w(x) = \sum_{i=1}^n \ln(w_i) + \text{ldet}(X^{e/2}(C - W^{-e})X^{e/2} + W^{-e})$$

Usando a regra da cadeia e (3.3), obtemos

$$\begin{aligned} \frac{\partial}{\partial w_i} \text{ldet } M_w(x) &= \frac{1}{w_i} + (X^{e/2}(C - W^{-e})X^{e/2} + W^{-e})_{ii}^{-1} w_i^{-2} (x_i - 1) \\ &= \frac{1}{w_i} (1 - (1 - x_i) M_w^{-1}(x)_{ii}). \end{aligned}$$

No entanto,

$$M_w(x) \preceq I \Rightarrow M_w^{-1} \succeq I \Rightarrow \text{Diag}(M_w^{-1}(x)) = I \circ M_w^{-1}(x) \succeq I \circ I = I \Rightarrow M_w^{-1}(x)_{ii} \geq 1.$$

Logo,

$$\frac{\partial}{\partial w_i} \text{ldet } M_w(x) \leq \frac{1}{w_i}(1 - (1 - x_i)) = \frac{x_i}{w_i}$$

Finalmente,

$$\frac{d}{d\gamma} \text{ldet } f_\gamma(x) = \sum_{i=1}^n \frac{\partial}{\partial w_i} \text{ldet } M_w(x)|_{w=\gamma e} - \frac{s}{\gamma} \leq \sum_{i=1}^n \frac{x_i}{\gamma} - \frac{s}{\gamma} = 0$$

Onde usamos o fato de que $e^T x = s$. ■

Fixando o índice i em S

Consideraremos agora o caso em que o problema filho é gerado ao fixarmos o índice i em S no problema pai.

Neste caso as funções objetivos para os problemas pai e filho são dadas respectivamente pelas seguintes funções:

$$\bar{f}_{\gamma_1}(\bar{x}) = \text{ldet}(\bar{X}^{e/2}(\gamma_1 \bar{C} - I)\bar{X}^{e/2} + I) - s \ln(\gamma_1) \quad (5.4)$$

e

$$f_{\gamma_2}(x) = \ln(\bar{C}_{ii}) + \text{ldet}(X^{e/2}(\gamma_2 C - I)X^{e/2} + I) - (s - 1) \ln(\gamma_2) \quad (5.5)$$

onde:

- C é o complemento de Schur de \bar{C}_{ii} em \bar{C} , ou seja:

$$C = \bar{C}_{TT} - \frac{\bar{C}_{Ti}\bar{C}_{iT}}{\bar{C}_{ii}}$$

onde $T = \{1, \dots, i - 1\} \cup \{i + 1, \dots, n\}$

- $\gamma_1 = 1/\rho(\bar{C})$ e $\gamma_2 = 1/\rho(C)$

Teorema 5.3.2 *Sejam $\bar{f}_{\gamma_1}(\cdot)$ e $f_{\gamma_2}(\cdot)$ definidos por (5.4) e (5.5). Então,*

$$\bar{f}_{\gamma_1}(\bar{x}^*) \geq f_{\gamma_2}(x^*)$$

Demonstração: Seja:

$$\hat{x}_j = \begin{cases} x_j^*, & \text{if } j < i \\ 1, & \text{if } j = i \\ x_{j-1}^*, & \text{if } j > i \end{cases}$$

Como \hat{x} é uma solução viável para o problema pai, é suficiente mostrar que $\bar{f}_{\gamma_1}(\hat{x}) \geq f_{\gamma_2}(x^*)$.

Seja:

$$M_{\gamma_1}(\hat{x}) = \hat{X}^{e/2}(\gamma_1 \bar{C} - I)\hat{X}^{e/2} + I$$

Logo:

$$\bar{f}_{\gamma_1}(\hat{x}) = \text{ldet}(M_{\gamma_1}(\hat{x})) - s \ln(\gamma_1) \quad (5.6)$$

Como $\hat{x}_i = 1$, $[M_{\gamma_1}(\hat{x})]_{ii} = \gamma_1 \bar{C}_{ii}$ e, conseqüentemente,

$$\text{ldet}(M_{\gamma_1}(\hat{x})) = \ln(\gamma_1 \bar{C}_{ii}) + \text{ldet}(B)$$

onde:

$$B = [M_{\gamma_1}(\hat{x})]_{TT} - \frac{[M_{\gamma_1}(\hat{x})]_{Ti}[M_{\gamma_1}(\hat{x})]_{iT}}{\gamma_1 \bar{C}_{ii}}$$

Podemos facilmente verificar que:

$$B = X^{*e/2}(\gamma_1 C - I)X^{*e/2} + I$$

Portanto,

$$\begin{aligned} \bar{f}_{\gamma_1}(\hat{x}) &= \ln(\gamma_1 \bar{C}_{ii}) + \text{ldet}(X^{*e/2}(\gamma_1 C - I)X^{*e/2} + I) - s \ln(\gamma_1) \\ &= \ln(\bar{C}_{ii}) + \text{ldet}(X^{*e/2}(\gamma_1 C - I)X^{*e/2} + I) - (s - 1) \ln(\gamma_1) \end{aligned}$$

Como $\frac{\bar{C}_{Ti}\bar{C}_{iT}}{\bar{C}_{ii}} \succeq 0$,

$$\bar{C}_{TT} \succeq \bar{C}_{TT} - \frac{\bar{C}_{Ti}\bar{C}_{iT}}{\bar{C}_{ii}}$$

e, sendo \bar{C}_{TT} uma submatriz principal de \bar{C} , nós temos:

$$\rho(\bar{C}) \geq \rho(\bar{C}_{TT}) \geq \rho(C)$$

Portanto,

$$\gamma_1 \leq \gamma_2$$

Neste ponto é então suficiente mostrar que:

$$\frac{d}{d\gamma} f_{\gamma}(x) \leq 0$$

para $\gamma > 0$.

O restante da demonstração é então, análoga ao do Teorema 5.3.1. ■

Limites Baseados nos Problemas Complementares

Fixando o índice i fora de S

Consideraremos o caso em que o problema filho é gerado ao fixarmos o índice i fora de S no problema pai.

As funções objetivos para os problema pai e filho são dadas respectivamente por:

$$\bar{f}_{\gamma_1}(\bar{x}) = \text{ldet}(\bar{X}^{e/2}(\gamma_1 \bar{C}^{-1} - I)\bar{X}^{e/2} + I) - s \ln(\gamma_1) \quad (5.7)$$

e

$$f_{\gamma_2}(x) = \text{ldet}(X^{e/2}(\gamma_2 C^{-1} - I)X^{e/2} + I) - s \ln(\gamma_2) \quad (5.8)$$

onde:

- $C^{-1} = [\bar{C}_{TT}]^{-1}$ e $T = \{1, \dots, i-1\} \cup \{i+1, \dots, n\}$
- $\gamma_1 = 1/\rho(\bar{C}^{-1})$ e $\gamma_2 = 1/\rho(C^{-1})$

Teorema 5.3.3 *Sejam $\bar{f}_{\gamma_1}(\cdot)$ e $f_{\gamma_2}(\cdot)$ definidos por (5.7) e (5.8). Então,*

$$\bar{f}_{\gamma_1}(\bar{x}^*) \geq f_{\gamma_2}(x^*)$$

Demonstração: Seja:

$$\hat{x}_j = \begin{cases} x_j^*, & \text{if } j < i \\ 0, & \text{if } j = i \\ x_{j-1}^*, & \text{if } j > i \end{cases}$$

Como \hat{x} é uma solução viável para o problema pai, é suficiente mostrar que $\bar{f}_{\gamma_1}(\hat{x}) \geq f_{\gamma_2}(x^*)$. Temos:

$$\begin{aligned} \bar{f}_{\gamma_1}(\hat{x}) &= \text{ldet}(\hat{X}^{e/2}(\gamma_1 \bar{C}^{-1} - I)\hat{X}^{e/2} + I) - s \ln(\gamma_1) \\ &= \text{ldet}(X^{*e/2}(\gamma_1 [\bar{C}^{-1}]_{TT} - I)X^{*e/2} + I) - s \ln(\gamma_1) \end{aligned}$$

e

$$f_{\gamma_2}(x^*) = \text{ldet}(X^{*e/2}(\gamma_2 [\bar{C}_{TT}]^{-1} - I)X^{*e/2} + I) - s \ln(\gamma_2)$$

No restante da demonstração, verificaremos as seguintes inequações:

$$\bar{f}_{\gamma_1}(\hat{x}) \geq \text{ldet}(X^{*e/2}(\gamma_1 [\bar{C}_{TT}]^{-1} - I)X^{*e/2} + I) - s \ln(\gamma_1) \geq f_{\gamma_2}(x^*) \quad (5.9)$$

Para provar a primeira inequação em (5.9) observamos que:

$$\begin{aligned} &[\bar{C}^{-1}]_{TT} \succeq [\bar{C}_{TT}]^{-1} \\ \Rightarrow X^{*e/2}(\gamma_1 [\bar{C}^{-1}]_{TT} - I)X^{*e/2} + I &\succeq X^{*e/2}(\gamma_1 [\bar{C}_{TT}]^{-1} - I)X^{*e/2} + I \\ \Rightarrow \det(X^{*e/2}(\gamma_1 [\bar{C}^{-1}]_{TT} - I)X^{*e/2} + I) &\geq \det(X^{*e/2}(\gamma_1 [\bar{C}_{TT}]^{-1} - I)X^{*e/2} + I) \end{aligned}$$

Finalmente, para provar a segunda inequação em (5.9), notamos que $\rho(\bar{C}^{-1}) \geq \rho([\bar{C}^{-1}]_{TT}) \geq \rho([\bar{C}_{TT}]^{-1})$ e consequentemente, $\gamma_1 \leq \gamma_2$. Sendo assim, é agora, suficiente mostrar que

$$\frac{d}{d\gamma} f_{\gamma}(x) \leq 0$$

para $\gamma > 0$, o qual já foi verificado na demonstração do Teorema 5.3.1. ■

5.3.2 A Opção “Diagonal”

Suponha agora que D e γ foram escolhidos pela segunda opção sugerida na Seção 3.3.4

Na prática, escolhamos $D = \text{Diag}(C)$, e em seguida, com o propósito de ter $D - C$ semidefinida positiva, multiplicamos C pela constante $\beta = 1/\rho(D^{-1/2}CD^{-1/2})$.

O fator γ é então escolhido de forma a termos $\gamma \in [1/d_{max}, 1/d_{min}]$. De fato, γ é uma função de um parâmetro $\theta \in [0, 1]$ dada por:

$$\gamma(\theta) = \theta \frac{1}{d_{max}} + (1 - \theta) \frac{1}{d_{min}} \quad (5.10)$$

Deste ponto em diante, consideraremos a escolha de $\theta = 1$ em (5.10).

Limites Baseados no Problema Original

Fixando o índice i fora de S

Consideraremos o caso em que o problema filho é gerado ao fixarmos o índice i fora de S no problema pai.

As funções objetivos para os problemas pai e filho são dadas respectivamente por:

$$\bar{f}_{\gamma_1, \beta_1}(\bar{x}) = \text{ldet}(\bar{X}^{e/2}(\beta_1 \gamma_1 \bar{C} - \gamma_1 D_1) \bar{X}^{e/2} + (\gamma_1 D_1)^{\bar{x}}) - s \ln(\gamma_1 \beta_1) \quad (5.11)$$

e

$$f_{\gamma_2, \beta_2}(x) = \text{ldet}(X^{e/2}(\beta_2 \gamma_2 C - \gamma_2 D_2) X^{e/2} + (\gamma_2 D_2)^x) - s \ln(\gamma_2 \beta_2) \quad (5.12)$$

onde:

- C é obtida de \bar{C} removendo-se suas i -ésimas linha e coluna.
- $D_1 = \text{Diag}(\bar{C})$ e $D_2 = \text{Diag}(C)$
- $\gamma_1 = 1/d_{1max}$ e $\gamma_2 = 1/d_{2max}$
- $\beta_1 = 1/\rho(D_1^{-\frac{1}{2}} \bar{C} D_1^{-\frac{1}{2}})$ e $\beta_2 = 1/\rho(D_2^{-\frac{1}{2}} C D_2^{-\frac{1}{2}})$

Teorema 5.3.4 *Sejam $\bar{f}_{\gamma_1}(\cdot)$ e $f_{\gamma_2}(\cdot)$ definidas por (5.11) e (5.12). Então,*

$$\bar{f}_{\gamma_1, \beta_1}(\bar{x}^*) \geq f_{\gamma_2, \beta_2}(x^*)$$

Demonstração: Seja:

$$\hat{x}_j = \begin{cases} x_j^*, & \text{if } j < i \\ 0, & \text{if } j = i \\ x_{j-1}^*, & \text{if } j > i \end{cases}$$

Observamos novamente que \hat{x} é uma solução viável para o problema pai. Consequentemente, é suficiente mostrar que $\bar{f}_{\gamma_1, \beta_1}(\hat{x}) \geq f_{\gamma_2, \beta_2}(x^*)$.

No entanto,

$$\begin{aligned}\bar{f}_{\gamma_1, \beta_1}(\hat{x}) &= \text{ldet}(\hat{X}^{e/2}(\beta_1 \gamma_1 \bar{C} - \gamma_1 D_1) \hat{X}^{e/2} + (\gamma_1 D_1)^{\hat{x}}) - s \ln(\gamma_1 \beta_1) \\ &= \text{ldet}(X^{*e/2}(\beta_1 \gamma_1 C - \gamma_1 D_2) X^{*e/2} + (\gamma_1 D_2)^{x^*}) - s \ln(\gamma_1 \beta_1)\end{aligned}$$

Onde a última inequação resulta do fato de que $\hat{x}_i = 0$.

Observamos agora que:

- $D_2^{-\frac{1}{2}} C D_2^{-\frac{1}{2}}$ é obtida de $D_1^{-\frac{1}{2}} \bar{C} D_1^{-\frac{1}{2}}$ ao remover-se suas i -ésimas linha e coluna. Sendo assim, $\beta_1 \leq \beta_2$.
- $d_{1max} \geq d_{2max}$ e consequentemente, $\gamma_1 \leq \gamma_2$.

Logo, como $0 < \beta_1 \leq \beta_2$ e $0 < \gamma_1 \leq \gamma_2$, é agora, suficiente mostrar que:

$$\frac{\partial}{\partial \gamma} f_{\gamma, \beta}(x) \leq 0 \quad \text{and} \quad \frac{\partial}{\partial \beta} f_{\gamma, \beta}(x) \leq 0$$

para $\gamma > 0$ e $\beta > 0$.

Onde, definindo $d := d_2$, com o intuito de simplificar a expressão, temos:

$$f_{\gamma, \beta}(x) = \text{ldet}(X^{e/2}(\beta \gamma C - \gamma D) X^{e/2} + (\gamma D)^x) - s \ln(\gamma \beta)$$

Temos ainda que:

$$\frac{\partial}{\partial \gamma} f_{\gamma, \beta}(x) = \frac{\partial}{\partial \gamma} \text{ldet}(X^{e/2}(\beta \gamma C - \gamma D) X^{e/2} + (\gamma D)^x) - \frac{s}{\gamma} \quad (5.13)$$

Logo, a demonstração de que $\frac{\partial}{\partial \gamma} f_{\gamma, \beta}(x) \leq 0$ é análoga à demonstração do Lema 3.3.8.

Neste ponto, para provar que $\frac{\partial}{\partial \beta} f_{\gamma, \beta}(x) \leq 0$, observamos que:

$$\frac{\partial}{\partial \beta} f_{\gamma, \beta}(x) = \frac{\partial}{\partial \beta} \text{ldet}(X^{e/2}(\beta \gamma C - \gamma D) X^{e/2} + (\gamma D)^x) - \frac{s}{\beta} \quad (5.14)$$

Como na demonstração do Lema 3.3.8 consideraremos um vetor $w > 0$ e definiremos

$$\begin{aligned}M_w(x) &:= \gamma X^{e/2}(W^{e/2} C W^{e/2} - D) X^{e/2} + (\gamma D)^x \\ &= W^{e/2}(\gamma X^{e/2}(C - W^{-e} D) X^{e/2} + W^{-e}(\gamma D)^x) W^{e/2}.\end{aligned}$$

Logo,

$$\text{ldet} M_w(x) = \sum_{i=1}^n \ln(w_i) + \text{ldet}(\gamma X^{e/2}(C - W^{-e} D) X^{e/2} + W^{-e}(\gamma D)^x)$$

Utilizando a regra da cadeia e (3.3), obtemos

$$\begin{aligned}
\frac{\partial}{\partial w_i} \text{ldet } M_w(x) &= \frac{1}{w_i} + (\gamma X^{e/2}(C - W^{-e}D)X^{e/2} + W^{-e}(\gamma D)^x)_{ii}^{-1} (x_i \gamma d_i - (\gamma d_i)^{x_i}) w_i^{-2} \\
&= \frac{1}{w_i} (1 + [M_w^{-1}(x)]_{ii} (x_i \gamma d_i - (\gamma d_i)^{x_i})) \\
&= \frac{1}{w_i} (1 + (x_i (\gamma d_i)^{1-x_i} - 1) [M_w^{-1}(x)]_{ii} (\gamma d_i)^{x_i})
\end{aligned}$$

No entanto,

$$M_w(x) \preceq (\gamma D)^x \Rightarrow M_w^{-1}(x) \succeq (\gamma D)^{-x} \Rightarrow (\gamma D)^x \circ M_w^{-1}(x) \succeq I \Rightarrow (\gamma d_i)^{x_i} [M_w^{-1}(x)]_{ii} \geq 1.$$

Portanto, $(x_i (\gamma d_i)^{1-x_i} - 1) [M_w^{-1}(x)]_{ii} (\gamma d_i)^{x_i} \leq (x_i (\gamma d_i)^{1-x_i} - 1)$.

Consequentemente,

$$\frac{\partial}{\partial w_i} \text{ldet } M_w(x) \leq \frac{1}{w_i} (1 + x_i (\gamma d_i)^{1-x_i} - 1) = \frac{1}{w_i} (x_i (\gamma d_i)^{1-x_i}) \leq \frac{x_i}{w_i}$$

Onde as três últimas inequações resultam do fato de que $(\gamma d_i) \leq 1$ e $0 \leq x_i \leq 1$.

Finalmente,

$$\frac{\partial}{\partial \beta} \text{ldet } f_{\gamma, \beta}(x) = \sum_{i=1}^n \frac{\partial}{\partial w_i} \text{ldet } M_w(x) |_{w=\beta e} - \frac{s}{\beta} \leq \sum_{i=1}^n \frac{x_i}{\beta} - \frac{s}{\beta} = 0$$

Onde usamos o fato de que $e^T x = s$. ■

Fixando o índice i em S

Consideraremos agora o caso em que o problema filho é gerado ao fixarmos o índice i em S no problema pai.

As funções objetivos dos problemas pai e filho são dadas respectivamente por:

$$\bar{f}_{\gamma_1, \beta_1}(\bar{x}) = \text{ldet}(\bar{X}^{e/2}(\beta_1 \gamma_1 \bar{C} - \gamma_1 D_1) \bar{X}^{e/2} + (\gamma_1 D_1)^{\bar{x}}) - s \ln(\gamma_1 \beta_1) \quad (5.15)$$

e

$$f_{\gamma_2, \beta_2}(x) = \ln(\bar{C}_{ii}) + \text{ldet}(X^{e/2}(\beta_2 \gamma_2 C - \gamma_2 D_2) X^{e/2} + (\gamma_2 D_2)^x) - (s-1) \ln(\gamma_2 \beta_2) \quad (5.16)$$

onde:

- C é o complemento de Schur de \bar{C}_{ii} em \bar{C} , ou seja:

$$C = \bar{C}_{TT} - \frac{\bar{C}_{Ti} \bar{C}_{iT}}{\bar{C}_{ii}}$$

onde $T = \{1, \dots, i-1\} \cup \{i+1, \dots, n\}$

- $D_1 = \text{Diag}(\bar{C})$ e $D_2 = \text{Diag}(C)$
- $\gamma_1 = 1/d_{1_{max}}$ e $\gamma_2 = 1/d_{2_{max}}$
- $\beta_1 = 1/\rho(D_1^{-\frac{1}{2}}\bar{C}D_1^{-\frac{1}{2}})$ e $\beta_2 = 1/\rho(D_2^{-\frac{1}{2}}CD_2^{-\frac{1}{2}})$

Neste caso assumiremos que $0 < \beta_1 \leq \beta_2$.

Teorema 5.3.5 *Sejam $\bar{f}_{\gamma_1, \beta_1}(\cdot)$ e $f_{\gamma_2, \beta_2}(\cdot)$ definidas por (5.15) e (5.16). Então, assumindo-se $0 < \beta_1 \leq \beta_2$, temos*

$$\bar{f}_{\gamma_1, \beta_1}(\bar{x}^*) \geq f_{\gamma_2, \beta_2}(x^*)$$

Demonstração: Seja:

$$\hat{x}_j = \begin{cases} x_j^*, & \text{if } j < i \\ 1, & \text{if } j = i \\ x_{j-1}^*, & \text{if } j > i \end{cases}$$

Observamos que \hat{x} é uma solução viável para o problema pai. Logo, é suficiente mostrar que $\bar{f}_{\gamma_1, \beta_1}(\hat{x}) \geq f_{\gamma_2, \beta_2}(x^*)$.

Como na demonstração do Teorema 5.3.2, definimos:

$$M_{\gamma_1, \beta_1}(\hat{x}) = \hat{X}^{e/2}(\beta_1 \gamma_1 \bar{C} - \gamma_1 D_1) \hat{X}^{e/2} + (\gamma_1 D_1)^{\hat{x}}$$

Logo:

$$\bar{f}_{\gamma_1, \beta_1}(\hat{x}) = \text{ldet}(M_{\gamma_1, \beta_1}(\hat{x})) - s \ln(\gamma_1 \beta_1) \quad (5.17)$$

Como $\hat{x}_i = 1$, $[M_{\gamma_1, \beta_1}(\hat{x})]_{ii} = \beta_1 \gamma_1 \bar{C}_{ii}$ e conseqüentemente,

$$\text{ldet}(M_{\gamma_1, \beta_1}(\hat{x})) = \ln(\beta_1 \gamma_1 \bar{C}_{ii}) + \text{ldet}(B)$$

onde:

$$B = [M_{\gamma_1, \beta_1}(\hat{x})]_{TT} - \frac{[M_{\gamma_1, \beta_1}(\hat{x})]_{Ti} [M_{\gamma_1, \beta_1}(\hat{x})]_{iT}}{\beta_1 \gamma_1 \bar{C}_{ii}}$$

Verificamos facilmente que:

$$B = X^{*e/2}(\beta_1 \gamma_1 C - \gamma_1 [D_1]_{TT}) X^{*e/2} + (\gamma_1 [D_1]_{TT})^{x^*}$$

Portanto,

$$\begin{aligned} \bar{f}_{\gamma_1, \beta_1}(\hat{x}) &= \ln(\beta_1 \gamma_1 \bar{C}_{ii}) + \text{ldet}(X^{*e/2}(\beta_1 \gamma_1 C - \gamma_1 [D_1]_{TT}) X^{*e/2} + (\gamma_1 [D_1]_{TT})^{x^*}) - s \ln(\gamma_1 \beta_1) \\ &= \ln(\bar{C}_{ii}) + \text{ldet}(X^{*e/2}(\beta_1 \gamma_1 C - \gamma_1 [D_1]_{TT}) X^{*e/2} + (\gamma_1 [D_1]_{TT})^{x^*}) - (s-1) \ln(\gamma_1 \beta_1) \end{aligned}$$

Observamos ainda que:

$$\begin{aligned} [D_1]_{TT} &= \text{Diag}(\bar{C}_{TT}) \\ D_2 &= \text{Diag}(C) = \text{Diag}(\bar{C}_{TT} - \frac{\bar{C}_{Ti} \bar{C}_{iT}}{\bar{C}_{ii}}) \\ &= \text{Diag}(\bar{C}_{TT}) - \text{Diag}(\frac{\bar{C}_{Ti} \bar{C}_{iT}}{\bar{C}_{ii}}) \end{aligned}$$

Logo, definindo $D := [D_1]_{TT} - D_2$, temos:

$$D = \text{Diag} \left(\frac{\bar{C}_{Ti}\bar{C}_{iT}}{\bar{C}_{ii}} \right)$$

E conseqüentemente para $j \in \{1, \dots, n-1\}$,

$$\begin{aligned} D_{jj} &= \frac{\bar{C}_{ji}\bar{C}_{ij}}{\bar{C}_{ii}} = \frac{\bar{C}_{ij}^2}{\bar{C}_{ii}} \\ \Rightarrow \gamma_1 D_{jj} &= \frac{\bar{C}_{ij}^2}{d_{1max}\bar{C}_{ii}} \leq \frac{\bar{C}_{ij}^2}{\bar{C}_{jj}\bar{C}_{ii}} < 1 \end{aligned}$$

onde a última inequação resulta do fato de que \bar{C} é uma matriz definida positiva. Como $0 \leq x_j^* \leq 1$ e $\gamma_1 d_j < 1$, temos agora que:

$$\begin{aligned} (\gamma_1 d_j)^{x_j^*} &\geq \gamma_1 x_j d_j \\ \Rightarrow (\gamma_1 D)^{x^*} &\geq \gamma_1 X^{*e/2} D X^{*e/2} \\ \Rightarrow (\gamma_1 [D_1]_{TT})^{x^*} - X^{*e/2} \gamma_1 [D_1]_{TT} X^{*e/2} &\geq (\gamma_1 D_2) x^* - X^{*e/2} \gamma_1 D_2 X^{*e/2} \\ \Rightarrow \bar{f}_{\gamma_1, \beta_1}(\hat{x}) &\geq \ln(\bar{C}_{ii}) + \text{l det}(X^{*e/2}(\beta_1 \gamma_1 C - \gamma_1 D_2) X^{*e/2} + (\gamma_1 D_2)^{x^*}) \\ &\quad - (s-1) \ln(\gamma_1 \beta_1) \end{aligned}$$

Finalmente notamos que:

- Sendo $\bar{C}_{kk} - \frac{\bar{C}_{ki}\bar{C}_{ik}}{\bar{C}_{ii}} := \max_j (\bar{C}_{jj} - \frac{\bar{C}_{ji}\bar{C}_{ij}}{\bar{C}_{ii}})$, temos $d_{1max} \geq \bar{C}_{kk} \geq d_{2max}$. Conseqüentemente, $\gamma_1 \leq \gamma_2$

Logo, sendo $0 < \gamma_1 \leq \gamma_2$ e considerando $0 < \beta_1 \leq \beta_2$, é suficiente mostrar que:

$$\frac{\partial}{\partial \gamma} f_{\gamma, \beta}(x) \leq 0 \quad \text{and} \quad \frac{\partial}{\partial \beta} f_{\gamma, \beta}(x) \leq 0$$

para $\gamma > 0$ e $\beta > 0$, os quais podem ser demonstrados usando-se os mesmos argumentos utilizados na demonstração do Teorema 5.3.4, considerando agora o fato de que $e^T x = s-1$.

