



MÉTODOS DE FEIXES INEXATOS APLICADOS À PROGRAMAÇÃO ESTOCÁSTICA

Wellington Luis de Oliveira

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia de Sistemas e Computação.

Orientadores: Susana Scheimberg de Makler
Claudia Alejandra Sagastizábal

Rio de Janeiro
Janeiro de 2011

MÉTODOS DE FEIXES INEXATOS APLICADOS À PROGRAMAÇÃO
ESTOCÁSTICA

Wellington Luis de Oliveira

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ
COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE)
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.

Examinada por:



Profª. Susana Scheimberg de Makler, D.Sc.



Dra. Claudia Alejandra Sagastizábal, D.Habil.



Prof. Clóvis Caesar Gonzaga, D.Sc



Prof. Mikhail Vladimir Solodov, Ph.D.



Prof. Nelson Maculan Filho, D.Habil.



Prof. Paulo Roberto Oliveira, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
JANEIRO DE 2011

Oliveira, Welington Luis de

Métodos de Feixes Inexatos Aplicados à Programação Estocástica/Welington Luis de Oliveira. – Rio de Janeiro: UFRJ/COPPE, 2011.

XVI, 183 p.: il.; 29, 7cm.

Orientadores: Susana Scheimberg de Makler

Claudia Alejandra Sagastizábal

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2011.

Referências Bibliográficas: p. 164 – 171.

1. Otimização estocástica. 2. Otimização não diferenciável. 3. Análise convexa. 4. Métodos de feixes proximal. 5. Métodos de nível. 6. Técnicas de decomposição. I. Makler, Susana Scheimberg de *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Aos meus pais, Dete e Luiz.

Agradecimentos

Dedico minhas primeiras palavras de agradecimento às minhas orientadoras Profa. Dra. Susana Scheimberg de Makler e Dra. Claudia Sagastizábal. Agradeço à Profa. Susana a oportunidade dada, sua orientação e confiança. Sua ajuda, paciência e boa vontade foram importantes em vários aspectos deste trabalho. Dedico sincero e especial agradecimento à Dra. Claudia pela sugestão do tema desta tese, pelo apoio fundamental em momentos importantes, pela confiança e, sobretudo, pela orientação incondicional em muitos aspectos da minha formação profissional. Seu constante apoio, que por diversas vezes fora além da dinâmica acadêmica, foi fundamental para tornar real este ideal.

Agradeço à Alessandra Maria de Andrade (agora bióloga!) a dedicação, paciência e ajuda ao longo destes dois anos e tantos meses. Obrigado pela liberdade dada para buscar meus próprios interesses, mesmo quando estes, em certo sentido, se apresentaram incompreensíveis ou perigosos.

Agradeço aos meus amigos Alberto Ordine, Carlos Henrique, Francisco Faria, Michel Pompeu e Natal Elson, o incentivo, os momentos de descontração e, sobretudo, a amizade. Também agradeço aos meus amigos dos tempos de mestrado Cássio Alves e Leonardo Moraes, a amizade e incentivo.

Agradeço aos meus parentes de forma geral, em especial à minha avó Maria Antônia, aos meus pais Maria Bernadete e Luiz e aos meus irmãos Tatiane e Erivelton, que torceram e acreditaram em mim e foram compreensivos quando, por vários momentos, não pude estar presente em momentos importantes de suas vidas.

Finalmente, agradeço a todos os professores/pesquisadores que cordialmente aceitaram compor a banca examinadora desta tese e também aos professores e funcionários do PESC. Sou particularmente grato ao professor Csabá I. Fábíán do Kecs-kemét College (Hungria) pela sugestão de incorporar o nosso oráculo parcialmente inexato ao método de nível.

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

MÉTODOS DE FEIXES INEXATOS APLICADOS À PROGRAMAÇÃO ESTOCÁSTICA

Wellington Luis de Oliveira

Janeiro/2011

Orientadores: Susana Scheimberg de Makler
Claudia Alejandra Sagastizábal

Programa: Engenharia de Sistemas e Computação

Problemas de programação estocástica aparecem em muitas aplicações práticas. Em geral, a formulação determinística equivalente desses problemas podem resultar em problemas de otimização muito grandes, que não podem ser resolvidos diretamente. Para o caso particular dos programas estocásticos com recursos, são consideradas técnicas de decomposição que podem lidar com aproximações na solução dos subproblemas. Do ponto de vista da otimização não diferenciável, essas técnicas consistem na aplicação de métodos de feixes empregados com oráculos inexatos, que realizam avaliações aproximadas da função objetivo e de um subgradiente, com erros de precisão limitados, mas possivelmente desconhecidos.

Ao invés de forçar a terminação antecipada na solução dos subproblemas, para definir os oráculos inexatos, são selecionados alguns subproblemas para os quais a solução é exata. Os demais subproblemas são aproximados por um processo rápido, que não envolve a solução de um problema de otimização. As informações aproximadas fornecidas pelos oráculos são utilizadas para construir linearizações inexatas no programa mestre, que são bem administradas pelos métodos de feixes inexatos recentemente desenvolvidos, e pelos métodos de nível inexatos propostos nesta tese.

Além do desenvolvimento teórico, garantindo o controle na aproximação efetuada, são apresentados resultados numéricos promissores, quando comparados com o método de feixes exato e com a decomposição de Benders clássica.

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

INEXACT BUNDLE METHODS FOR STOCHASTIC PROGRAMMING

Wellington Luis de Oliveira

January/2011

Advisors: Susana Scheimberg de Makler
Claudia Alejandra Sagastizábal

Department: Systems Engineering and Computer Science

Stochastic programming problems arise in many practical situations. In general, the deterministic equivalents of these problems can be very large and may not be solvable directly by general-purpose optimization approaches. For the particular case of stochastic programs with recourse, we consider decomposition approaches that can handle inexactness in the subproblem solution. From a nonsmooth optimization perspective, these variants amount to applying bundle methods to oracles that give inaccurate values for the objective function and a subgradient.

Rather than forcing early termination of the subproblems optimization to define inexact oracles, we select a small subset of scenarios for which the subproblem solution is exact, and replace the information for the remaining scenarios by a fast procedure that does not involve solving an optimization problem. The inaccurate oracle information creates inexact cuts in the master program, that are well handled by the recently introduced inexact bundle methods, and by the proposed inexact level methods.

The proposed approaches are validated by encouraging numerical results on several stochastic linear programs found in the literature.

Sumário

Lista de Figuras	xi
Lista de Tabelas	xiii
Lista de Símbolos	xiv
Lista de Abreviaturas	xvi
1 Introdução	1
1.1 Métodos de Otimização Estocástica com Resolução Inexata	5
1.2 Objetivo e Contribuições do Trabalho	9
1.3 Organização do Trabalho	10
2 Programação Estocástica	13
2.1 Considerações Gerais	13
2.2 Programação Estocástica em Dois Estágios	15
2.2.1 Condições de Otimalidade	20
2.3 Programação Estocástica em Multiestágios	22
2.3.1 Condições de Otimalidade	27
2.3.2 Explosão da Dimensionalidade	28
3 Método de Feixes Proximal Inexato	30
3.1 Considerações Iniciais	31
3.1.1 Comparação entre os Métodos Exato e Inexato	33
3.2 Desenvolvimento Teórico	36
3.3 Condições de Otimalidade Inexata	39
3.3.1 Algoritmo Inexato	40
3.3.2 Análise de Convergência	41
3.3.3 Determinação dos Iterados	47
3.4 Atualização do Passo Proximal	49

4	Método de Feixes Proximal Parcialmente Inexato	51
4.1	Considerações Iniciais	51
4.1.1	Comparação entre os Métodos Inexato e Parcialmente Inexato	53
4.2	Desenvolvimento Teórico	55
4.2.1	Algoritmo Parcialmente Inexato	56
4.2.2	Análise de Convergência	57
4.3	Terminação Finita	58
5	Método de Nível Proximal Inexato	67
5.1	Considerações Iniciais	68
5.1.1	Comparação entre os Métodos Exato e Inexato	69
5.2	Desenvolvimento Teórico	75
5.3	Algoritmo Inexato	77
5.3.1	Análise de Convergência	79
6	Método de Nível Parcialmente Inexato	83
6.1	Considerações Iniciais	83
6.1.1	Comparação entre os Métodos Inexato e Parcialmente Inexato	85
6.2	Desenvolvimento Teórico	87
6.3	Algoritmo Parcialmente Inexato	89
6.3.1	Análise de Convergência	92
7	Métodos de Feixes Inexatos Aplicados à Programação Linear Esto-	
	cástica em Dois Estágios	95
7.1	Programação Linear Estocástica em Dois Estágios	95
7.2	Critério de Colinearidade	97
7.2.1	Caso 1: Programas com Recurso e Custo Fixos	97
7.2.2	Caso 2: Programas Lineares Estocásticos Gerais	101
8	Métodos de Feixes Inexatos Aplicados à Programação Não Linear	
	Estocástica em Dois Estágios	105
8.1	Programação Não Linear Estocástica em Dois Estágios	105
8.1.1	Redução Ótima de Cenários	107
8.2	Redução Ótima de Cenários Sequencial	110
8.3	Seleção Sucessiva de Cenários	113
8.4	Classificação em Grupos	115
9	Métodos de Feixes Inexatos Aplicados à Programação Linear Esto-	
	cástica em Multiestágios	120
9.1	Decomposição Aninhada	121
9.1.1	Decomposição Aninhada de Benders	123

9.1.2	Decomposição Aninhada e Métodos de Feixes	125
9.2	Aproximação por Dois Estágios de Programas em Multiestágios . . .	126
9.2.1	Aproximação por Multiproblemas em Dois Estágios	129
9.3	Decomposição Dual	132
9.3.1	Considerações Iniciais	132
9.3.2	Relaxação das Restrições de Mensurabilidade	134
9.3.3	Critério de Colinearidade	136
10	Resultados Numéricos	140
10.1	Programas Lineares em Dois Estágios	140
10.1.1	Principais Características dos Casos	140
10.1.2	Resultados para SH10	143
10.1.3	Resultados para os Problemas com Medida de Risco	146
10.1.4	Avaliação do Desempenho das Técnicas	148
10.1.5	Métodos Parcialmente Inexatos	153
10.2	Programas Lineares em Multiestágios	157
11	Considerações Finais e Direções Futuras	160
11.1	Direções Futuras	162
	Referências Bibliográficas	164
A	Análise Convexa e Teoria de Probabilidade	172
A.1	Análise Convexa	172
A.2	Probabilidades	174
B	Perfil da Performance	176
C	Exemplos de Programas Lineares Estocásticos em Dois Estágios	178
C.1	Planejamento da Expansão e Operação Termoelétrica - <i>ExpTerm</i> . . .	178
C.2	Programa Linear Estocástico com Recurso Completo - <i>SH10</i>	182

Lista de Figuras

1.1	Decomposição em programas mestre e escravos.	2
1.2	Aproximações por planos cortantes.	6
2.1	Árvore de cenários.	25
3.1	Métodos de feixes proximal exato (esquerda) e inexato (direita).	35
3.2	Processo de atenuação do ruído. A imprecisão do oráculo é excessiva.	36
4.1	Método de feixes parcialmente inexato.	55
5.1	Métodos de nível proximal exato (esquerda) e inexato (direita). Iteração 1.	72
5.2	Métodos de nível proximal exato (esquerda) e inexato (direita). Iterações 2 e 3.	73
5.3	Métodos de nível proximal exato (esquerda) e inexato (direita). Iteração 4.	74
6.1	Método de nível parcialmente inexato. Iterações 1 e 2.	86
6.2	Método de nível parcialmente inexato. Iteração 3.	87
8.1	Aprimoramento sucessivo do processo estocástico.	111
8.2	Seleção sucessiva de cenários.	113
9.1	Árvore com 5 períodos, 81 cenários, e 121 nós.	125
9.2	Aproximação por dois estágios de programas em multiestágios.	128
9.3	Árvore que permite um número menor de restrições de mensurabilidade.	139
10.1	Intervalo de confiança para as variáveis de primeiro estágio - SH10.	145
10.2	Desempenho das técnicas para todos os problemas.	150
10.3	Desempenho das técnicas para todos os problemas (combinação entre acurácia e redução de CPU).	151
10.4	Desempenho das técnicas para os problemas com custo q fixo.	152
10.5	Desempenho das técnicas para os problemas com custo q fixo (combinação entre acurácia e redução de CPU).	153

10.6	Desempenho das técnicas para os problemas com custo q fixo - métodos parcialmente inexatos.	157
10.7	Desempenho dos métodos de feixes para problemas em multiestágios.	158

Lista de Tabelas

10.1	Parâmetros	143
10.2	Valor ótimo e tempo de CPU para SH10.	143
10.3	Qualidade da solução - SH10.	144
10.4	Redução de tempo de CPU - SH10.	144
10.5	Limites para o valor ótimo - SH10.	145
10.6	Variáveis duais de segundo estágio - SH10.	146
10.7	Qualidade da solução - InvestmentRisk.	147
10.8	Redução de tempo de CPU - InvestmentRisk.	147
10.9	Qualidade da solução - SH10Risk.	147
10.10	Redução de tempo de CPU - SH10Risk.	147
10.11	Parâmetros	149
10.12	Média dos erros percentuais (11 casos).	149
10.13	Média das reduções de tempo de CPU (11 casos).	149
10.14	Probabilidade da melhor performance.	151
10.15	Probabilidade da melhor performance - q fixo.	152
10.16	Redução do tempo de CPU - métodos parcialmente inexatos.	155
10.17	Média das reduções de tempo de CPU - métodos parcialmente inexatos.	156
10.18	Qualidade da solução e redução do tempo de CPU - PlanFin3.	158
10.19	Média dos erros e redução de CPU (%).	159
C.1	<i>ExpTerm</i> . Parâmetros das usinas 1-5.	181
C.2	<i>ExpTerm</i> . Parâmetros das usinas 6-10.	181
C.3	<i>ExpTerm</i> . Demanda.	181

Lista de Símbolos

Ξ	Conjunto dos parâmetros incertos, p. 1
\mathcal{F}	σ -álgebra, p. 1
(Ξ, \mathcal{F})	Espaço amostral, p. 1
P	Distribuição de probabilidades, p. 1
(Ξ, \mathcal{F}, P)	Espaço de probabilidades, p. 1
ξ	Variável aleatória, p. 1
ξ	Evento da variável aleatória, denominado cenário, p. 1
\mathbb{E}	Operador do valor esperado, p. 15
\mathcal{X}	Conjunto viável, p. 1
D	Diâmetro do conjunto \mathcal{X} , p. 68
$\mathcal{N}_{\mathcal{X}}(z)$	Cone das direções normais em \mathcal{X} , no ponto z , p. 47
$i_{\mathcal{X}}(z)$	Função indicadora do conjunto \mathcal{X} , p. 31
$\text{ri}(\mathcal{X})$	Interior relativo do conjunto \mathcal{X} , p. 37
$P_{\mathcal{X}}(z)$	Projeção (ortogonal) do ponto z sobre o conjunto \mathcal{X} , p. 7
$Q(z, \xi)$	Função de recurso no ponto z e cenário ξ , p. 17
$\mathcal{Q}(z)$	Valor esperado da função de recurso no ponto z , p. 17
\mathcal{R}	Conjunto dos números reais, p. 1
$\partial f(z)$	Subdiferencial da função f , no ponto z , p. 3
$\text{dom } f$	Domínio da função f , p. 15
$\check{f}_k(z)$	Modelo de planos cortantes no ponto z , p. 32

Λ	Constante de Lipschitz, p. 69
J^k	Conjunto de índices do feixe de informações, p. 32
N	Número de cenários, p. 2
$f^N(z)$	Função definida a partir de N cenários, no ponto z , p. 2
$\varepsilon_f, \varepsilon_g, \varepsilon_z$	Erros de precisão do oráculo inexato, p. 31
ε_{roc}	Tolerância para a redução de cenários, p. 109
ε_{cos}	Tolerância para o critério de colinearidade, p. 98
ζ_{inex}	Parâmetro para o oráculo parcialmente inexato, p. 51
V_k	Medida de otimalidade para os métodos de feixes, p. 38
Δ_k	Brecha de otimalidade, p. 75
I^E	Conjunto de índices dos subproblemas que são resolvidos de maneira exata, p. 98
$c(\mathbb{J})$	Métrica de Fortet-Mourier, p. 108
$d_\lambda(\xi, \tilde{\xi})$	Métrica utilizada para calcular a distância entre os cenários ξ e $\tilde{\xi}$, p. 113

Lista de Abreviaturas

asci	Aleatória semicontínua inferiormente, p. 17
c.p.1	Com probabilidade 1, p. 20
q.c.	Quase certamente, p. 15
sci	Semicontínua inferiormente, p. 17
DAB	Decomposição aninhada de Benders, p. 120
DIB	Decomposição inexata de Benders, p. 154
DIN	Decomposição inexata de nível, p. 154
MFE	Método de feixes proximal exato, p. 141
MFI	Método de feixes proximal inexato, p. 4
MFPI	Método de feixes proximal parcialmente inexato, p. 11
MNE-LNN	Método de nível exato, p. 154
MNE	Método de nível proximal exato, p. 154
MNI	Método de nível proximal inexato, p. 4
MNPI	Método de nível parcialmente inexato, p. 11
MPC	Método de planos cortantes, p. 141
PP	Perfil da performance, p. 148
ROCSeq	Redução ótima de cenários sequencial, p. 110
ROC	Redução ótima de cenários, p. 106
SAA	Sample average approximation, p. 2
SSC	Seleção sucessiva de cenários, p. 113

Capítulo 1

Introdução

Em muitas situações realísticas tem-se a necessidade de tomar decisões na presença de incertezas e, além disso, almeja-se que tais decisões sejam realizadas atendendo algum critério de otimalidade. Uma área da ciência dedicada aos problemas de tomadas de decisões sob incertezas é a denominada *programação estocástica* que, sendo uma área multidisciplinar, envolve conceitos e resultados de otimização, probabilidades, estatística, e computação.

Em termos gerais, a programação estocástica auxilia na determinação (sob incertezas) de uma estratégia $x \in \mathcal{R}^n$, de modo que um objetivo $f(x)$ seja otimizado. Afim de definir formalmente o campo de estudo deste trabalho, seja $t \in \{1, \dots, T\}$ um índice de tempo correspondente ao horizonte de T estágios. Dados os conjuntos compactos $\Xi_t \subset \mathcal{R}^{s_t}$ para $t = 1, \dots, T$, seja \mathcal{F} a σ -álgebra gerada pelo conjunto $\Xi := \Xi_1 \times \dots \times \Xi_T$. O espaço (Ξ, \mathcal{F}) é chamado de *espaço amostral*, e $\xi = (\xi_1, \dots, \xi_T) \in (\Xi, \mathcal{F})$ é denominado *cenário*. Dado um estágio de tempo $t \in \{1, \dots, T\}$, cada realização ξ_t é um evento da variável aleatória ξ_t ; desde modo, o cenário ξ é uma realização do processo estocástico $\boldsymbol{\xi} := \{\xi_t\}_{t=1}^T$. O problema de otimização baseado no *valor esperado*, e considerando o *processo estocástico* $\boldsymbol{\xi}$ contínuo no *espaço de probabilidades* (Ξ, \mathcal{F}, P) , tem a forma

$$\min_{x \in \mathcal{X}} f(x), \quad \text{com} \quad f(x) := \int_{\Xi} f(x, \xi) dP(\boldsymbol{\xi}), \quad (1.1)$$

onde o conjunto viável $\mathcal{X} \subset \mathcal{R}^n$ é assumido ser não vazio, fechado e convexo, e a função $f : \mathcal{X} \times \Xi \rightarrow \mathcal{R}$ é suposta ser convexa, e *aleatoriamente semicontínua inferiormente*. Adicionalmente, ao longo deste trabalho é suposto que $f(x, \cdot)$ seja *mensurável e P -integrável*.

Quando o processo estocástico é contínuo e não há uma representação analítica para o valor esperado que define a função objetivo em (1.1), o problema é intratável computacionalmente, devido às limitações numéricas para o cálculo da integral (em geral multidimensional). Desta forma, o processo estocástico $\boldsymbol{\xi}$ é discretizado em um

número finito N de *cenários* (ou eventos), e para $p(\xi^i)$ a probabilidade de ocorrência do cenário ξ^i , o valor esperado é aproximado pelo somatório

$$f^N(x) := \sum_{i=1}^N f(x, \xi^i) p(\xi^i).$$

Naturalmente, quanto mais fina for a discretização de $\boldsymbol{\xi}$, i.e., quanto maior for o valor de N , melhor será a aproximação do processo estocástico. Se os cenários $\xi^i \in \Xi$ são equiprováveis, $p(\xi^i) = \frac{1}{N}$, e sorteados aleatoriamente segundo a distribuição de probabilidades P , pela *Lei dos Grandes Números* [1, Seção 7.2] tem-se a relação

$$\lim_{N \rightarrow \infty} f^N(x) \rightarrow f(x),$$

para $x \in \mathcal{X}$ um ponto arbitrário. Deste modo, para que o problema de otimização estocástica

$$\min_{x \in \mathcal{X}} f^N(x), \tag{1.2}$$

denominado *aproximação na média amostral*, ou *Sample Average Approximation - SAA* - em inglês [1, Capítulo 5], tenha um conjunto solução e valor ótimo próximos daqueles fornecidos pelo problema (1.1), é preciso que N seja suficientemente grande. Por outro lado, quanto maior for o número N , mais dispendioso será o cálculo de $f^N(x)$, uma vez que dados $x \in \mathcal{X}$ e $\xi^i \in \Xi$, calcular o valor da função $f(x, \xi^i)$ envolve geralmente a resolução de um ou mais problemas de otimização, chamados de subproblemas ou programas escravos, como ilustrado na Figura 1.1.

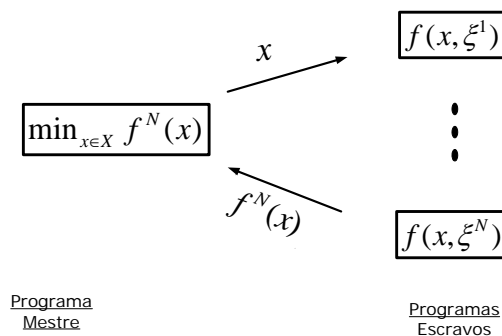


Figura 1.1: Decomposição em programas mestre e escravos.

Para resolver numericamente o problema de otimização estocástica (ou simplesmente programa estocástico) (1.1) quando o processo estocástico é contínuo, ou quando é discreto, mas tem um número muito elevado de cenários (por exemplo 10^4), deve-se escolher entre as seguintes opções, de natureza oposta:

- uma representação pobre (com poucos cenários) do processo estocástico $\boldsymbol{\xi}$ é considerada, e o problema de otimização é resolvido de forma exata; ou

- uma representação suficientemente boa (com muitos cenários) do processo estocástico ξ é considerada, e a solução do problema de otimização é inexata.

Este trabalho está focado em técnicas de decomposição para programas estocásticos, que permitem um bom compromisso entre resolubilidade e acurácia na representação das incertezas.

Em programação estocástica, frequentemente precisa-se resolver problemas de grande porte com uma estrutura diagonal em blocos, favorável às técnicas de decomposição, tais como a *decomposição de Dantzig-Wolfe* e a *decomposição de Benders* [2, Capítulo 11.1]. Em particular, o método *L-shaped* [3] pode ser visto como uma decomposição de Benders aplicada aos programas lineares estocásticos em dois estágios. Ao longo deste trabalho o método L-shaped é considerado adotando o ponto de vista da otimização não diferenciável. Neste sentido, este pode ser visto como um método de *planos cortantes* para minimizar uma função convexa e não diferenciável f , obtida pela soma da função objetivo de primeiro estágio com a função de recurso do segundo estágio. Usualmente a função de recurso é difícil de ser avaliada, porque envolve o cálculo de uma integral multidimensional e, portanto, muitos subproblemas de segundo estágios devem ser resolvidos, como ilustrado na Figura 1.1.

O algoritmo de planos cortantes apresentado em [4, 5] define o novo iterado $z^{k+1} \in \mathcal{X}$ a partir do conhecimento do valor da função $f(z^j)$ e de um subgradiente $g(z^j) \in \partial f(z^j)$ dos iterados anteriores z^j , para $j = 1, \dots, k$. Fixado um ponto z , o conhecimento de $f(z)$ e $g(z)$ é referenciado neste trabalho como a informação do *oráculo*, [2, Capítulo 9.3]. Apesar de bastante confiável, o método de planos cortantes tem duas principais desvantagens:

- (i) instabilidade durante o processo de otimização, i.e., iterações sucessivas do método podem não proporcionar um progresso no processo de otimização, no sentido que o decréscimo no valor da função objetivo é pequeno (ou mesmo negativo), enquanto pode haver uma grande diferença entre os iterados da variável de decisão;
- (ii) não existe um desenvolvimento teórico que permita a remoção de cortes (linearização de f) garantindo a convergência do método. Assim sendo, o problema de otimização tem cada vez mais restrições e, deste modo, é cada vez mais difícil de ser resolvido.

Estas duas desvantagens são eliminadas pelos *métodos de feixes* [6, vol. II], os quais podem ser vistos como variantes estabilizadoras do algoritmo de planos cortantes, que mantêm-se convergentes mesmo após a remoção de alguns cortes.

Uma versão regularizada do método L-shaped proposta em [7] explora, no campo da programação estocástica, as ideias do método de feixes. Os métodos propostos

neste trabalho para a programação estocástica em dois estágios diferem do *L-shaped regularizado* de [7] em um ponto importante: em vez de considerar cortes exatos, i.e, linearizações da forma

$$f(z^k) + g(z^k)^\top (\cdot - z^k),$$

são considerados cortes *inexatos*, dados por

$$f_z^k + g_z^{k\top} (\cdot - z^k).$$

Dado $z^k \in \mathcal{X}$, os valores f_z^k e g_z^k são obtidos por um *oráculo inexato* que estima o valor exato da função e do subgradiente com erro de precisão $\varepsilon > 0$ limitado, mas possivelmente desconhecido. Para lidar com um oráculo inexato, é empregado neste trabalho o *método de feixes proximal inexato* - MFI, desenvolvido por Kiwiel [8]; e o *método de nível proximal inexato* - MNI, que é fundamentado no trabalho [9], do mesmo autor. O MFI determina assintoticamente uma ε -solução com um mecanismo similar ao método clássico de feixes, exceto por um esquema de *atenuação do ruído* quando a inexatidão dos cortes é detectada. O MNI determina assintoticamente uma ε -solução procedendo similarmente ao método de nível proximal (exato), segundo formulado por Kiwiel [9].

Mais especificamente para a programação estocástica, quando o número de cenários N é muito grande (por exemplo 10^4), torna-se então interessante (se não necessário) aproximar a função utilizando estimativas f_z do valor $f^N(z)$, para $z \in \mathcal{X}$. É desejável que tais estimativas estejam próximas do valor $f^N(z)$, i.e., que

$$f_z \in [f^N(z) - \varepsilon_f, f^N(z) + \varepsilon_g], \quad (1.3)$$

para $\varepsilon_f, \varepsilon_g \geq 0$ dois erros de precisão tão pequenos quanto possível. Como será visto no Capítulo 3 (resp. 5), para que o MFI (resp. MNI) convirja, é importante que o erro de aproximação $\varepsilon = \varepsilon_f + \varepsilon_g$ seja limitado (mas não precisa ser assintoticamente nulo).

Utilizando estas aproximações, seja k um contador de iteração e $x^k \in \mathcal{X}$ um centro de estabilidade dado. A cada iteração, o problema considerado pelo MFI para obter o novo iterado z^{k+1} é dado por

$$\min_{z \in \mathcal{X}} \phi_k(z), \quad \text{com} \quad \phi_k(z) := \check{f}_k(z) + \frac{1}{2t_k} \|z - x^k\|^2, \quad (1.4)$$

onde $\check{f}_k(\cdot) := \max_{j \in J^k} \{f_z^j + g_z^{j\top}(\cdot - z^j)\}$ é um modelo de planos cortantes inexato da função f^N , t_k é um passo proximal, e J^k é um conjunto de índices correspondentes a algumas (ou todas) iterações passadas. Já o MNI define o novo iterado resolvendo

o problema de projeção

$$\min_{z \in X^k} \|z - x^k\|^2,$$

para $X^k \subset \mathcal{X}$ um conjunto não vazio, convexo, e fechado; definido por $X^k := \{z \in \mathcal{X} : \check{f}_k(z) \leq f_{niv}^k\}$, com f_{niv}^k o *parâmetro do nível*, que é atualizado iterativamente de forma apropriada.

Ao longo deste trabalho, o método de feixes proximal inexato e o método de nível proximal inexato são, por conveniência, referenciados por *métodos de feixes inexatos*.

Fundamentado nos Capítulos 3, 7 e 8, o trabalho [10] considera a aplicação do MFI aos programas estocásticos em dois estágios. Outras técnicas especiais de otimização para resolver de maneira aproximada este tipo de problemas são bem conhecidas. A seguir é apresentada uma síntese de alguns métodos que tem sido utilizados ao longo das últimas décadas, por diferentes autores.

1.1 Métodos de Otimização Estocástica com Resolução Inexata

O principal interesse em utilizar métodos capazes de lidar com oráculos inexatos em programação estocástica se deve a possibilidade de redução do tempo computacional para resolver os problemas de otimização do tipo (1.2).

Em programação estocástica existem diversas maneiras de modelar e manipular as realizações futuras do evento incerto. Por exemplo, quando os parâmetros incertos se prolongam por mais de um estágio t (caso multiestágios), uma árvore de cenários é normalmente considerada para representar o processo estocástico [11, pp. 21-22]. Quando há muitos estágios, esta representação pode atingir proporções colossais. Assim sendo, alguns métodos de otimização estocástica avançam na árvore de cenários através de apenas algumas de suas ramificações, escolhidas de forma aleatória. Esta é, por exemplo, a estratégia adotada pela *decomposição dinâmica dual estocástica* - SDDP - desenvolvida em [12]; pela *decomposição aninhada abreviada* - AND [13]; pela *decomposição com amostragem reduzida* - ReSa [14]; ou ainda, pelo *método de planos cortantes com amostragem parcial* - CUPPS, desenvolvido em [15]. Todos estes métodos aproximam a função objetivo por um modelo de planos cortantes inexato, como apresentado na Figura 1.2(a). O programa mestre é resolvido pelo método de planos cortantes, e por isso, as mesmas desvantagens (i) e (ii) mencionadas anteriormente estão presentes nos métodos citados acima.

Contudo, o caso em multiestágios não é o único em que métodos inexatos de otimização são considerados. Na programação linear estocástica em dois estágios este tipo de estratégia já foi explorada por Zakeri, Philpott e Ryan [16] para a

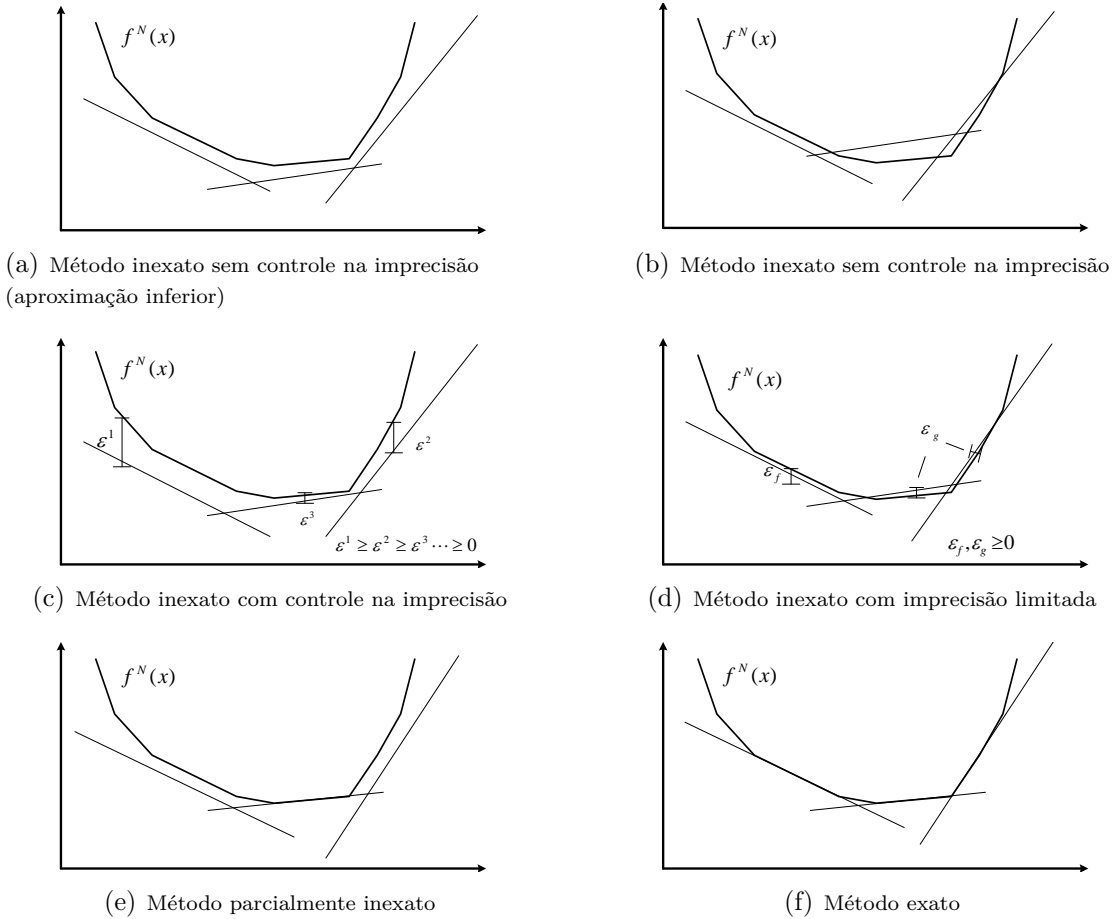


Figura 1.2: Aproximações por planos cortantes.

decomposição de Benders; porém, sendo um método de planos cortantes, esta abordagem também apresenta as desvantagens (i) e (ii). No método proposto pelos autores, denominado *decomposição inexacta de Benders*, o conjunto de cenários é fixo ao longo do processo de otimização. Zakeri, Philpott e Ryan consideram os problemas de otimização linear estocástica em que cada subproblema é de grande porte, no sentido de possuir muitas variáveis e restrições (este é, por exemplo, o caso em que um programa com multiestágios é reformulado como um programa em dois estágios). Tais subproblemas são resolvidos de maneira aproximada pelo *método primal-dual de pontos interiores* para a programação linear [2, Capítulo 4], fornecendo uma ε^k -solução e um ε^k -subgradiente. O erro $\varepsilon^k \geq 0$, fixado a priori, serve como teste de parada para a resolução de cada subproblema: uma ε^k -solução é encontrada quando a brecha de dualidade de cada subproblema é inferior a ε^k . Deste modo, para z^k e ξ^i dados, quanto menor for a tolerância $\varepsilon^k \geq 0$, maior será o esforço computacional exigido para calcular (aproximadamente) a função $f(z^k, \xi^i)$ (em compensação, menor será a inexactidão do método).

O algoritmo apresentado em [16] é inicializado com um valor $\varepsilon^1 > 0$. O método iterativamente seleciona $0 \leq \varepsilon^{k+1} \leq \varepsilon^k$ e faz com que o erro ε^k tenda a zero,

encontrando assim uma solução assintoticamente exata do problema de otimização linear estocástica considerado. Este método é ilustrado na Figura 1.2(c).