■

5.4 Aspectos da Implementação

Nesta seção consideraremos alguns aspectos da implementação da relaxação não linear para o *Problema de Máxima Entropia* que estão bastante relacionados à obtenção de bons limites superiores.

Na Seção 3.3.4 sugerimos três métodos de escolha para D e γ . Nos dois últimos métodos, a escolha de γ não é única, ela fica limitada ao intervalo $[1/d_{max}, 1/d_{min}]$. A teoria apresentada

até o momento não nos fornece dados suficientes para afirmar qual ou quais destas opções de D e γ funcionarão melhor na prática. Através da implementação dos três métodos e da variação de γ no referido intervalo, podemos então antecipar, a partir dos limites obtidos para alguns subproblemas, quais das opções deverão levar em geral à solução mais rápida do problema original.

Nos gráficos e tabelas abaixo, “EIG” refere-se aos limites baseados nos autovalores, “NLP” refere-se aos limites baseados na relaxação não linear com o problema original, utilizando-se C , e “INV” aos limites obtidos com o problema complementar, utilizando-se C^{-1} . “Id” (identidade), “Di” (diagonal), e “Tr” (traço) referem-se aos três métodos de escolha de D e γ sugeridos na Seção 3.3.4.

Para ilustrar os resultados obtidos, consideramos alguns dos subproblemas resolvidos, e mostramos inicialmente nas tabelas 5.1 a 5.6, comparações entre os limites superiores e tempo de execução associado, para diferentes escolhas de D e γ . Estabelecemos a princípio, $\gamma = 0.9(1/d_{max}) + 0.1(1/d_{min})$ nas duas últimas opções. Observamos que nestes exemplos, $M = \emptyset$, ou seja, estes problemas são irrestritos.

Primeiramente consideramos um exemplo de [48] e apresentamos seus resultados nas tabelas 5.1 e 5.2. Os dados originais destes problemas derivam de 52 estações de monitoria de chuva ácida nos Estados Unidos. Após fixar 36 estações, nos resta o problema irrestrito de escolher s entre as 16 estações restantes de forma a maximizar a entropia condicional. Consideramos os problemas para todos os valores de s de 1 a 15 e os denotamos por 52/16/ s . Em seguida consideramos um exemplo de [42] (também resolvido em [48]), e apresentamos os resultados obtidos nas tabelas 5.3 e 5.4. Neste caso, os dados do problema derivam de uma rede de 63 estações de observação ambiental nos Estados Unidos, para o controle da concentração de sulfato. Cada matriz de ordem 27 utilizada corresponde a matriz de covariância condicional gerada ao fixarmos uma escolha particular de 36 estações. O problema foi resolvido para os valores de s de 1 a 26, portanto nossos problemas são problemas de expansão de uma rede, nos quais queremos aumentar a rede de 36 estações através da escolha ótima de mais s , dentre um conjunto de 27 estações potenciais. Denotamos estes problemas por 63/27/ s .

Por fim, no último exemplo considerado, os dados também são estimados de uma rede de estações de monitoria ambiental que controla a concentração de sulfato. Esta rede, com 36 estações, foi estudada por N.D. Le e J.V. Zidek, da Universidade de British Columbia, e foi novamente considerada em [48]. No problema resolvido, 9 estações estão fixas na solução, e escolhemos s dentre as 27 restantes, onde s varia de 1 a 26. Tais problemas são denominados 36/27/ s . Os resultados obtidos para estes problemas são apresentados nas tabelas 5.5 e 5.6.

Através das tabelas 5.1 a 5.6, verificamos que o uso da estratégia “Di”, em geral resulta em melhores limites do que o uso das estratégias “Id”, e “Tr” resulta em ainda melhores. Vale notar entretanto, que o aumento no tempo de solução quando a estratégia “Tr” é empregada, pode não ser compensado pela melhoria dos limites. Na próxima seção abordaremos novamente esta questão e veremos que a estratégia “Tr” realmente não compensa nos exemplos

Tab 5.1: Limites obtidos para os problemas 52/16/s

s	NLP-Id	NLP-Di	NLP-Tr	Melhor Limite
1	1.02434	1.04708	1.02916	Id
2	1.88448	1.90551	1.93076	Id
3	2.67476	2.69554	2.72065	Id
4	3.40870	3.42738	3.45722	Id
5	4.07455	4.09519	4.10890	Id
6	4.67618	4.69966	4.67641	Id
7	5.22501	5.24080	5.19029	Tr
8	5.71907	5.71833	5.64616	Tr
9	6.15430	6.12344	6.04100	Tr
10	6.53255	6.45252	6.37260	Tr
11	6.85364	6.70187	6.61341	Tr
12	7.10372	6.85175	6.76372	Tr
13	7.25166	6.91452	6.80833	Tr
14	7.25916	6.87951	6.76404	Tr
15	7.02028	6.69536	6.60701	Tr

Tab 5.2: Tempo em *wall seconds* para obtenção dos limites para os problemas 52/16/s

s	NLP-Id	NLP-Di	NLP-Tr	Menor Tempo
1	0.20	0.21	0.37	Id
2	0.15	0.15	0.29	Id,Di
3	0.14	0.15	0.30	Id
4	0.15	0.15	0.28	Id,Di
5	0.14	0.13	0.28	Di
6	0.15	0.15	0.27	Id,Di
7	0.15	0.14	0.30	Di
8	0.16	0.15	0.29	Di
9	0.15	0.14	0.30	Di
10	0.15	0.14	0.30	Di
11	0.14	0.16	0.31	Id
12	0.15	0.14	0.31	Di
13	0.15	0.15	0.30	Id,Di
14	0.14	0.14	0.28	Id,Di
15	0.13	0.14	0.28	Id

Tab 5.3: Limites obtidos para os problemas 63/27/*s*

s	NLP-Id	NLP-Di	NLP-Tr	Melhor Limite
1	1.10079	1.24358	1.14407	Id
2	2.18645	2.29896	2.20890	Id
3	3.21071	3.25822	3.22004	Id
4	4.12330	4.16169	4.13396	Id
5	4.98760	5.02244	5.00495	Id
6	5.81511	5.84858	5.84621	Id
7	6.61373	6.64541	6.65456	Id
8	7.38641	7.41633	7.42026	Id
9	8.12893	8.15820	8.14591	Id
10	8.83997	8.86533	8.84134	Id
11	9.51716	9.53678	9.49735	Tr
12	10.1662	10.1678	10.1084	Tr
13	10.7677	10.7584	10.6803	Tr
14	11.3410	11.3052	11.2195	Tr
15	11.8793	11.8013	11.7199	Tr
16	12.3824	12.2497	11.1687	Tr
17	12.8522	12.6538	12.5583	Tr
18	13.2864	13.0095	12.8940	Tr
19	13.6789	13.3100	13.1736	Tr
20	14.0213	13.5460	13.3980	Tr
21	14.3031	13.7202	13.5617	Tr
22	14.5097	13.8347	13.6501	Tr
23	14.6215	13.8837	13.6772	Tr
24	14.6010	13.8533	13.6512	Tr
25	14.4036	13.7147	13.5455	Tr
26	13.9268	13.4178	13.3111	Tr

Tab 5.4: Tempo em *wall seconds* para obtenção dos limites para os problemas 63/27/*s*

s	NLP-Id	NLP-Di	NLP-Tr	Menor Tempo
1	0.66	0.81	1.30	Id
2	0.64	0.67	1.20	Id
3	0.63	0.68	1.25	Id
4	0.68	0.76	1.25	Id
5	0.72	0.76	1.25	Id
6	0.68	0.72	1.25	Id
7	0.67	0.68	1.24	Id
8	0.75	0.68	1.24	Di
9	0.67	0.72	1.21	Id
10	0.71	0.67	1.29	Di
11	0.72	0.68	1.21	Di
12	0.80	0.67	1.25	Di
13	0.76	0.64	1.16	Di
14	0.72	0.67	1.25	Di
15	0.76	0.68	1.29	Di
16	0.67	0.67	1.24	Id,Di
17	0.68	0.71	1.29	Id
18	0.72	0.68	1.24	Di
19	0.73	0.76	1.29	Id
20	0.67	0.72	1.25	Id
21	0.70	0.68	1.26	Di
22	0.71	0.67	1.24	Di
23	0.68	0.68	1.20	Id,Di
24	0.72	0.67	1.25	Di
25	0.67	0.67	1.21	Id,Di
26	0.72	0.72	1.16	Id,Di

Tab 5.5: Limites obtidos para os problemas 36/27/s

s	NLP-Id	NLP-Di	NLP-Tr	Melhor Limite
1	6.44125	6.77244	6.76233	Id
2	12.7589	13.2493	13.1646	Id
3	19.0237	19.6548	19.4243	Id
4	25.2509	26.0073	25.6196	Id
5	31.4445	32.3117	31.7548	Id
6	37.6056	38.5679	37.8410	Id
7	43.7321	44.7748	43.8822	Id
8	49.8216	50.9340	49.8754	Id
9	55.8706	57.0464	55.8253	Tr
10	61.8755	63.1092	61.7293	Tr
11	67.8318	69.1207	67.5819	Tr
12	73.7343	75.0764	73.3766	Tr
13	79.5773	80.9696	79.1094	Tr
14	85.3534	86.7923	84.7832	Tr
15	91.0544	92.5348	90.3914	Tr
16	96.6698	98.1855	95.9236	Tr
17	102.187	103.730	101.367	Tr
18	107.590	109.150	106.707	Tr
19	112.857	114.423	111.922	Tr
20	117.961	119.517	116.988	Tr
21	122.867	124.391	121.870	Tr
22	127.518	128.987	126.518	Tr
23	131.834	133.213	130.861	Tr
24	135.687	136.928	134.781	Tr
25	138.823	139.866	138.052	Tr
26	140.617	141.374	140.133	Tr

Tab 5.6: Tempo em *wall seconds* para obtenção dos limites para os problemas 36/27/*s*

s	NLP-Id	NLP-Di	NLP-Tr	Menor Tempo
1	0.69	0.76	1.25	Id
2	0.50	0.64	1.21	Id
3	0.56	0.64	1.17	Id
4	0.42	0.59	1.20	Id
5	0.34	0.59	1.18	Id
6	0.25	0.55	1.14	Id
7	0.26	0.55	1.12	Id
8	0.25	0.47	1.04	Id
9	0.26	0.38	1.00	Id
10	0.26	0.30	1.00	Id
11	0.25	0.26	1.00	Id
12	0.26	0.25	1.03	Di
13	0.25	0.26	0.80	Id
14	0.25	0.25	0.79	Id,Di
15	0.25	0.26	0.79	Id
16	0.26	0.26	0.79	Id,Di
17	0.34	0.25	0.78	Di
18	0.34	0.26	0.78	Di
19	0.38	0.25	0.79	Di
20	0.42	0.26	0.79	Di
21	0.51	0.26	0.79	Di
22	0.51	0.25	0.79	Di
23	0.55	0.26	0.78	Di
24	0.59	0.25	0.95	Di
25	0.63	0.39	1.08	Di
26	0.63	0.60	1.13	Di

apresentados, quando consideramos a resolução do problema inteiro CMESP pelo método de *branch-and-bound*.

A obtenção dos resultados apresentados nas tabelas 5.1 a 5.6, nos permitiu comparar os limites obtidos pela relaxação não linear NLP com os limites baseados em autovalores, utilizados em [48]. Foi através desta comparação que verificamos que a relaxação NLP nos fornece limites bastante superiores aos limites obtidos pela estratégia “EIG”, mas apenas para valores de s aproximadamente menores do que $n/2$. Para ilustrar os resultados obtidos, consideramos os mesmos subproblemas abordados nas tabelas 5.1 e 5.2, e mostramos nas figuras 5.2 a 5.4, o *gap* entre o limite obtido através da relaxação NLP e a solução ótima do problema inteiro associado, para diferentes valores de s e para as três opções de escolha de D e γ . Plotamos também o *gap* obtido quando os limites considerados são baseados nos autovalores da matriz de covariância. Vemos claramente que para valores altos de s , a estratégia “EIG” é superior à estratégia “NLP”, quando a relaxação na sua forma original é utilizada. Foi por este motivo que a relaxação complementar $\overline{\text{NLP}}$ definida na Seção 3.3.5 foi introduzida. Esta relaxação, ao contrário da relaxação original, nos fornece em geral limites melhores do que os obtidos pela estratégia “EIG”, para valores de s aproximadamente maiores do que $n/2$. Sendo assim, o procedimento que passou a ser seguido pela estratégia “NLP” foi o de calcular os limites baseados em ambas as relaxações, NLP e $\overline{\text{NLP}}$, e então considerar o melhor entre os dois. Os resultados obtidos com este procedimento são apresentados nas figuras 5.5 a 5.7. Para cada valor de s consideramos então, o limite baseado no problema original e o limite baseado no problema complementar. Podemos ver com clareza que, para estes problemas, a combinação dos dois limites obtidos é bem melhor do que os limites baseados nos autovalores.

Nos exemplos considerados até o momento, estabelecemos $\gamma = 0.9(1/d_{max}) + 0.1(1/d_{min})$, quando D e γ são determinados pela estratégia “Di” ou pela estratégia “Tr”. Nas figuras 5.8 a 5.10 vemos que esta é realmente uma escolha razoável ao calcularmos os limites para os referidos exemplos utilizando diferentes valores γ . Nos exemplos considerados, γ é dado pela expressão

$$(1 - \alpha) \frac{1}{d_{max}} + (\alpha) \frac{1}{d_{min}},$$

onde α varia de 0 a 1, com um intervalo de 0.1.

Uma outra questão relacionada à implementação do método proposto, é a de como escolher o tamanho do passo de Newton em cada iteração do algoritmo de barreira logarítmica visto na Seção 3.3.7.1. Na implementação por nós realizada, o tamanho do passo pode ser determinado por dois procedimentos distintos. No primeiro, calcula-se o tamanho do passo que nos leva até a fronteira da região viável e o denotamos por α_{max} . Em seguida estabelecemos $\alpha = \theta \alpha_{max}$ para algum $0 < \theta < 1$. Verificamos então se a função barreira logarítmica dada por (3.61), decresce quando tomamos um passo de tamanho α . Se este for o caso, confirmamos o tamanho do passo como α . Caso contrário, estabelecemos repetidamente $\alpha := 0.9\alpha$ até alcançarmos um ponto em que a função barreira tenha decrescido.

O segundo procedimento para determinar o tamanho do passo de Newton é mais sofisti-

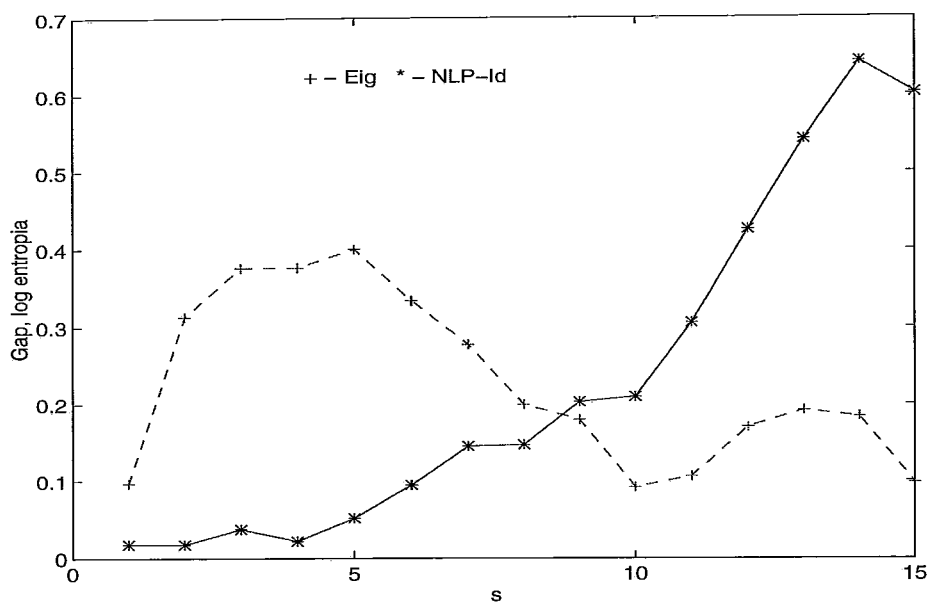


Fig 5.2: Gaps entre soluções ótimas inteiras e limites gerados pelo método “Identidade” e pelo método dos autovalores para os problemas 52/16/s

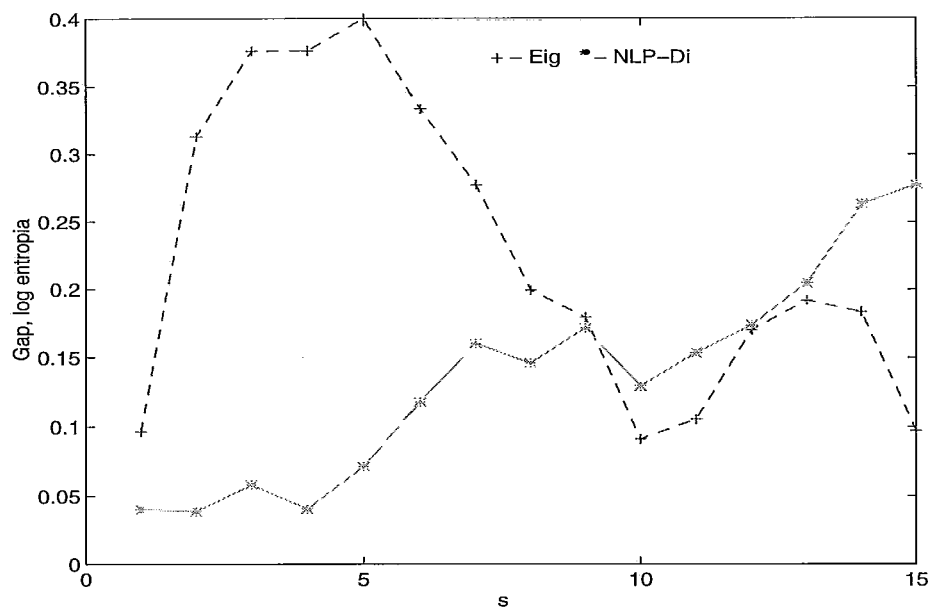


Fig 5.3: Gaps entre soluções ótimas inteiras e limites gerados pelo método “Diagonal” e pelo método dos autovalores para os problemas 52/16/s

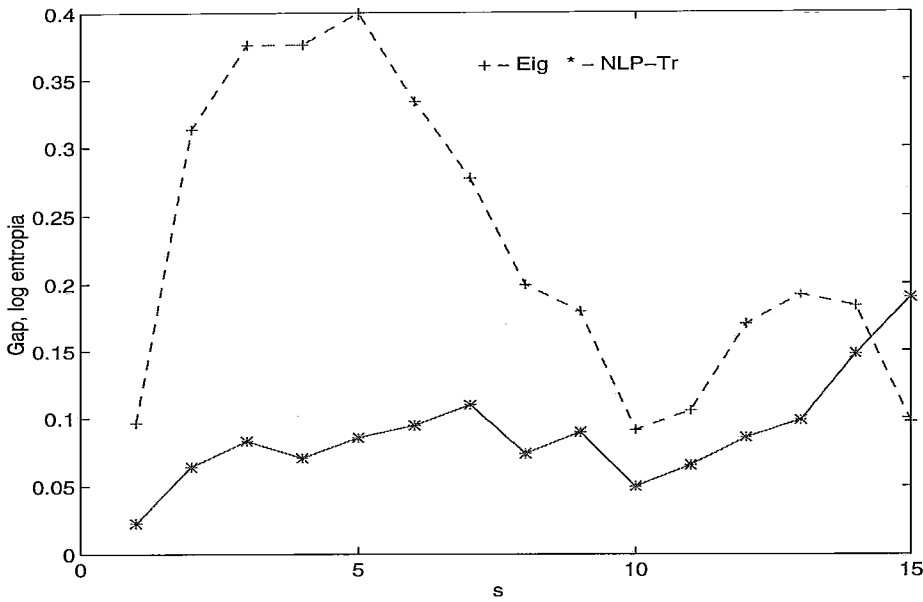


Fig 5.4: Gaps entre soluções ótimas inteiras e limites gerados pelo método “Traço Mínimo” e pelo método dos autovalores para os problemas 52/16/s

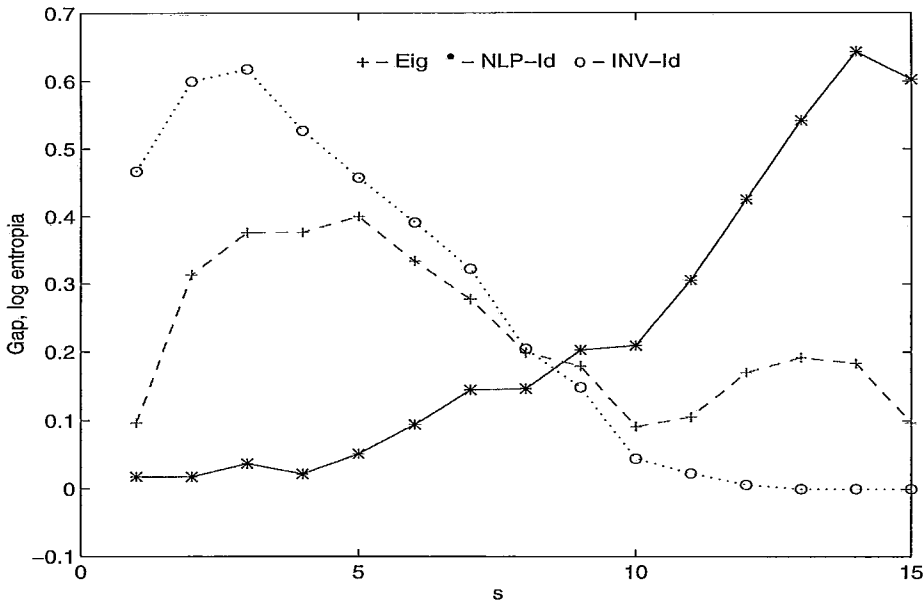


Fig 5.5: Gaps entre soluções ótimas inteiras e limites gerados pelo método “Identidade”, utilizando-se relaxações originais e complementares, e pelo método dos autovalores para os problemas 52/16/s

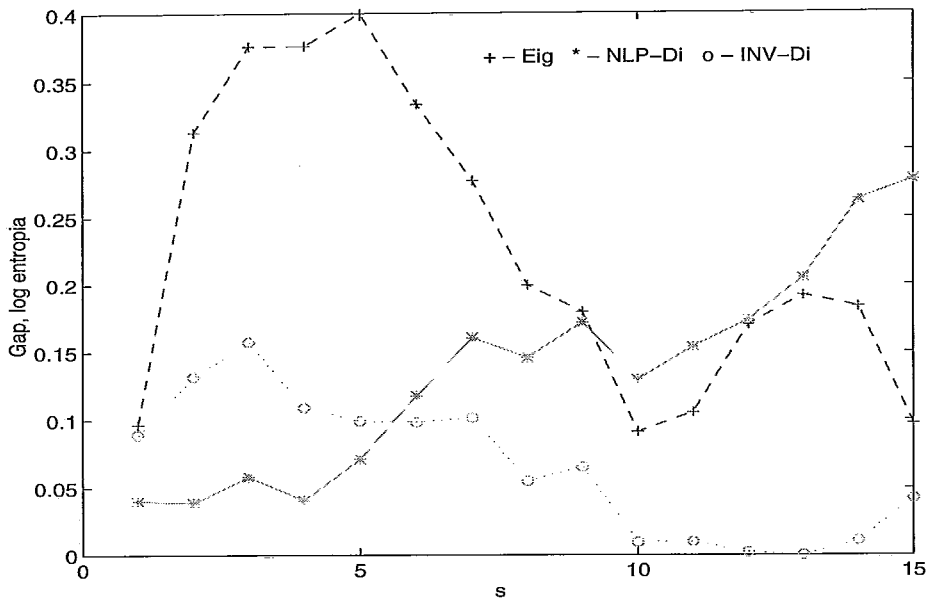


Fig 5.6: Gaps entre soluções ótimas inteiras e limites gerados pelo método “Diagonal”, utilizando-se relaxações originais e complementares, e pelo método dos autovalores para os problemas 52/16/s

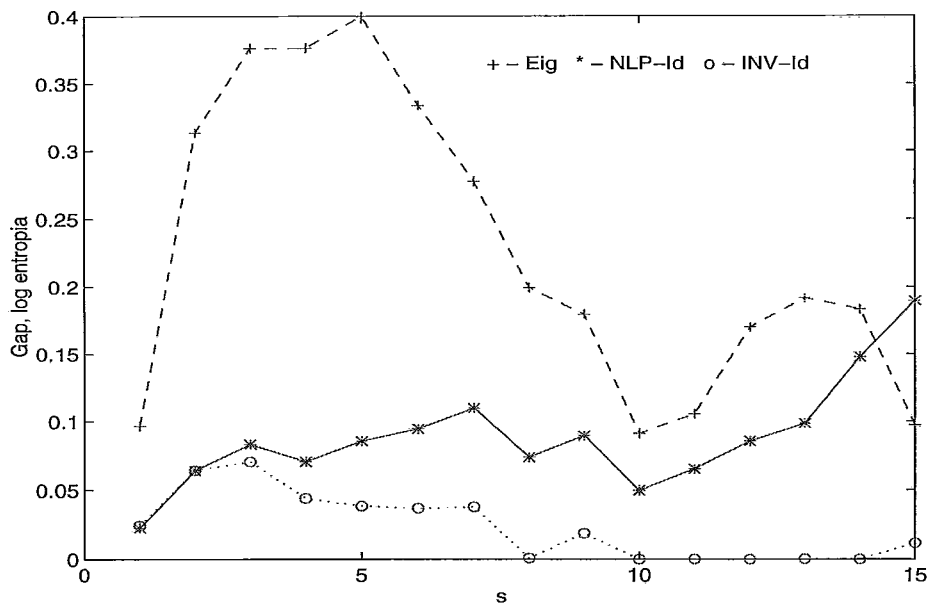


Fig 5.7: Gaps entre soluções ótimas inteiras e limites gerados pelo método “Traço Mínimo”, utilizando-se relaxações originais e complementares, e pelo método dos autovalores para os problemas 52/16/s

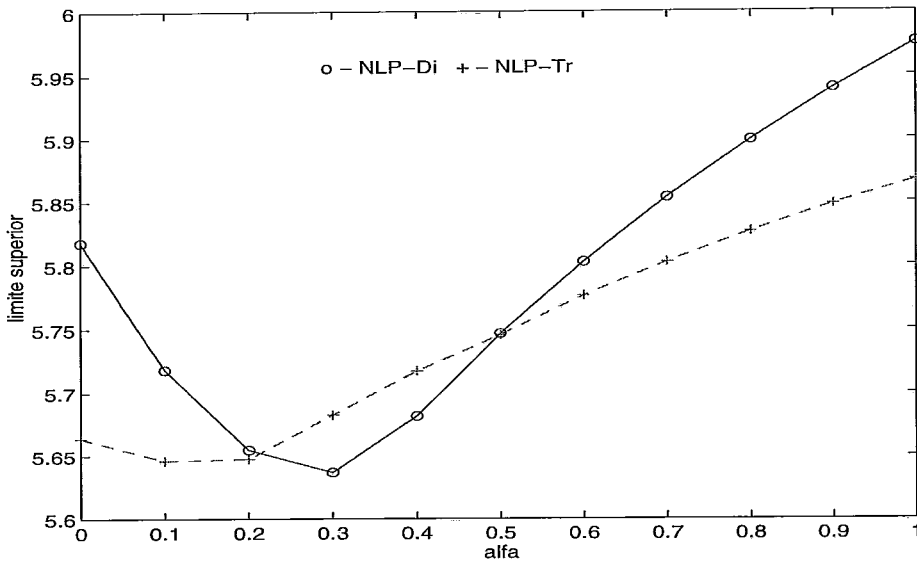


Fig 5.8: Limites obtidos estabelecendo-se $\gamma = (1 - \alpha)1/d_{max} + (\alpha)1/d_{min}$ para os problemas 52/16/s

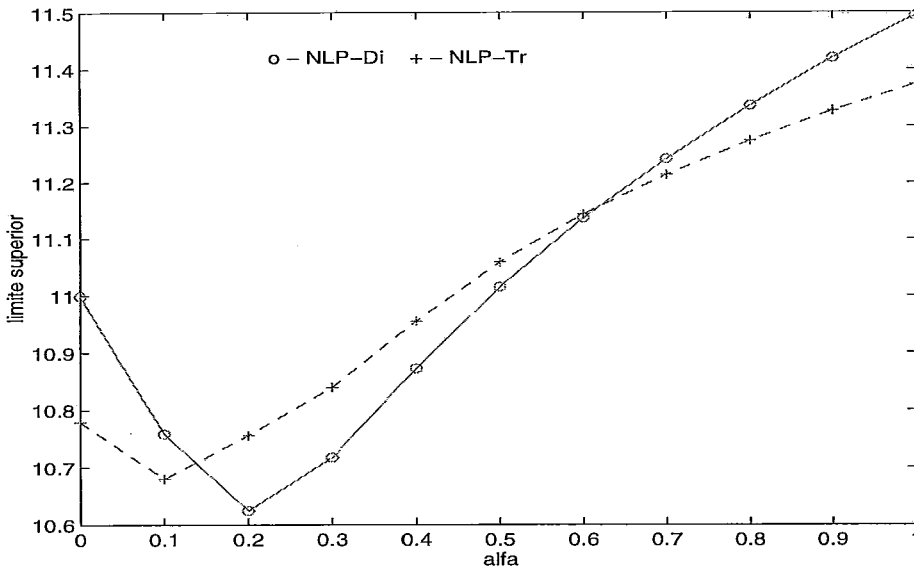


Fig 5.9: Limites obtidos estabelecendo-se $\gamma = (1 - \alpha)1/d_{max} + (\alpha)1/d_{min}$ para os problemas 63/27/s

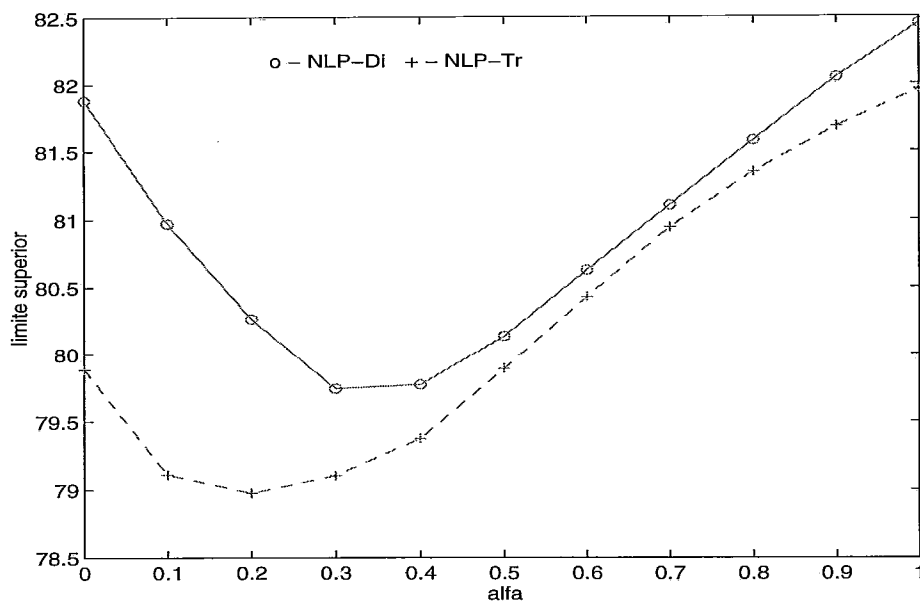


Fig 5.10: Limites obtidos estabelecendo-se $\gamma = (1-\alpha)1/d_{max} + (\alpha)1/d_{min}$ para os problemas 36/27/s

cado do que o primeiro, de forma a consumir maior tempo de execução quando utilizado. No entanto, ele poderá ser válido no caso em que o primeiro procedimento produz um decréscimo na função barreira muito pequeno, quando comparado ao decréscimo por ele produzido. A sofisticação atribuída a este procedimento é dada pela introdução de uma busca linear realizada sobre uma aproximação de segunda ordem da função barreira (3.61), na direção de Newton denotada por p . Sendo assim, dado o ponto corrente x e a direção de Newton p , definimos:

$$\begin{aligned}\tau(\alpha) &:= F(x + \alpha p, \mu) \\ &\approx -[f(x) + \alpha p^T \nabla f(x) + \frac{\alpha^2}{2} p^T \nabla^2 f(x) p] - \mu \sum_{j=1}^n \log(x_j + \alpha p_j) - \mu \sum_{j=1}^n \log(1 - (x_j + \alpha p_j)) \\ &\quad - \mu \sum_{i=1}^m \log(b_i - a_i^T(x + \alpha p))\end{aligned}$$

e

$$\begin{aligned}\tau'(\alpha) &:= \frac{d}{d\alpha} F(x + \alpha p, \mu) \\ &\approx -p^T \nabla f(x) - \alpha p^T \nabla^2 f(x) p - \mu \left[\sum_{j=1}^n \frac{p_j}{x_j + \alpha p_j} + \sum_{j=1}^n \frac{-p_j}{1 - (x_j + \alpha p_j)} + \sum_{i=1}^m \frac{-a_i^T p}{b_i - a_i^T(x + \alpha p)} \right]\end{aligned}$$

Com a finalidade de estabelecer o tamanho do passo α que minimiza a aproximação da função barreira na direção de Newton $\tau(\cdot)$, procuramos então o zero de $\tau'(\cdot)$, através do seguinte procedimento:

Procedimento para encontrar o intervalo onde é realizada a busca linear que determina o tamanho do passo de Newton

Início

$$\alpha := 0.9\alpha_{max};$$

Se $\tau'(\alpha) < 0$ **então**

$$I = [0.9\alpha_{max}, \alpha_{max}]$$

Senão

Enquanto $\tau'(\alpha) \geq 0$ **faça** $\alpha = 0.7\alpha$

$$I = [\alpha, \frac{\alpha}{0.7}]$$

Fim

onde α_{max} é novamente o passo que nos leva até a fronteira da região viável.

Finalmente, dado o intervalo inicial I , aplicamos o conhecido algoritmo de bissecção para minimização unidimensional. Neste caso, não faz sentido trabalhar com uma precisão muito

grande, já que estamos avaliando uma aproximação da função barreira. Sendo assim, dividimos o intervalo I em apenas 10 iterações no algoritmo de bisseção, como mostra o procedimento abaixo:

Algoritmo de Bisseção

Dados:

Intervalo inicial $I = [v, w]$, onde $\tau'(v) < 0$ e $\tau'(w) > 0$,

Início

Repita 10 vezes:

$$\lambda := (v + w)/2$$

Se $\tau'(\lambda) < 0$ **então** $v := \lambda$

Senão $w := \lambda$

Fim

Nas tabelas 5.7 a 5.9, comparamos o tempo de execução necessário para obtenção de limites quando utilizamos o primeiro procedimento, com o tempo necessário quando utilizamos a busca linear. Observamos que a busca linear não produz diminuição nos tempos, e concluímos então que, ao menos para estes subproblemas, o esforço computacional relacionado à busca linear não é compensado pela diminuição no número de iterações internas realizadas pelo algoritmo.

Por fim, mostraremos ainda nesta seção como se comporta na prática o procedimento discutido na Seção 3.3.6 para fixar variáveis em zero ou um, com base na solução dual. Na tabela 5.10 mostramos os resultados obtidos para os problemas 52/16/ s , com s variando de 1 a 15. A tabela contém o número de variáveis elegíveis restantes após a aplicação do procedimento e o número de relaxações resolvidas. Para melhor se verificar a surpreendente eficiência do procedimento, mostramos no gráfico 5.11, o *gap* entre o limite obtido através da relaxação NLP e a solução ótima do problema inteiro associado, para os diferentes valores de s , antes e após fixar-se variáveis. Observamos que para a maior parte dos problemas chegamos à solução ótima do problema inteiro associado, sem nem mesmo ter que aplicar o algoritmo de *branch-and-bound*. No caso dos problemas em que s está próximo de $n/2$, apesar de não chegarmos à solução ótima, melhoramos bastante o limite gerado. É interessante observar que estes problemas, nos quais $s \approx n/2$, são ressaltados na literatura pela sua maior dificuldade. Tal dificuldade é naturalmente esperada já que no caso em que $s \approx n/2$, existe um número maior de combinações dos índices da matriz que levam a submatrizes com a dimensão desejada.

Tab 5.7: Tempo de execução em *wall seconds* com e sem busca linear (BL) para os problemas 52/16/s

s	NLP-I		NLP-Di		NLP-Tr	
	sem BL	com BL	sem BL	com BL	sem BL	com BL
1	0.20	0.21	0.22	0.23	0.38	0.39
2	0.15	0.17	0.15	0.16	0.28	0.30
3	0.14	0.16	0.15	0.17	0.29	0.30
4	0.16	0.16	0.14	0.19	0.28	0.30
5	0.15	0.17	0.14	0.17	0.29	0.31
6	0.16	0.17	0.15	0.16	0.29	0.30
7	0.15	0.17	0.13	0.15	0.28	0.30
8	0.17	0.19	0.15	0.16	0.29	0.31
9	0.16	0.18	0.13	0.15	0.30	0.31
10	0.16	0.17	0.15	0.16	0.30	0.32
11	0.15	0.15	0.16	0.18	0.31	0.29
12	0.15	0.17	0.15	0.16	0.29	0.31
13	0.16	0.16	0.14	0.17	0.29	0.31
14	0.15	0.17	0.14	0.15	0.29	0.30
15	0.14	0.17	0.14	0.16	0.28	0.30

Tab 5.8: Tempo de execução em *wall seconds* com e sem busca linear (BL) para os problemas 63/27/s

s	NLP-I		NLP-Di		NLP-Tr	
	sem BL	com BL	sem BL	com BL	sem BL	com BL
1	0.69	0.75	0.77	0.82	1.26	1.30
2	0.53	0.59	0.63	0.70	1.25	1.27
3	0.57	0.53	0.63	0.64	1.16	1.22
4	0.44	0.50	0.59	0.61	1.21	1.22
5	0.36	0.32	0.59	0.65	1.17	1.18
6	0.28	0.33	0.55	0.60	1.12	1.09
7	0.27	0.32	0.55	0.56	1.12	1.14
8	0.27	0.28	0.46	0.48	1.04	1.02
9	0.27	0.28	0.38	0.30	0.99	0.96
10	0.28	0.27	0.30	0.30	1.01	0.97
11	0.28	0.28	0.25	0.26	1.00	0.96
12	0.27	0.28	0.26	0.25	1.03	0.87
13	0.28	0.28	0.25	0.26	0.79	0.83
14	0.27	0.28	0.26	0.26	0.78	0.78
15	0.28	0.28	0.25	0.26	0.78	0.80
16	0.28	0.28	0.26	0.26	0.78	0.79
17	0.35	0.32	0.25	0.26	0.78	0.79
18	0.37	0.36	0.26	0.25	0.79	0.78
19	0.41	0.37	0.25	0.26	0.78	0.79
20	0.44	0.36	0.26	0.26	0.79	0.79
21	0.53	0.45	0.25	0.25	0.79	0.78
22	0.53	0.45	0.26	0.26	0.78	0.79
23	0.57	0.54	0.25	0.25	0.79	0.79
24	0.61	0.67	0.25	0.26	0.95	0.92
25	0.65	0.67	0.37	0.34	1.08	1.08
26	0.66	0.63	0.60	0.61	1.12	1.18

Tab 5.9: Tempo de execução em *wall seconds* com e sem busca linear (BL) para os problemas 36/27/s

s	NLP-I		NLP-Di		NLP-Tr	
	sem BL	com BL	sem BL	com BL	sem BL	com BL
1	0.70	0.77	0.78	0.81	1.28	1.30
2	0.65	0.71	0.67	0.74	1.20	1.27
3	0.65	0.70	0.68	0.73	1.25	1.26
4	0.69	0.71	0.75	0.78	1.24	1.26
5	0.73	0.80	0.75	0.83	1.25	1.26
6	0.69	0.75	0.72	0.77	1.25	1.31
7	0.69	0.75	0.67	0.78	1.24	1.32
8	0.69	0.79	0.68	0.74	1.24	1.30
9	0.69	0.75	0.71	0.78	1.21	1.27
10	0.73	0.80	0.68	0.77	1.28	1.36
11	0.73	0.75	0.67	0.73	1.21	1.30
12	0.81	0.84	0.67	0.74	1.24	1.27
13	0.77	0.79	0.64	0.70	1.16	1.32
14	0.73	0.79	0.69	0.74	1.25	1.31
15	0.78	0.80	0.67	0.69	1.29	1.31
16	0.69	0.70	0.68	0.74	1.25	1.31
17	0.69	0.75	0.71	0.78	1.29	1.36
18	0.73	0.79	0.68	0.74	1.24	1.31
19	0.73	0.76	0.75	0.78	1.29	1.35
20	0.69	0.75	0.74	0.74	1.25	1.31
21	0.69	0.75	0.69	0.78	1.25	1.32
22	0.73	0.75	0.67	0.74	1.24	1.31
23	0.70	0.75	0.67	0.74	1.20	1.31
24	0.73	0.75	0.67	0.74	1.24	1.26
25	0.69	0.76	0.68	0.69	1.22	1.27
26	0.73	0.75	0.71	0.75	1.17	1.23

Tab 5.10: Fixando variáveis nos problemas 52/16/*s*

<i>s</i>	Gap Inicial	Gap Final	# Variáveis Restantes	# Relaxações Resolvidas
1	0.0227335	-1.06581e-14	0	2
2	0.0210214	-3.55271e-15	0	2
3	0.0402948	1.42109e-14	0	3
4	0.0229261	3.55271e-14	0	3
5	0.0542458	4.97380e-14	0	4
6	0.1001310	0.07021e-00	7	2
7	0.1428670	0.10066e-00	8	2
8	0.1282060	0.08596e-00	8	2
9	0.1543350	0.10639e-00	8	2
10	0.1118480	-2.48690e-14	0	2
11	0.1359720	-3.55271e-15	0	3
12	0.1559820	-1.77636e-14	0	3
13	0.1869520	3.55271e-14	0	3
14	0.2456580	-4.26326e-14	0	4
15	0.2603190	0	0	4

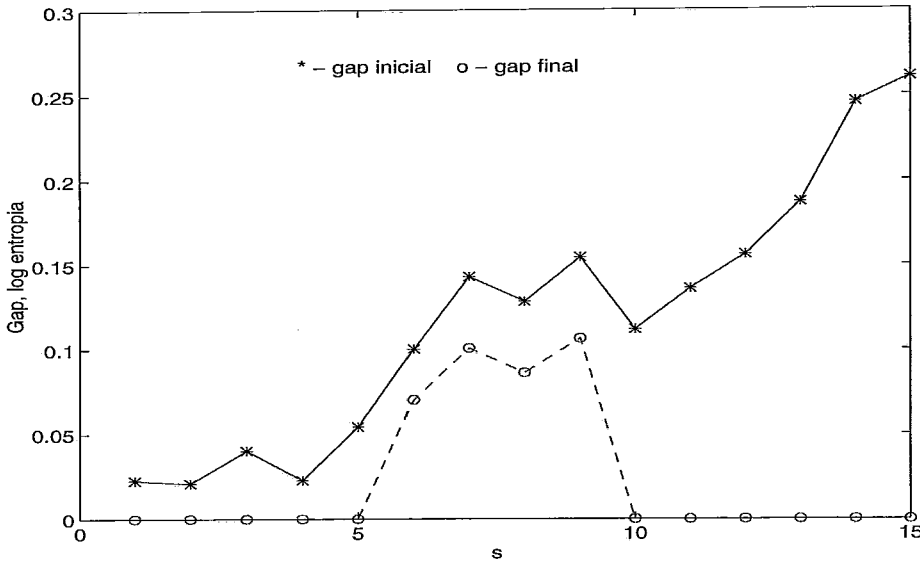


Fig 5.11: Gaps antes e depois de fixar-se variáveis nos problemas 52/16/s

5.5 Resultados Numéricos

Implementamos o algoritmo de *branch-and-bound* em C. Nosso programa faz uso do pacote de álgebra linear LAPACK 2.0 [6], o qual é escrito em FORTRAN 77, e do pacote de programação linear CPLEX 3.0 [19], o qual é escrito em C. As rotinas para a obtenção do limite superior requerem o cálculo de complementos de Schur, bases para espaço nulo de matrizes, inversão de matrizes, resolução de equações lineares, sistemas de autovalores, etc.. Para resolvê-los, usamos uma combinação de rotinas do LAPACK, e de rotinas específicas por nós implementadas. Para solucionar os problemas de programação linear dos itens 2 e 8 do algoritmo de *branch-and-bound*, utilizamos CPLEX.