Fundamentados no *método de nível inexato* proposto por Fábíán [17], Fábíán e Szóke [18] propuseram a *decomposição inexata de nível* para a programação linear estocástica em dois estágios. A decomposição inexata de nível também considera o tipo de aproximação introduzida em [16] para calcular as linearizações da função objetivo, com a diferença importante que o método utilizado para resolver o problema é do tipo método de feixes¹ [19], em vez do método de planos cortantes considerado em [16]. Novamente, o erro $\varepsilon^k \geq 0$ é conhecido e iterativamente tende a zero fazendo com que o método convirja para uma solução assintoticamente exata do problema, como na Figura 1.2(c).

Para resolver o problema (1.2) (com $T = 2$), os métodos apresentados em [16, 18] proporcionam uma redução do esforço computacional somente nas iterações iniciais. Com efeito, visto que $\varepsilon^k \rightarrow 0$, na medida em que o método avança ao longo das iterações, todos os N subproblemas são resolvidos de maneira cada vez mais exata.

Oráculos inexatos tem sido também utilizados em parceria com algoritmos que utilizam amostragens. Em [20] é apresentada a *decomposição estocástica - DE* - desenvolvida por Higle e Sen. A DE é um método de amostragem para os problemas de otimização linear estocástica em dois estágios, e é um tipo de método de planos cortantes. Em cada iteração k do método é sorteado um novo cenário ξ^k , e resolvido o seu subproblema associado obtendo a variável dual u^k . O método armazena estas variáveis ao longo do processo iterativo no conjunto $\mathcal{D}_k = \mathcal{D}_{k-1} \cup \{u^k\}$, sendo $\mathcal{D}_0 := \emptyset$. Para economizar esforço computacional, \mathcal{D}_k é utilizado como o conjunto viável para aproximar as soluções duais dos demais subproblemas associados aos cenários ξ^j , para $j = 1, \dots, k-1$. Deste modo, o método proposto por Higle e Sen resolve um único subproblema a cada iteração, e constrói um modelo de planos cortantes inexato, como ilustrado na Figura 1.2(b).

Baseados na decomposição estocástica, Au, Higle e Sen [21] utilizam subgradiantes $p^k \in \partial \check{f}_k(z^k)$ para definir o novo iterado do método pela regra $z^{k+1} := P_{\mathcal{X}}(z^k - t_k p^k)$ (em vez de resolver um programa mestre), onde $t_k > 0$ é o tamanho do passo, e $P_{\mathcal{X}}(\cdot)$ é o operador de projeção sobre o conjunto \mathcal{X} . Como p^k é um subgradiente de $\check{f}_k(z^k)$, p^k é um *subgradiente inexato* de $f^N(z^k)$ (com $N = k$), pela propriedade de convexidade. Por este motivo, o método proposto em [21] é chamado de *método de subgradiente inexato*. Assim como a DE, uma deficiência do método de subgradiente inexato é a ausência de um teste de parada confiável.

Com o objetivo de controlar o número de hiperplanos (cortes) utilizados para definir o modelo de planos cortantes, e obter assim uma subsequência de pontos $\{x^k\} \subseteq \{z^k\} \subset \mathcal{X}$ para os quais a função objetivo tenha valores monotonamente

¹Método de Nível ou “Level Method”.

decrecentes, Higel e Sen estenderam a DE para a denominada *decomposição estocástica regularizada* [22]. O método define cada iterado pela regra $z^{k+1} := x^k + td^k$, onde $t > 0$ é uma constante, e d^k é uma direção obtida resolvendo um problema quadrático, similar ao problema quadrático considerado pelo método de feixes proximal. O ponto x^k é chamado de *passo de incumbência*, e é essencialmente o chamado *passo sério* do método de feixes (ver Capítulo 3).

A decomposição estocástica regularizada pode ser interpretada como um método de feixes “aleatório”. Devido à aleatoriedade na escolha dos problemas de segundo estágio a serem resolvidos, os cortes inexatos satisfazem estimativas estatísticas que, com probabilidade um, epi-convergem à função f para uma subsequência de passos de incumbência. Para testar a otimalidade de um ponto $x^k \in \mathcal{X}$ dado, é utilizada uma heurística que reconstrói (várias vezes) o modelo de planos cortantes em x^k , utilizando o conjunto de variáveis duais \mathcal{D}_k . Se uma proporção significativa das distâncias dos modelos reconstruídos ao valor $f^N(x^k)$ (com $N = k$) for inferior a uma tolerância dada, o ponto x^k é aceito como uma solução aproximada do problema considerado.

Apesar dos passos de incumbência serem os passos sérios do método de feixes, as técnicas propostas neste trabalho são conceitualmente bastante diferentes da decomposição estocástica regularizada. Ao invés de sortear cenários a partir de um conjunto infinito de eventos, as técnicas propostas assumem que o processo estocástico subjacente possui um número grande, porém finito de cenários, e estuda diferentes alternativas para construir cortes inexatos de modo que a imprecisão na solução seja controlada. Deste modo, os N cenários que definem os subproblemas são considerados fixos e, portanto, não são alterados ao longo das iterações. Em vez de escolher aleatoriamente os cenários para os quais os subproblemas serão resolvidos, a abordagem proposta neste trabalho seleciona os cenários a partir de algum critério de proximidade que permite agrupar cenários de maneira consistente às hipóteses de convergência dos métodos de feixes inexatos. Desde que tais hipóteses são fracas, muitas estratégias de agrupamentos são possíveis.

Em comparação aos trabalhos [16–18], as estratégias propostas nesta tese permitem um esforço computacional reduzido em todo o processo iterativo de resolução do problema (1.2), mesmo quando o número N é muito grande. No entanto, para certas classes de problemas, o que se pode garantir é a obtenção de soluções aproximadas de boa qualidade, ao invés de soluções exatas. Os tipos de aproximações consideradas neste trabalho são exemplificados nas Figuras 1.2(d) (sendo ε_g igual a zero em uma classe especial de problemas estudados na Seção 7.2) e 1.2(e) (que proporciona soluções assintoticamente exatas). As soluções aproximadas do problema (1.2) são obtidas sem a necessidade de resolver (pelos menos exatamente) todos os N subproblemas $f(z^k, \xi^i)$, para cada $z^k \in \mathcal{X}$.

1.2 Objetivo e Contribuições do Trabalho

O principal objetivo deste trabalho consiste em prover ferramentas matemáticas para a obtenção de soluções de boa qualidade, para os programas estocásticos convexos da forma (1.2), mesmo quando o processo estocástico subjacente é representado por uma quantidade consideravelmente grande de cenários. Com esta finalidade, são desenvolvidos oráculos inexatos que são empregados com alguns métodos inexatos de otimização convexa não diferenciável.

A abordagem mais empregada em programação estocástica consiste em obter uma representação razoavelmente pobre do processo estocástico (com poucos cenários), e aplicar métodos exatos de otimização para resolver o problema de otimização resultante. Neste trabalho, o enfoque é outro: são assumidas representações suficientemente boas do processo estocástico (considerando muitos cenários), e o problema de otimização resultante é resolvido de maneira aproximada. O principal foco é, deste modo, os métodos de otimização, e não os métodos de geração de cenários. Por este motivo, é suposto que os cenários que representam o processo estocástico sejam fornecidos por métodos de geração apropriados.

Mais especificamente, a área de atuação deste trabalho é a otimização convexa não diferenciável, onde são considerados tanto os aspectos teóricos, quanto os aspectos computacionais. Como pode ser verificado ao longo deste texto, o conteúdo tratado não está restrito a nenhum programa computacional específico, ou a alguma aplicação específica da programação estocástica. Os resultados numéricos utilizados para validar os métodos de otimização considerados são, em sua grande maioria, encontrados na literatura, e são bastante abrangentes, no sentido que envolvem diferentes áreas de aplicação, como por exemplo, a fabricação de produtos para atender uma demanda aleatória, ou o gerenciamento financeiro de uma carteira de investimentos, assumindo medidas de risco.

A seguir são evidenciadas as principais contribuições deste trabalho.

1. Aplicação do método de feixes proximal inexato, proposto em [8], à programação estocástica. O oráculo considerado difere um pouco do oráculo assumido em [8]. Por este motivo, é apresentada a análise de convergência do método, adaptada a tais exigências.
2. Aplicação do método de feixes proximal parcialmente inexato, proposto em [23], à programação estocástica. Ao contrário do experimento numérico apresentado em [23] para um problema de otimização inteira, o método de feixes proximal parcialmente inexato utilizando um oráculo específico para a programação linear estocástica em dois estágios apresenta resultados bastante satisfatórios.

3. Assumindo hipóteses adicionais, é mostrado que o método de feixes proximal parcialmente inexato tem terminação finita quando a função objetivo f e o conjunto viável \mathcal{X} são poliedrais.
4. Apresentação de uma abordagem inexata para o método de nível proximal desenvolvido em [9]. O método resultante é denominado método de nível proximal inexato, e é um método alternativo ao método de feixes proximal inexato.
5. Tendo verificado que o desempenho numérico do método de feixes proximal parcialmente inexato é fortemente dependente da qualidade do oráculo parcialmente inexato utilizado (este pode ter sido o motivo pelo qual Kiwiel [23] não obteve resultados satisfatórios com o método), é desenvolvido neste trabalho o método de nível parcialmente inexato. Este não é um método proximal, e dispensa o conceito de passos sérios e nulos. A análise detalhada da convergência do método é apresentada. Os resultados numéricos comprovam que o método de nível parcialmente inexato é menos susceptível a qualidade do oráculo, do que o método proposto em [23].
6. São propostas seis possíveis abordagens para os oráculos inexatos em programação estocástica. Para o caso linear, alguns destes oráculos são fundamentados em ideias relativamente simples, que analisam apenas o ângulo formado pelas inclinações das funções objetivo dos programas lineares definidos para cenários diferentes. Outros oráculos são um pouco mais sofisticados, e envolvem o conceito de métricas probabilísticas em espaços de funções mensuráveis.
7. É realizada uma análise numérica comparando os métodos e oráculos considerados, com os principais métodos utilizados em programação estocástica. Ao todo são considerados doze problemas numéricos, e vinte abordagens distintas, que são variantes do método de planos cortantes, do método de feixes proximal, e do método de nível.

Todas as abordagens inexatas e métodos apresentados tem um compromisso fiel entre a teoria e a prática. Todas as técnicas consideradas nesta tese são desenvolvidas e apresentadas prezando pelo formalismo matemático, e pela prática computacional.

Além das contribuições citadas acima, este trabalho apresenta uma revisão geral dos conceitos e resultados envolvidos em programação estocástica.

1.3 Organização do Trabalho

O presente trabalho está organizado como se segue: são desenvolvidos no Capítulo 2 alguns conceitos e resultados teóricos importantes na programação estocástica (com

recurso) em dois estágios e em multiestágios. Tais resultados não fazem parte das contribuições deste trabalho, e podem ser obtidos em [1, 11, 24].

É apresentado no Capítulo 3 o *método de feixes proximal inexato* - MFI, bem como o seu algoritmo e a análise de convergência. Este capítulo é fundamentado no trabalho de Kiwiel, [8].

Baseado em [23], é apresentado no Capítulo 4 o *método de feixes proximal parcialmente inexato* - MFPI. Para este método é assumido um oráculo parcialmente inexato, no sentido que, para determinados iterados $z^k \in \mathcal{X}$, o valor exato da função $f(z^k)$ e um subgradiente $g(z^k) \in \partial f(z^k)$ são disponibilizados. Diferentemente do MFI, o MFPI determina assintoticamente uma solução exata (quando existe) para o problema de otimização considerado.

Fundamentado no método de nível proximal desenvolvido por Kiwiel [9], é introduzido no Capítulo 5 o *método de nível proximal inexato* - MNI, que é uma alternativa ao MFI.

Assumindo um oráculo parcialmente inexato, é proposto no Capítulo 6 o *método de nível parcialmente inexato* - MNPI, bem como a sua análise de convergência. Assim como o MFPI, o MNPI determina assintoticamente uma solução ótima do problema de otimização considerado.

É considerada no Capítulo 7 a aplicação dos métodos de feixes inexatos (MFI e MNI), e dos métodos de feixes parcialmente inexatos (MFPI e MNPI), aos programas lineares estocásticos em dois estágios. São propostos dois oráculos: um inexato, e outro parcialmente inexato. É demonstrado que ambos estes oráculos satisfazem as hipóteses exigidas por cada método.

A aplicação dos métodos de feixes inexatos e parcialmente inexatos aos programas não lineares estocásticos convexos em dois estágios é considerada no Capítulo 8. Neste capítulo são propostos dois oráculos inexatos baseados em seleção de cenários, e um oráculo parcialmente inexato baseado na *desigualdade de Jensen*, para os programas estocásticos convexos.

Os programas lineares estocásticos em multiestágios são considerados no Capítulo 9. É realizada uma revisão da *decomposição aninhada* [25], e proposta uma abordagem dual para a aplicação dos métodos de feixes inexatos e parcialmente inexatos.

São apresentados no Capítulo 10 alguns problemas práticos de otimização estocástica, e são comparadas as suas soluções numéricas obtidas pelas técnicas propostas neste trabalho, e pelos principais métodos encontrados na literatura.

São reservadas ao Capítulo 11 as principais conclusões do trabalho.

Finalmente, com o objetivo de tornar o presente trabalho autocontido, estão incluídos no final do texto algumas definições e metodologias relevantes. O Apêndice A contém algumas definições e resultados de análise convexa, utilizados ao longo deste

trabalho. Para tratar a programação estocástica com o devido formalismo matemático se fazem necessários alguns conceitos importantes da teoria de probabilidades, também apresentados no Apêndice A. O Apêndice B contém uma exposição sucinta do *perfil da performance*, um método gráfico desenvolvido em [26] para avaliar e comparar o desempenho computacional de um método de otimização. O Apêndice C contém a descrição, a formulação, e os parâmetros de dois programas lineares estocásticos utilizados para validar numericamente os métodos propostos.

Capítulo 2

Programação Estocástica

Um *programa estocástico* é um problema de otimização onde a função objetivo e/ou o conjunto viável dependem de parâmetros incertos, porém, com uma distribuição de probabilidades (conjunta) conhecida, e independente da variável de decisão.

Inúmeros problemas da vida real são modelados como programas estocásticos. Dois exemplos clássicos são:

- o *planejamento de sistemas hidrotérmicos de geração de energia elétrica*, que possui como fonte de incertezas os preços dos combustíveis, a demanda de energia elétrica, e as afluências aos reservatórios de água das usinas hidroelétricas;
- o *gerenciamento financeiro de uma carteira de investimentos*, cuja fonte principal de incertezas são as taxas de retorno de cada ativo financeiro que compõe a carteira.

Este capítulo está dividido em três seções, segundo as particularidades dos problemas de otimização estocástica com recursos. A Seção 2.1 contém uma descrição das principais características de um programa estocástico. São considerados na Seção 2.2 os programas estocásticos em dois estágios. A extensão deste tipo de problemas a um número maior de estágios é denominada programação estocástica em multiestágios, e é considerada na Seção 2.3. O material deste capítulo segue as notações e conteúdos de [1, Capítulos 2 e 3].

2.1 Considerações Gerais

As características principais de um programa estocástico (com recurso) podem ser resumidas como apresentadas em [15]:

- (i) um estágio geralmente representa um (ou mais) período(s) de tempo;
- (ii) o primeiro estágio, em geral, não envolve incertezas;

- (iii) no início de cada estágio as incertezas são reveladas, sendo que os eventos incertos dos estágios seguintes são conhecidos em um sentido probabilístico;
- (iv) no primeiro estágio as decisões devem ser tomadas antes das realizações futuras;
- (v) na medida em que os eventos futuros vão sendo revelados, diretivas de correção (recursos) são retornadas ao primeiro estágio.

Em relação ao item (i), em geral considera-se como *período* a discretização intrínseca da modelagem do problema, enquanto *estágio* é a discretização considerada para o método de otimização. O primeiro estágio em programação estocástica geralmente corresponde ao tempo inicial, instante em que as decisões devem ser tomadas. Por isso, todos os parâmetros que definem o problema no primeiro estágio são observáveis (ou pelo menos estimados com um alto grau de acurácia), e deste modo são determinísticos. Estes comentários esclarecem os itens (ii) e (iii).

Os itens (iv) e (v) correspondem, respectivamente, aos programas estocásticos com recursos, [1, Capítulo 2].

Em [15] os métodos de programação estocástica são divididos em duas classes: aqueles que definem todo o conjunto de incertezas mediante uma árvore de cenários, denominados *métodos baseados em árvore*; e aqueles que consideram amostragem durante o processo de otimização, denominados *métodos baseados em amostragem*. Uma técnica comumente empregada com os métodos baseados em árvore é a *Sample Average Approximation* - SAA - apresentada em [1, Capítulo 5], que consiste em resolver o programa estocástico subjacente para diferentes árvores de cenários, de modo que se obtenha um intervalo de confiança para o valor ótimo do problema (1.1).

Em geral, os métodos de otimização para resolver programas lineares estocásticos se dividem nas duas categorias, sendo os métodos baseados em amostragem mais comuns no caso multiestágios. Com frequência os métodos desta classe consideram uma árvore de cenários muito grande, e deste modo, são realizadas amostragens dos cenários que serão percorridos, de modo que o problema (9.3) seja aproximado, [12, 13, 15, 27]. O presente trabalho pertence à categoria dos métodos baseados em árvore.

Para definir matematicamente um programa estocástico, sejam \mathcal{F} a σ -álgebra gerada pelo conjunto fechado Ξ , e ξ uma possível realização da variável aleatória $\boldsymbol{\xi} : (\Omega, \hat{\mathcal{F}}) \rightarrow (\Xi, \mathcal{F})$, definida em um espaço de probabilidades apropriado $(\Omega, \hat{\mathcal{F}}, \hat{P})$. A realização $\boldsymbol{\xi}(\omega) = \xi$ é chamada de evento aleatório, ou simplesmente, *cenário*. Definindo a medida de probabilidades em Ξ por $P(\mathcal{A}) := \hat{P}(\{\omega \in \Omega : \boldsymbol{\xi}(\omega) \in \mathcal{A}\})$, para todo subconjunto $\mathcal{A} \subset \Xi$, tem-se o *espaço de probabilidades* (Ξ, \mathcal{F}, P) induzido por $\boldsymbol{\xi}$.