Nós resolvemos uma variedade de problemas restritos e irrestritos. Nas tabelas 5.11 e 5.12, expomos os resultados do uso de dez diferentes matrizes de covariância (C0-C9), e cinco conjuntos de restrições (A0/A2/A5/A10/A15). As matrizes de covariância e as restrições são as mesmas que foram utilizadas em [53]. As matrizes de covariância são matrizes de ordem 30 que foram derivadas de uma matriz de ordem 63 usada em [42]. A matriz de ordem 63 foi derivada de dados de sulfato obtidos de uma rede de 63 estações de observação ambiental nos Estados Unidos. Cada matriz de ordem 30 utilizada corresponde a matriz de covariância condicional gerada ao fixarmos uma escolha particular de 33 estações. Todos os problemas foram resolvidos com $s = 15$, portanto nossos problemas são problemas de expansão de uma rede nos quais queremos aumentar a rede de 33 estações através da escolha ótima de mais 15, dentre um conjunto de 30 estações potenciais. (Observamos que em geral, para um dado n , a resolução de problemas CMESP é mais difícil quando s é aproximadamente $n/2$). Cada conjunto de restrições A_m contém m restrições. Usando todas as combinações

Tab 5.11: Número de limites calculados na resolução de 50 problemas

	Limites	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	Média	MAD
A0	EIG	241	266	782	1161	2178	328	1161	956	4837	565	1247.5	904.0
	NLP-Id	79	146	84	213	80	300	237	70	161	723	209.3	127.2
	NLP-Di	65	88	66	85	70	110	79	46	165	107	88.1	23.5
	NLP-Tr	57	62	58	73	46	84	69	46	113	65	67.3	14.0
A2	EIG	227	246	960	4344	3667	319	903	4835	3182	513	1919.6	1669.9
	NLP-Id	77	128	108	914	207	389	211	549	76	655	331.4	236.3
	NLP-Di	65	68	52	382	101	79	57	135	56	103	109.8	59.5
	NLP-Tr	57	56	52	286	55	59	57	87	56	63	82.8	41.5
A5	EIG	197	553	907	1618	3833	144	1236	728	5632	505	1535.3	1295.4
	NLP-Id	71	534	110	288	131	258	276	101	206	617	259.2	135.6
	NLP-Di	47	80	60	128	95	82	86	47	148	107	88.0	25.2
	NLP-Tr	43	58	52	100	53	52	72	43	78	65	61.6	13.7
A10	EIG	258	643	919	1023	4769	244	574	1819	2790	477	1351.6	1064.6
	NLP-Id	744	606	122	198	251	434	160	275	124	507	342.1	184.5
	NLP-Di	64	86	82	90	125	94	54	95	104	105	89.9	14.7
	NLP-Tr	54	54	52	80	69	56	52	79	56	65	61.7	9.2
A15	EIG	37	147	85	85	289	217	67	143	347	47	146.4	82.9
	NLP-Id	63	231	65	81	91	329	45	57	79	97	113.8	66.5
	NLP-Di	29	67	29	39	41	91	29	29	53	29	43.6	16.0
	NLP-Tr	29	49	29	29	29	53	29	29	29	29	33.4	7.0

Tab 5.12: Tempo em *wall seconds* para solucionar-se 50 problemas

	Limites	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	Média	MAD
A0	EIG	112	116	335	502	942	139	503	417	2092	245	540.4	390.7
	NLP-Id	16	22	15	33	13	45	41	13	19	106	32.3	19.2
	NLP-Di	16	12	12	13	12	21	15	8	17	18	14.5	2.9
	NLP-Tr	20	20	17	25	13	29	23	14	36	22	21.8	5.0
A2	EIG	137	148	511	2766	2399	221	490	2887	1881	314	1175.3	1046.3
	NLP-Id	26	34	33	229	62	112	63	143	21	180	90.4	60.7
	NLP-Di	20	18	14	102	32	28	16	42	15	31	31.8	16.1
	NLP-Tr	31	22	20	134	23	27	25	39	22	29	37.0	19.9
A5	EIG	149	367	537	992	2414	110	762	479	3622	326	975.6	820.1
	NLP-Id	25	158	36	86	42	80	85	30	57	182	78.1	40.1
	NLP-Di	20	25	23	49	30	27	28	13	43	35	29.3	7.9
	NLP-Tr	20	26	22	47	22	23	34	19	35	36	28.6	7.7
A10	EIG	225	572	659	897	4273	226	411	1330	1924	441	1096.0	847.9
	NLP-Id	250	207	48	68	99	145	63	87	39	174	117.9	60.8
	NLP-Di	21	29	29	30	41	33	17	29	33	38	30.0	4.9
	NLP-Tr	27	27	23	41	30	27	24	37	24	34	29.3	5.0
A15	EIG	76	214	157	142	401	236	148	225	477	86	216.3	94.8
	NLP-Id	16	49	18	24	23	66	14	17	19	27	27.4	12.1
	NLP-Di	8	18	8	13	11	24	8	8	14	8	11.8	4.3
	NLP-Tr	11	20	11	13	11	22	11	12	12	13	13.6	2.9

das dez matrizes e dos cinco conjuntos de restrições, trabalhamos com cinquenta problemas distintos. Cada um dos cinquenta problemas foi resolvido com o uso dos quatro métodos para obtenção do limite superior. Um dos métodos foi o método baseado em autovalores utilizado por Lee em [53]. Também utilizamos os limites gerados pela relaxação NLP com as três diferentes escolhas de D e γ sugeridas na Seção 3.3.4. (Para cada método de escolha de D e γ , calculamos o limite com o subproblema original e com o subproblema complementar descrito na Seção 3.3.5. Consideramos então o melhor entre os dois limites gerados.)

Executamos os programas numa HP 9000/715 com freqüência 50MH, uma workstation bastante modesta. Nas Tabelas 5.11 e 5.12, expomos estatísticas (número de limites superiores calculados, e tempo total de solução) para a resolução ótima destes problemas, sem empregarmos ainda a metodologia para fixar variáveis nos valores zero e um, descrita na Seção 3.3.6.

Nas tabelas, como na seção anterior, “EIG” refere-se aos limites baseados nos autovalores e “NLP” refere-se aos limites obtidos com a relaxação não linear. “Id” (identidade), “Di” (diagonal), e “Tr” (traço) referem-se aos três métodos de escolha de D e γ sugeridos na Seção 3.3.4. Nas duas últimas colunas são colocadas a média e a média do desvio absoluto (MAD), dos valores de cada linha. Através das tabelas verificamos com clareza que quando usamos os limites obtidos com a relaxação não linear a performance é bastante superior do que quando usamos os limites baseados nos autovalores. O uso da estratégia “Di”, em geral resulta em menos cálculos de limites superiores do que o uso das estratégias “Id,” e “Tr” resulta em menos ainda. Entretanto, nesses problemas, o aumento no tempo de solução quando a estratégia “Tr” é empregada, em geral não é compensado pela redução do número de cálculo de limites. É interessante também notar que o desvio relativo (MAD/Média) para ambas as estatísticas, é menor quando a relaxação não linear é utilizada, principalmente com as estratégias “Di” e “Tr” .

Nas tabelas 5.13 e 5.14 observamos os surpreendentes resultados obtidos ao aplicarmos aos mesmos cinquenta problemas, o procedimento descrito na Seção 3.3.6 para fixar variáveis nos valores zero e um baseados na solução dual do problema. O procedimento é aplicado repetidamente a um mesmo subproblema, ou seja, após fixar-se todas as possíveis variáveis num dado subproblema, resolve-se o subproblema reduzido que foi gerado e aplica-se novamente o procedimento. Este processo é então repetido até que nenhuma variável a mais possa ser fixada.

Tab 5.13: Número de limites calculados - Fixando variáveis - na resolução de 50 problemas

	Limites	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	Média	MAD
A0	NLP-Id	15	8	11	20	9	14	34	9	39	542	70.1	94.4
	NLP-Di	6	6	6	18	7	13	6	5	24	15	10.6	5.5
	NLP-Tr	11	5	11	10	7	14	7	6	11	11	9.3	2.4
A2	NLP-Id	16	14	11	293	18	32	33	140	29	512	109.8	123.1
	NLP-Di	6	7	6	53	19	9	6	28	18	15	16.7	10.2
	NLP-Tr	11	5	8	31	8	7	7	13	6	11	10.7	4.6
A5	NLP-Id	11	108	12	77	31	29	75	13	81	490	92.7	82.5
	NLP-Di	6	15	9	44	23	19	15	8	29	17	18.5	8.2
	NLP-Tr	10	8	10	11	15	10	12	9	13	11	10.9	1.5
A10	NLP-Id	232	157	21	73	47	77	47	57	40	447	119.8	95.3
	NLP-Di	18	31	10	39	34	21	7	18	25	19	22.2	8.0
	NLP-Tr	11	9	8	11	20	7	7	16	7	11	10.7	3.1
A15	NLP-Id	42	173	47	60	66	300	28	44	58	72	89.0	59.0
	NLP-Di	5	41	8	18	26	52	5	6	23	6	19.0	13.2
	NLP-Tr	5	13	5	6	9	31	4	4	7	5	8.9	5.3

Tab 5.14: Tempo em *wall seconds* - Fixando variáveis - para solucionar-se 50 problemas

	Limites	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	Média	MAD
A0	NLP-Id	5.0	2.1	2.0	4.1	2.4	3.1	7.4	4.3	4.6	72.4	10.7	12.3
	NLP-Di	1.8	1.5	1.7	2.8	1.9	2.3	1.6	1.6	2.7	2.8	2.1	0.5
	NLP-Tr	3.4	2.6	3.0	3.4	2.7	3.9	3.5	2.5	3.5	4.4	3.3	0.5
A2	NLP-Id	4.9	5.2	4.1	80.2	7.3	12.0	11.9	41.2	7.6	128.4	30.3	31.8
	NLP-Di	4.1	2.7	2.8	14.5	7.2	3.6	2.6	6.3	4.7	4.7	5.3	2.4
	NLP-Tr	5.1	3.5	4.1	9.2	4.3	4.5	4.7	5.3	3.8	6.0	5.0	1.1
A5	NLP-Id	4.3	36.9	4.5	26.9	11.1	14.3	24.4	4.7	23.0	142.4	29.3	24.2
	NLP-Di	2.8	4.6	3.5	11.5	6.3	6.1	4.1	3.0	9.6	5.3	5.7	2.2
	NLP-Tr	5.2	4.3	5.2	5.9	6.1	5.0	5.0	4.5	6.3	6.3	5.4	0.6
A10	NLP-Id	80.0	65.1	13.5	29.2	18.8	31.2	19.9	22.9	14.5	145.0	44.0	31.6
	NLP-Di	6.6	12.2	4.4	13.0	10.3	7.9	5.4	5.3	9.6	6.4	8.1	2.5
	NLP-Tr	5.4	5.3	5.4	6.8	11.0	4.7	4.9	5.9	5.4	6.9	6.2	1.2
A15	NLP-Id	14.6	40.5	14.1	17.4	18.2	60.3	10.2	14.8	15.5	19.8	22.5	11.1
	NLP-Di	2.5	11.8	3.0	5.4	7.6	12.5	2.4	2.6	7.9	2.8	5.8	3.3
	NLP-Tr	3.5	5.7	3.7	4.1	4.1	10.3	3.5	3.7	4.0	3.7	4.6	1.3

Capítulo 6

Um Algoritmo de Trajetória Central de Passos Longos para o Problema de Programação Semidefinida

6.1 Introdução

Neste capítulo apresentaremos a extensão de um algoritmo de pontos interiores para programação linear (LP) ao problema de programação semidefinida (SDP). O algoritmo considerado é o algoritmo de trajetória central de passos longos. Programação semidefinida é uma generalização de programação linear, na qual a restrição de que um vetor está no ortante positivo de R^m é substituída pela restrição de que uma matriz $m \times m$ é positiva semidefinida. Como é atualmente sabido, muitos elementos da teoria de algoritmos de pontos interiores para LP são estendidos de uma forma bastante natural a SDP. Tais extensões foram inicialmente sugeridas por Alizadeh [1], e Nesterov e Nemirovskii [68]. Ambos [1] e [68] consideram a extensão de algoritmos “primais” de pontos interiores para SDP; [1] utiliza um algoritmo projetivo de redução potencial, enquanto a teoria geral de [68] permite a análise de uma variedade de algoritmos de redução potencial e de trajetória central. O grande interesse recente em problemas SDP está ligado a aplicações não-triviais de otimização combinatória (veja por exemplo [1]), teoria de controle (veja por exemplo Vandenberghe e Boyd [91]), e outras áreas. Veja [91] para um excelente apanhado sobre a teoria e as aplicações de SDP.

Nos dois últimos anos tem havido muitas publicações sobre métodos de pontos interiores para SDP. A maior parte desta pesquisa considera a extensão de métodos dos tipos primal-dual de trajetória central e preditor-corretor de LP para SDP. Veja por exemplo [51], [55], [61], [76], [90], [95], e trabalhos neles referidos. Nesterov e Todd [69] estenderam a teoria de [68] para algoritmos do tipo primal-dual para problemas formulados sobre “cones auto-escalados”, os quais englobam os problemas SDP. No caso de LP, métodos do tipo primal-dual são em geral reconhecidos como os métodos de pontos interiores mais eficientes na

prática. É natural portanto que se imagine que tais algoritmos estendidos a SDP sejam também bastante eficientes. É importante notar no entanto, que pelo menos para algumas formulações de SDP (por exemplo a formulação utilizada em [91]), métodos primais requerem menos trabalho por iteração do que algoritmos primais-duais, uma vez que o cálculo da direção para a matriz de variáveis duais é evitado.

Neste capítulo analisamos o algoritmo primal de trajetória central de passos longos para o SDP. O algoritmo de trajetória central de passos longos é uma generalização do algoritmo de trajetória central de passos curtos, na qual o algoritmo passa por algumas iterações entre as “visitas” à vizinhança da “trajetória central”. Este algoritmo foi inicialmente proposto por Gonzaga [40] e Roos e Vial [81], e posteriormente estendido para programação quadrática em [9]. Veja em [52] e [22] outras extensões para programação convexa com restrições lineares, e programação convexa geral, respectivamente. O método é bastante parecido com a técnica clássica SUMT de Fiocco e McCormick [32] (veja também [8]), embora [32] anteceda o tipo de análise de complexidade usualmente empregada nos trabalhos atuais.

Nesterov e Nemirovskii [68, 3.2.6] estenderam a metodologia dos algoritmos de trajetória central de passos longos à otimização linear sobre um domínio convexo em R^n , tal que a função barreira satisfaça a condição de auto-concordância. Tal classe de problemas engloba o SDP, entretanto, estamos aqui interessados em estender a elegante análise de [81], a qual baseia-se numa demonstração particularmente simples de convergência quadrática do método de Newton aplicada a função barreira, iniciando num ponto perto da trajetória central. O resultado mais importante de nosso trabalho é a extensão desta análise de convergência quadrática de uma forma bastante natural à SDP. Utilizamos diretamente as técnicas de [81] (veja também [24]) no desenvolvimento da análise de complexidade do algoritmo. Para um problema SDP, no qual restringe-se uma matriz $m \times m$ a ser semidefinida positiva, nosso resultado final é a complexidade de $O(m \ln(t))$ ou $O(\sqrt{m} \ln(t))$ iterações para reduzir o gap primal-dual inicial por um fator t , dependendo de como o parâmetro de penalização da função barreira é reduzido.

Após a elaboração dos resultados aqui apresentados, ficamos cientes de que resultados a eles relacionados foram também obtidos por L. Faybusovich. [28] obtem uma extensão do resultado de convergência quadrática de Roos-Vial para programação semidefinida, e usa o resultado para estabelecer a complexidade do algoritmo de trajetória central de passos curtos. (Resultados similares aos de [28] foram também obtidos por Bininsheng He, independentemente).

Em [29], Faybusovich descreve e analisa o algoritmo de trajetória central de passos longos para programação semidefinida utilizando uma metodologia parecida com a que utilizamos aqui. As principais diferenças entre este trabalho e [29] são:

- as formulações para o problema nos dois trabalhos são duais uma à outra;
- as demonstrações apresentadas aqui são em geral, mais simples do que as apresentadas em [29].

6.2 O Problema de Programação Semidefinida

Consideraremos o seguinte problema de minimizar uma função linear de uma variável $x \in R^n$ sujeito a uma inequação matricial:

$$\begin{array}{ll} \text{PSP} & \min \quad c^T x \\ & \text{s.t.} \quad F(x) \succeq 0, \end{array}$$

onde

$$F(x) \equiv F_0 + \sum_{i=1}^n x_i F_i. \quad (6.1)$$

Os dados do problema são o vetor $c \in R^n$, e $n + 1$ matrizes simétricas $F_0, \dots, F_n \in R^{m \times m}$. A inequação $F(x) \succeq 0$ significa que $F(x)$ é uma matriz semidefinida positiva, i.e., $y^T F(x) y \geq 0$ para todo $y \in R^m$.

Denominamos a inequação $F(x) \succeq 0$ uma inequação matricial linear e o problema PSP um problema de programação semidefinida. Um problema de programação semidefinida é um problema de otimização convexa, uma vez que sua função objetivo e suas restrições são convexas, ou seja, se $F(x) \succeq 0$ e $F(y) \succeq 0$ então, para todo λ tal que $0 \leq \lambda \leq 1$,

$$F(\lambda x + (1 - \lambda)y) = \lambda F(x) + (1 - \lambda)F(y) \succeq 0 \quad (6.2)$$

É sabido que associado com **PSP**, existe o problema dual:

$$\begin{array}{ll} \text{PSD} & \max \quad d(Z) = -Tr(F_0 Z) \\ & \text{s.t.} \quad Tr(F_i Z) = c_i, \quad i = 1, \dots, n \\ & \quad \quad Z \succeq 0, \end{array}$$

onde $Tr(\cdot)$ denota o traço de uma matriz. Neste problema a variável é a matriz $Z = Z^T \in R^{m \times m}$, a qual é sujeita a n restrições de igualdade e à condição de ser semidefinida positiva.

Os resultados duais para programação linear são bastante fortes. Os valores ótimos dos problemas primal e dual são sempre iguais, a não ser no caso em que pelo menos um dos problemas é inviável. No caso da programação semidefinida, no entanto, os resultados duais são bem mais fracos. A principal propriedade do problema de programação semidefinida dual, é o fornecimento de um limite inferior para o valor ótimo do problema primal. Para x viável a **PSP** e Z viável a **PSD**, temos

$$c^T x - d(Z) = \sum_{i=1}^n Tr(Z F_i) x_i + Tr(Z F_0) = Tr(Z F(x)) \geq 0, \quad (6.3)$$

onde usamos o fato de que $Tr(AB) \geq 0$ quando $A = A^T \succeq 0$ e $B = B^T \succeq 0$. Logo, temos:

$$c^T x \geq -Tr(F_0 Z), \quad (6.4)$$

ou seja, o valor da função objetivo do problema primal, em qualquer ponto primal viável x , é maior ou igual ao valor da função objetivo do problema dual em qualquer ponto dual viável Z .

Denominamos a diferença entre os valores das funções objetivos primal e dual nos pontos x e Z respectivamente, como gap primal-dual η associado a x e Z , ou seja

$$\eta := c^T x + Tr(F_0 Z) = Tr(F(x) Z). \quad (6.5)$$

Concluimos então que temos apenas a dualidade fraca entre **PSP** e **PSD**. Sendo v^* e d^* os valores ótimos de **PSP** e de **PSD**, respectivamente, segue-se de (6.3) que $v^* \geq d^*$.

Em geral é possível que os problemas **PSP** e **PSD** tenham um gap primal-dual não nulo; ou seja, ambos os problemas podem ter soluções ótimas, tal que $v^* > d^*$. No entanto, é sabido que se ambos, **PSP** e **PSD**, são estritamente viáveis (i.e., existe x tal que $F(x) \succ 0$, e $Z = Z^T \succ 0$ tal que $Tr(F_i Z) = c_i, i = 1, \dots, n$), então $v^* = d^*$. O seguinte teorema formaliza este resultado. Sua demonstração pode ser encontrada em [68] ou em [80].

Teorema 6.2.1 *Temos $v^* = d^*$ caso pelo menos uma das seguintes condições é satisfeita.*

1. *O problema primal PSP é estritamente viável, isto é, existe x tal que $F(x) \succ 0$.*
2. *O problema dual PSD é estritamente viável, isto é, existe Z tal que $Z = Z^T \succ 0$.*

Se as duas condições forem satisfeitas, então os conjuntos ótimos X_{opt} e Z_{opt} são não vazios, onde:

$$\begin{aligned} X_{opt} &:= \{x | F(x) \succeq 0 \text{ e } c^T x = v^*\} \\ Z_{opt} &:= \{Z | Z \succeq 0, Tr(F_i Z) = c_i, i = 1, \dots, n, \text{ e } -Tr(F_0 Z) = d^*\}. \end{aligned}$$

Assumiremos a partir deste ponto viabilidade primal e dual estritas. Sendo assim, existem soluções x^* e Z^* tais que

$$c^T x^* = -Tr(F_0 Z^*) = d^* = v^*. \quad (6.6)$$

A partir de (6.3), verificamos que $Tr(Z^* F(x^*)) = 0$. Como $F(x^*) \succeq 0$ e $Z^* \succeq 0$, concluimos que $Z^* F(x^*) = 0$, onde usamos o fato de que se A e B são matrizes simétricas semidefinidas positivas, então $Tr(AB)$ implica em $AB = 0$. A condição $Z F(x) = 0$ é conhecida como condição de complementaridade e significa que o espaço imagem das matrizes simétricas Z e $F(x)$ são ortogonais.

As condições de otimalidade para o problema de programação semidefinida PSP, assumindo-se viabilidade primal e dual estritas são dadas por:

x é solução ótima de PSP se e somente se existir Z tal que

$$\begin{aligned} F(x) &\succeq 0 \\ Z &\succeq 0, \quad Tr(F_i Z) = c_i, \quad i = 1, \dots, n \\ Z F(x) &= 0 \end{aligned}$$

Para uma apresentação mais detalhada sobre a teoria de dualidade para programação semidefinida o leitor é referido à [1] ou [91]. Veja também [2], [78], e trabalhos neles referidos, para uma apresentação ainda mais detalhada a respeito da teoria de dualidade para SDP. Quase todas as extensões de algoritmos de pontos interiores a SDP assumem viabilidade primal e dual estritas. Vale notar no entanto, que Freund [35] obteve resultados de complexidade sem esta hipótese.

6.3 A Trajetória Central

Seja

$$\phi(x) \equiv \begin{cases} -\text{ldet}(F(x)) & \text{se } F(x) \succ 0, \\ +\infty & \text{caso contrário,} \end{cases}$$

onde $\text{ldet}(\cdot) = \ln(\det(\cdot))$. A função $\phi(\cdot)$ é a barreira para $\{x \mid F(x) \succ 0\}$, i.e., $\phi(x)$ é finita se e somente se $F(x) \succ 0$, e tende a ser infinita a medida que $F(x)$ aproxima-se de ser singular. Nesterov e Nemirovskii [68] mostraram que quando $F(x) \succ 0$, $\phi(\cdot)$ é analítica, estritamente convexa, e satisfaz a condição de auto-concordância. Consideremos agora a função barreira definida no interior da região viável de **PSP**:

$$f(x, \mu) = \frac{c^T x}{\mu} - \text{ldet}(F(x)).$$

O gradiente e a Hessiana de $f(\cdot, \mu)$ em um dado ponto (x, μ) são portanto, dados por (veja [91])

$$g_i = g_i(x, \mu) = \nabla_x f(x, \mu)_i = \frac{c_i}{\mu} - \text{Tr} \left(F(x)^{-1} F_i \right) = \frac{c_i}{\mu} - \text{Tr} \left(F(x)^{-\frac{1}{2}} F_i F(x)^{-\frac{1}{2}} \right), \quad (6.7)$$

$$\begin{aligned} H_{ij} = H_{ij}(x, \mu) = \nabla_x^2 f(x, \mu)_{ij} &= \text{Tr} \left(F(x)^{-1} F_i F(x)^{-1} F_j \right) \\ &= \text{Tr} \left(\left(F(x)^{-\frac{1}{2}} F_i F(x)^{-\frac{1}{2}} \right) \left(F(x)^{-\frac{1}{2}} F_j F(x)^{-\frac{1}{2}} \right) \right). \end{aligned} \quad (6.8)$$

para $i, j = 1, \dots, n$. Nestas expressões $F(x)^{-\frac{1}{2}}$ significa a raiz quadrada simétrica da matriz $F(x)$. Ao invés de $F(x)^{-\frac{1}{2}}$, expressões análogas usam a fatorização de Cholesky de $F(x)$.

Baseados no fato de que $\phi(\cdot)$ é estritamente convexa e na hipótese de viabilidade estrita, podemos concluir que a função $f(\cdot, \mu)$ tem um único mínimo em um ponto $x(\mu)$ para todo $\mu > 0$. Seja $Z(\mu) = \mu F(x(\mu))^{-1}$. De $\nabla_x f(x(\mu), \mu) = 0$, concluímos que $c_i = \text{Tr}(\mu F(x(\mu))^{-1} F_i)$, e portanto $Z(\mu)$ é viável a **PSD**. Temos ainda que, por (6.3), $c^T x(\mu) - d(Z(\mu)) = \mu \text{Tr}(I) = m\mu$. O conjunto de mínimos $\{x(\mu) \mid \mu > 0\}$ é conhecido como trajetória central para **PSP**, e $\{Z(\mu) \mid \mu > 0\}$ é a trajetória central para **PSD**.

Teorema 6.3.1 *Os valores da função objetivo primal $c^T x(\mu)$ são não crescentes em μ , e os valores da função objetivo dual $d(Z(\mu))$ são não decrescentes em μ .*

Demonstração: Em [32] podemos encontrar o resultado clássico de que $c^T x(\mu)$ é monótona. Incluímos aqui o argumento para que a demonstração fique completa. Seja $\mu' < \mu$. Temos então por definição

$$f(x(\mu), \mu) \leq f(x(\mu'), \mu), \quad (6.9)$$

$$f(x(\mu'), \mu') \leq f(x(\mu), \mu'). \quad (6.10)$$

Em seguida, adicionando (6.9) a (6.10) temos

$$\begin{aligned} \frac{c^T x(\mu)}{\mu} + \frac{c^T x(\mu')}{\mu'} &\leq \frac{c^T x(\mu')}{\mu} + \frac{c^T x(\mu)}{\mu'} \\ \Rightarrow c^T x(\mu) \left(\frac{1}{\mu} - \frac{1}{\mu'} \right) &\leq c^T x(\mu') \left(\frac{1}{\mu} - \frac{1}{\mu'} \right) \\ \Rightarrow c^T x(\mu) &\geq c^T x(\mu'). \end{aligned}$$

O mesmo argumento pode ser aplicado a $d(Z(\mu))$, considerando -se o problema barreira dual:

$$\begin{aligned} \min \quad \bar{f}(Z, \mu) &= \frac{\text{Tr}(F_0 Z)}{\mu} - \text{ldet}(Z) \\ \text{s.t.} \quad \text{Tr}(F_i Z) &= c_i, \quad i = 1, \dots, n \end{aligned} \quad (6.11)$$

Se demonstrarmos que $Z(\mu)$ é uma solução ótima de (6.11), teremos mostrado também que $d(Z(\mu))$ é monótona. Como a função objetivo em (6.11) é convexa em Z , e as restrições são lineares, as condições de otimalidade de KKT são necessárias e suficientes. Pode-se facilmente verificar que as condições de KKT para (6.11) são:

$$\frac{F_0}{\mu} - Z^{-1} = \sum_{i=1}^n \alpha_i F_i \quad (6.12)$$

$$\text{Tr}(F_i Z) = c_i, \quad i = 1, \dots, n, \quad (6.13)$$

para $\alpha_i \in \mathbb{R}$, $i = 1, \dots, n$. Concluimos então que $Z(\mu) = \mu F(x(\mu))^{-1}$ satisfaz (6.13) a partir de $\nabla_x f(x(\mu), \mu) = 0$, como visto acima, e (6.12) é satisfeito pela definição de $F(x(\mu))$. Sendo assim, $Z(\mu)$ resolve (6.11), como requerido. ■

6.4 Propriedades de Pontos Próximos da Trajetória Central

Consideramos três medidas para a distância entre x , um ponto interior a **PSP**, e um ponto na trajetória central $x(\mu)$.

A primeira é uma versão para programação semidefinida da medida introduzida por Roos e Vial [82] para programação linear, estendida para programação quadrática em [9] e [8], e

posteriormente estendida a programação convexa em [52]. Esta medida é definida por

$$\delta(x, \mu) = \min_{\substack{Z \mid \text{Tr}(F_i Z) = c_i \\ i=1, \dots, n}} \left\| \frac{F(x)^{\frac{1}{2}} Z F(x)^{\frac{1}{2}}}{\mu} - I \right\|, \quad (6.14)$$

onde a variável Z é uma matriz simétrica $m \times m$. As outras duas medidas estão associadas com o passo de Newton para $f(\cdot, \mu)$ em x . Utilizamos $p = p(x, \mu)$ para denominar o passo de Newton na variável x , e $p_F = p_F(x, \mu)$ para denominar o passo resultante na matriz $F(\cdot)$. Sendo g e H o gradiente e a Hessiana de $f(\cdot, \mu)$ em um dado ponto (x, μ) , como em (6.7) e em (6.8), temos

$$p = -H^{-1}g, \quad p_F = \sum_{i=1}^n p_i F_i. \quad (6.15)$$

O segundo e terceiro critérios são:

$$\|p\|_H \equiv \sqrt{p^T H p}, \quad \text{and} \quad \|F(x)^{-\frac{1}{2}} p_F F(x)^{-\frac{1}{2}}\|. \quad (6.16)$$

O critério $\|p\|_H$, bastante empregado em [68], é atualmente padrão na análise de algoritmos de pontos interiores. A medida $\|F(x)^{-\frac{1}{2}} p_F F(x)^{-\frac{1}{2}}\|$ pode ser considerada como a norma do passo de Newton, na matriz $F(\cdot)$ depois de um escalonamento simétrico que transforma a matriz $F(x)$ na identidade.

É conhecido o fato de que para programação linear as três medidas acima são todas iguais. Mostraremos agora que de fato esta igualdade permanece verdadeira para o caso mais geral, de programação semidefinida. Iniciaremos a demonstração com as duas medidas em (6.16).

Lema 6.4.1 *Para um dado x e μ , $\|p\|_H = \|F(x)^{-\frac{1}{2}} p_F F(x)^{-\frac{1}{2}}\|$.*

Demonstração: Seja $F = F(x)$. Então

$$\begin{aligned} \|F^{-\frac{1}{2}} p_F F^{-\frac{1}{2}}\|^2 &= \left\| \sum_{i=1}^n p_i F^{-\frac{1}{2}} F_i F^{-\frac{1}{2}} \right\|^2 \\ &= \text{Tr} \left(\left(\sum_{i=1}^n p_i F^{-\frac{1}{2}} F_i F^{-\frac{1}{2}} \right) \left(\sum_{j=1}^n p_j F^{-\frac{1}{2}} F_j F^{-\frac{1}{2}} \right) \right) \\ &= \sum_{i=1}^n \sum_{j=1}^n p_i p_j \text{Tr} \left(F^{-\frac{1}{2}} F_i F^{-\frac{1}{2}} F_j F^{-\frac{1}{2}} \right) \\ &= \sum_{i=1}^n \sum_{j=1}^n p_i H_{ij} p_j \\ &= p^T H p, \end{aligned}$$

como pretendido. ■

A seguir, definimos

$$Z(x, \mu) = \mu \left(F(x)^{-1} - F(x)^{-1} p_F F(x)^{-1} \right).$$

Observamos que

$$\| F(x)^{\frac{1}{2}} Z(x, \mu) F(x)^{\frac{1}{2}} / \mu - I \| = \| F(x)^{-\frac{1}{2}} p_F F(x)^{-\frac{1}{2}} \|. \quad (6.17)$$

Dessa forma, teremos a igualdade das três medidas se pudermos provar que $Z(x, \mu)$ soluciona o problema de minimização que define $\delta(x, \mu)$, em (6.14). Este é de fato o caso, como estabeleceremos nos dois próximos lemas. Por conveniência, para (x, μ) fixo, definimos

$$h(Z) = \| F(x)^{\frac{1}{2}} Z F(x)^{\frac{1}{2}} - \mu I \|^2,$$

e consideramos então o problema equivalente:

$$\begin{aligned} \min \quad & h(Z) \\ \text{s.t.} \quad & \text{Tr}(F_i Z) = c_i, \quad i = 1, \dots, n. \end{aligned} \quad (6.18)$$

Em (6.18) ignoramos o requerimento explícito da simetria de Z , uma vez que a expressão de $h(\cdot)$ garante que a solução de (6.18) seja naturalmente simétrica.

Lema 6.4.2 *As condições de otimalidade de primeira ordem para (6.18) são*

$$F(x) Z F(x) - \mu F(x) = \sum_{i=1}^n \alpha_i F_i, \quad (6.19)$$

$$\text{Tr}(F_i Z) = c_i, \quad i = 1, \dots, n, \quad (6.20)$$

onde $\alpha_i \in R$, $i = 1, \dots, n$.

Demonstração: Seja $F = F(x)$. Temos então

$$\begin{aligned} h(Z) &= \| F^{\frac{1}{2}} Z F^{\frac{1}{2}} - \mu I \|^2 \\ &= \sum_{i=1}^m \sum_{j=1}^m \left(F^{\frac{1}{2}} Z F^{\frac{1}{2}} - \mu I \right)_{ij}^2 \\ &= \sum_{i=1}^m \sum_{j=1}^m \left(F_{i\bullet}^{\frac{1}{2}} Z F_{\bullet j}^{\frac{1}{2}} \right)^2 + \sum_{i=1}^m \left(\mu^2 - 2\mu F_{i\bullet}^{\frac{1}{2}} Z F_{\bullet i}^{\frac{1}{2}} \right), \end{aligned}$$

onde $F_{i\bullet}^{\frac{1}{2}}$ e $F_{\bullet j}^{\frac{1}{2}}$ são a linha i e a coluna j de $F^{\frac{1}{2}}$, respectivamente. Observamos que para quaisquer vetores u e v , $u^T Z v = \sum_{i=1}^m \sum_{j=1}^m u_i Z_{ij} v_j$, logo

$$\frac{\partial u^T Z v}{\partial Z_{kl}} = u_k v_l, \quad k, l = 1, \dots, m. \quad (6.21)$$

Aplicando a regra da cadeia e (6.21), obtemos

$$\begin{aligned}
\frac{\partial h(Z)}{\partial Z_{kl}} &= \sum_{i=1}^m \sum_{j=1}^m 2 \left(F_{i\bullet}^{\frac{1}{2}} Z F_{\bullet j}^{\frac{1}{2}} \right) F_{ik}^{\frac{1}{2}} F_{jl}^{\frac{1}{2}} - 2\mu \sum_{i=1}^m F_{ik}^{\frac{1}{2}} F_{il}^{\frac{1}{2}} \\
&= 2 \sum_{i=1}^m \sum_{j=1}^m \left(F^{\frac{1}{2}} Z F^{\frac{1}{2}} \right)_{ij} F_{ik}^{\frac{1}{2}} F_{jl}^{\frac{1}{2}} - 2\mu \sum_{i=1}^m F_{ik}^{\frac{1}{2}} F_{il}^{\frac{1}{2}} \\
&= 2F_{k\bullet}^{\frac{1}{2}} \left(F^{\frac{1}{2}} Z F^{\frac{1}{2}} - \mu I \right) F_{\bullet l}^{\frac{1}{2}} \\
&= 2(FZF - \mu F)_{kl}.
\end{aligned} \tag{6.22}$$

Temos ainda que $Tr(F_i Z) = \sum_{j=1}^m (F_i)_{j\bullet} Z_{\bullet j}$, e portanto

$$\frac{\partial Tr(F_i Z) - c_i}{\partial Z_{kl}} = (F_i)_{kl}. \tag{6.23}$$

O lema é então uma consequência direta de (6.22), (6.23), e da definição de (6.18). \blacksquare

Lema 6.4.3 Para um dado x e μ ,

$$\delta(x, \mu) = \| F(x)^{-\frac{1}{2}} p_F F(x)^{-\frac{1}{2}} \|. \tag{6.24}$$

Demonstração: Seja $F = F(x)$. Considerando (6.17) é suficiente mostrar que $Z(x, \mu)$ é uma solução ótima de (6.18). Temos ainda que, como $h(\cdot)$ é uma função convexa, é suficiente então mostrar que $Z(x, \mu)$ satisfaz as condições de otimalidade de primeira ordem (6.19) e (6.20). Como $Hp = -g$, temos

$$g_i = - \sum_{j=1}^n H_{ij} p_j = - \sum_{j=1}^n Tr \left(F^{-1} F_j F^{-1} F_i \right) p_j,$$

e portanto, considerando (6.7),

$$\begin{aligned}
\frac{c_i}{\mu} - Tr \left(F^{-1} F_i \right) &= - \sum_{j=1}^n Tr \left(F^{-1} F_j F^{-1} F_i \right) p_j \\
&= - Tr \left(F^{-1} \sum_{j=1}^n (p_j F_j) F^{-1} F_i \right) \\
&= - Tr \left(F^{-1} p_F F^{-1} F_i \right).
\end{aligned}$$

Segue-se que

$$c_i = \mu Tr \left((F^{-1} - F^{-1} p_F F^{-1}) F_i \right) = Tr(Z(x, \mu) F_i), \tag{6.25}$$

logo, $Z(x, \mu)$ satisfaz (6.20). Além disso, $FZ(x, \mu)F - \mu F = -\mu p_F$, logo $Z(x, \mu)$ irá satisfazer (6.19) se houver α_i tal que

$$-\mu p_F = \sum_{i=1}^n \alpha_i F_i.$$

Finalmente, a existência de tais α_i resulta diretamente de (6.15). ■

A seguir consideramos a convergência da medida $\delta(\cdot, \mu)$ quando o método de Newton é aplicado a $f(x, \mu)$, para x numa vizinhança da trajetória central. O teorema seguinte estende a demonstração de Roos e Vial [82] de convergência quadrática de $\delta(\cdot, \mu)$ de programação linear ao caso mais geral, de programação semidefinida.

Teorema 6.4.4 *Suponha que $\delta(x, \mu) < 1$, e defina $x^* = x + p$. Então*

$$F(x^*) \succ 0, \quad e \quad \delta(x^*, \mu) \leq \delta(x, \mu)^2. \quad (6.26)$$

Demonstração: Seja $F = F(x)$. Então

$$\begin{aligned} F(x^*) &= F(x + p) \\ &= F + p_F \\ &= F^{\frac{1}{2}}(I + F^{-\frac{1}{2}}p_F F^{-\frac{1}{2}})F^{\frac{1}{2}}. \end{aligned}$$

No entanto, $|\rho(F^{-\frac{1}{2}}p_F F^{-\frac{1}{2}})| \leq \|F^{-\frac{1}{2}}p_F F^{-\frac{1}{2}}\| < 1$, e conseqüentemente $F(x^*) \succ 0$. A seguir, por definição, temos

$$\begin{aligned} \delta(x^*, \mu)^2 &\leq \left\| \frac{(F + p_F)^{\frac{1}{2}}Z(x, \mu)(F + p_F)^{\frac{1}{2}}}{\mu} - I \right\|^2 \\ &= \sum_{i=1}^m \lambda_i^2 \left(\frac{(F + p_F)^{\frac{1}{2}}Z(x, \mu)(F + p_F)^{\frac{1}{2}}}{\mu} - I \right) \\ &= \sum_{i=1}^m \lambda_i^2 \left(\frac{(F + p_F)Z(x, \mu)}{\mu} - I \right). \end{aligned}$$

Entretanto,

$$\begin{aligned} \frac{(F + p_F)Z(x, \mu)}{\mu} - I &= (F + p_F)F^{-1}(I - p_F F^{-1}) - I = -p_F F^{-1}p_F F^{-1} \\ &= -F^{\frac{1}{2}}(F^{-\frac{1}{2}}p_F F^{-1}p_F F^{-\frac{1}{2}})F^{-\frac{1}{2}}, \end{aligned}$$

logo, nós temos

$$\begin{aligned} \delta(x^*, \mu) &\leq \left(\sum_{i=1}^m \lambda_i^2 (F^{-\frac{1}{2}}p_F F^{-1}p_F F^{-\frac{1}{2}}) \right)^{\frac{1}{2}} \\ &= \left(\sum_{i=1}^m \lambda_i^4 (F^{-\frac{1}{2}}p_F F^{-\frac{1}{2}}) \right)^{\frac{1}{2}} \\ &\leq \sum_{i=1}^m \lambda_i^2 (F^{-\frac{1}{2}}p_F F^{-\frac{1}{2}}) \\ &= \|F^{-\frac{1}{2}}p_F F^{-\frac{1}{2}}\|^2, \end{aligned}$$

e, com base no Lema 6.4.3, $\|F^{-\frac{1}{2}}p_F F^{-\frac{1}{2}}\|^2 = \delta(x, \mu)^2$. ■

O Teorema 6.4.4 pode ser utilizado para gerar limites para possíveis mudanças na função objetivo de **PSP** e na função barreira, entre um ponto aproximadamente centralizado x e um ponto na trajetória central $x(\mu)$. A técnica de demonstração nos dois próximos lemas foi introduzida para programação linear em [24], e é atualmente bastante conhecida.

Lema 6.4.5 *Se $\delta = \delta(x, \mu) < 1$, então*

$$f(x, \mu) - f(x(\mu), \mu) \leq \frac{\delta^2}{1 - \delta^2}$$

Demonstração: A função barreira $f(\cdot, \mu)$ é convexa, logo

$$f(x, \mu) - f(x + p, \mu) \leq -g^T p = \|p\|_H^2 = \delta(x, \mu)^2 \quad (6.27)$$

Seja agora $x^0 := x$ e seja x^0, x^1, x^2, \dots a sequência de pontos obtidos através de repetidos passos de Newton, iniciando-se em x^0 . Utilizando (6.27) e Teorema 6.4.4, obtemos

$$\begin{aligned} f(x, \mu) - f(x(\mu), \mu) &= \sum_{i=0}^{\infty} (f(x^i, \mu) - f(x^{i+1}, \mu)) \\ &\leq \sum_{i=0}^{\infty} \delta^{2^{i+1}} \\ &\leq \sum_{i=0}^{\infty} \delta^{2(i+1)} \\ &= \frac{\delta^2}{1 - \delta^2}, \end{aligned}$$

como pretendido. ■

Lema 6.4.6 *Se $\delta = \delta(x, \mu) < 1$, então*

$$|c^T x - c^T x(\mu)| \leq \frac{\delta(1 + \delta)}{1 - \delta} \mu \sqrt{m}$$

Demonstração: Consideramos $c^T(x + p) = c^T x + c^T p$. Usando (6.7), temos

$$\begin{aligned} c^T p &= \sum_{i=1}^n \mu (g_i + \text{Tr}(F^{-1} F_i)) p_i \\ &= \mu g^T p + \mu \text{Tr}(F^{-1} \sum_{i=1}^n F_i p_i) \\ &= -\mu \delta(x, \mu)^2 + \mu \text{Tr}(F^{-1} p_F) \end{aligned} \quad (6.28)$$

No entanto,

$$\begin{aligned}
| \text{Tr}(F^{-1}p_F) | &= \left| \sum_{i=1}^m \lambda_i(F^{-1}p_F) \right| \leq \sum_{i=1}^m \left| \lambda_i(F^{-\frac{1}{2}}p_FF^{-\frac{1}{2}}) \right| \\
&\leq \sqrt{m} \left(\sum_{i=1}^m \lambda_i^2(F^{-\frac{1}{2}}p_FF^{-\frac{1}{2}}) \right)^{\frac{1}{2}} \\
&= \sqrt{m} \| F^{-\frac{1}{2}}p_FF^{-\frac{1}{2}} \| = \sqrt{m}\delta(x, \mu)
\end{aligned} \tag{6.29}$$

Combinando (6.28) e (6.29), chegamos a

$$| c^T x - c^T(x + p) | \leq \mu\sqrt{m}(\delta(x, \mu) + \delta(x, \mu)^2). \tag{6.30}$$

Como na demonstração do Lema 6.4.5, consideramos agora uma sequência de pontos x^0, x^1, x^2, \dots obtidos através de repetidos passos de Newton iniciados em $x_0 = x$. Utilizando então (6.30) e Teorema 6.4.4 repetidamente, obtemos

$$\begin{aligned}
| c^T x - c^T x(\mu) | &= \sum_{i=0}^{\infty} (c^T x - c^T x^{i+1}) \\
&\leq \sum_{i=0}^{\infty} \mu\sqrt{m}(\delta(x, \mu)^{2^i} + \delta(x, \mu)^{2^{i+1}}) \\
&= \sum_{i=0}^{\infty} \mu\sqrt{m}\delta^{2^i} (1 + \delta) \\
&= \mu\sqrt{m} (1 + \delta) \sum_{i=0}^{\infty} \delta^{2^i} \\
&\leq \mu\sqrt{m} (1 + \delta) \sum_{i=0}^{\infty} \delta^{i+1} \\
&= \frac{\delta(1 + \delta)}{1 - \delta} \mu\sqrt{m},
\end{aligned}$$

como pretendido. ■

O último lema desta seção provê um limite para $c^T x - v^*$ quando $\delta(x, \mu) < 1$. O limite gerado, que utiliza o fato de que $Z(x, \mu)$ é viável para **PSD**, é um pouco melhor do que o limite que seria deduzido do Lema 6.4.6.

Lema 6.4.7 *Se $\delta = \delta(x, \mu) < 1$, então $Z(x, \mu)$ é viável para **PSD**, e*

$$c^T x - v^* \leq c^T x - d(Z(x, \mu)) \leq m\mu(1 + \delta/\sqrt{m}).$$

Demonstração: Considerando (6.25), verificamos que $Z(x, \mu)$ satisfaz as restrições de igualdade de **PSD**. Além disso,

$$Z(x, \mu) = \mu F(x)^{-\frac{1}{2}} (I - F(x)^{-\frac{1}{2}} p_F F(x)^{-\frac{1}{2}}) F(x)^{-\frac{1}{2}},$$

e

$$\rho(F(x)^{-\frac{1}{2}}p_F F(x)^{-\frac{1}{2}}) \leq \|F(x)^{-\frac{1}{2}}p_F F(x)^{-\frac{1}{2}}\| = \delta(x, \mu) < 1,$$

logo, é claro que $Z(x, \mu) \succeq 0$. Sendo assim, $Z(x, \mu)$ é viável para **PSD**, e conseqüentemente

$$c^T x - v^* \leq c^T x - d(Z(x, \mu)).$$

Finalmente, a partir de (6.3) temos

$$\begin{aligned} c^T x - d(Z(x, \mu)) &= \text{Tr}(Z(x, \mu)F(x)) \\ &= \mu \text{Tr}(I - F(x)^{-1}p_F) \\ &= m\mu - \mu \text{Tr}(F(x)^{-1}p_F) \\ &\leq m\mu + \sqrt{m}\mu\delta(x, \mu), \end{aligned} \tag{6.31}$$

onde a última inequação utiliza (6.29). ■

6.5 Complexidade de um Algoritmo de Passos Longos

Nesta seção descrevemos e analisamos o simples algoritmo de trajetória central de passos longos para **PSP**. Iniciamos com a descrição do algoritmo:

Trajетória Central de Passos Longos para **PSP**

Dados:

$\epsilon > 0$ é o parâmetro de precisão;

$0 < \tau < 1$ é o parâmetro de proximidade;

$0 < \theta < 1$ é o parâmetro de redução;

μ_0 é o valor inicial do parâmetro de penalização;

x^0 é um ponto estritamente viável;

Início

$x := x_0;$

$\mu := \mu_0;$

Enquanto $m\mu > \frac{\epsilon}{1+\tau/\sqrt{m}}$ **faça**

Início (passos externos)

$\mu := (1 - \theta)\mu;$

Enquanto $\delta(x, \mu) \geq \tau$ **faça**

Início (passos internos)

$\tilde{\alpha} := \arg \min_{\alpha > 0} \{f(x + \alpha p(x, \mu), \mu) : x + \alpha p(x, \mu) \text{ é um ponto interior}\}$

$x := x + \tilde{\alpha} p(x, \mu)$

Fim (passos internos)

Fim (passos externos)

Fim

O algoritmo é inicializado com um ponto x^0 tal que $F(x^0) \succ 0$, e com o valor inicial do parâmetro de penalização μ . Na análise de complexidade abaixo, assumiremos que $\delta(x^0, \mu_0) \leq \tau$; esta hipótese pode ser facilmente enfraquecida para uma hipótese na magnitude de $f(x^0, \mu_0) - f(x(\mu_0), \mu_0)$. Veja [68, 3.2.3] para um estudo do problema de inicialização para encontrar uma solução (x^0, μ_0) . Nas iterações internas assumimos que o tamanho do passo $\alpha > 0$ provê um decréscimo em $f(\cdot, \mu)$ que é pelo menos igual ao garantido pelo Lema 6.5.3, abaixo. Primeiramente, consideramos o número de iterações externas necessárias à redução do gap primal-dual inicial por um fator $t > 1$.

Lema 6.5.1 *Suponha que $\delta(x^0, \mu_0) \leq \tau$, e seja $t > 1$. Seja $\epsilon_0 = c^T x^0 - d(Z(x^0, \mu_0))$, e $\epsilon = \epsilon_0/t$. Então o algoritmo termina em $K = O(\ln(t)/\theta)$ iterações, e o ponto x^K final satisfaz a $c^T x^K - v^* \leq \epsilon$.*

Demonstração: Por construção temos $\mu_k = (1 - \theta)^k \mu_0$ para cada $k \geq 0$, logo o algoritmo termina após K iterações externas, tal que

$$K \ln(1 - \theta) \leq \ln \left(\frac{\epsilon}{m\mu_0(1 + \tau/\sqrt{m})} \right). \quad (6.32)$$

No entanto, $\ln(1 - \theta) \leq -\theta$, e (6.31) implica em $\epsilon_0 \geq m\mu_0(1 - \tau/\sqrt{m})$, logo (6.32) acontecerá para

$$K \geq -\frac{1}{\theta} \ln \left(\frac{\epsilon(1 - \tau/\sqrt{m})}{\epsilon_0(1 + \tau/\sqrt{m})} \right) = O \left(\frac{\ln(t)}{\theta} \right).$$

Finalmente $c^T x^K - v^* \leq \epsilon$ segue imediatamente do critério de parada das iterações externas do algoritmo, e do Lema 6.4.7. ■

Para completar a análise de complexidade, precisamos de um limite superior no número de iterações internas por iteração externa. Para obter tal limite superior, nós necessitamos primeiramente de um limite inferior no decréscimo de $f(\cdot, \mu)$ que pode ser obtido em cada iteração. Lema 6.5.3 é uma analogia a um já conhecido resultado para programação linear; veja por exemplo [41, Lemma 5.8]. Começamos com a inequação logarítmica.