Uma formulação amplamente utilizada em programação estocástica é a minimização do valor esperado das funções parametrizadas pelos cenários, $f : \mathcal{X} \times \Xi \rightarrow \mathcal{R}$, segundo a distribuição de probabilidades P . Procura-se então resolver o problema

$$\min_{x \in \mathcal{X}} \mathbb{E}[f(x, \xi)], \quad \text{com } \mathcal{X} \subset \mathcal{R}^n. \quad (2.1)$$

Assume-se a hipótese de que a função $f(\cdot, \xi)$ seja *quase certamente*¹ (q.c.) semicontínua inferiormente para garantir que o valor esperado $\mathbb{E}[f(x, \xi)]$ seja semicontínuo inferiormente, [1] (ver Definição 2.4). Deste modo, se $\text{dom } \mathbb{E}[f(x, \xi)] \cap \mathcal{X} \subset \mathcal{R}^n$ é não vazio, e \mathcal{X} é um conjunto compacto, o problema (2.1) possui ao menos uma solução ótima x^* .

A formulação do problema (2.1) é justificada pela *Lei dos Grandes Números*, e tem a seguinte interpretação: em média, a solução x^* é uma decisão ótima para qualquer possível realização $\xi \in \Xi$.

Existem também formulações que consideram somente eventos extremos, e resolvem o problema (2.1) considerando um único cenário $\tilde{\xi} \in \Xi$, e $f(x) = f(x, \tilde{\xi})$ (por exemplo, o pior caso). Uma deficiência evidente desta formulação é a obtenção de soluções excessivamente otimistas ou pessimistas, de acordo com o cenário $\tilde{\xi}$ escolhido.

Um compromisso com as possíveis variações dos valores $f(x, \xi)$, para $\xi \in \Xi$ e $x \in \mathcal{X}$ fixo, pode ser obtido considerando uma função não decrescente e convexa $\rho : \mathcal{L}_p(\Xi, \mathcal{F}, P) \rightarrow \mathcal{R}$, que determina uma medida de risco proporcionada pela decisão x . O espaço $\mathcal{L}_p(\Xi, \mathcal{F}, P)$ para $p \in [1, \infty)$ é determinado pelas funções \mathcal{F} -mensuráveis $\varphi : \Xi \rightarrow \mathcal{R}$, tais que $\mathbb{E}|\varphi(\xi)|^p < \infty$. Com este desenvolvimento, uma formulação *avessa ao risco* para o problema (2.1) é dada por

$$\min_{x \in \mathcal{X}} \rho(\mathbb{E}[f(x, \xi)]).$$

Um estudo detalhado acerca das medidas de risco ρ é apresentado em [1, Capítulo 6]. Desde que ρ seja convexa, a metodologia desenvolvida neste trabalho pode ser aplicada ao problema acima.

A seguir é considerada a programação estocástica em dois estágios.

2.2 Programação Estocástica em Dois Estágios

Seja $\mathcal{X} \subset \mathcal{R}^n$ um conjunto não vazio e independente dos parâmetros incertos, denominado conjunto viável de primeiro estágio. Um programa estocástico em dois estágios pode ser genericamente representado por (2.1), com a função $f(\cdot, \xi)$ dada

¹Em termos probabilísticos, a expressão “quase certamente” significa que um resultado vale a menos de um conjunto com probabilidade zero.

por

$$f(x, \xi) := f_1(x) + \inf_{y \in \mathcal{X}(x, \xi)} f_2(y, \xi), \quad (2.2)$$

onde $\mathcal{X}(x, \xi) \subset \mathcal{R}^{n_2}$ é o conjunto viável de segundo estágio determinado pela multifunção $\mathcal{X} : \mathcal{R}^n \times \Xi \rightrightarrows \mathcal{R}^{n_2}$. A variável x representa o vetor de decisões que devem ser tomadas antes da realização do evento incerto ξ . Este primeiro período, em que não existem incertezas, é denominado *primeiro estágio*. As funções $f_1 : \mathcal{R}^n \rightarrow \mathcal{R}$ e $f_2 : \mathcal{R}^{n_2} \times \Xi \rightarrow \mathcal{R}$ são denominadas funções de primeiro e segundo estágio, respectivamente.

Segundo [28], “os modelos em dois estágios permitem que se faça uma escolha inicial (dita de primeiro estágio) antes de se conhecer o valor dos parâmetros incertos. Após o conhecimento dos valores dos mesmos, o agente de decisão faz novas escolhas (ditas de segundo estágio) que visam corrigir possíveis efeitos negativos gerados pela decisão de primeiro estágio (por este motivo, as decisões de segundo estágio também são chamadas de ações corretivas)”.

O problema (2.1)-(2.2) é dito ser um *programa linear estocástico em dois estágios* se \mathcal{X} é um conjunto poliedral,

$$f_1(x) := c^\top x, \quad f_2(y, \xi) := q^\top y, \text{ e}$$

$$\mathcal{X}(x, \xi) := \{y \in \mathcal{R}_+^{n_2} : Tx + Wy = h\},$$

onde os vetores c, q, h e as matrizes T, W possuem dimensões compatíveis, e compõem o cenário² $\xi := (q, h, T, W)$ do problema de segundo estágio.

Definição 2.1 *O problema (2.1)-(2.2) é dito ser um programa com recurso fixo se a matriz de recurso W não é aleatória.*

□

Definição 2.2 *O problema (2.1)-(2.2) é denominado um programa com recurso relativamente completo se o conjunto viável para as variáveis de segundo estágio definido pela multifunção $\mathcal{X}(x, \xi)$ é não vazio para todo $x \in \mathcal{X}$ e para q.c. todo $\xi \in \Xi$. Além disso, se as propriedades valem para todo $x \in \mathcal{R}^n$, diz-se que (2.1)-(2.2) é um programa com recurso completo.*

□

Uma condição importante que deve ser satisfeita pelos problemas gerais de minimização (em otimização contínua) é a semicontinuidade inferior da função objetivo.

²Com esta notação, ξ é um evento da variável aleatória $\xi(\omega) := (q(\omega), h(\omega), T(\omega), W(\omega))$.

Definição 2.3 Uma função $f : \mathcal{X} \rightarrow \mathcal{R}$ é dita ser *semicontínua inferiormente* (*sci*) se $f^{-1}((\alpha, \infty]) := \{x \in \mathcal{X} : f(x) > \alpha\}$ é um conjunto aberto em \mathcal{X} , para todo $\alpha \in \mathcal{R}$.

□

Em programação estocástica, a noção de semicontinuidade inferior é estendida de modo que a aleatoriedade do problema seja incorporada.

Definição 2.4 Uma função $f : \mathcal{X} \times \Xi \rightarrow \mathcal{R}$ é dita ser *aleatória semicontínua inferiormente* (*asci*) se valem concomitantemente as três relações seguintes:

- (i) a função $f(\cdot, \xi)$ é *q.c. semicontínua inferiormente*;
- (ii) a multifunção $\xi \rightarrow \text{dom } f(\cdot, \xi)$ é *mensurável*;
- (iii) a função $f(x, \cdot)$ é *mensurável* para cada $x \in \mathcal{X}$ fixo.

□

Se para $\xi \in \Xi$, um evento fixo, a função $f(\cdot, \xi)$ é semicontínua inferiormente, própria e convexa no conjunto convexo \mathcal{X} , então $f(\cdot, \xi)$ é uma função fechada, e o conjunto de nível $L_{\mathcal{X}}(\alpha) := \{x \in \mathcal{X} : f(x, \xi) \leq \alpha\}$ é convexo e fechado, [29, p. 263]. Se ξ é qualquer evento em um conjunto $\mathcal{A} \subset \Xi$ tal que $P(\mathcal{A}) = 1$, então $f(\cdot, \xi)$ é *q.c. semicontínua inferiormente*. Além disso, se o conjunto viável para as variáveis de segundo estágio definido pela multifunção $\mathcal{X}(\cdot, \xi)$ é compacto e não vazio, então a condição (i) juntamente com a hipótese de que $f(\cdot, \xi)$ é própria e convexa garantem a existência (exceto, possivelmente, para um conjunto com probabilidade nula em Ξ) de ao menos uma solução para o problema $\min_{x \in \mathcal{X}} f(x, \xi)$. Os itens (ii) e (iii) da Definição 2.4 asseguram que o conjunto viável e a função $f(x, \cdot)$ estão bem definidos no espaço de probabilidades (Ξ, \mathcal{F}, P) .

Segue das considerações acima a seguinte representação para o problema (2.1)-(2.2)

$$\min_{x \in \mathcal{X}} f(x) \quad \text{com} \quad f(x) = f_1(x) + \mathcal{Q}(x), \quad (2.3a)$$

e $\mathcal{Q}(x) := \mathbb{E}[Q(x, \xi)]$, onde a função de recurso é a solução ótima do problema de otimização

$$Q(x, \xi) := \inf_{y(\xi) \in \mathcal{X}(x, \xi)} f_2(y(\xi), \xi) \quad (2.3b)$$

Como resultado desta formulação, a variável de segundo estágio descreve as decisões como funções de x e dos eventos incertos $\xi \in \Xi$, e não como simples vetores em \mathcal{R}^{n_2} . Assim sendo, a função $\xi \mapsto y(\xi)$ é assumida pertencer a um espaço de funções apropriado. Quando a variável aleatória ξ possui seus $r \geq 1$ momentos finitos, e $y(\xi)$ é uma função contínua em Ξ , então $y(\xi) \in \mathcal{L}_r(\Xi, \mathcal{F}, P; \mathcal{R}^{n_2})$, [30, Seção 1]. Por este motivo, quando o conjunto Ξ possui infinitos eventos ξ , a formulação (2.3) corresponde a um problema de otimização de dimensão infinita.

Entretanto, em algumas situações, a função de recurso (média)

$$\mathcal{Q}(x) = \mathbb{E}[Q(x, \xi)] := \int_{\Xi} Q(x, \xi) dP(\xi)$$

possui uma representação explícita, compacta, e até mesmo diferenciável (ver Exemplo 2.1 a seguir). Nesta classe de problemas, a variável de decisão $(x, y) \in \mathcal{R}^{n \times n_2}$ tem dimensão finita, e o problema (2.3) limita-se aos problemas de otimização linear ou não linear, podendo então ser resolvidos por métodos especializados. É importante destacar que uma representação explícita para o valor esperado $\mathbb{E}[Q(x, \xi)]$ é, na maioria das aplicações, impossível de se obter. Nesta situação, o problema (2.3) é intratável computacionalmente devido à exigência de calcular a integral (geralmente multidimensional) correspondente ao valor esperado. Com a intenção de tornar os cálculos manejáveis, discretiza-se o espaço amostral Ξ em um número finito de eventos $\{\xi^1, \dots, \xi^N\}$. Este processo é chamado de *geração de cenários*. Para detalhes sobre este tema, recomendam-se as referências [11, 31, 32] e [33].

É importante ressaltar que se o conjunto suporte Ξ possui infinitos elementos, a hipótese de recurso relativamente completo não é suficiente para garantir a viabilidade do programa estocástico quando são considerados quaisquer eventos finitos $\{\xi^1, \dots, \xi^N\} \subset \Xi$ para representar $f(\cdot) = f_1(\cdot) + \mathcal{Q}(\cdot)$. Afim de verificar esta afirmação, suponha que para todo conjunto $\mathcal{A} \subsetneq \Xi$ tal que $P(\mathcal{A}) = 1$, a condição $\mathcal{X}(x, \xi) \neq \emptyset$ para todo $\xi \in \mathcal{A}$ e $x \in \mathcal{X}$ é válida, i.e., a hipótese de recurso relativamente completo é verificada. Suponha também que é realizada uma discretização finita do espaço amostral (Ξ, \mathcal{F}) , e um cenário $\tilde{\xi} \in \Xi \cap \mathcal{A}^c$ é obtido. Neste sentido, sem hipóteses adicionais não há garantias de que o conjunto $\mathcal{X}(x, \tilde{\xi})$ seja não vazio para ao menos algum $x \in \mathcal{X}$. Se $\mathcal{X}(x, \tilde{\xi})$ é vazio para todo $x \in \mathcal{X}$, então $Q(x, \tilde{\xi}) = \infty$. Como a probabilidade pontual $p(\tilde{\xi})$ é positiva (devido a discretização finita), tem-se que $\mathcal{Q}(x) = \sum_{i=1}^N p(\xi^i) Q(x, \xi^i) = \infty$ para todo $x \in \mathcal{X}$. Em outras palavras, a condição de recurso relativamente completo não assegura que o conjunto $\bigcap_{\xi \in \Xi} \mathcal{X}(x, \xi)$ seja não vazio para ao menos um ponto $x \in \mathcal{X}$. O Exemplo 2.1 ilustra esta situação.

Exemplo 2.1 *Seja o seguinte programa linear estocástico em dois estágios*

$$\min_{0 \leq x \leq 1/2} f(x) \quad \text{com} \quad f(x) = -x + \mathbb{E}[Q(x, \xi)], \quad (2.4)$$

sendo que $\xi \in \Xi := [0, 1]$ tem função de densidade dada por $g(\xi) = 2\xi$, e

$$Q(x, \xi) := \min y \quad \text{s.a.} \quad y\xi = 1 - x, \quad y \geq 0.$$

O problema (2.4) tem recurso relativamente completo, e o recurso dado por $\xi = W \in [0, 1]$ é aleatório. É importante notar que para $\xi = 0$, o conjunto viável do programa

de segundo estágio é vazio, e para cada $x \in [0, 1/2]$ tem-se que

$$Q(x, \xi) := \begin{cases} (1-x)/\xi & \text{se } \xi \in (0, 1], \\ \infty & \text{se } \xi = 0. \end{cases}$$

Apesar de o problema $Q(\cdot, \xi)$ ser inviável para $\xi = 0$, o valor esperado $\mathbb{E}[Q(x, \xi)]$ está bem definido, e é dado por

$$\mathbb{E}[Q(x, \xi)] := \int_0^1 Q(x, \xi)g(\xi)d\xi = \int_0^1 \left(\frac{1-x}{\xi}\right) 2\xi d\xi = 2(1-x).$$

Assim sendo, o problema de otimização considerado se resume a

$$\min_{0 \leq x \leq 1/2} f(x) \quad \text{com} \quad f(x) = 2 - 3x,$$

com solução e valor ótimo iguais a $1/2$. No entanto, para qualquer discretização de Ξ em N cenários tal que o elemento zero seja um evento com probabilidade não nula (i.e., $\xi^i = 0$ para algum $i \in \{1, \dots, N\}$, e $p(\xi_i) > 0$), o problema (2.4) é inviável.

□

Felizmente, existem condições que asseguram a viabilidade do problema (2.3) utilizando qualquer discretização (finita ou não) do conjunto suporte Ξ .

A Proposição 2.1 a seguir, demonstrada em [11, Teorema 3], fornece condições para que o problema (2.3) seja viável, independentemente da discretização de Ξ .

Proposição 2.1 *Suponha que o problema (2.3) tem recurso fixo: W é uma matriz determinística. Sejam os conjuntos*

$$K_2 = \{x : \mathcal{Q}(x) < \infty\} \quad \text{e} \quad K_2^P = \{x : \cap_{\xi \in \Xi} \mathcal{X}(x, \xi) \neq \emptyset\},$$

onde $\mathcal{X}(x, \xi) := \{y \in \mathcal{R}_+^{n_2} : Tx + Wy = h\}$. Se a variável aleatória ξ que define o cenário $\xi = (q, h, T)$ tem variância finita, então

$$K_2 = K_2^P.$$

□

É importante notar que a hipótese de recurso relativamente completo garante que o conjunto $\mathcal{X} \cap K_2$ seja não vazio. No entanto, para garantir a viabilidade do problema (2.3) para toda discretização (finita ou não) do conjunto suporte Ξ , é necessário assumir que $\mathcal{X} \cap K_2^P \neq \emptyset$.

Quando o problema (2.3b) é um programa linear, a Proposição 2.2 a seguir estabelece condições suficientes para que a função $\mathcal{Q}(\cdot)$ seja bem definida.

Proposição 2.2 *Considere o programa linear definido pelo problema (2.3b), com função $f_2(\cdot, \xi)$ afim. Além disso, suponha que:*

- (i) o problema (2.3) tenha recurso fixo;
- (ii) a variável ξ que define $\xi = (q, h, T) \in \Xi$ possui variância finita;
- (iii) o conjunto de segundo estágio $\mathcal{X}(x, \xi)$ é q.c. não vazio para todo $x \in \mathcal{X}$.

Então a função $Q(x) = \mathbb{E}[Q(x, \xi)]$ é bem definida e $Q(x) > -\infty$ para todo $x \in \mathcal{R}^n$. Adicionalmente, Q é convexa, sci e Lipschitz contínua no conjunto³ convexo e fechado:

$$\text{dom } Q := \{x \in \mathcal{R}^n : h - Tx \in \text{pos } W, \text{ c.p.1}\}.$$

Prova. Dada a condição (ii), valem as seguintes desigualdades, [1]

$$\mathbb{E}[\|q\| \|h\|] < \infty \quad \text{e} \quad \mathbb{E}[\|q\| \|T\|] < \infty.$$

Portanto, o resultado segue da Proposição 2.7 em [1]. ■

A seguir são assumidas as seguintes hipóteses acerca do problema (2.1):

- h1.** o conjunto de primeiro estágio \mathcal{X} é não vazio, convexo e compacto.
- h2.** para cada $\xi = (h, T) \in \Xi$ as imagens da multifunção $\mathcal{X}(\cdot, \xi)$ definem conjuntos compactos não vazios dados por $\mathcal{X}(x, \xi) := \{y \in \mathcal{R}_+^{n_2} : Tx + Wy = h\}$;
- h3.** as funções parametrizadas $f(\cdot, \xi)$ são próprias e convexas em \mathcal{X} ;
- h4.** a função $f : \mathcal{X} \times \Xi \rightarrow \mathcal{R}$ é asci, i.e. $f(\cdot, \xi)$ é q.c. sci;
- h5.** o problema (2.3) possui recurso relativamente completo.

As hipóteses h1-h5 permitem definir as condições de otimalidade para os programas estocásticos em dois estágios, apresentadas a seguir.

2.2.1 Condições de Otimalidade

As hipóteses h1-h5 asseguram que $\mathcal{X} \cap K_2 \neq \emptyset$ é compacto, e deste modo, garantem a existência de uma solução ótima para o problema (2.3).

Dado $u \in \mathcal{R}^{m_2}$, a função Lagrangiana do subproblema de segundo estágio $Q(x, \cdot)$ definido em (2.3b) é dada por

$$\mathcal{L}(y, u; x, \xi) := f_2(y, \xi) + u^\top (Tx + Wy - h).$$

³Define-se o conjunto $\text{pos } W$ por $\{\chi : \chi = Wy, y \geq 0\}$.

Sob as hipóteses h2 e h3, o problema (2.3b) tem uma solução ótima $y^*(\xi)$. Dada h5, pelo [29, Teorema 28.2 e Corolário 28.2.2] existe um multiplicador de Lagrange ótimo $u^*(\xi) := \arg \sup_u \mathcal{L}(y^*(\xi), u; x, \xi)$, e não há brecha de dualidade [29, Teorema 28.3], i.e.,

$$\inf_{y \geq 0} \sup_u \mathcal{L}(y, u; x, \xi) = \sup_u \inf_{y \geq 0} \mathcal{L}(y, u; x, \xi). \quad (2.5)$$

Deste modo,

$$\begin{aligned} Q(x, \xi) &= \sup_u \inf_{y \geq 0} \mathcal{L}(y, u; x, \xi) \\ &= \sup_u \{u^\top(Tx - h) + \inf_{y \geq 0} [f_2(y, \xi) + u^\top W y]\} \\ &= \sup_u \{u^\top(h - Tx) + \sup_{y \geq 0} [f_2(y, \xi) - u^\top W y]\} \\ &= \sup_u \{u^\top(h - Tx) - \theta(u)\}, \end{aligned}$$

onde $\theta : \mathcal{R}^{m_2} \rightarrow \mathcal{R} \cup \{-\infty, \infty\}$ é a função dual estendida dada por

$$\theta(u) := - \inf_{y \geq 0} \{f_2(y, \xi) - u^\top W y\}. \quad (2.6)$$

Para verificar a convexidade da função $Q(\cdot, \xi)$ faz-se uso da Proposição 2.3 a seguir.

Proposição 2.3 *Dadas as hipóteses h1-h3, a função $Q(\cdot, \xi) : \mathcal{X} \rightarrow \mathcal{R}$ é convexa para todo $\xi \in \Xi$.*

Prova. Sejam x e z em \mathcal{X} e $\lambda \in [0, 1]$. Então, por h1, o ponto $x_\lambda := \lambda x + (1 - \lambda)z$ pertence ao conjunto \mathcal{X} , e segue de h2 e h3 a relação

$$\begin{aligned} Q(x_\lambda, \xi) &= \sup_u \{u^\top(h - T x_\lambda) - \theta(u)\} \\ &\leq \sup_u \{u^\top(\lambda(h - T x) + (1 - \lambda)(h - T z)) - \theta(u)\} \\ &\leq \lambda \sup_u \{u^\top(h - T x) - \theta(u)\} + (1 - \lambda) \sup_u \{u^\top(h - T z) - \theta(u)\} \\ &= \lambda Q(x, \xi) + (1 - \lambda) Q(z, \xi), \end{aligned}$$

i.e., $Q(\cdot, \xi)$ é uma função convexa. ■

A seguir verifica-se a convexidade de $Q(x, \cdot)$, para $x \in \mathcal{X}$ um ponto dado.

Proposição 2.4 *Dadas as hipóteses h1-h3, suponha adicionalmente que a função de segundo estágio $f_2 : \mathcal{R}^{n_2} \times \Xi \rightarrow \mathcal{R}$ independe dos cenários, i.e., $f_2(y, \xi) = f_2(y)$. Então a função $Q(x, \cdot) : \Xi \rightarrow \mathcal{R}$ é convexa para cada $x \in \mathcal{X}$ fixo.*

Prova. Desde que a função f_2 e a matriz W são determinísticas, a função $\theta(\cdot)$ definida (2.6) é independente do cenário $\xi \in \Xi$. Deste modo, tomando

$$\xi_\lambda := \lambda \hat{\xi} + (1 - \lambda)\check{\xi}, \quad \text{para } \lambda \in [0, 1] \text{ e } \hat{\xi}, \check{\xi} \in \Xi,$$

o resultado enunciado é obtido de modo análogo à demonstração da Proposição 2.4. ■

Como mencionado, existe um multiplicador de Lagrange $u^*(\xi)$ ótimo tal que $Q(x, \xi) = \sup_u \{u^\top(h - Tx) - \theta(u)\} = \{u^*(\xi)^\top(h - Tx) - \theta(u^*(\xi))\}$. Assim sendo, a derivada direcional de $Q(x, \xi)$ em relação a x é $-T^\top u^*(\xi)$, demonstrando deste modo, o resultado da Proposição 2.5 a seguir.

Proposição 2.5 *Seja $u^*(\xi)$ uma solução do problema dual $\sup_u \{u^\top(h - Tx) - \theta(u)\}$, para a função $\theta(\cdot)$ dada em (2.6). Então,*

$$-T^\top u^*(\xi) \in \partial Q(x, \xi).$$

□

Supondo que $\mathcal{X} \cap K_2^P \neq \emptyset$, considera-se a seguir um número finito N de cenários $\xi^i \in \Xi$. Seja $P = (p_1, \dots, p_N)$ uma medida de probabilidades discreta. Então, pelas representações (2.1), e (2.3b), pode-se escrever a função objetivo do problema (2.3) por

$$f^N(x) := f_1(x) + \sum_{i=1}^N p_i Q(x, \xi^i).$$

Pelo teorema de Moreau-Rockafellar [1, Teorema 7.4],

$$\partial f^N(x) \supseteq \partial f_1(x) + \sum_{i=1}^N p_i \partial Q(x, \xi^i), \text{ para todo } x \in \text{dom } f^N.$$

Então, se $g_1 \in \partial f_1(x)$ tem-se pela Proposição 2.5 que $g_1 - \sum_{i=1}^N p_i T^\top u^*(\xi^i) \in \partial f(x)$.

Pelo Teorema 3.4.13 de [34], o ponto $x^* \in \mathcal{X}$ é uma solução ótima do problema (2.3) se, e somente se, existe um ponto $p_{\mathcal{X}}$ no cone normal $\mathcal{N}_{\mathcal{X}}(x^*)$ tal que

$$g_1 - \sum_{i=1}^N p_i T^\top u^*(\xi^i) - p_{\mathcal{X}} = 0. \quad (2.7)$$

As hipóteses h1, h4 e h5 resultam que $\text{dom } f$ tem interior relativo não vazio, e a relação (2.7) pode ser obtida mesmo quando a distribuição P é contínua, i.e., quando infinitos cenários $\xi \in \Xi$ são considerados (ver [1, Teorema 7.47]).

A seguir são considerados os problemas de otimização estocástica em multiestágios.

2.3 Programação Estocástica em Multiestágios

Um programa estocástico em multiestágios é um problema de otimização em que as incertezas dos parâmetros vão se revelando ao longo de T estágios de tempo

($T > 2$). As variáveis de decisão são temporalmente dependentes, mas independentes da realização de eventos futuros, [1].

Sejam $t \in \{1, \dots, T\}$ um índice de tempo correspondente ao horizonte de T estágios, (Ξ, \mathcal{F}, P) o espaço de probabilidades, com $\xi := (\xi_1, \dots, \xi_T) \in \Xi$ um cenário do processo estocástico ξ , e \mathcal{F} é a σ -álgebra gerada por $\Xi := \Xi_1 \times \dots \times \Xi_T$.

Para $t = 1, \dots, T$ e cada $\xi \in \Xi$, seja $x_t(\xi) \in \mathcal{R}^{n_t}$ uma função de ξ . Utiliza-se a notação $\xi_{[t]} := (\xi_1, \dots, \xi_t)$ para representar a trajetória do cenário ξ até o estágio t ; definindo $\eta_t := \sum_{i=1}^t n_i$, resulta que a *estratégia* (também chamada de política) $x(\xi_{[t]}) := (x_1(\xi), \dots, x_t(\xi)) \in \mathcal{R}^{\eta_t}$ é uma função dos cenários $\xi \in \Xi$. Para $t = 2, \dots, T$, seja $\mathcal{X}_t : \mathcal{R}^{\eta_{t-1}} \times \Xi_{[t]} \rightrightarrows \mathcal{R}^{n_t}$ a multifunção que define o conjunto viável para o estágio t . A estratégia $x_t(\xi)$ é dita ser \mathcal{F} -mensurável (ou simplesmente mensurável) quando $x(\xi_{[t]}) \in \mathcal{X}_t(x_{t-1}, \xi_{[t]})$. Com este desenvolvimento, um programa estocástico em multiestágios pode ser escrito usando a *formulação estática* a seguir:

$$\begin{cases} \inf_{x_1, x_2(\xi), \dots, x_T(\xi)} & \mathbb{E}[f_1(x_1) + f_2(x_2(\xi), \xi_2) + \dots + f_T(x_T(\xi), \xi_T)] \\ \text{s.a} & x_1 \in \mathcal{X}_1, x_t(\xi) \in \mathcal{X}_t(x_{t-1}, \xi_{[t]}), \text{ para } t = 2, \dots, T. \end{cases} \quad (2.8)$$

Utilizando a notação $\mathbb{E}_{|\xi_{[t]}}[\cdot]$ para representar o valor esperado (com relação a probabilidade P) condicionado à trajetória (ou passado) $\xi_{[t]}$, uma maneira alternativa de se representar um programa estocástico em multiestágios emprega a *formulação aninhada*, [1, Capítulo 3]:

$$\min_{x_1 \in \mathcal{X}_1} \mathbb{E}[f(x_1, \xi)], \quad (2.9a)$$

com

$$\mathbb{E}[f(x_1, \xi)] := f_1(x_1) + \mathbb{E}_{|\xi_{[1]}} \left[\inf_{x_2 \in \mathcal{X}_2(x_1, \xi_{[2]})} f_2(x_2, \xi_2) + \mathbb{E}_{|\xi_{[2]}} \left[\inf_{x_3 \in \mathcal{X}_3(x_2, \xi_{[3]})} f_3(x_3, \xi_3) + \dots + \mathbb{E}_{|\xi_{[T-1]}} \left[\inf_{x_T \in \mathcal{X}_T(x_{T-1}, \xi_{[T]})} f_T(x_T, \xi_T) \right] \dots \right] \right]. \quad (2.9b)$$

A função $f(\cdot, \xi)$ assim definida é uma extensão do caso em dois estágios. Se a distribuição de probabilidades é temporalmente independente, o valor esperado $\mathbb{E}_{|\xi_{[t]}}[\cdot]$ pode ser substituído por $\mathbb{E}_{P_t}[\cdot]$, onde P_t é a distribuição de probabilidades definida no *espaço amostral* (Ξ_t, \mathcal{F}_t) . Esta formulação é computacionalmente interessante, por induzir nitidamente uma decomposição por estágios.

Finalmente, também pode-se representar o problema (2.9) com uma *formulação dinâmica*:

$$\min_{x_1 \in \mathcal{X}_1} f(x_1), \quad \text{com } f(x_1) = f_1(x_1) + \mathcal{Q}_2(x_1, \xi_1), \quad (2.10a)$$

onde a função \mathcal{Q}_2 é definida por recorrência. Mais precisamente, para $t = 1, \dots, T-1$, as funções de recurso são definidas por

$$\mathcal{Q}_{t+1}(x_t, \xi_{[t+1]}) := \inf_{x_{t+1} \in \mathcal{X}_{t+1}(x_t, \xi_{[t+1]})} \{f_{t+1}(x_{t+1}, \xi_{t+1}) + \mathcal{Q}_{t+2}(x_{t+1}, \xi_{[t+1]})\}, \quad (2.10b)$$

a partir das quais são definidas as funções

$$\mathcal{Q}_{t+1}(x_t, \xi_{[t]}) := \begin{cases} \mathbb{E}_{|\xi_{[t]}}[\mathcal{Q}_{t+1}(x_t, \xi_{[t+1]})] & \text{se } t \in \{1, \dots, T-1\} \\ 0 & \text{se } t = T. \end{cases} \quad (2.10c)$$

Em [33, Capítulo 2] são estabelecidas algumas condições que devem ser satisfeitas por um programa estocástico de modo que as formulações (2.10) e (2.8) sejam equivalentes em termos de solução e valor ótimo, [33, Teorema 2.6]. Tais condições são geralmente verificadas em problemas práticos, e se referem, por exemplo, à compacidade do conjunto suporte Ξ , à hipótese de recurso relativamente completo, e à exigência de que a função $f : \mathcal{X} \times \Xi \rightarrow$ seja *asci*.

Novamente, se a distribuição de probabilidades condicionada é temporalmente independente, pode-se escrever $\mathcal{Q}_{t+1}(x_t, \xi_{[t]})$ como $\mathcal{Q}_{t+1}(x_t)$.

Exemplo 2.2 (*Independência temporal.*)

Para $t = 2, \dots, T$, suponha que a variável aleatória ξ_t represente o resultado do lançamento de uma moeda. Suponha que:

$$\xi_t(\omega) = \xi_t = \begin{cases} 1 & \text{se o resultado do lançamento é cara, com probabilidade } p \\ 0 & \text{se o resultado do lançamento é coroa, com probabilidade } 1 - p. \end{cases}$$

Portanto, no instante de tempo t existem apenas 2^{t-1} trajetórias diferentes de resultados de lançamentos $\xi_{[t]}$. No entanto, os possíveis eventos da variável ξ_{t+1} não se alteram, e são, portanto, 0 e 1. Logo, $P(\xi_{t+1}(\omega) = 1 | \xi_{[t]} = \xi_{[t]}) = p$ e $P(\xi_{t+1}(\omega) = 0 | \xi_{[t]} = \xi_{[t]}) = 1 - p$, para cada uma das 2^{t-1} trajetória $\xi_{[t]}$, i.e., a distribuição de probabilidades condicionada é temporalmente independente.

Um programa estocástico em multiestágios é dito ser linear quando

$$f_1(x_1) := c^\top x_1, \quad \mathcal{X}_1 := \{x_1 \in \mathcal{R}_+^n : A_1 x_1 = b_1\}, \quad f_t(x_t, \xi_t) := c_t^\top x_t, \text{ e}$$

$$\mathcal{X}_t(x_{t-1}, \xi_t) := \{x_t \in \mathcal{R}_+^{n_t} : B_t x_{t-1} + A_t x_t = b_t\}, \text{ para } t = 2, \dots, T,$$

onde os vetores c_t , b_t , e as matrizes A_t , B_t (com dimensões compatíveis) definem o cenário $\xi_t := (c_t, A_t, B_t, b_t) \in \Xi_t$ da variável aleatória $\xi_t(\omega) := (c_t(\omega), A_t(\omega), B_t(\omega), b_t(\omega))$, com $\xi_1 = (c_1, A_1, b_1)$ determinístico. O seguinte exemplo ilustra um problema de otimização linear estocástica em multiestágios.

Exemplo 2.3 Considere o seguinte programa em três estágios:

$$\left\{ \begin{array}{l} \min_{x,r} \quad \mathbb{E} \left[\sum_{t=1}^3 \xi_t x_t \right] \\ \text{s.a} \quad (x_t, r_t) \in \mathcal{R}_+^2 \\ \quad (x_t, r_t) \text{ é } \mathcal{F}_t \text{- mensurável} \\ \quad r_t - r_{t-1} = x_t, \quad t = 2, 3 \\ \quad r_1 = 0, r_3 = 5, \end{array} \right. \quad (2.11)$$

onde $\Xi := \{\xi^1, \xi^2, \xi^3, \xi^4\}$, e os cenários (eventos) ξ^i , $i = 1, \dots, 4$ são equiprováveis ($p_i = 1/4$) e representados pela árvore de cenários da Figura 2.1. Cada valor c_i é

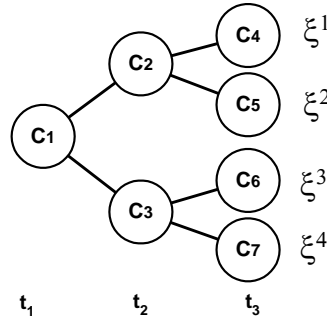


Figura 2.1: Árvore de cenários.

chamado de nó da árvore. Todo cenário ξ representa a trajetória do evento incerto desde o nó raiz c_1 até as “folhas da árvore”, representadas pelos nós c_j , com $j = 4, \dots, 7$. Deste modo, o cenário 2 é representado por $\xi^2 := \{\xi_1^2 = c_1, \xi_2^2 = c_2, \xi_3^2 = c_5\}$. Dada a árvore de cenários da Figura 2.1, a σ -álgebra \mathcal{F} gerada pelo conjunto Ξ é composta pelas partes de Ξ , i.e., $\mathcal{F} := \mathcal{P}(\Xi)$. Para $t = 1$ e $t = 2$, as σ -álgebras são dadas, respectivamente, por $\mathcal{F}_1 := \{\emptyset, \Xi\}$ e $\mathcal{F}_2 := \{\emptyset, \{c_4, c_5\}, \{c_6, c_7\}, \Xi\}$. A sequência $\mathcal{F}_1 \subset \mathcal{F}_2 \subset \mathcal{F}$ é chamada de filtração da σ -álgebra \mathcal{F} .

Desconsiderando as restrições de mensurabilidade do problema (2.11), e resolvendo-o para cada cenário ξ , têm-se 4 soluções para cada estágio:

- estágio 1 proporciona $x_1(\xi^1), x_1(\xi^2), x_1(\xi^3)$ e $x_1(\xi^4)$;
- estágio 2 proporciona $x_2(\xi^1), x_2(\xi^2), x_2(\xi^3)$ e $x_2(\xi^4)$;
- estágio 3 proporciona $x_3(\xi^1), x_3(\xi^2), x_3(\xi^3)$ e $x_3(\xi^4)$.