Lema 6.5.2 *Se $x \in R^m$ e $\|x\| \leq 1$, então*

$$\sum_{i=1}^m \ln(1 + x_i) - \sum_{i=1}^m x_i \geq \|x\| + \ln(1 - \|x\|).$$

Demonstração:

$$\begin{aligned}
\sum_{i=1}^n \ln(1 + x_i) - \sum_{i=1}^n x_i &= \sum_{i=1}^n x_i - \sum_{k=2}^{\infty} \frac{(-1)^k}{k} \sum_{i=1}^n x_i^k - \sum_{i=1}^n x_i \\
&\geq - \sum_{k=2}^{\infty} \frac{1}{k} \sum_{i=1}^n (x_i^2)^{k/2} \\
&\geq - \sum_{k=2}^{\infty} \frac{1}{k} \|x\|^k \\
&= \|x\| + \ln(1 - \|x\|),
\end{aligned}$$

onde a segunda inequação resulta do fato de que

$$\sum_{i=1}^n a_i^p \leq \left(\sum_{i=1}^n a_i \right)^p$$

se $a_i \geq 0$, para $i = 1, \dots, n$ e $p \geq 1$ ■

Lema 6.5.3 *Seja $\delta = \delta(x, \mu)$. Definimos $\bar{\alpha} := 1/(1 + \delta)$. Então $f(x, \mu) - f(x + \bar{\alpha}p, \mu) \geq \delta - \ln(1 + \delta)$.*

Demonstração: Seja λ_i , $i = 1, \dots, m$ os autovalores de $F(x)^{-1}p_F$. Então para $\alpha > 0$, suficientemente pequeno,

$$\begin{aligned}
f(x, \mu) - f(x + \alpha p, \mu) &= \frac{c^T x}{\mu} - \text{ldet}(F(x)) - \frac{c^T(x + \alpha p)}{\mu} + \text{ldet}(F(x + \alpha p)) \\
&= -\alpha \frac{c^T p}{\mu} - \text{ldet}(F(x)) + \text{ldet}(F(x) + \alpha p_F) \\
&= -\alpha \sum_{i=1}^n \frac{c_i p_i}{\mu} + \text{ldet}(I + \alpha F(x)^{-1} p_F) \\
&= -\alpha \sum_{i=1}^n \left(g_i + \text{Tr}(F(x)^{-1} F_i) \right) p_i + \sum_{i=1}^m \ln(1 + \alpha \lambda_i) \\
&= \alpha \delta(x, \mu)^2 - \alpha \text{Tr}(F(x)^{-1} p_F) + \sum_{i=1}^m \ln(1 + \alpha \lambda_i) \\
&= \alpha \delta(x, \mu)^2 - \alpha \sum_{i=1}^m \lambda_i + \sum_{i=1}^m \ln(1 + \alpha \lambda_i).
\end{aligned}$$

No entanto $\sum_{i=1}^m \lambda_i^2 = \|F(x)^{-\frac{1}{2}} p_F F(x)^{-\frac{1}{2}}\|^2 = \delta^2$, logo, para $0 \leq \alpha < 1/\delta$, Lema 6.5.2 implica em

$$f(x, \mu) - f(x + \alpha p, \mu) \geq \alpha \delta^2 + \alpha \delta + \ln(1 - \alpha \delta). \quad (6.33)$$

Ao derivarmos podemos verificar que o lado direito de (6.33) é máximo em $\bar{\alpha} = 1/(1 + \delta)$, e substituindo $\alpha = \bar{\alpha}$ em (6.33) temos a demonstração completa. ■

Baseados no Lema 6.5.3 podemos verificar que se $\delta(x, \mu) > \tau > 0$, então um decréscimo de $\Delta = \tau - \ln(1 + \tau) > 0$ é definitivamente possível em $f(\cdot, \mu)$. Considerando este fato, podemos agora limitar o número de iterações internas necessárias em uma iteração externa do algoritmo. A demonstração do seguinte teorema é análoga a demonstrações de resultados equivalentes para programação linear e quadrática, em [8], [9], e [24].

Teorema 6.5.4 *Seja $\Delta = \tau - \ln(1 + \tau)$, $E_f = \tau^2/(1 - \tau^2)$, $E_c = \tau(1 + \tau)/(1 - \tau)$. Então o número de iterações internas para cada iteração externa é no máximo*

$$N = \frac{1}{\Delta} \left(\frac{\theta}{1 - \theta} (\theta m + 2E_c \sqrt{m}) + E_f \right).$$

Demonstração: Considerando o Lema 6.5.3, sabemos que se N iterações internas são realizadas em k iterações externas, então

$$f(x^{k-1}, \mu_k) - f(x^k, \mu_k) \geq N\Delta, \quad (6.34)$$

onde ambos (x^{k-1}, μ_{k-1}) e (x^k, μ_k) satisfazem o critério de proximidade $\delta(x, \mu) \leq \tau$. Temos ainda que,

$$\begin{aligned} f(x, \mu_k) &= f(x, \mu_{k-1}) + \frac{c^T x}{\mu_k} - \frac{c^T x}{\mu_{k-1}} \\ &= f(x, \mu_{k-1}) + \frac{\theta}{1 - \theta} \frac{c^T x}{\mu_{k-1}}, \end{aligned} \quad (6.35)$$

para qualquer x . Combinando (6.34) e (6.35) obtemos

$$f(x^{k-1}, \mu_{k-1}) - f(x^k, \mu_{k-1}) + \frac{\theta}{1 - \theta} \frac{c^T x^{k-1} - c^T x^k}{\mu_{k-1}} \geq N\Delta. \quad (6.36)$$

No entanto, ambos, x^{k-1} e x^k satisfazem o critério de proximidade, para μ_{k-1} e μ_k respectivamente, logo

$$f(x^{k-1}, \mu_{k-1}) - f(x^k, \mu_{k-1}) \leq E_f, \quad (6.37)$$

pelo Lema 6.4.5. Temos ainda que Lema 6.4.6 obtém

$$\begin{aligned} c^T x^{k-1} - c^T x^k &\leq c^T x(\mu_{k-1}) + E_c \mu_{k-1} \sqrt{m} - c^T x(\mu_k) + E_c \mu_k \sqrt{m} \\ &\leq m(\mu_{k-1} - \mu_k) + E_c \sqrt{m}(\mu_{k-1} + \mu_k) \\ &\leq \mu_{k-1}(\theta m + 2E_c \sqrt{m}), \end{aligned} \quad (6.38)$$

onde a segunda inequação é obtida ao subtrairmos e adicionarmos $d(Z(\mu_{k-1}))$, e usarmos o fato de que $d(Z(\mu_{k-1})) \leq d(Z(\mu_k))$, o qual resulta do Lema 6.3.1. O resultado do teorema é então obtido combinando-se (6.36), (6.37), e (6.38). ■

Combinando-se o Lema 6.5.1 com o Teorema 6.5.4, obtemos imediatamente limites abrangentes para a complexidade do algoritmo de trajetória central de passos longos para **PSP**:

Corolário 6.5.5 *Suponha que $\delta(x^0, \mu_0) \leq \tau$, e seja $t > 1$. Seja $\epsilon_0 = c^T x^0 - d(Z(x^0, \mu_0))$, e $\epsilon = \epsilon_0/t$. Então o número total de passos de Newton necessários ao algoritmo para a obtenção de um ponto x^K tal que $c^T x^K - v^* \leq \epsilon$ é $O(m \ln(t))$ se $\theta = O(1)$, e $O(\sqrt{m} \ln(t))$ se $\theta = O(1/\sqrt{m})$.*

6.6 Viabilidade e Solução Inicial para o Problema de Programação Semidefinida

Na Seção 6.2 vimos como a teoria da dualidade para programação semidefinida nos permite obter um limite inferior para o problema primal PSP. Passamos a assumir neste ponto, que tanto o problema primal PSP quanto o dual PSD são estritamente viáveis. Este é realmente o caso na maioria dos problemas práticos. Nesta seção, no entanto, estabeleceremos um critério para determinar a inviabilidade do problema PSP que pode ser utilizado sempre que não houver certeza da viabilidade do mesmo e descreveremos em seguida um método para encontrar uma solução primal estritamente viável para o caso em que esta não seja conhecida.

6.6.1 Viabilidade

Como na teoria de inequações lineares, a forma padrão de determinar inviabilidade para o problema de programação semidefinida é através do uso de teoremas alternativos. Veja, por exemplo, [80], [13] ou [43].

Par um dado conjunto de inequações lineares $Ax + b \geq 0$, exatamente uma das seguintes afirmações é verdadeira:

1. Existe x tal que $Ax + b \geq 0$.
2. Existe $y \geq 0$ tal que $A^T y = 0$ e $b^T y < 0$.

Podemos interpretar y que satisfaz o segundo conjunto de restrições como um certificado de inviabilidade do primeiro conjunto.

Um teorema similar pode ser aplicado às inequações matriciais lineares $F(X) \succeq 0$ que restringem o problema PSP.

Exatamente uma das seguintes afirmativas é verdadeira:

1. Existe x tal que $F(x) \succeq 0$.
2. Existe $Z = Z^T \succeq 0$ tal que $Tr(F_i Z) = 0, i = 1, \dots, n$, e $Tr(F_0 Z) < 0$.

O teorema resulta de um simples argumento geométrico apresentado em [43]. O conjunto

$$\{x \in R^n | F_0 + \sum_{i=1}^n x_i F_i \succeq 0\}$$

é vazio se e somente se existe um hiperplano separando o conjunto de matrizes semidefinidas positivas $X \succeq 0$ do subespaço $\{X | X = F_0 + \sum_{i=1}^n v_i F_i\}$. Em outras palavras, existe uma matriz não nula $Z = Z^T$ que satisfaz $Tr(ZX) > Tr(Z\tilde{X})$ para todo $X \succeq 0$ e para todo \tilde{X} da forma $\tilde{X} = F_0 + \sum_{i=1}^n v_i F_i$. Esta condição é equivalente às seguintes condições em Z :

$$\begin{aligned} Z &\succeq 0 \\ Tr(F_i Z) &= 0 \quad i = 1, \dots, n \\ Tr(F_0 Z) &\prec 0 \end{aligned}$$

As matrizes Z que certificam a inviabilidade estão relacionadas ao problema dual PSD. Assumindo que o conjunto de soluções duais viáveis não é vazio, toda matriz $Z \succeq 0$ tal que $Tr(F_i Z) = 0$ produz um raio no conjunto de soluções duais viáveis. Se $Tr(F_0 Z) < 0$ então a função objetivo dual é crescente ao longo do raio e o problema dual é ilimitado superiormente. Em outras palavras, o fato do problema dual ser ilimitado implica na inviabilidade do problema primal.

É importante notar ainda que é possível que ambos os problemas primal e dual sejam inviáveis, como se verifica no seguinte exemplo:

$$\begin{aligned} &\text{minimizar} && -x \\ &\text{sujeito à} && \begin{bmatrix} x & 0 \\ 0 & -1 \end{bmatrix} \succeq 0, \end{aligned}$$

com dual

$$\begin{aligned} &\text{maximizar} && z \\ &\text{sujeito à} && \begin{bmatrix} -1 & 0 \\ 0 & z \end{bmatrix} \succeq 0, \end{aligned}$$

6.6.2 Solução Inicial

Veremos agora qual procedimento seguir quando uma solução estritamente viável para o problema PSP não é conhecida.

O método “big- M ” é um método padrão em programação não linear (veja por exemplo, [13] ou [7]). No caso em questão o método consiste em adicionar termos ao problema primal PSP de forma a obter o seguinte problema:

$$\begin{aligned} &\text{minimizar} && c^T x + Mt \\ &\text{sujeito à} && F(x) + tI \succeq 0 \\ &&& t \geq 0. \end{aligned} \tag{6.39}$$

Para obter uma solução estritamente viável para (6.40), escolhamos qualquer x^0 e a seguir escolhamos

$$t^0 > -\min\{\lambda_{\min}(F(x^0)), 0\}.$$

O problema dual à (6.40) é

$$\begin{array}{ll} \text{maximizar} & -Tr(F_0Z) \\ \text{sujeito à} & Tr(F_iZ) = c_i, \quad i = 1, \dots, n \\ & Tr(Z) + z = M \\ & Z \succeq 0, z \geq 0, \end{array}$$

ou, se eliminarmos a variável de folga z ,

$$\begin{array}{ll} \text{maximizar} & -Tr(F_0Z) \\ \text{sujeito à} & Tr(F_iZ) = c_i, \quad i = 1, \dots, n \\ & Tr(Z) \leq M. \end{array}$$

Verificamos portanto que o problema dual PSD foi modificado pela introdução de um limite superior para o traço de Z .

Capítulo 7

Implementação do Algoritmo de Trajetória Central de Passos Longos para o Problema de Programação Semidefinida

7.1 Introdução

Na Seção 6.5 apresentamos o algoritmo de trajetória central de passos longos para programação semidefinida.

Ao implementarmos este algoritmo um aspecto importante a ser considerado é o de como calcular o tamanho do passo de Newton em cada iteração interna. Um outro aspecto também relacionado com a implementação do algoritmo, é o de como evitar que problemas numéricos prejudiquem o comportamento do algoritmo de forma a comprometer a credibilidade da solução por ele gerada.

Estes e outros aspectos relacionados à implementação do algoritmo analisado no capítulo anterior, serão discutidos na próxima seção.

Em seguida, na última seção deste capítulo, apresentaremos os resultados numéricos obtidos com a resolução de dois importantes problemas de programação semidefinida, para os quais utilizamos como dados de entrada, as matrizes de covariância consideradas na Seção 5.4. Desta forma, como veremos mais adiante, a solução destes problemas nos indicam boas opções para a matriz D a ser utilizada na formulação da relaxação não linear para o *Problema da Amostra de Máxima Entropia Restrito*, introduzida na Seção 3.3, quando o terceiro método de escolha para D e γ , apresentado na Seção 3.3.4, é utilizado.

7.2 Aspectos da Implementação

O aspecto mais relevante da implementação dos algoritmos de pontos interiores para programação semidefinida é a instabilidade numérica. Nos artigos sobre programação semidefinida encontrados atualmente na literatura, a análise de complexidade dos algoritmos é um aspecto mais abordado do que os resultados computacionais obtidos através da sua implementação. Nos artigos que contêm resultados computacionais, no entanto, a instabilidade numérica é em geral um assunto abordado. Alizadeh, Haeberly e Overton, por exemplo, mostraram em [3], que para problemas de programação semidefinida não degenerados, o número de condição da matriz envolvida na determinação da direção primal do método primal-dual apresentado, quando avaliada na trajetória central, é limitado superiormente por uma constante vezes $1/\mu$. Conseqüentemente, espera-se que quando $\mu \rightarrow 0$, a imprecisão no cálculo da direção aumente. Alizadeh et al. observaram uma perda significativa de viabilidade perto da solução ótima do problema, devida à instabilidade numérica na implementação de seu método. Todd, Toh e Tütüncü [89] apresentaram expressões alternativas para a determinação das direções primal e dual, que levariam a obtenção de uma maior precisão na solução. No entanto, mesmo simplificando-se tais expressões, a instabilidade continua sendo um empecilho que pode levar a inviabilidade da solução.

Na implementação do método primal analisado no capítulo anterior, a instabilidade numérica foi logo observada. Os principais cálculos onde a instabilidade leva a erros consideráveis são os cálculos da direção de Newton e do tamanho do passo a ser dado em cada iteração. No procedimento que determina o tamanho do passo, o ponto x e a direção de Newton p , nos são fornecidos. Determinamos então inicialmente qual o maior valor de α para o qual

$$F(x + \alpha p) \succeq 0. \quad (7.1)$$

No entanto, temos

$$F(x + \alpha p) = F(x) + \alpha p_F = F(x)^{\frac{1}{2}}(I + \alpha F(x)^{-\frac{1}{2}} p_F F(x)^{-\frac{1}{2}}) F(x)^{\frac{1}{2}}.$$

Portanto, para que (7.1) seja satisfeita basta que

$$1 + \alpha \lambda_{\min} \left(F(x)^{-\frac{1}{2}} p_F F(x)^{-\frac{1}{2}} \right) \geq 0.$$

Conseqüentemente, definindo-se $\lambda_{\min} := \lambda_{\min} \left(F(x)^{-\frac{1}{2}} p_F F(x)^{-\frac{1}{2}} \right)$, concluímos que se $\lambda_{\min} \geq 0$, qualquer valor de α leva a uma solução viável e neste caso tomamos $\alpha = 1$. No caso em que $\lambda_{\min} < 0$, o maior valor de α é dado por $-\frac{1}{\lambda_{\min}}$ e para garantir que o novo ponto seja interior tomamos o passo dado por

$$\sigma \alpha p, \quad (7.2)$$

onde $0 < \sigma < 1$. Em seguida, partindo-se de $\sigma \alpha$ determinamos o tamanho do passo através de busca linear ou simplesmente tomando o primeiro passo em que a função objetivo decresce, num procedimento análogo ao descrito na Seção 5.4 para o Problema da Amostra de Máxima Entropia Restrito.

Através da descrição do procedimento de obtenção do maior passo viável, observamos que erros na determinação de p_F ou de $F(x)^{-\frac{1}{2}}$ podem de fato levar a soluções inviáveis. Para compreender os resultados obtidos com a implementação, passamos então a observar o erro no cálculo da inversa de $F(x)^{\frac{1}{2}}$ (de fato utilizamos a fatorização de Cholesky de $F(x)$ ao invés de sua raiz quadrada) e o erro na inversa da Hessiana de $F(x)$, que obviamente implica no erro da direção de Newton e de p_F . Como esperado os problemas numéricos observados com a implementação estão diretamente relacionados a estes erros. Na próxima seção apresentaremos os resultados obtidos com a implementação.

7.3 Resultados Numéricos

Implementamos o algoritmo de trajetória central de passos longos para programação semidefinida em C . Nosso programa faz uso do pacote de álgebra linear LAPACK 2.0 [6], o qual é escrito em FORTRAN 77. Executamos os programas numa HP 9000/715 com frequência 50MH. Os resultados numéricos obtidos são apresentados nas tabelas abaixo. Expomos os resultados produzidos pela solução de dois problemas classificados como problemas de programação semidefinida.

No primeiro, dada uma matriz simétrica definida positiva C , minimiza-se o traço da matriz diagonal D , tal que $D - C \succeq 0$. Este problema tem aplicação em estatística na análise do fator de mínimo traço (veja [92]); como por exemplo no problema de teste educacional (veja [33, 34]) e em problemas de modelagem empírica (veja [77]). O problema também aplica-se na obtenção da matriz D da relaxação proposta para o Problema da Amostra de Máxima Entropia Restrita, pelo terceiro método proposto na Seção 3.3.4.

A formulação do problema é dada por:

$$\begin{aligned} & \text{minimizar} && \sum_{i=1}^n d_i \\ & \text{sujeito à} && \text{diag}(d) - C \succeq 0 \end{aligned}$$

onde a matriz definida positiva $C = C^T \in R^{n \times n}$ é dada, e as variáveis são as componentes de $d \in R^n$.

No segundo problema resolvido, dada uma matriz simétrica definida positiva C , minimiza-se o maior autovalor da matriz $D - C$, sujeito à $D - C \succeq 0$, onde D é uma matriz diagonal. Muitos pesquisadores têm trabalhado no problema de minimizar o maior autovalor de uma matriz simétrica, o qual pode ser considerado um problema de programação semidefinida. Pesquisas realizadas especificamente sobre este problema podem ser encontradas em Cullum, Donath e Wolfe [21], Goh e Teo [39], Panier [74], Allwright [5], Overton [70, 71], Overton e Womersley [72, 73], Ringertz [79], Fan e Nekooie [27], Fan [26], Hiriart-Urruty e Ye [44], Shapiro e Fan [86] e Pataki [75]. Vale ainda observar que este pode também ser o problema resolvido na obtenção da matriz D da relaxação para o Problema da Amostra de Máxima Entropia, quando o terceiro método da Seção 3.3.4 é utilizado.

O problema pode ser formulado como:

$$\begin{aligned} & \text{minimizar } \lambda I \\ & \text{sujeito à } \lambda I - (\text{diag}(d) - C) \succeq 0 \\ & \text{diag}(d) - C \succeq 0, \end{aligned}$$

onde a matriz definida positiva $C = C^T \in R^{n \times n}$ é dada, e as variáveis são as componentes de $d \in R^n$ e $\lambda \in R$.

Para ilustrar os resultados obtidos, solucionamos os problemas onde C representa as matrizes de covariância de ordem 36, 52 e 63 descritas na Seção 5.4. Inicialmente utilizamos uma precisão $\epsilon = 10^{-6}$ e em seguida, $\epsilon = 10^{-8}$. Utilizamos vários valores para σ definido em (7.2). As tabelas 7.1 a 7.6 retratam o aspecto da instabilidade numérica na solução do problema de traço mínimo. Nestas tabelas #Iti representa o número de iterações internas, ou seja, o número de passos de Newton, realizados a cada iteração externa (Ite) do algoritmo. As tabelas também contêm o maior erro no cálculo da inversa da Hessiana de $F(x)$ e de $F(x)^{\frac{1}{2}}$, em cada iteração externa. Esses erros são medidos respectivamente da seguinte maneira:

$$\begin{aligned} \text{err1} &= \frac{\|Hp + g\|_{\infty}}{\|g\|_{\infty}} \\ \text{err2} &= \|F(x)^{-\frac{1}{2}} F(x) F(x)^{-\frac{1}{2}} - I\|_{\infty} \end{aligned}$$

Finalmente, nas três últimas linhas das tabelas, encontramos o número total de passos de Newton dados durante a execução do algoritmo e os maiores valores de err1 e err2 observados para cada valor de σ . Observamos ainda que '–' representa situações nas quais a iteração externa não pode ser concluída por causa da geração pelo algoritmo de uma solução inviável ou de uma matriz Hessiana H definida negativa. Já o símbolo ∞ representa as situações em que a precisão desejada nunca é alcançada pelo algoritmo.

Pela teoria pode-se esperar que o número de iterações internas decresça a medida que σ aumente, já que sendo σ maior, a região onde a função objetivo é avaliada é maior, e portanto o seu decréscimo também deveria ser maior. No entanto, como podemos observar nas tabelas, este fato nem sempre ocorre. Na verdade, conforme $\mu \rightarrow 0$, observamos um aumento nos erros err1 e err2. Estes erros levam então a um desvio na direção de Newton ou a uma incorreção no cálculo de $\lambda_{\min} \left(F(x)^{-\frac{1}{2}} p_F F(x)^{-\frac{1}{2}} \right)$, que afetam bastante o comportamento do algoritmo, principalmente quando chegamos muito próximo da fronteira, ou seja, quando σ é muito grande. Neste caso o desvio na direção de Newton pode fazer com que a precisão ϵ seja muito dificilmente alcançada e o número de passos de Newton na última iteração do algoritmo fique muito grande (veja as tabelas 7.1 e 7.5). Já a incorreção no cálculo de $\lambda_{\min} \left(F(x)^{-\frac{1}{2}} p_F F(x)^{-\frac{1}{2}} \right)$ pode levar a soluções inviáveis, o que provoca a interrupção do algoritmo.