No entanto, como ilustrado na Figura 2.1, a solução do nó c_4 está relacionada com a solução do nó c_1 , e é diretamente dependente da solução do nó c_2 . Analogamente, o nó c_7 depende dos nós c_1 e c_3 . Deste modo, as soluções $x_1(\xi^i)$ devem ser idênticas para $i = 1, \dots, 4$. Quando este resultado é verificado, diz-se que a estratégia $x_1(\xi^1)$

é \mathcal{F}_1 -mensurável. Mais especificamente, para qualquer $r \in \mathcal{R}$,

$$x_1^{-1}(r) = \begin{cases} \emptyset, & \text{se } x_1(\xi^i) \neq r, \text{ para } i = 1, \dots, 4; \\ \Xi, & \text{se } x_1(\xi^i) = r, \text{ para } i = 1, \dots, 4. \end{cases}$$

Deste modo, qualquer que seja $r \in \mathcal{R}$, tem-se que $x_1^{-1}(r) \in \mathcal{F}_1$. Note que se tivesse $x_1(\xi^1) = r$ e $x_1(\xi^i) \neq r$ para $i = 2, 3, 4$, a variável x_1 não seria \mathcal{F}_1 -mensurável, pois $\{\xi^1\} \notin \mathcal{F}_1$ e $\{\xi^2, \xi^3, \xi^4\} \notin \mathcal{F}_1$. A estratégia $x(\xi)$ é \mathcal{F} -mensurável quando $x(\xi_t)$ é \mathcal{F}_t -mensurável para $t = 1, \dots, T$.

Assim sendo, a formulação do problema (2.11) considerando explicitamente as restrições de mensurabilidade é, para $x_t^i := x(\xi_t^i)$:

$$\left\{ \begin{array}{l} \min_{x,r} \left\{ \begin{array}{l} (c_1x_1^1 + c_2x_2^1 + c_4x_3^1)/4 + (c_1x_1^2 + c_2x_2^2 + c_5x_3^2)/4 \\ (c_1x_1^3 + c_3x_2^3 + c_6x_3^3)/4 + (c_1x_1^4 + c_3x_2^4 + c_7x_3^4)/4 \end{array} \right. + \end{array} \right. \\ \left. \begin{array}{l} \text{s.a} \quad (x_t^i, r_t^i) \in \mathcal{R}_+, t = 1, 2, 3 \text{ e } i = 1, \dots, 4 \\ x_1^i = x_1^j, i, j = 1, \dots, 4, x_2^1 = x_2^2, x_2^3 = x_2^4 \\ r_t^i - r_{t-1}^i = x_t^i, t = 2, 3 \text{ e } i = 1, \dots, 4 \\ r_1^i = 0, r_3^i = 5, i = 1, \dots, 4. \end{array} \right. \quad (i) \quad (2.12)$$

As equações definidas em (i) representam a restrição de mensurabilidade do problema. □

Para que um problema de otimização estocástica em multiestágios seja computacionalmente abordável, é preciso que se formulem as restrições de mensurabilidade de maneira a torná-las implementáveis, como ilustrado no Exemplo 2.3. Por este motivo, as restrições (i) em (2.12) são chamadas de restrições de implementabilidade.

Definição 2.5 *Em geral, se dois cenários têm a mesma sequência de nós entre o primeiro e o t -ésimo estágios, eles têm a mesma parcela de informações durante estes períodos. Consequentemente, decisões associadas a tais cenários devem ser idênticas até o t -ésimo período. Este requerimento é conhecido como condição de implementabilidade, ou “nonanticipativity” em inglês.*

□

Portanto, em programação estocástica o conceito de mensurabilidade é equivalente ao conceito de implementabilidade.

Assim como para os programas em dois estágios, nesta seção são assumidas as seguintes hipóteses para os programas em multiestágios:

H1. o conjunto de primeiro estágio \mathcal{X}_1 é não vazio, convexo e compacto.

H2. para cada $\xi_t := (c_t, b_t, B_t) \in \Xi$ a imagem da multifunção $\mathcal{X}_t(\cdot, \xi_t)$ define um conjunto compacto para o t -ésimo estágio, dado por $\mathcal{X}_t(x_{t-1}, \xi_t) := \{x_t \in \mathcal{R}^{n_t} : B_t x_{t-1} + A_t x_t = b_t\}$, para $t = 2, \dots, T$;

H3. a função objetivo em cada estágio t é linear:

$$f_t(x_t, \xi_t) := \begin{cases} c_t^\top x_t & \text{se } x_t \geq 0 \\ \infty & \text{se } x_t \not\geq 0. \end{cases}$$

H4. o problema (2.9) possui recurso relativamente completo, e possui um número finito N de cenários.

A seguir são dadas as condições de otimalidade para os problemas em multies-tágios.

2.3.1 Condições de Otimalidade

É considerada nesta seção a formulação dinâmica (2.10) do problema (2.9). Dadas as hipóteses H2 e H3, segue da Proposição 2.5 que a função Q_T é convexa. Utilizando o princípio de indução desde o T -ésimo até o primeiro estágio, conclui-se que $f(x_1) = f_1(x_1) + Q_2(x_1)$ é uma função convexa e, dada H1, (2.10) é um programa convexo. Adicionalmente, se vale H4, o problema (2.10) possui uma solução ótima.

A Lagrangiana associada ao problema $Q_t(x_{t-1}, \xi_{[t]})$ é dada por

$$\mathcal{L}(x_t, u; x_{t-1}, \xi_{[t]}) = f_t(x_t, \xi_t) + Q_{t+1}(x_t, \xi_{[t]}) + u^\top (B_t x_{t-1} + A_t x_t - b_t).$$

No resultado a seguir, $\mathcal{D}_t(x_{t-1}, \xi_{[t]})$ denota o conjunto das soluções ótimas do problema dual: $\sup_u \mathcal{L}(x_t, u; x_{t-1}, \xi_{[t]})$, para $(x_{t-1}, \xi_{[t]})$ fixos.

Proposição 2.6 *Se valem as hipóteses H2, H3 e H4, então dados $(x_{t-1}, \xi_{[t]})$ fixos:*

- (i) $Q_t(x_{t-1}, \xi_{[t]}) = \inf_{x_t} \sup_u \mathcal{L}(x_t, u; x_{t-1}, \xi_{[t]}) = \sup_u \inf_{x_t} \mathcal{L}(x_t, u; x_{t-1}, \xi_{[t]})$, i.e., não há brecha de dualidade;
- (ii) $x_t^*(\xi_t)$ é uma solução ótima de (2.10b) se, e somente se, existe $u_t^*(\xi_t) \in \mathcal{D}_t(x_{t-1}, \xi_{[t]})$ satisfazendo $0 \in \partial \mathcal{L}(x_t^*(\xi_t), u_t^*(\xi_t); x_{t-1}, \xi_{[t]})$;
- (iii) a função $Q_t(x_{t-1}, \xi_{[t]})$ é subdiferenciável em x_{t-1} e $\partial Q_t(x_{t-1}, \xi_{[t]}) = -B_t^\top \mathcal{D}_t(x_{t-1}, \xi_{[t]})$.

□

Sob as hipóteses H2 e H4, a função $Q_t(\cdot, \xi_{[t]})$ é c.p.1 finita⁴, e vale a condição **D1** de

⁴Porque N é finito.

[1, p. 78]. Deste modo, a proposição enunciada é um caso particular da Proposição 3.3 de [1], a qual é rigorosamente demonstrada por seus autores.

Sendo a função $Q_t(\cdot, \xi_{[t]})$ finita, valem a condição **D3** de [1] e o teorema seguinte (ver [1, Teorema 3.5]).

Teorema 2.1 *Sob as hipóteses H2 e H4, uma solução $x_t^*(\xi^t)$ é ótima se, e somente se, existe um mapeamento mensurável $u_t^*(\xi_t)$, $t = 1, \dots, T$, tal que a seguinte condição é válida*

$$0 \in \partial f_t(x_t^*(\xi_t)) - A_t^\top u_t^*(\xi_t) - \mathbb{E}_{|\xi_{[t]}}[B_{t+1}^\top u_{t+1}^*(\xi_{t+1})],$$

para q.c. todo $\xi_{[t]} \in \Xi_{[t]}$ e $t = 1, \dots, T$, onde o termo em $T + 1$ é zero.

□

O Teorema 2.1 assegura que uma estratégia $x(\xi)$ \mathcal{F} -mensurável é ótima se cada um dos vetores $x_t(\xi)$ que a compõem é uma solução ótima do subproblema que define $Q_{t+1}(x_t, \xi_{[t]})$. Este é então um resultado importante que possibilita a aplicação de técnicas de decomposição para resolver o problema (2.10).

Para um cenário fixo ξ , a ideia principal das técnicas de decomposição é construir uma envoltória convexa de hiperplanos suporte para a função $f_t(x_t, \xi_t) + Q_{t+1}(x_t, \xi_{[t]})$, utilizando linearizações de primeira ordem para diferentes iterados $x_t^k(\xi)$, onde k é um contador de iteração.

Contudo, mesmo com a estrutura favorável às técnicas de decomposição, os programas estocásticos são difíceis de serem abordados, devido as dimensões elevadas das variáveis e restrições, como explicado a seguir.

2.3.2 Explosão da Dimensionalidade

Quando a distribuição de probabilidades P é contínua (Ξ é um conjunto infinito) foi mencionado na Subseção 2.2 que o artifício empregado para viabilizar a resolução numérica do programa estocástico é a discretização do conjunto Ξ em um número finito de cenários, usando, por exemplo, uma decomposição *SAA* do problema. Discretizando cada conjunto Ξ_t em $|\Xi_t|$ cenários, o problema (2.10) pode ser decomposto em $N := \sum_{t=1}^T N_t$ subproblemas, onde $N_1 = 1$ e $N_t := N_{t-1}|\Xi_t|$ para $t = 2, \dots, T$. Neste sentido, a quantidade de subproblemas que definem o programa estocástico aumenta (exponencialmente) com o número de estágios. Este fenômeno é conhecido como *maldição da dimensionalidade*⁵, e depende do tamanho da árvore de cenários, não do espaço de estados como na programação dinâmica [11, pp. 69-72].

⁵Em inglês, curse of dimensionality; termo cunhado pelo pesquisador e professor Richard Ernest Bellman, e bastante usado na área de programação dinâmica.

Para que a resolução do problema (2.10) (equivalentemente o problema (1.1)) seja possível em termos de tempo computacional, é preciso considerar valores moderados para N_t , $t = 1, \dots, T$. O exemplo seguinte ilustra esta afirmação.

Exemplo 2.4 *Suponha que o problema (1.1) tenha dois parâmetros incertos e independentes entre si, com cada um deles possuindo duas ocorrências possíveis. Deste modo, no segundo estágio existem $2 + 2$ cenários possíveis, i.e., $N_2 = 4$. Pela independência dos parâmetros segue que $N_t = 4^{t-1}$. Se $T = 10$ estágios, calcular o valor de $f_1(x) + \mathcal{Q}_2(x)$ para $x \in \mathcal{X}$ um ponto dado, exige a resolução de quase 350 mil problemas de otimização (mais especificamente, $N = 349.525$).*

□

Por outro lado, quando N_t é um número pequeno, a amostra de cenários ξ_t^i (para $i = 1, \dots, N_t$) se apresenta pobre do ponto de vista estatístico. Para que o problema de otimização estocástica (9.3) tenha um conjunto solução e um valor ótimo próximos daqueles do problema (1.1), é preciso que os valores N_t , para $t = 2, \dots, T$, sejam suficientemente grandes. Portanto, tem-se dois interesses contraditórios:

- (i) N_t precisa ser moderadamente pequeno para que o problema (9.3) seja abordável computacionalmente;
- (ii) N_t necessita ser suficientemente grande para que os problemas (9.3) e (1.1) sejam parecidos, em termos de conjunto solução e valor ótimo.

Fundamentados nestes interesses conflitantes, os trabalhos [35–40] propõem maneiras eficientes de representar o processo estocástico contínuo $\{\xi_t\}_{t=1}^T$ por meio de árvores de cenários de tamanhos moderados. Idealmente, procura-se uma árvore com relativamente poucos cenários, capaz de representar suficientemente bem o processo estocástico.

Mesmo com técnicas especiais de geração de cenários, em muitas situações, principalmente na formulação multiestágios, a complexidade computacional para resolver o programa estocástico subjacente é alta. Neste sentido, torna-se necessário considerar uma árvore de porte razoável e reduzir o esforço computacional para resolver o problema de otimização através de aproximações das função objetivo $f^N(\cdot) := \sum_{i=1}^N f(\cdot, \xi^i)p(\xi^i)$, como apresentado no capítulo seguinte. Maiores informações sobre a complexidade de programas estocásticos podem ser obtidas em [41].

Capítulo 3

Método de Feixes Proximal Inexato

Em determinados problemas de otimização convexa, a avaliação inexata da função e do subgradiente para construir uma linearização inexata da função tem-se apresentado uma alternativa interessante (se não necessária) para resolver o problema desejado, com um esforço computacional aceitável. Trabalhos recentes combinam este tipo de procedimento com os métodos de feixes, sendo deste modo, denominados métodos de feixes inexatos, [8, 10, 17, 42–47].

Diferentemente dos trabalhos acima citados, Hintermüller [48] considera aproximações somente no cálculo dos subgradientes, sendo o valor da função calculado de maneira exata. A principal vantagem do método proposto em [48] é que o conhecimento explícito da imprecisão não é necessário. Além disso, o método proposto por Hintermüller é assintoticamente exato. Solodov [44] apresenta uma regra para controlar a imprecisão do método, que é assumida ser limitada. Kiwiel [8] assume apenas que os erros de precisão são limitados, e em [45, 46] o método é aplicado a problemas duais que resultam de *otimização inteira*, [49, Capítulo 10]. Miller [43] desenvolveu um método de feixes inexato aplicado à função de máximo autovalor. Em [45] é apresentado um método de feixes parcialmente inexato que, para determinadas iterações do algoritmo, considera os valores exatos da função e de um subgradiente.

Neste capítulo é introduzido o *método de feixes proximal inexato* - MFI - proposto por Kiwiel [8]. Em essência, o MFI resolve problemas de otimização convexa não diferenciável exigindo somente o cálculo da função e de seu subgradiente com uma precisão limitada, mas possivelmente desconhecida, $\varepsilon > 0$. Como já mencionado, este método se torna atrativo em situações em que calcular o valor funcional e um subgradiente da função objetivo exige um esforço computacional elevado. Este é, por exemplo, o caso da programação estocástica quando há um número elevado de cenários para representar as incertezas.

3.1 Considerações Iniciais

Seja o problema de minimização $\inf_{z \in \mathcal{X}} f(z)$, onde a função fechada e convexa $f : \mathcal{R}^n \rightarrow \mathcal{R} \cup \{\infty\}$ é difícil de ser avaliada, e o conjunto $\mathcal{X} \subseteq \mathcal{R}^n$ é convexo e fechado. Além disso, é assumido que $\emptyset \neq \mathcal{X} \subseteq \text{dom } f$, e \mathcal{X} possui uma estrutura simples, por exemplo, um poliedro. Este problema pode ser reescrito como um programa convexo irrestrito

$$f_* = \inf f_{\mathcal{X}}(z), \quad \text{com} \quad f_{\mathcal{X}}(z) := f(z) + i_{\mathcal{X}}(z), \quad (3.1)$$

e $i_{\mathcal{X}}$ é a *função indicadora* de \mathcal{X} , i.e., $i_{\mathcal{X}}(z) = 0$ se $z \in \mathcal{X}$, $+\infty$ se $z \notin \mathcal{X}$, que é uma função *sci* por ser $\mathcal{X} \neq \emptyset$ convexo e fechado. Neste capítulo a função $f(z)$ pode ser, por exemplo, $f(z) := f_1(z) + \mathcal{Q}_2(z)$, como no Capítulo 2.

Para resolver o problema (3.1) de forma aproximada e com um custo computacional aceitável, Kiwiel [8] assume a disponibilidade de um oráculo inexato:

para todo $z \in \mathcal{X}$ dado existem dois erros de precisão $\varepsilon_f \geq 0$ e $\varepsilon_g \geq 0$ independentes de z , limitados, mas possivelmente desconhecidos, tais que o *valor aproximado* f_z e um *subgradiente aproximado* g_z , proporcionam a linearização aproximada de f :

$$\bar{f}_z(\cdot) := f_z + (\cdot - z)^\top g_z \leq f(\cdot) + \varepsilon_f, \quad \text{com} \quad \bar{f}_z(z) = f_z \geq f(z) - \varepsilon_f. \quad (3.2)$$

Deste modo, oráculo retorna (f_z, g_z) , onde o valor $f_z \in [f(z) - \varepsilon_f, f(z) + \varepsilon_g]$ estima $f(z)$, enquanto g_z é um $(\varepsilon_f + \varepsilon_g)$ -subgradiente de f no ponto $z \in \mathcal{X}$, i.e., $g_z \in \partial_{\varepsilon} f(z)$, com

$$\partial_{\varepsilon} f(z) := \{g : f(\cdot) + (\cdot - z)^\top g - \varepsilon\}, \quad \text{e} \quad \varepsilon := \varepsilon_f + \varepsilon_g.$$

Ao longo deste capítulo são assumidas hipóteses para o oráculo inexato que são mais fracas do que (3.2):

para todo $z \in \mathcal{X}$ existem dois erros de precisão $\varepsilon_f \geq 0$ e $\varepsilon_g \geq 0$ independentes de z , limitados, mas possivelmente desconhecidos, tais que o *valor aproximado* f_z e um *subgradiente aproximado* g_z satisfazem:

$$\begin{aligned} f_z &\in [f(z) - \varepsilon_f, f(z) + \varepsilon_g] \\ g_z &\in \partial_{\varepsilon_f + \varepsilon_g} f(z). \end{aligned} \quad (3.3)$$

Serão apresentados nos Capítulos 7, 8 e 9 oráculos que satisfazem (3.3), para a programação estocástica.

Segue de (3.3) que

$$\bar{f}_z(\cdot) := f_z + (\cdot - z)^\top g_z \leq f(\cdot) + \varepsilon_f + 2\varepsilon_g, \quad \text{com} \quad \bar{f}_z(z) = f_z \geq f(z) - \varepsilon_f,$$

i.e, a linearização inexata $\bar{f}_z(\cdot)$ é menos precisa do que aquela obtida a partir da hipótese usada por Kiwiel (cf. equação (3.2)). Este fato faz com que o método de feixes utilizado neste trabalho seja levemente diferente daquele proposto em [8] (porém, essencialmente idêntico).

O MFI utilizado neste trabalho gera uma sequência $\{z^k\} \subset \mathcal{X}$ de candidatos à solução ótima de (3.1) utilizando os valores aproximados para a função

$$f_z^k := f_{z^k},$$

para os subgradientes

$$g_z^k := g_{z^k},$$

e para as linearizações

$$f_k := \bar{f}_{z^k}$$

satisfazendo (3.3):

$$f_z^k + (\cdot - z^k)^\top g_z^k = f_k(\cdot) \leq f(\cdot) + \varepsilon_f + 2\varepsilon_g, \quad \text{com} \quad f_k(z^k) = f_z^k \geq f(z^k) - \varepsilon_f. \quad (3.4)$$

O MFI utiliza na k -ésima iteração o modelo de planos cortantes inexato de f :

$$\check{f}_k(\cdot) := \max_{j \in J^k} f_j(\cdot), \quad \text{com} \quad J^k \subset \{1, \dots, k\}, \quad (3.5)$$

e determina o novo iterado resolvendo o problema

$$z^{k+1} := \arg \min \phi_k(\cdot), \quad \text{com} \quad \phi_k(\cdot) := \check{f}_k(\cdot) + i_{\mathcal{X}}(\cdot) + \frac{1}{2t_k} \|\cdot - x^k\|^2, \quad (3.6)$$

onde $t_k > 0$ é o passo proximal que controla a influência do termo quadrático $\|\cdot - x^k\|^2$, e o centro de estabilidade $x^k := z^{k(l)}$ proporciona o valor $f_x^k = f_z^{k(l)}$ para algum $k(l) \leq k$.

Dada esta notação, a seguir são apresentadas as principais diferenças entre o método de feixes proximal exato apresentado em [6], e o método de feixes proximal inexato considerado neste trabalho.

3.1.1 Comparação entre os Métodos Exato e Inexato

São descritas na tabela a seguir as etapas principais dos algoritmos dos métodos de feixes proximais exato e inexato.

<ul style="list-style-type: none"> • Método de Feixes Proximal Exato - MFE, [6] 1. $J^k \subset \{1, \dots, k\}, \{(z^i, f(z^i), g(z^i))\}_{i \in J^k}$ 2. $\check{f}_k(\cdot) = \max_{j \in J^k} \{f(z^j) + g(z^j)^\top (\cdot - z^j)\}$ 3. $\phi_k(z) = \check{f}_k(z) + i_{\mathcal{X}}(z) + \frac{1}{2t_k} \ z - x^k\ ^2$ 4. $z^{k+1} = \arg \min \phi_k(z),$ 5. cálculo de $(f(z^{k+1}), g(z^{k+1}))$, com $g(z^{k+1}) \in \partial f(z^{k+1})$ 6. Se $f(z^{k+1}) \leq f(x^k) - \kappa v_k$, com $\kappa \in (0, 1)$ e $v_k = f(x^k) - \check{f}_k(z^{k+1})$, então $x^{k+1} = z^{k+1}$; caso contrário, $x^{k+1} = x^k$ 7. $k = k + 1$ e volte ao passo 1. • todo ponto de acumulação de $\{x^k\}$ é uma solução ótima. 	<ul style="list-style-type: none"> • Método de Feixes Proximal Inexato - MFI, [10] 1. $J^k \subset \{1, \dots, k\}, \{(z^i, f_z^i, g_z^i)\}_{i \in J^k}$ 2. $\check{f}_k(\cdot) = \max_{j \in J^k} \{f_z^j + g_z^{j\top} (\cdot - z^j)\}$ 3. $\phi_k(z) = \check{f}_k(z) + i_{\mathcal{X}}(z) + \frac{1}{2t_k} \ z - x^k\ ^2$ 4. $z^{k+1} = \arg \min \phi_k(z),$ 5. Se a imprecisão é excessiva, aumente t_k e volte ao passo 4 6. cálculo de $f_z^{k+1} \in [f(z^{k+1}) - \varepsilon_f, f(z^{k+1}) + \varepsilon_g]$ e $g_z^{k+1} \in \partial_{\varepsilon_f + \varepsilon_g} f(z^{k+1})$ 7. Se $f_x^{k+1} \leq f_x^k - \kappa v_k$, então $x^{k+1} = z^{k+1}$; caso contrário, $x^{k+1} = x^k$ 8. $k = k + 1$ e volte ao passo 1. • todo ponto de acumulação de $\{x^k\}$ é uma $2(\varepsilon_f + \varepsilon_g)$-solução.
---	---

O algoritmo do MFI é apresentado com detalhes na Seção 3.3.1.

Supondo que o problema (3.1) possua solução ótima, se o oráculo inexato satisfaz (3.3), então todo ponto de acumulação da sequência $\{x^k\}$ é uma $2(\varepsilon_f + \varepsilon_g)$ -solução.

Pode-se verificar que a qualidade da solução aproximada obtida pelo MFI é dependente dos erros de precisão ε_f e ε_g . Além disso, o esforço computacional do MFI depende do esforço computacional do oráculo inexato para calcular o par aproximado (f_z, g_z) , para $z \in \mathcal{X}$ um ponto dado. Fica então evidenciada a importância de se empregar, conjuntamente com o MFI, oráculos rápidos que proporcionam erros de precisão ε_f e ε_g tão pequenos quanto possível.

Como o método de feixes proximal (exato) considera um oráculo exato, então a desigualdade

$$f(x^k) - \phi_k(z^{k+1}) \geq 0$$

é satisfeita para qualquer iteração k do algoritmo do método. Mas, se o oráculo for inexato, o algoritmo do MFI pode proporcionar

$$f_x^k - \phi_k(z^{k+1}) < 0. \quad (3.7)$$

A desigualdade acima é utilizada para identificar quando a imprecisão do oráculo é

excessiva. Sempre que a desigualdade (3.7) é satisfeita, o algoritmo do MFI realiza uma etapa adicional, denominada *atenuação do ruído* (item 5 acima), que consiste em aumentar o parâmetro proximal t_k e o obter z^{k+1} até que $f_x^k - \phi_k(z^{k+1}) \geq 0$, ou até determinar x^k , uma $2(\varepsilon_f + \varepsilon_g)$ -solução para o problema (3.1).

Seja

$$v_k := f_x^k - \check{f}_k(z^{k+1}) \quad (3.8)$$

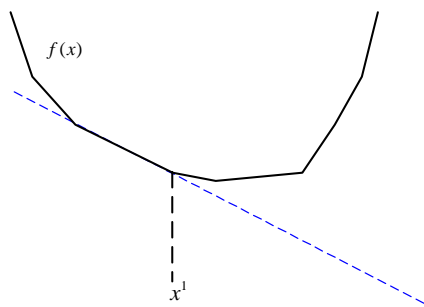
o *decrécimo predito* pelo modelo de planos cortantes \check{f}_k , com

$$f(x^k) - \varepsilon_f \leq f_x^k \leq f(x^k) + \varepsilon_g, \quad (3.9)$$

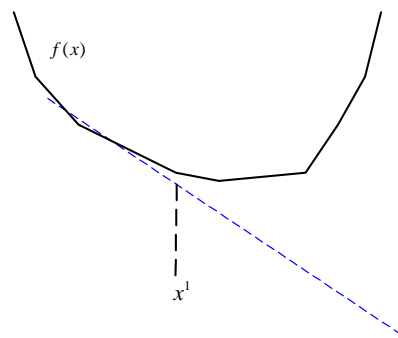
devido a hipótese (3.3). Um *passo sério* $x^{k+1} := z^{k+1}$ com $f_x^{k+1} := f_z^{k+1}$ ocorre se $f_z^{k+1} \leq f_x^k - \kappa v_k$, para $\kappa \in (0, 1)$ fixo. Caso contrário, um *passo nulo* $x^{k+1} := x^k$ é utilizado juntamente com f_{k+1} para aprimorar o modelo de planos cortantes \check{f}_{k+1} , considerando $J^{k+1} \subseteq J^k \cup \{k+1\}$. Portanto, a ideia principal do método é resolver o problema de otimização considerado procedendo como se fosse um método de feixes proximal convencional, até que (3.7) seja satisfeita. Então o método diminui a influência do termo quadrático e obtém um novo candidato à solução.

A Figura 3.1 contém uma ilustração gráfica para os métodos de feixes proximal exato e inexato, quando o conjunto viável é a reta real, i.e., $\mathcal{X} = \mathcal{R}^1$.

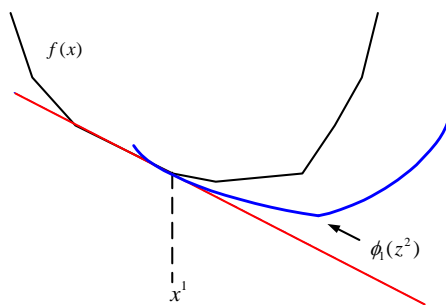
O esquema apresentado no lado direito da Figura 3.1 não ilustra o procedimento realizado pelo algoritmo do MFI quando a imprecisão do oráculo é excessiva. Este procedimento é exemplificado pela Figura 3.2.



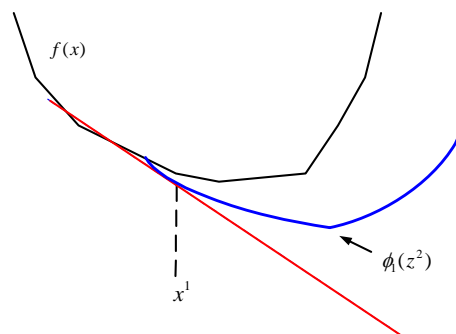
(a) Método exato ($k=1$)



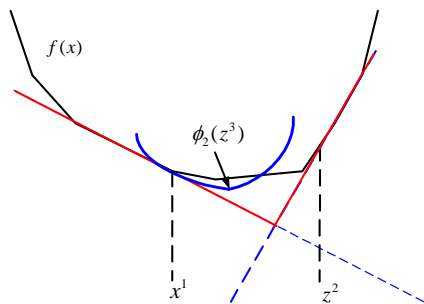
(b) Método inexacto ($k=1$)



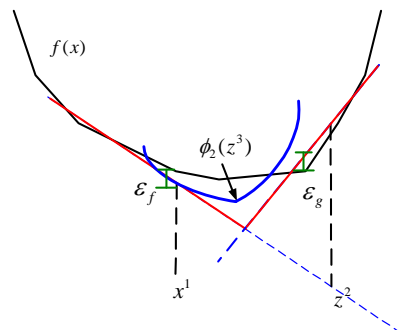
(c) Método exato ($k=2$)



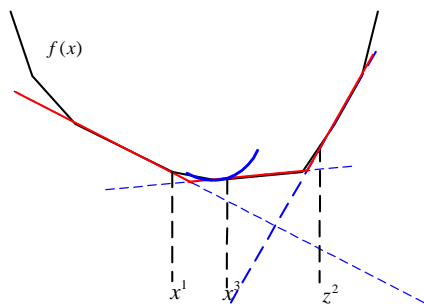
(d) Método inexacto ($k=2$)



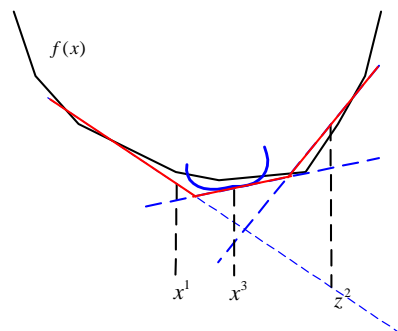
(e) Método exato ($k=3$)



(f) Método inexacto ($k=3$)

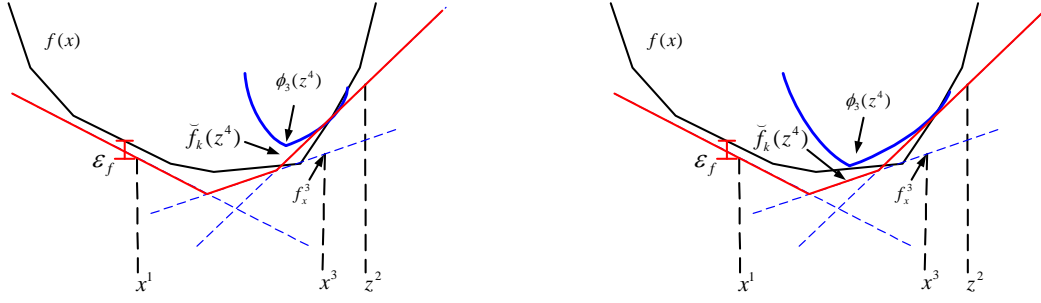


(g) Método exato ($k=4$)



(h) Método inexacto ($k=4$)

Figura 3.1: Métodos de feixes proximal exato (esquerda) e inexacto (direita).



(a) $f_x^k < \check{f}_k(z^{k+1}) + \frac{1}{2t_k} \|x^k - z^{k+1}\|^2 = \phi_k(z^{k+1})$
o parâmetro proximal t_k deve ser aumentado.

(b) $f_x^k > \check{f}_k(z^{k+1}) + \frac{1}{2t_k} \|x^k - z^{k+1}\|^2 = \phi_k(z^{k+1})$
o parâmetro proximal t_k é satisfatório.

Figura 3.2: Processo de atenuação do ruído. A imprecisão do oráculo é excessiva.

A seguir é dado o desenvolvimento teórico do método de feixes proximal inexato.

3.2 Desenvolvimento Teórico

Inicia-se esta seção com o Lema 3.1.

Lema 3.1 *Sejam $\mathcal{X} \subset \mathcal{R}^n$ um conjunto não vazio, convexo e fechado. Então o problema convexo*

$$\min_{z \in \mathcal{R}^n} \phi_k(z), \quad \text{com} \quad \phi_k(\cdot) := \check{f}_k(\cdot) + i_{\mathcal{X}}(\cdot) + \frac{1}{2t_k} \|\cdot - x^k\|^2, \quad (3.10)$$

possui uma única solução $z^{k+1} \in \mathcal{X}$. Seja também $\mathcal{N}_{\mathcal{X}}(z^{k+1})$ o cone das direções normais em \mathcal{X} no ponto z^{k+1} . Então existem $p_f^k \in \partial \check{f}_k(z^{k+1})$ e $p_{\mathcal{X}}^k \in \mathcal{N}_{\mathcal{X}}(z^{k+1})$, tais que a solução z^{k+1} é dada por

$$z^{k+1} := x^k - t_k(p_f^k + p_{\mathcal{X}}^k), \quad \text{para} \quad p_f^k = \sum_{j \in J^k} \nu_j^* g_z^j, \quad (3.11)$$

onde $\nu_j^* \geq 0$ para todo $j \in J^k$ é tal que $\sum_{j \in J^k} \nu_j^* = 1$, e

$$\nu_j^* \left\{ \check{f}_k(z^{k+1}) - [f_z^j + g_z^{j\top}(z^{k+1} - z^j)] \right\} = 0, \quad \text{para todo} \quad j \in J^k.$$

Prova. A função $\phi_k(\cdot)$ é fortemente convexa no domínio de ϕ_k , [6, Proposição IV.1.1.2]. Como o conjunto \mathcal{X} é convexo, fechado e não vazio, o problema (3.10) possui uma única solução ótima z^{k+1} , [34, Corolário 3.4.2]. A condição necessária e suficiente para a otimalidade do ponto z^{k+1} é dada por $0 \in \partial \phi_k(z^{k+1})$, ver [29, p. 264]. Dado que $\text{dom } \phi_k \subseteq \mathcal{X}$ é não vazio e convexo, então $\text{ri}(\mathcal{X}) \neq \emptyset$ e vale o

Teorema de Moreau-Rockafellar [29, Teorema 23.8] que proporciona

$$\partial\phi_k(z^{k+1}) = \partial\check{f}_k(z^{k+1}) + \frac{z^{k+1} - x^k}{t_k} + \partial i_{\mathcal{X}}(z^{k+1}).$$

Assim sendo, a condição de otimalidade se escreve como

$$0 \in \partial\check{f}_k(z^{k+1}) + \frac{z^{k+1} - x^k}{t_k} + \partial i_{\mathcal{X}}(z^{k+1}),$$

ou seja, devem existir $p_f^k \in \partial\check{f}_k(z^{k+1})$ e $p_{\mathcal{X}}^k \in \partial i_{\mathcal{X}}(z^{k+1})$ tais que $z^{k+1} = x^k - t_k(p_f^k + p_{\mathcal{X}}^k)$. Segue de [29, p. 215] que $\partial i_{\mathcal{X}}(z^{k+1}) = \mathcal{N}_{\mathcal{X}}(z^{k+1})$, pois $z^{k+1} \in \mathcal{X}$. Logo, a primeira parte do lema está provada. A seguir será mostrado que $p_f^k = \sum_{j \in J^k} \nu_j^* g_z^j$.