No gráfico 7.1 ilustramos a influência do erro numérico sobre o número de passos de Newton executados pelo algoritmo para resolver o problema de traço mínimo com $n = 36$ e

Tab 7.1: Resultados numéricos para o problema de Traço Mínimo com $n = 36$ e $\epsilon = 10^{-6}$

Ite		σ							
		0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
1	#Iti	1	1	1	1	1	1	1	1
	err1	5.515e-16	4.003e-16	4.495e-16	4.003e-16	4.003e-16	4.100e-16	4.180e-16	4.003e-16
	err2	8.882e-16	8.882e-16	7.772e-16	6.661e-16	4.441e-16	8.882e-16	4.441e-16	4.441e-16
2	#Iti	5	4	4	3	3	2	2	2
	err1	7.216e-16	4.504e-16	3.622e-16	4.753e-16	4.300e-16	5.815e-16	3.357e-16	6.283e-16
	err2	3.109e-15	1.776e-15	2.959e-15	2.776e-15	4.885e-15	1.998e-15	1.144e-14	3.997e-15
3	#Iti	3	3	2	2	2	2	2	2
	err1	2.685e-16	4.856e-16	2.662e-16	4.463e-16	4.812e-16	1.474e-15	1.020e-15	4.798e-15
	err2	4.996e-14	4.996e-14	3.175e-14	1.713e-14	5.229e-14	1.598e-14	5.707e-14	1.046e-13
4	#Iti	3	2	2	2	2	2	2	2
	err1	2.256e-16	3.11e-16	2.752e-16	2.956e-16	3.615e-16	9.162e-16	1.286e-15	1.545e-14
	err2	4.432e-13	2.031e-13	1.544e-13	3.102e-13	2.807e-13	4.776e-13	6.206e-13	1.493e-11
5	#Iti	2	3	3	3	2	2	2	2
	err1	3.018e-16	2.248e-16	2.785e-16	2.948e-16	4.263e-16	7.577e-16	1.512e-15	2.118e-13
	err2	3.446e-12	8.459e-12	3.282e-12	4.581e-12	2.853e-12	9.337e-12	8.717e-12	9.619e-11
6	#Iti	3	2	2	2	3	2	2	3
	err1	2.359e-16	3.402e-16	4.855e-16	4.255e-16	4.285e-16	1.051e-15	1.231e-15	1.635e-12
	err2	3.318e-11	2.479e-11	1.547e-11	7.304e-11	1.156e-10	5.599e-11	5.557e-11	3.118e-08
7	#Iti	2	3	3	2	2	2	2	2
	err1	2.954e-16	3.506e-16	2.106e-16	2.741e-16	3.568e-16	2.101e-15	1.753e-15	1.176e-11
	err2	2.254e-10	6.545e-10	4.652e-10	2.481e-10	4.748e-10	1.629e-10	4.442e-10	9.116e-07
8	#Iti	3	2	2	3	2	2	2	3
	err1	3.015e-16	2.431e-16	2.504e-16	2.347e-16	4.088e-16	1.636e-15	1.556e-15	1.261e-10
	err2	4.385e-09	3.800e-09	3.050e-09	1.003e-08	2.215e-09	5.459e-09	4.859e-09	0.000397
9	#Iti	1	1	1	1	1	1	1	∞
	err1	2.396e-16	2.549e-16	2.101e-16	2.097e-16	2.620e-16	8.346e-16	1.013e-15	1.873e-09
	err2	8.914e-09	6.201e-09	5.743e-09	1.410e-08	1.586e-08	9.912e-09	1.314e-08	0.031846
	#Tot Iti	23	21	20	19	18	16	16	∞
	Max err1	7.216e-16	4.856e-16	4.855e-16	4.753e-16	4.812e-16	2.101e-15	1.753e-15	1.873e-09
	Max err2	8.914e-09	6.201e-09	5.743e-09	1.410e-08	1.586e-08	9.912e-09	1.314e-08	0.031846

Tab 7.2: Resultados numéricos para o problema de Traço Mínimo com $n = 36$ e $\epsilon = 10^{-8}$

Ite		σ							
		0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
1	#Iti	1	1	1	1	1	1	1	1
	err1	5.515e-16	4.003e-16	4.495e-16	4.003e-16	4.003e-16	4.100e-16	4.180e-16	4.003e-16
	err2	8.882e-16	8.882e-16	7.772e-16	6.661e-16	4.441e-16	8.882e-16	4.441e-16	4.441e-16
2	#Iti	5	4	4	3	3	2	2	2
	err1	7.216e-16	4.504e-16	3.622e-16	4.753e-16	4.300e-16	5.815e-16	3.357e-16	6.283e-16
	err2	3.109e-15	1.776e-15	2.959e-15	2.776e-15	4.885e-15	1.998e-15	1.144e-14	3.997e-15
3	#Iti	3	3	2	2	2	2	2	2
	err1	2.685e-16	4.856e-16	2.662e-16	4.463e-16	4.812e-16	1.474e-15	1.020e-15	4.798e-15
	err2	4.996e-14	4.996e-14	3.175e-14	1.713e-14	5.229e-14	1.598e-14	5.707e-14	1.046e-13
4	#Iti	3	2	2	2	2	2	2	2
	err1	2.256e-16	3.111e-16	2.752e-16	2.956e-16	3.615e-16	9.162e-16	1.286e-15	1.545e-14
	err2	4.432e-13	2.031e-13	1.544e-13	3.102e-13	2.807e-13	4.776e-13	6.206e-13	1.493e-11
5	#Iti	2	3	3	3	2	2	2	2
	err1	3.018e-16	2.248e-16	2.785e-16	2.948e-16	4.263e-16	7.577e-16	1.512e-15	2.118e-13
	err2	3.446e-12	8.859e-12	3.282e-12	4.581e-12	2.853e-12	9.337e-12	8.717e-12	9.619e-11
6	#Iti	3	2	2	2	3	2	2	3
	err1	2.359e-16	3.402e-16	4.855e-16	4.255e-16	4.285e-16	1.051e-15	1.231e-15	1.635e-12
	err2	3.318e-11	2.479e-11	1.547e-11	7.304e-11	1.156e-10	5.599e-11	5.557e-11	3.118e-08
7	#Iti	2	3	3	2	2	2	2	2
	err1	2.954e-16	3.506e-16	2.106e-16	2.741e-16	3.568e-16	2.101e-15	1.753e-15	1.176e-11
	err2	2.254e-10	6.545e-10	4.652e-10	2.481e-10	4.748e-10	1.629e-10	4.442e-10	9.116e-07
8	#Iti	3	2	2	3	2	2	2	3
	err1	3.015e-16	2.431e-16	2.504e-16	3.347e-16	4.088e-16	1.636e-15	1.556e-15	1.261e-10
	err2	4.385e-09	3.800e-09	3.050e-09	1.003e-08	2.215e-09	5.459e-09	4.859e-09	0.000397
9	#Iti	2	3	2	2	2	2	2	3
	err1	2.846e-16	2.549e-16	2.766e-16	2.375e-16	6.064e-16	2.315e-15	1.013e-15	1.490e-09
	err2	2.507e-08	6.938e-08	3.250e-08	2.699e-08	2.080e-08	3.806e-08	9.703e-08	0.012532
10	#Iti	3	2	3	2	2	2	2	-
	err1	2.785e-16	2.568e-16	3.599e-16	3.030e-16	3.483e-16	2.692e-15	9.513e-16	-
	err2	6.287e-07	3.256e-07	6.263e-07	3.356e-07	2.973e-07	1.881e-07	5.405e-07	-
11	#Iti	1	1	1	1	1	1	1	-
	err1	2.241e-16	2.633e-16	2.430e-16	1.939e-16	2.708e-16	2.342e-15	3.070e-15	-
	err2	1.342e-06	5.100e-07	8.787e-07	1.014e-06	1.177e-06	1.519e-06	2.891e-06	-
	#Tot Iti	28	26	25	23	22	20	20	-
	Max err1	7.216e-16	4.856e-16	4.855e-16	4.753e-16	6.064e-16	2.692e-15	3.070e-15	1.490e-09
	Max err2	1.342e-06	5.100e-07	8.787e-07	1.014e-06	1.177e-06	1.519e-06	2.891e-06	0.012532

Tab 7.3: Resultados numéricos para o problema de Traço Mínimo com $n = 52$ e $\epsilon = 10^{-6}$

Ite		σ							
		0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
1	#Iti	1	1	1	1	1	1	1	1
	err1	4.289e-16	4.091e-16	4.091e-16	4.731e-16	5.188e-16	2.286e-16	4.091e-16	4.091e-16
	err2	6.661e-16	7.772e-16	9.992e-16	8.882e-16	1.110e-15	8.882e-16	7.772e-16	5.551e-16
2	#Iti	6	5	5	4	4	3	3	2
	err1	4.813e-16	5.805e-16	4.787e-16	9.608e-16	1.105e-15	5.783e-16	1.050e-15	1.854e-15
	err2	5.107e-15	1.421e-14	1.932e-14	4.663e-15	3.020e-14	2.909e-14	2.255e-13	9.104e-15
3	#Iti	3	3	2	2	1	2	1	1
	err1	8.165e-16	1.131e-15	3.205e-16	1.526e-15	4.407e-15	6.716e-15	2.284e-15	8.842e-15
	err2	4.330e-14	2.165e-13	2.881e-13	2.085e-13	3.293e-13	8.003e-13	3.784e-13	8.686e-13
4	#Iti	2	2	2	2	2	1	2	2
	err1	3.517e-16	3.760e-16	2.898e-16	4.711e-16	7.861e-16	6.432e-15	1.656e-14	5.930e-14
	err2	4.316e-13	2.594e-12	2.285e-12	1.562e-12	6.704e-12	1.689e-12	1.604e-11	1.115e-11
5	#Iti	3	2	2	1	1	1	1	2
	err1	3.232e-16	2.829e-16	2.737e-16	4.009e-16	4.553e-16	1.896e-14	2.486e-13	3.947e-13
	err2	5.789e-12	1.794e-12	1.411e-11	3.990e-12	1.261e-11	4.620e-12	1.756e-11	2.723e-08
6	#Iti	2	2	2	2	1	1	3	2
	err1	2.815e-16	4.899e-16	2.672e-16	3.434e-16	4.050e-16	7.151e-15	2.708e-13	3.444e-12
	err2	1.040e-10	7.885e-11	7.496e-10	4.434e-10	2.180e-11	2.289e-11	1.344e-08	3.130e-07
7	#Iti	3	3	1	2	2	2	3	2
	err1	2.882e-16	3.773e-16	4.545e-16	2.722e-16	5.715e-16	2.420e-15	1.748e-12	2.833e-11
	err2	1.177e-09	2.720e-10	2.337e-09	8.925e-10	2.313e-09	2.735e-09	1.371e-07	3.035e-05
8	#Iti	3	2	2	1	1	1	3	2
	err1	3.479e-16	4.437e-16	3.585e-16	4.651e-16	2.434e-16	4.259e-15	2.371e-11	4.634e-10
	err2	9.986e-09	1.034e-08	4.374e-09	4.867e-09	5.815e-09	7.989e-09	1.601e-05	0.013820
9	#Iti	2	2	2	2	2	1	3	-
	err1	3.399e-16	3.101e-16	2.712e-16	3.624e-16	2.514e-16	3.606e-15	1.563e-10	-
	err2	1.779e-07	1.604e-07	1.883e-07	1.323e-07	4.084e-07	3.502e-07	0.003843	-
10	#Iti	1	1	1	1	1	1	-	-
	err1	2.780e-16	2.593e-16	3.326e-16	2.633e-16	4.394e-16	7.465e-16	-	-
	err2	5.106e-07	5.548e-07	2.636e-07	6.525e-07	2.624e-06	1.759e-06	-	-
	#Tot Iti	26	23	20	18	16	14	-	
	Max err1	8.165e-16	1.131e-15	4.787e-16	1.525e-15	4.407e-15	1.896e-14	1.563e-10	4.634e-10
	Max err2	5.106e-07	5.548e-07	2.636e-07	6.525e-07	2.624e-06	1.759e-06	0.003843	0.013820

Tab 7.4: Resultados numéricos para o problema de Traço Mínimo com $n = 52$ e $\epsilon = 10^{-8}$

Ite		σ							
		0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
1	#Iti	1	1	1	1	1	1	1	1
	err1	4.289e-16	4.091e-16	4.091e-16	4.731e-16	5.188e-16	4.286e-16	4.091e-16	4.091e-16
	err2	6.661e-16	7.772e-16	9.992e-16	8.882e-16	1.110e-15	8.882e-16	7.772e-16	5.551e-16
2	#Iti	6	5	5	4	4	3	3	2
	err1	4.813e-16	5.805e-16	4.787e-16	9.608e-16	1.105e-15	5.783e-16	1.050e-15	1.854e-15
	err2	5.107e-15	1.421e-14	1.932e-14	4.663e-15	3.020e-14	2.909e-14	2.255e-13	9.104e-15
3	#Iti	3	3	2	2	1	2	1	1
	err1	8.165e-16	1.131e-15	3.205e-16	1.525e-15	4.407e-15	6.716e-15	2.584e-15	8.452e-15
	err2	4.330e-14	2.165e-13	2.881e-13	2.085e-13	3.293e-13	8.003e-13	3.784e-13	8.686e-13
4	#Iti	2	2	2	2	2	1	2	2
	err1	3.517e-16	3.760e-16	2.898e-16	4.711e-16	7.861e-16	6.432e-15	1.656e-14	5.930e-14
	err2	4.317e-13	2.594e-12	2.285e-12	1.562e-12	6.704e-12	1.189e-12	1.604e-11	1.115e-11
5	#Iti	3	2	2	1	1	1	1	2
	err1	3.232e-16	2.829e-16	2.737e-16	4.009e-16	4.553e-16	1.896e-14	2.486e-13	3.947e-13
	err2	5.789e-12	1.794e-12	1.411e-11	3.990e-12	1.261e-11	4.620e-12	1.756e-11	2.723e-08
6	#Iti	2	2	2	2	1	1	3	2
	err1	2.815e-16	4.899e-16	2.672e-16	3.434e-16	4.050e-16	7.151e-15	1.748e-12	3.444e-12
	err2	1.040e-10	7.884e-11	7.496e-10	4.334e-10	2.180e-11	2.289e-11	1.371e-07	3.130e-07
7	#Iti	3	3	1	2	2	2	3	2
	err1	2.882e-16	3.773e-16	4.545e-16	2.722e-16	5.715e-16	2.420e-15	2.371e-11	2.833e-11
	err2	1.177e-09	2.720e-10	2.337e-09	8.925e-10	2.313e-09	2.735e-09	1.601e-05	3.035e-05
8	#Iti	3	2	2	1	1	1	3	2
	err1	3.479e-16	4.437e-16	3.585e-16	4.651e-16	2.434e-16	4.259e-15	1.563e-10	4.634e-10
	err2	9.986e-09	1.034e-08	4.374e-09	4.867e-09	5.815e-09	7.989e-09	0.003843	0.013820
9	#Iti	2	2	2	2	2	1	3	-
	err1	3.399e-16	3.101e-16	2.712e-16	3.624e-16	2.514e-16	3.606e-15	2.813e-09	-
	err2	1.779e-07	1.604e-07	1.883e-07	1.33e-07	4.084e-07	3.502e-07	0.294948	-
10	#Iti	3	2	2	2	1	1	-	-
	err1	2.780e-16	2.837e-16	3.326e-16	2.650e-16	4.394e-16	7.465e-16	-	-
	err2	1.102e-06	1.622e-06	1.543e-06	4.309e-06	2.264e-06	1.759e-06	-	-
11	#Iti	2	2	2	1	2	2	-	-
	err1	3.746e-16	2.823e-16	5.839e-16	2.729e-16	2.608e-16	3.735e-16	-	-
	err2	1.082e-05	7.360e-06	1.525e-05	7.121e-06	2.098e-05	4.427e-06	-	-
12	#Iti	1	1	1	1	1	1	-	-
	err1	2.628e-16	4.933e-16	2.827e-16	2.912e-16	2.850e-16	3.200e-16	-	-
	err2	1.368e-05	2.663e-05	0.000107	3.681e-05	0.000292	0.000122	-	-
	#Tot Iti	31	27	24	21	19	17	-	-
	Max err1	8.165e-16	1.131e-15	4.787e-16	1.525e-15	4.407e-15	1.896e-14	2.813e-09	4.634e-10
	Max err2	1.368e-05	2.663e-05	0.000107	3.681e-05	0.000292	0.000122	0.294948	0.013820

Tab 7.5: Resultados numéricos para o problema de Traço Mínimo com $n = 63$ e $\epsilon = 10^{-6}$

Ite		σ							
		0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
1	#Iti	1	1	1	1	1	1	1	1
	err1	4.899e-16	4.899e-16	6.043e-16	4.899e-16	4.987e-16	4.899e-16	4.899e-16	4.899e-16
	err2	8.882e-16	1.110e-15	1.332e-15	1.110e-15	1.221e-15	1.332e-15	6.661e-16	6.661e-16
2	#Iti	7	6	5	5	4	4	3	2
	err1	3.042e-15	3.712e-15	5.998e-15	3.173e-15	1.495e-15	3.582e-15	5.891e-15	2.957e-15
	err2	9.992e-15	.887e-13	2.853e-14	3.886e-14	1.776e-14	1.221e-14	5.973e-14	2.987e-14
3	#Iti	3	3	3	2	3	2	2	2
	err1	2.049e-15	3.218e-14	8.762e-15	5.809e-15	9.581e-15	2.305e-14	1.616e-14	4.599e-14
	err2	8.105e-14	2.887e-13	2.494e-13	2.649e-13	1.033e-12	4.117e-13	1.465e-12	2.280e-11
4	#Iti	3	2	2	2	2	2	3	2
	err1	5.115e-15	3.567e-15	1.798e-14	1.776e-14	7.797e-14	4.971e-14	4.352e-13	1.937e-13
	err2	2.831e-12	1.583e-12	3.352e-12	3.725e-13	4.464e-12	3.109e-12	2.004e-10	6.296e-10
5	#Iti	2	3	2	2	2	4	3	2
	err1	6.523e-16	1.818e-15	4.380e-15	8.394e-15	9.145e-12	2.677e-13	9.948e-13	2.206e-12
	err2	1.023e-11	4.715e-12	1.608e-11	5.459e-11	6.767e-11	5.699e-10	1.026e-08	1.115e-07
6	#Iti	3	2	2	1	3	4	3	2
	err1	5.650e-16	3.635e-16	2.713e-15	2.213e-15	1.802e-12	2.512e-12	1.132e-11	2.275e-11
	err2	1.017e-10	5.785e-11	2.510e-10	8.278e-11	1.840e-09	5.380e-08	9.225e-07	1.266e-05
7	#Iti	2	2	2	2	5	4	3	2
	err1	6.186e-16	6.556e-16	7.072e-16	5.091e-16	3.779e-12	2.123e-11	9.603e-11	1.602e-10
	err2	1.739e-09	1.782e-09	6.324e-10	4.310e-09	2.383e-08	7.335e-06	0.000281	0.001420
8	#Iti	3	2	2	2	5	4	3	3
	err1	6.703e-16	7.953e-16	6.176e-16	5.840e-16	2.744e-11	1.944e-10	1.104e-09	1.906e-09
	err2	2.158e-08	1.372e-08	1.288e-08	3.812e-08	5.941e-06	0.000522	0.005804	0.01468
9	#Iti	2	2	2	1	5	4	4	-
	err1	5.739e-16	5.523e-16	5.131e-16	5.948e-16	3.670e-10	2.436e-09	9.141e-09	-
	err2	1.237e-07	1.312e-07	3.605e-08	4.400e-08	0.000180	0.031290	1.455740	-
10	#Iti	1	1	1	1	∞	-	-	-
	err1	5.877e-16	7.205e-16	5.520e-16	5.247e-16	2.860e-09	-	-	-
	err2	1.078e-06	1.516e-08	1.714e-06	9.284e-07	0.254820	-	-	-
	#Tot Iti	27	24	22	19	∞	-	-	-
	Max err1	3.042e-15	3.218e-14	1.798e-14	1.776e-14	2.860e-09	2.436e-09	9.141e-09	1.906e-09
	Max err2	1.078e-06	1.312e-07	1.714e-06	9.284e-07	0.254820	0.118604	1.455740	0.014682

Tab 7.6: Resultados numéricos para o problema de Traço Mínimo com $n = 63$ e $\epsilon = 10^{-8}$

Ite		σ							
		0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
1	#Iti	1	1	1	1	1	1	1	1
	err1	4.899e-16	4.899e-16	6.043e-16	4.899e-16	4.987e-16	4.899e-16	4.899e-16	4.899e-16
	err2	8.882e-16	1.110e-15	1.332e-15	1.110e-15	1.221e-15	1.332e-15	6.661e-16	6.661e-16
2	#Iti	7	6	5	5	4	4	3	2
	err1	3.042e-15	3.712e-15	6.000e-15	3.17e-15	1.495e-15	3.582e-15	5.891e-15	2.957e-15
	err2	9.992e-15	8.549e-15	2.853e-14	3.886e-14	1.776e-14	1.221e-14	5.973e-14	2.987e-14
3	#Iti	3	3	3	2	3	2	2	2
	err1	2.049e-15	3.218e-14	8.761e-15	5.809e-15	9.581e-15	2.305e-14	1.66e-14	4.599e-14
	err2	8.105e-14	2.887e-13	2.494e-13	2.649e-13	1.033e-12	4.117e-13	1.465e-12	2.280e-11
4	#Iti	3	2	2	2	2	2	3	2
	err1	5.115e-16	3.567e-15	1.798e-14	1.776e-14	7.797e-14	4.971e-14	4.352e-13	1.937e-13
	err2	2.831e-12	1.583e-12	3.352e-12	3.725e-13	4.464e-12	3.109e-12	2.004e-10	6.296e-10
5	#Iti	2	3	2	2	2	4	3	2
	err1	6.523e-16	1.818e-5	4.380e-15	8.394e-15	9.145e-12	2.677e-13	1.133e-11	2.206e-12
	err2	1.023e-11	4.715e-12	1.608e-11	5.459e-11	6.766e-11	5.699e-10	9.225e-07	1.115e-07
6	#Iti	3	2	2	1	3	4	3	2
	err1	5.650e-16	3.635e-16	2.713e-15	2.213e-15	1.802e-12	2.512e-12	9.603e-11	2.275e-11
	err2	1.017e-10	5.785e-11	2.510e-10	8.278e-11	1.840e-09	5.380e-08	0.000281	1.266e-05
7	#Iti	2	2	2	2	5	4	3	2
	err1	6.186e-16	6.556e-16	7.072e-16	5.091e-16	3.779e-12	2.123e-11	9.603e-11	1.906e-09
	err2	1.739e-09	1.782e-09	6.324e-10	4.310e-09	2.383e-08	7.335e-06	0.000281	0.014682
8	#Iti	3	2	2	2	5	4	3	-
	err1	6.703e-16	7.953e-16	6.176e-16	5.840e-16	2.744e-11	1.944e-10	1.104e-09	-
	err2	2.158e-08	1.372e-08	1.288e-08	3.812e-08	5.941e-06	0.000522	0.005804	-
9	#Iti	2	2	2	1	5	4	4	-
	err1	5.739e-16	5.523e-16	5.131e-16	5.948e-16	3.670e-10	2.436e-09	9.141e-09	-
	err2	1.237e-07	1.312e-07	3.605e-08	4.400e-08	0.000180	0.0312899	1.455740	-
10	#Iti	3	2	2	2	5	-	-	-
	err1	5.877e-16	7.205e-16	5.520e-16	6.772e-16	2.095e-09	-	-	-
	err2	1.279e-06	3.952e-07	1.714e-06	1.245e-06	0.037665	-	-	-
11	#Iti	2	3	2	2	-	-	-	-
	err1	5.972e-16	6.254e-16	5.540e-16	6.351e-16	-	-	-	-
	err2	1.030e-05	2.491e-05	2.023e-05	5.157e-06	-	-	-	-
12	#Iti	1	1	1	1	-	-	-	-
	err1	5.308e-16	5.320e-16	5.943e-16	5.448e-16	-	-	-	-
	err2	1.103e-05	6.475e-05	9.836e-05	8.504e-05	-	-	-	-
	#Tot Iti	32	29	26	23	-	-	-	-
	Max err1	5.115e-15	3.218e-14	1.799e-14	1.776e-14	2.094e-09	2.436e-09	9.141e-09	1.906e-09
	Max err2	1.103e-05	6.475e-05	9.836e-05	8.504e-05	0.037665	0.031290	1.455740	0.014682

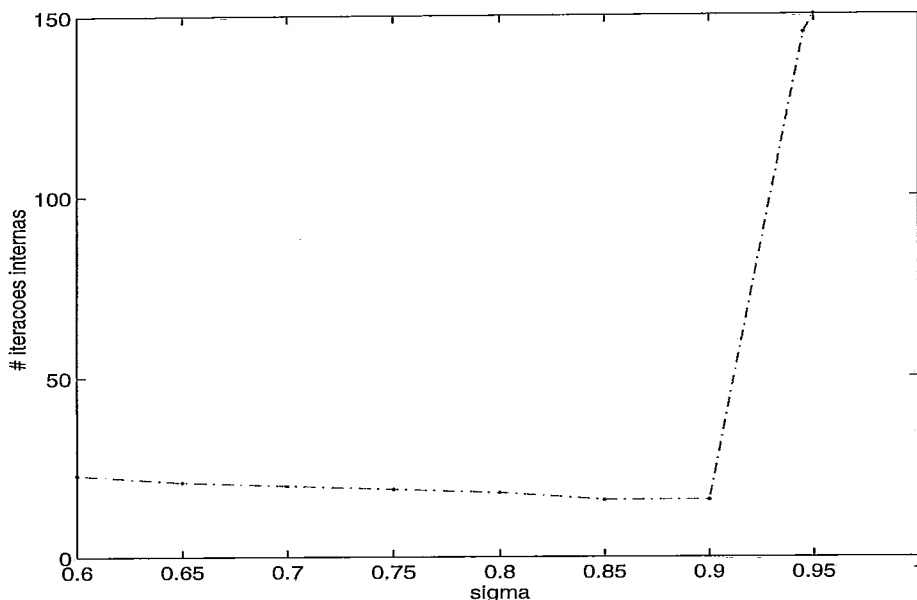


Fig 7.1: Aplicação do algoritmo de passos longos ao problema de Traço Mínimo com $n = 36$ e $\epsilon = 10^{-6}$ para diferentes valores de σ

Tab 7.7: Resultados finais para o problema de Traço Mínimo com $\epsilon = 10^{-8}$

	n		
	36	52	63
#Iti	20	17	23
Tempo	2.55	6.35	14.07
σ	0.85	0.85	0.75

$\epsilon = 10^{-6}$. É interessante notar através do gráfico como enquanto o erro pode ser desprezado, o aumento de σ leva a uma diminuição no número de passos de Newton. Já quando o erro passa a ser considerável o número de passos é infinito, ou seja, a solução com a precisão desejada nunca é alcançada pelo algoritmo.

Finalmente, concluindo esta seção, vale observar que apesar da instabilidade numérica abordada acima, todos os exemplos de problemas de traço mínimo e de máximo autovalor considerados, foram solucionados pelo algoritmo com a precisão desejada $\epsilon = 10^{-8}$. Nas tabelas 7.7e 7.8 apresentamos o número de iterações e o tempo em *wall seconds* necessários para resolver os problemas, nos quais C representa as três matrizes de covariância abordadas na Seção 5.4, de ordem 36, 52 e 63. Os resultados apresentados foram obtidos com o valor de σ que levou ao melhor funcionamento do algoritmo, na resolução de cada problema teste. Este valor é especificado na terceira linha das tabelas.

Tab 7.8: Resultados finais para o problema de Máximo Autovalor com $\epsilon = 10^{-8}$

	n		
	36	52	63
#Iti	27	21	21
Tempo	7.30	16.22	28.62
σ	0.60	0.85	0.85

Capítulo 8

Conclusão

O *Problema da Amostra de Máxima Entropia Restrito* é um problema com vasta aplicação prática. O primeiro método a buscar a solução ótima deste problema foi proposto por Lee em 1995 [53] e utiliza um algoritmo de *branch-and-bound*. O sucesso na obtenção da solução ótima pelo método de Lee está bastante relacionado com a disponibilidade de bons limites superiores e inferiores para o problema. Devido à complexidade do procedimento de geração de limites superiores, o método de Lee se mostra ineficaz a medida que o número de restrições do problema cresce.

A primeira etapa do trabalho apresentado nesta tese visa o desenvolvimento de um método de obtenção de limites superiores para o *Problema da Amostra de Máxima Entropia* que incorpore naturalmente a imposição de restrições ao problema, muito comuns em situações reais.

A relaxação não linear apresentada na Seção 3.3 é uma relaxação bastante original para o problema em questão. Os resultados obtidos até então com o uso do algoritmo de *branch-and-bound* completo e com limites superiores obtidos através da relaxação, são bastante encorajadores.

De uma forma geral, os limites gerados pela relaxação parecem estar bem mais próximos do valor ótimo dos problemas do que os limites utilizados por Lee, baseados nos autovalores da matriz de covariância.

O único ponto negativo na relaxação é que para problemas irrestritos, limites baseados na relaxação consomem mais tempo para serem calculados do que limites baseados em autovalores, para subproblemas do mesmo tamanho. Entretanto, este aumento de tempo é quase sempre compensado pela diminuição no número de subproblemas gerados quando a relaxação é utilizada, devido à melhoria na qualidade dos limites. Além disto, o tempo para calcular os limites baseados nos autovalores crescem bem mais rapidamente do que o tempo para calcular os limites baseados na relaxação, conforme o número de restrições aumenta.

Para solucionar cinquenta problemas reais considerados no Capítulo 5, o uso da relaxação

com a matriz D escolhida pela opção “Diagonal” (NLP-Di), reduz o número total de limites calculados por um fator de aproximadamente 15, e o tempo total requerido por um fator de aproximadamente 34, comparando com o uso dos limites baseados em autovalores. Cabe observar que esta redução foi obtida sem o emprego de nenhum critério para fixar variáveis na solução. Na prática, o procedimento descrito na Seção 3.3.6 para fixar variáveis nos valores zero e um, baseado na solução dual da relaxação NLP, mostrou-se surpreendentemente eficaz. Empregando-se o procedimento na solução ótima dos cinquenta problemas, calculando-se novamente o número total de limites calculados para resolver todos os problemas com o uso de NLP-Di e comparando-se com o método dos autovalores, o fator de redução cresce de 15 para 71. Se considerarmos o tempo total de resolução, o fator de redução cresce de 34 para 148.

Atualmente estamos resolvendo problemas restritos com o uso de uma implementação paralela. A utilização de uma máquina paralela Convex nos permitiu solucionar problemas bem maiores, com por exemplo $n=124$, para os quais a solução ótima nunca havia sido obtida. Pretendemos agora experimentar a interrupção do cálculo do limite superior sempre que um determinado valor for alcançado. Este valor é na verdade um limite inferior para o problema. Desta forma, não haverá desperdício com o cálculo de limites superiores quando a poda da árvore enumerativa não é possível. Apesar de atraente, este procedimento tem algumas contra-indicações. A primeira é dada pela existência de estratégias úteis de geração de subproblemas baseadas nos limites superiores. Uma segunda contra-indicação é dada pela dependência do procedimento de geração de limites inferiores, nos limites superiores. Estas duas contra-indicações sugerem a busca do melhor limite superior possível, mesmo que estes não levem a poda da árvore enumerativa do *branch-and-bound*.

Como contribuição a estudos subsequentes, o trabalho desenvolvido fornece um método original e com bom desempenho nos testes realizados, para gerar limites para problemas em que a maximização ou a minimização de determinantes de submatrizes de uma dada matriz é objetivada. Aplicações para esta relaxação com as quais pretendemos trabalhar são, por exemplo, problemas em grafos. Como mencionado no Capítulo 2 o *Problema da Amostra de Máxima Entropia* pode ser transformado no problema de encontrar um conjunto estável em um grafo. Pretendemos também estender a relaxação de forma a aplicá-la ao problema de encontrar cliques em grafos.

A segunda parte do trabalho apresentado na tese aborda a apresentação e análise de um algoritmo de pontos interiores para programação semidefinida. O trabalho considera o algoritmo primal de trajetória central de passos longos e consiste basicamente na extensão da elegante análise de convergência quadrática de Roos e Vial, de programação linear para programação semidefinida. Os aspectos mais relevantes deste trabalho são a simplicidade e a clareza das demonstrações apresentadas, que comprovam a convergência quadrática do algoritmo para a resolução de problemas de programação semidefinida.

A motivação para este estudo encontra-se no terceiro método de escolha de parâmetros existentes na formulação da relaxação não linear para o *Problema da Amostra de Máxima Entropia Restrito* (CMESP), descrito na Seção 3.3.4. Neste método, a matriz D que compõe a

relaxação é escolhida de forma a minimizar aproximadamente o traço de $D - C$ onde $D - C \succeq 0$. O problema de determinação de D é portanto, um típico problema de programação semidefinida.

A necessidade de resolver este problema toda vez que um subproblema é gerado pelo algoritmo de *branch-and-bound* que resolve CMESP, nos leva a busca de um algoritmo eficaz. Os métodos de pontos interiores se mostram atraentes já que confirmam a cada dia a sua eficiência na resolução de problemas de programação linear e já que se estendem de uma forma bastante natural a programação semidefinida, como tem sido verificado nos inúmeros trabalhos publicados a esse respeito nos últimos dois anos.

O algoritmo primal de trajetória central de passos longos requer pouco trabalho computacional para a obtenção da solução ótima do problema, quando comparado com outros algoritmos de pontos interiores para programação semidefinida. A sua implementação levou a resultados bastante satisfatórios para a nossa aplicação, principalmente por trabalharmos com uma solução de precisão relativamente pequena ($\epsilon = 10^{-3}$). A medida que a precisão desejada para a solução ótima do problema aumenta, problemas numéricos podem prejudicar o comportamento do algoritmo, principalmente quando nos aproximamos muito da fronteira da região viável do problema.

A instabilidade numérica é de fato um aspecto considerado na maioria dos trabalhos que abordam a implementação de métodos de pontos interiores para programação semidefinida. Alguns trabalhos apontam maneiras de contornar os problemas numéricos, de forma a obter uma maior precisão na solução. Sob este enfoque, pretendemos agora justamente implementar um procedimento que evite que pontos muito próximos da fronteira sejam gerados, mas apenas quando erros numéricos ameaçam o comportamento do algoritmo. Dessa forma, a velocidade de convergência não fica comprometida por uma redução muito pequena no valor da função barreira a cada iteração do algoritmo, e por outro lado, introduzimos um controle no tamanho do erro numérico de forma a não impossibilitar a obtenção da solução ótima do problema.

Referências Bibliográficas

- [1] F. Alizadeh (1995), “Interior point methods in semidefinite programming with applications to combinatorial optimization,” *SIAM J. Optimization* 5, 13-51.
- [2] F. Alizadeh, J.-P.A. Haeberly, and M.L. Overton (1995), “Complementarity and nondegeneracy in semidefinite programming,” RUTCOR, Rutgers University (New Brunswick, NJ).
- [3] F. Alizadeh, J.-P.A. Haeberly, and M.L. Overton (1996), “Primal–Dual Interior–Point Methods for Semidefinite Programming: Convergence Rates, Stability and Numerical Results”, working paper, update versions available by anonymous ftp to cs.nyu.edu in the file pub/local/overton/pdsdp.ps.gz.
- [4] J. Allwright (1988), *Positive semidefinite matrices: characterization via conical hulls and least-squares solution of a matrix equation*, SIAM Journal on Control and Optimization, 26, pp.537-556.
- [5] J. Allwright (1989), *On maximizing the minimum eigenvalue of a linear combination of symmetric matrices*, SIAM J. on Matrix Analysis and Applications, 10, pp. 347-382.
- [6] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov and D. Sorensen (1992), *LAPACK Users’ Guide*. SIAM, Philadelphia.
- [7] K.M. Anstreicher (1991), “A combined phase I - phase II scaled potential algorithm for linear programming”, *Mathematical Programming*, 52:429-439.
- [8] K.M. Anstreicher (1996), “On long step path following and SUMT for linear and quadratic programming,” to appear in *SIAM J. Optimization*.
- [9] K.M. Anstreicher, D. den Hertog, C. Roos, and T. Terlaky (1993), “A long–step barrier method for convex quadratic programming,” *Algorithmica* 10, 365-382.
- [10] K.M. Anstreicher and M. Fampa (1996), A long-step path following algorithm for semidefinite programming problems. To appear in: “Proceedings of the Fields Institute Workshop on Semidefinite Programming and Interior-Point Approaches for Combinatorial Optimization Problems”, to be published by AMS.

- [11] K.M. Anstreicher, M. Fampa, J. Lee and J. Williams (1996), “Continuous Relaxations for Constrained Maximum–Entropy Sampling”, in: *Integer Programming and Combinatorial Optimization, Lecture Notes in Computer Science 1084*, W.H. Cunningham, S.T. McCormick, and M. Queyranne, eds., Springer Verlag, pp. 234-248.
- [12] D.A. Bayer and J.C. Lagarias (1989), The Nonlinear Geometry of Linear Programming I. Affine and Projective Scaling Trajectories III. Projective Legendre Transform Coordinates and Hilbert Geometry, *Transactions of the American Mathematical Society* 314, 2, 499-581.
- [13] M.S. Bazaraa, H.D. Sherali, and C.M. Shetty (1993), *Nonlinear Programming. Theory and Algorithms*. Wiley, second edition.
- [14] R. Bellman and K. Fan, *On systems of linear inequalities in Hermitian matrix variables*, in *Convexity*, V.L. Klee, ed., vol. 7 Proceedings of Symposia in Pure Mathematics, American Mathematical Society, 1963, pp. 1-11.
- [15] D. Blackwell (1951), Comparison of experiments. In *Proceedings of the 2nd Berkeley Symposium*, 93-102, University of California Press, Berkeley.
- [16] L. Boltzmann (1877), Beziehung zwischen dem zweiten Hauptatz der Wärmetheorie und der Wahrscheinlichkeitsrechnung resp. den Sätzen über das Wärmeleichgewicht (Complexionen-Theorie). *Wien Ber.* 76², p. 373.
- [17] W.F. Caselton, L. Kan and J.V. Zidek (1991), Quality data network designs based on entropy. In *Statistics in the Environmental and Earth Science*, P. Guttorp and A. Walden, eds., Griffin, London.
- [18] W.F. Caselton and J. Zidek (1984), Optimal monitoring network designs. *Stat. Prob. Lett.* 2, 223-227.
- [19] CPLEX Optimization, Inc. (1994), *Using the CPLEX Callable Library*.
- [20] B. Craven and B. Mond (1981), *Linear programming with matrix variables*, *Linear Algebra and Appl.*, 38, pp. 73-80.
- [21] J. Cullum, W. Donath and P. Wolfe (1975), *The minimization of certain nondifferentiable sums of eigenvalue of symmetric matrices*, *Math. Programming Study*, 3, pp. 35-55.
- [22] D. den Hertog (1992), “Interior point approach to linear, quadratic, and convex programming,” Ph.D. Thesis, Dept. of Applied Mathematics, Delft University of Technology (Delft, Netherlands).
- [23] D. den Hertog, C. Roos, and T. Terlaky (1992), On the classical logarithmic barrier method for a class of smooth convex programming problems. *JOTA* 73, 1-25.

- [24] D. den Hertog, C. Roos, and J.-Ph. Vial (1992), "A complexity reduction for the long-step path-following algorithm for linear programming," *SIAM J. Optimization* 2, 71-87.
- [25] I. Dikin (1967), *Iterative solution of problems of linear and quadratic programming*, Soviet Math. Dokl., 8, pp. 674-675.
- [26] M.K.H. Fan (1993), *A quadratically convergent algorithm on minimizing the largest eigenvalue of a symmetric matrix*, Linear Algebra and Appl., 188-189, pp.231-253.
- [27] M.K.H. Fan and B. Nekoioe (1992), *On minimizing the largest eigenvalue of a symmetric matrix*, in Proc. IEEE Conf. on Decision and Control, Dec. 1992, pp. 134-139.
- [28] L. Faybusovich (1995), "On a matrix generalization of affine-scaling vector fields," *SIAM J. Matrix Anal. Appl.* 16, 886-897.
- [29] L. Faybusovich (1995), "Semi-definite programming: a path-following algorithm for a linear-quadratic functional," Dept. of Mathematics, University of Notre Dame (Notre Dame, IN), to appear in *SIAM J. Optimization*.
- [30] V. Fedorov, S. Leonov, M. Antonovsky and S. Pitovranov (1987), The experimental design of an observation network: software and examples. WP-87-05, International Institute for Applied Systems Analysis, Laxenburg, Austria.
- [31] V. Fedorov and W. Mueller (1989), Comparison of two approaches in the optimal design of an observation network. *Statistics* 20, 339-351.
- [32] A.V. Fiacco and G.P. McCormick (1968), *Nonlinear Programming, Sequential Unconstrained Minimization Techniques*, John Wiley, New York; reprinted as *Classics in Applied Mathematics Vol. 4*, (SIAM, Philadelphia, 1990).
- [33] R. Fletcher (1981), *A nonlinear programming problem in statistics (educational testing)*, SIAM Journal on Scientific and Statistical Computing, 2(3:257-267, September 1981.
- [34] R. Fletcher (1985), *Semidefinite matrix constrains in optimization*, SIAM J. Control and Opt., 23, pp. 493-513.
- [35] R.M. Freund (1994), "Complexity of an algorithm for finding an approximate solution of a semi-definite program with no regularity assumption," Working Paper OR 302-94, Operations Research Center, MIT (Cambridge, MA).
- [36] K.R. Frisch (1955), The Logarithm Potential Method for Solving Linear Programming Problems, Memorandum, Institute of Economics, Oslo, Norway.
- [37] K.R. Frisch (1955), The Logarithm Potential Method of Convex Programming, Memorandum, Institute of Economics, Oslo, Norway.

- [38] M.R. Garey and D.S. Johnson (1979), "Computers and Intractability: A Guide to the Theory of NP-Completeness", W.H. Freeman.
- [39] C. Goh and D. Teo (1988), *On minimax eigenvalue problems via constrained optimization*, Journal of Optimization Theory and Applications, 57, pp. 59-68.
- [40] C.C. Gonzaga (1991), "Large-step path-following methods for linear programming, Part I: Barrier function method," *SIAM J. Optimization* 1, 268-279.
- [41] C.C. Gonzaga (1992), "Path-following methods for linear programming," *SIAM Review* 34, 167-224.
- [42] P. Gutterp, N.D. Le, P.D. Sampson and J.V. Zidek (1992), Using Entropy in the redesign of an environmental monitoring network. Technical Report #116, Department of Statistics, The University of British Columbia, Vancouver, British Columbia.
- [43] J.-B. Hiriart-Urruty and C. Lemaréchal (1993), *Convex Analysis and Minimization Algorithms II: Advanced Theory and Bundle Methods*, volume 306 of *Grundlehren der mathematischen Wissenschaften*. springer-Verlag, New York.
- [44] J.-B. Hiriart-Urruty and D. Ye (1995), *Sensitivity analysis of all eigenvalues of a symmetric matrix*, NumerischeMathematik, 70, pp. 45-72.
- [45] R.A. Horn and C.R. Johnson (1985), *Matrix Analysis*. Cambridge University Press, Cambridge.
- [46] E.T. Jaynes (1963), Information theory and statistical mechanics. In "Statistical Physics Vol.", K.W. Ford, ed., pp. 102-218, W.A. Benjamin, New York.
- [47] N. Karmarkar (1984), *A new polynomial-time algorithm for linear programming*, Combinatorica, 4, pp. 373-395.
- [48] C.-W. Ko, J. Lee, and M. Queyranne (1995), An exact algorithm for maximum entropy sampling. *Operations Research* 43, 684-691.
- [49] M. Kojima, S. Kojima and S. Hara (1994), *Linear algebra for semidefinite programming*, tech. report, Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology, Oh-Okayama, Meguro-ku, Tokyo 152, Japan.
- [50] M. Kojima, N. Megiddo, T. Noma and A. Yoshise (1991), *A Unified Approach to Interior Point Algorithms for Linear Complementarity Problems*, Lecture Notes in Computer Science, Springer-Verlag.
- [51] M. Kojima, S. Shindoh, and S. Hara (1994), "Interior-point methods for the monotone semidefinite linear complementarity problem in symmetric matrices," Dept. of Information Sciences, Tokyo Institute of Technology (Tokyo, Japan).

- [52] K.O. Kortanek and J. Zhu (1993), "A polynomial barrier algorithm for linearly constrained convex programming problems," *MOR* 18, 116-127.
- [53] J. Lee (1995), Constrained maximum-entropy sampling. Working paper, Dept. of Mathematics, University of Kentucky, Lexington, KY.
- [54] B. Lieu and P. Huard (1966), *La méthode des centres dans un espace topologique*, Numerische Mathematik, 8, pp. 56-67.
- [55] C.-J. Lin and R. Saigal (1995), "A predictor corrector method for semi-definite linear programming," Dept. of Industrial and Operations Engineering, University of Michigan (Ann Arbor, MI).
- [56] D. Lindley (1956), On a measure of the information provided by an experiment, *Annals of Mathematical Statistics* 27.
- [57] D. Lindley (1957), Binomial sampling schemes and the concept of information, *Biometrika* 44, 179-186.
- [58] I.J. Lustig, R.E. Marsten and D.F. Shanno (1994), *Interior point methods for linear programming: Computational state of the art*, ORSA Journal on Computing, 6.
- [59] L. McLinden (1980), An Analogue of Moreau's Proximation Theorem, With Application to the Nonlinear Complementarity Problem, *Pacific Journal of Mathematics* 88, 1, 101-161.
- [60] N. Megiddo (1989), Pathways to the Optimal Set in Linear Programming, in: *Progress in Mathematical Programming, Interior Point and Related Methods*, 131-158, N. Megiddo ed., Springer Verlag, New York.
- [61] R.D.C. Monteiro (1995), "Primal-dual path following algorithms for semidefinite programming," School of Industrial and Systems Engineering, Georgia Institute of Technology (Atlanta, GA).
- [62] W.C. Mylander, R.L. Holmes and G.P. McCormick (1971), A Guide to SUMT-Version 4: the Computer Program Implementing the Sequential Unconstrained Minimization Technique of Nonlinear Programming, Research Paper RAC-P-63, Research Analysis Corporation, McLean, VA.
- [63] G. L. Nemhauser, and L. A. Wolsey (1981), Maximizing Submodular Set Functions: Formulations and Algorithms. *Studies on Graphs and Discrete Programming*, P. Hansen (ed.). North-Holland, Amsterdam, 279-301.
- [64] Y. Nesterov and A. Nemirovsky (1988), *A general approach to polynomial-time algorithms design for convex programming*, tech. report, Centr. Econ. & Math. Inst., USSR Acad. Sci., Moscow, USSR.

- [65] Y. Nesterov and A. Nemirovsky (1989), *Self-Concordant Functions and Polynomial Time Methods in Convex Programming*, Report, Central Economical and Mathematical Institute, USSR Academy of Science, Moscow, USSR.
- [66] Y. Nesterov and A. Nemirovsky (1990), *Optimization over positive semidefinite matrices: Mathematical background and user's manual*, USSR Acad. Sci. Centr. Econ. & Math. Inst., 32 Krasikova St., Moscow 117418 USSR.
- [67] Y. Nesterov and A. Nemirovsky (1991), *Conic formulation of a convex programming problem and duality*, tech. report, Centr. Econ. & Math. Inst., USSR Academy of Sciences, Moscow USSR.
- [68] Y. Nesterov and A. Nemirovskii (1994), *Interior-Point Polynomial Algorithms in Convex Programming*, volume 13 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA.
- [69] Y. Nesterov and M.J. Todd (1995), "Primal-dual interior-point methods for self-scaled cones," Technical Report 1125, School of OR/IE, Cornell University (Ithaca, NY).
- [70] M. Overton (1988), *On minimizing the maximum eigenvalue of a symmetric matrix*, SIAM J. on Matrix Analysis and Applications, 9, pp. 256-268.
- [71] M. Overton (1992), *Large-scale optimization of eigenvalues*, SIAM J. Optimization, pp. 88-120.
- [72] M. Overton and R. Womersley (1992), *On the sum of the largest eigenvalues of a symmetric matrix*, SIAM J. on Matrix Analysis and Applications, pp. 41-45.
- [73] M. Overton and R. Womersley (1993), *Optimality conditions and duality theory for minimizing sums of the largest eigenvalues of symmetric matrices*, Mathematical Programming, 62, pp. 321-357.
- [74] E. Panier (1989), *On the need for special purpose algorithms for minimax eigenvalue problems*, Journal Opt. Theory Appl., 62, pp.279-287.
- [75] G. Pataki (1994), *On the multiplicity of optimal eigenvalues*, Tech. Report MSRR-604, Carnegie Mellon University. Graduate School of Industrial Administration.
- [76] F.A. Potra and R. Sheng (1995), "Homogenous interior-point algorithms for semidefinite programming," Dept. of Mathematics, University of Iowa (Iowa City, IA).
- [77] F. Pukelsheim (1993), *Optimal Design of Experiments*, Wiley.
- [78] M. Ramana, L. Tunçel, and H. Wolkowicz (1995), "Strong duality for semidefinite programming," Dept. of Combinatorics and Optimization, University of Waterloo (Waterloo, Ontario, Canada).

- [79] U.T. Ringertz (1991), *Optimal design of nonlinear shell structures*, Tech. Report FFA TN 1991-18, The Aeronautical Research Institute of Sweden, Stockholm.
- [80] R.T. Rockafellar (1970), *Convex Analysis*. Princeton Univ. Press, Princeton, second edition.
- [81] C. Roos and J.-Ph. Vial (1990), "Long steps with the logarithmic penalty barrier function in linear programming," in *Economic Decision Making: Games, Economics, and Optimization*, J. Gabszewicz, J.-F. Richard, and L. Wolsey, eds., Elsevier Science, Amsterdam, 433-441.
- [82] C. Roos and J.-Ph. Vial (1992), "A polynomial method of approximate centers for linear programming," *Math. Programming* 54, 295-305.
- [83] C.E. Shannon (1948), The mathematical theory of communication. *Bell Systems Technical Journal* 27, 379-423, 623-656.
- [84] C.E. Shannon and W. Weaver (1963), "The Mathematical Theory of Communication", (paperback edition), The University of Illinois Press, Urbana.
- [85] A. Shapiro (1985), *Extremal problems on the set of nonnegative definite matrices*, *Linear Algebra and Appl.*, 67, pp. 7-18.
- [86] A. Shapiro and M.K.H. Fan (1994), *On eigenvalue optimization*, *SIAM J. on Optimization*. To appear.
- [87] M.C. Shewry and H.P. Wynn (1987), Maximum entropy sampling. *Journal of Applied Statistics* 46, 165-170.
- [88] G. Sonnevend (1986), An "Analytical Centre" for Polyhedrons and New Classes of Global Algorithms for Linear (Smooth, Convex) Programming, *Lectures Notes in Control and Information Sciences* 84, 866-878.
- [89] M.J. Todd, K.C. Toh and R.H. Tütüncü (1996), On the Nesterov-Todd direction in semidefinite programming. Technical report, School of Operations Research and Industrial Engineering, Cornell University.
- [90] L. Vandenberghe and S. Boyd (1993), "Primal-dual potential reduction method for problems involving matrix inequalities." Information Systems Laboratory, Dept. of Electrical Engineering, Stanford University (Stanford, CA).
- [91] L. Vandenberghe and S. Boyd (1994), "Positive definite programming," Information Systems Laboratory, Dept. of Electrical Engineering, Stanford University (Stanford, CA).
- [92] G.A. Watson (1992). *Algorithm for minimum tracefactor analysis*. *SIAM J. on Matrix Analysis and Applications*, 13(4):1039-1053.

- [93] H. Wolkowicz (1981), *Some applications of optimization in matrix theory*, Linear Algebra and Appl., 40, pp. 101-118.
- [94] S. Wu and J.V. Zidek (1992), An entropy based review of selected NADP/NTN network sites for 1983-86. *Atmospheric Environment* **26A**, 2089-2103.
- [95] Y. Zhang (1995), "On extending primal-dual interior-point algorithms from linear programming to semidefinite programming," Dept. of Mathematics and Statistics, University of Maryland (Baltimore, MD).