O problema (3.10) pode ser reescrito como

$$\begin{cases} \min_{r,z} & r + i_{\mathcal{X}}(z) + \frac{1}{2t_k} \|z - x^k\|^2 \\ \text{s.a} & f_z^j + g_z^{j\top}(z - z^j) \leq r, \text{ para } j \in J^k. \end{cases} \quad (3.12)$$

A Lagrangiana do problema (3.12) é dada por

$$\mathcal{L}(r, z, \nu) := \begin{cases} r + \frac{1}{2t_k} \|z - x^k\|^2 + \sum_{j \in J^k} \nu_j [f_z^j + g_z^{j\top}(z - z^j) - r] & \text{se } \nu \geq 0, \text{ e } z \in \mathcal{X}, \\ -\infty & \text{se } \nu \not\geq 0, z \in \mathcal{X}, \\ +\infty & \text{se } z \notin \mathcal{X}. \end{cases}$$

Seja (r^*, z^{k+1}) a solução ótima do problema (3.12). Como $\text{ri}(\mathcal{X}) \neq \emptyset$, existem multiplicadores de Lagrange $\nu_j^* \geq 0$ tais que, para todo $j \in J^k$, $\nu_j^* \{r^* - [f_z^j + g_z^{j\top}(z^{k+1} - z^j)]\} = 0$, [29, Teoremas 28.2 e 28.3]. Pela estrutura do problema considerado, existe ao menos um índice ativo $i \in J^k$ tal que, $r^* = f_z^i + g_z^{i\top}(z - z^i)$, ou seja, $r^* = \check{f}_k(z^{k+1})$.

Para mostrar que $\sum_{j \in J^k} \nu_j^* = 1$, basta reescrever a Lagrangiana $\mathcal{L}(r, z, \nu)$ para $z \in \mathcal{X}$ e $\nu = \nu^*$:

$$\mathcal{L}(r, z, \nu^*) = r(1 - \sum_{j \in J^k} \nu_j^*) + \frac{1}{2t_k} \|z - x^k\|^2 + \sum_{j \in J^k} \nu_j^* [f_z^j + g_z^{j\top}(z - z^j)].$$

Pela otimalidade de z^{k+1} segue que $-\infty < \phi_k(z^{k+1}) = \inf\{\mathcal{L}(r, z, \nu^*) : (r, z) \in \mathcal{R}^{n+1}\}$. Portanto, $\sum_{j \in J^k} \nu_j^* = 1$.

Para todo índice ativo $i \in J^k$, g_z^i é um subgradiente de $\check{f}_k(z^{k+1})$. Então, $p_f^k = \sum_{j \in J^k} \nu_j^* g_z^j$ é uma combinação convexa dos subgradientes g_z^i , e deste modo $p_f^k \in \partial\check{f}_k(z^{k+1})$, como se queria demonstrar (note que p_f^k é único se, e somente se, os vetores $\{g_z^j\}_{j \in J^k}$ são linearmente independentes). ■

Quando o conjunto \mathcal{X} possui uma estrutura mais concreta, pode-se especificar com clareza o vetor $p_{\mathcal{X}}^k$ (ver Seção 3.3.3). De modo que se economize memória computacional sem prejudicar a convergência do método, pode-se eliminar algumas linearizações inativas f_j com $\nu_j^k = 0$, e tomar $J^{k+1} \supseteq \{j \in J^k : \nu_j^k \neq 0\} \cup \{k+1\}$ para definir o modelo de planos cortantes.

A relação (3.11) permite obter uma estimativa da otimalidade a partir das seguintes *linearizações agregadas* de \check{f}_k (e f) e $i_{\mathcal{X}}$, respectivamente:

$$\bar{f}_k(\cdot) := \check{f}_k(z^{k+1}) + (\cdot - z^{k+1})^\top p_f^k \leq \check{f}_k(\cdot) \leq f(\cdot) + \varepsilon_f + 2\varepsilon_g, \quad (3.13)$$

$$\bar{i}_{\mathcal{X}}^k(\cdot) := (\cdot - z^{k+1})^\top p_{\mathcal{X}}^k \leq i_{\mathcal{X}}(\cdot), \quad (3.14)$$

pois $p_f^k \in \partial \check{f}_k(z^{k+1})$ e $p_{\mathcal{X}}^k \in \partial i_{\mathcal{X}}(z^{k+1})$. Dado que $\check{f}_k \leq f + \varepsilon_f + 2\varepsilon_g$ e que $f_{\mathcal{X}} := f + i_{\mathcal{X}}$, a relação seguinte é obtida somando (3.13) e (3.14):

$$\bar{f}_{\mathcal{X}}^k(\cdot) := \bar{f}_k(\cdot) + \bar{i}_{\mathcal{X}}^k(\cdot) \leq \check{f}_{\mathcal{X}}^k(\cdot) := \check{f}_k(\cdot) + i_{\mathcal{X}}(\cdot) \leq f_{\mathcal{X}}(\cdot) + \varepsilon_f + 2\varepsilon_g. \quad (3.15)$$

Com esta notação,

$$\bar{f}_{\mathcal{X}}^k(\cdot) = \check{f}_k(z^{k+1}) + (\cdot - z^{k+1})^\top (p_f^k + p_{\mathcal{X}}^k) \quad (3.16)$$

é denominada *linearização agregada* da função $\check{f}_k + i_{\mathcal{X}}$. Sejam o *subgradiente agregado* e o *erro da linearização agregada* de $\check{f}_k + i_{\mathcal{X}}$ dados, respectivamente, por

$$p^k := p_f^k + p_{\mathcal{X}}^k = (x^k - z^{k+1})/t_k, \quad \text{e} \quad \alpha_k := f_x^k - \bar{f}_{\mathcal{X}}^k(x^k). \quad (3.17)$$

Então, considerado $x^k \in \mathcal{X}$ o último passo sério do método, o seguinte desenvolvimento segue da relação (3.16):

$$\begin{aligned} \bar{f}_{\mathcal{X}}^k(z) &= \check{f}_k(z^{k+1}) + (z - z^{k+1})^\top p^k + f_x^k - f_x^k + (x^k - z^{k+1})^\top p^k - (x^k - z^{k+1})^\top p^k \\ &= f_x^k + (z - x^k)^\top p^k + \check{f}_k(z^{k+1}) + (x^k - z^{k+1})^\top p^k - f_x^k \\ &= f_x^k + (z - x^k)^\top p^k + \bar{f}_{\mathcal{X}}^k(x^k) - f_x^k \\ &= f_x^k + (z - x^k)^\top p^k - \alpha_k. \end{aligned} \quad (3.18)$$

Como existem valores inexatos, a identificação de tanto um ponto ótimo quanto de um passo sério, ou nulo, precisa ser modificada. Primeiramente será mostrado que o valor

$$V_k := \max\{\|p^k\|, \alpha_k\} \quad (3.19)$$

é uma boa medida para a otimalidade inexata.

3.3 Condições de Otimalidade Inexata

Dada a igualdade (3.18) tem-se que

$$f_x^k + (z - x^k)^\top p^k - \alpha_k = \bar{f}_{\mathcal{X}}^k(z) \leq f_{\mathcal{X}}(z) + \varepsilon_f + 2\varepsilon_g, \quad (3.20)$$

e deste modo,

$$\begin{aligned} f_x^k &\leq f_{\mathcal{X}}(z) - (z - x^k)^\top p^k + \alpha_k + \varepsilon_f + 2\varepsilon_g \\ &\leq f_{\mathcal{X}}(z) + \|p^k\| \|z - x^k\| + \alpha_k + \varepsilon_f + 2\varepsilon_g \\ &\leq f_{\mathcal{X}}(z) + |V_k| \left(\|z - x^k\| + 1 \right) + \varepsilon_f + 2\varepsilon_g, \end{aligned} \quad (3.21)$$

com V_k a medida definida em (3.19). Desta forma, se $V_k = 0$ tem-se

$$f_{\mathcal{X}}(x^k) - \varepsilon_f \leq f_x^k \leq \inf f_{\mathcal{X}}(z) + \varepsilon_f + 2\varepsilon_g \Rightarrow f_{\mathcal{X}}(x^k) - f_* \leq 2(\varepsilon_f + \varepsilon_g), \quad (3.22)$$

i.e., o último centro de estabilidade x^k é uma 2ε -solução para o problema (3.1), com $\varepsilon := \varepsilon_f + \varepsilon_g$. Portanto, um bom critério de parada para o método é considerar $V_k \leq \delta_{\text{To1}}$, para δ_{To1} uma tolerância positiva.

Para verificar se o iterado z^{k+1} fornece um decréscimo significativo na função f , i.e., se z^{k+1} corresponde a um passo sério, deve-se considerar o decréscimo predito pelo modelo v_k , definido em (3.8). Além de (3.8), outra maneira de representar v_k é dada a seguir.

Proposição 3.1 *O decréscimo predito pelo modelo $v_k = f_x^k - \check{f}_k(z^{k+1})$ pode ser alternativamente escrito por*

$$v_k = \alpha_k + t_k \|p^k\|^2. \quad (3.23)$$

Prova. O resultado segue diretamente de (3.16) e (3.17):

$$\begin{aligned} v_k &= f_x^k - \check{f}_k(z^{k+1}) = f_x^k - [\bar{f}_{\mathcal{X}}^k(x^k) - (x^k - z^{k+1})^\top p^k] \\ &= f_x^k - \bar{f}_{\mathcal{X}}^k(x^k) + t_k \|p^k\|^2 = \alpha_k + t_k \|p^k\|^2. \end{aligned}$$

■

Segue de (3.23) que $v_k \geq \alpha_k$. Quando o oráculo é exato, o erro de linearização α_k é não negativo e a condição $V_k \rightarrow 0$ implica que $v_k \rightarrow 0$, ou seja, $f_x^k - \check{f}_k(z^{k+1}) \rightarrow 0$. No entanto, se o oráculo é inexato, pode ser que α_k seja estritamente negativo. A Proposição 3.2 estima um limite inferior para α_k .

Proposição 3.2 *O erro de linearização α_k é limitado inferiormente por*

$$-2\varepsilon = -2(\varepsilon_f + \varepsilon_g).$$

Prova. Segue de (3.17) que

$$\begin{aligned} \alpha_k &= f_x^k - \bar{f}_{\mathcal{X}}^k(x^k) \geq f_x^k - \check{f}_{\mathcal{X}}^k(x^k) \geq f_{\mathcal{X}}(x^k) - \varepsilon_f - \check{f}_{\mathcal{X}}^k(x^k) \\ &\geq f_{\mathcal{X}}(x^k) - \varepsilon_f - f_{\mathcal{X}}(x^k) - 2\varepsilon_g - \varepsilon_f = -2\varepsilon, \end{aligned}$$

como se queria mostrar. ■

Assim sendo, usando (3.23) e a Proposição 3.2, tem-se os limites superiores para o termo V_k :

$$V_k \leq \begin{cases} \max \left\{ \sqrt{2v_k/t_k}, v_k \right\} & \text{se } v_k \geq -\alpha_k \\ \sqrt{4\varepsilon/t_k} & \text{se } v_k < -\alpha_k. \end{cases} \quad (3.24)$$

Se $v_k < -\alpha_k$, as relações

$$\begin{aligned} v_k &< -\alpha_k = -f_x^k + \bar{f}_{\mathcal{X}}^k(x^k) = -f_x^k + (\check{f}_k(z^{k+1}) + (x^k - z^{k+1})^\top p^k) \\ &= -f_x^k + (\check{f}_k(z^{k+1}) + \frac{1}{t_k}(x^k - z^{k+1})^\top (x^k - z^{k+1})) \\ &= -f_x^k + \check{f}_k(z^{k+1}) + \frac{1}{t_k} \|x^k - z^{k+1}\|^2 \end{aligned}$$

forneem a desigualdade (equivalente a (3.7))

$$f_x^k < \check{f}_k(z^{k+1}) + \frac{1}{2t_k} \|x^k - z^{k+1}\|^2 = \phi_k(z^{k+1}). \quad (3.25)$$

Isto significa que a função objetivo do problema (3.10) está acima do valor aproximado no último passo sério, como mostrado na Figura 3.2. Esta é, então, a maneira de identificar quando a inexatidão do oráculo é excessiva.

Para encontrar um novo candidato ao passo sério, pela Figura 3.2 (à direita) basta diminuir a influência do termo quadrático (i.e., aumentar t_k), e obter z^{k+1} por (3.11). Com efeito, aumentar o passo proximal faz com que (por (3.23)) a desigualdade (3.25) tenha o sinal contrário.

A seguir é apresentado o algoritmo do MFI, segundo [8, Seção 2].

3.3.1 Algoritmo Inexato

Algoritmo 3.1 MÉTODO DE FEIXES PROXIMAL INEXATO

Passo 0 (*Inicialização*). *Selecione $x^1 \in \mathcal{X}$, um parâmetro $\kappa \in (0, 1)$, um limite $\tau_{min} > 0$ e um passo proximal $t_1 \in [\tau_{min}, \tau_1)$, para $\tau_1 \geq \tau_{min}$. Faça $z^1 := x^1$, $f_x^1 := f_z^1$, $g_x^1 := g_z^1$, $J^1 := \{1\}$, $\mathbf{a}^1 := 0$, $k = k(0) := 1$, $\ell := 0$ ($k(\ell) - 1$ determina a iteração do ℓ -ésimo passo sério).*

Passo 1 (ponto teste). Obtenha z^{k+1} e os multiplicadores de Lagrange ν_j^k tais como no Lema 3.1.

Passo 2 (teste de parada). Se $V_k = 0$ (ver (3.22)), pare ($f_x^k \leq f_* + 2\varepsilon$).

Passo 3 (atenuação do ruído). Se $v_k < -\alpha_k$, faça $\mathbf{t}_k := 10\mathbf{t}_k$, $\tau_k := \max\{\tau_k, \mathbf{t}_k\}$, $\mathbf{a}^k := k$, e volte ao passo 1; caso contrário faça $\tau_{k+1} = \tau_k$.

Passo 4 (teste de descida). Calcule f_z^{k+1} e g_z^{k+1} que satisfaçam (3.3). Se o seguinte teste de descida vale:

$$f_z^{k+1} \leq f_x^k - \kappa v_k, \quad (3.26)$$

faça $x^{k+1} := z^{k+1}$, $f_x^{k+1} := f_z^{k+1}$, $\mathbf{a}^{k+1} := 0$, $k(\ell + 1) := k + 1$ e aumente ℓ por 1 (passo sério); caso contrário faça $x^{k+1} := x^k$, $f_x^{k+1} := f_x^k$ e $\mathbf{a}^{k+1} := \mathbf{a}^k$ (passo nulo).

Passo 5 (seleção do feixe). Escolha $J^{k+1} \supset \hat{J}^k \cup \{k + 1\}$, com $\hat{J}^k := \{j \in J^k : \nu_j^k \neq 0\}$.

Passo 6 (atualização do passo proximal). Se $k(\ell) = k + 1$ (i.e., após um passo sério), selecione $\mathbf{t}_{k+1} \in [\mathbf{t}_k, \tau_{k+1}]$; caso contrário, faça $\mathbf{t}_{k+1} := \mathbf{t}_k$, ou escolha $\mathbf{t}_{k+1} \in [\max\{\tau_{min}, 0.1\mathbf{t}_k\}, \mathbf{t}_k]$ se $\mathbf{a}^{k+1} = 0$.

Passo 7 (ciclo). Faça $k = k + 1$ e volte ao Passo 1.

□

No Passo 5 do algoritmo a seleção das linearizações ativas \hat{J}^k pode, por exemplo, ser considerada somente quando z^{k+1} é um passo sério. É apresentado em [2, Algoritmo 10.11] um mecanismo eficiente para manter o conjunto J^k limitado, chamado de compressão do feixe. Uma regra para atualização do passo proximal no Passo 4 é apresentada na Seção 3.4. A seguir é dada a análise de convergência do método, baseada em [8].

3.3.2 Análise de Convergência

Sejam $f_x^\infty := \lim_{k \rightarrow \infty} f_x^k$, e $V_k' := \min_{j \in \{1, \dots, k\}} V_j$. Inicia-se a análise de convergência do algoritmo do MFI pelo Lema 3.2.

Lema 3.2 Dado o problema (3.1), suponha que:

A1 - existe $c \in \mathcal{R}$ tal que o conjunto de nível

$$L_{\mathcal{X}}(c) := \{z \in \mathcal{X} : f(z) \leq c\}$$

é não vazio e limitado.

Então, $f_x^\infty \leq f_* + \varepsilon_f + 2\varepsilon_g$ quando $\lim V'_k = 0$.

Prova. A sequência $\{f_x^{k(\ell)}\}$ gerada pelo Algoritmo 3.1 é monótona decrescente. Então, para $\ell \geq 1$

$$f(x^{k(\ell)}) - \varepsilon_f \leq f_x^{k(\ell)} \leq f_x^{k(1)} \leq f(x^1) + \varepsilon_g,$$

i.e., $\{x^{k(\ell)}\} \subset L_{\mathcal{X}}(f(x^1) + \varepsilon_f + \varepsilon_g)$.

Como f é convexa e $L_{\mathcal{X}}(c) \neq \emptyset$ é limitado, então $L_{\mathcal{X}}(t)$ é limitado para todo $t \in \mathcal{R}$, [34, Teorema 3.4.4]. Deste modo, a sequência de passos sérios $\{x^{k(\ell)}\}$ gerada pelo Algoritmo 3.1 é limitada. Então, se $\lim V'_k = 0$ a relação (3.21) fornece $f_x^\infty \leq f_{\mathcal{X}}(x) + \varepsilon_f + 2\varepsilon_g$. Logo, $f_x^\infty \leq \inf f_{\mathcal{X}}(x) + \varepsilon_f + 2\varepsilon_g = f_* + \varepsilon_f + 2\varepsilon_g$, como se queria demonstrar. ■

Se o conjunto $\mathcal{X} \subset \mathcal{R}^n$ é não vazio e compacto, ou a função $f : \mathcal{X} \rightarrow \mathcal{R}$ é coerciva, então a condição A1 é satisfeita.

Dada hipótese A1, para verificar que todo ponto de acumulação da sequência de passos sérios gerada pelo Algoritmo 3.1 é uma solução aproximada do problema (3.1), é suficiente mostrar, pelo Lema 3.2, que zero é um ponto de acumulação da sequência $\{V_k\}$. Os desenvolvimentos seguintes mostram que $\liminf V_k = 0$, para diferentes alternativas resultantes do Algoritmo 3.1.

Mais precisamente, se o procedimento do Algoritmo 3.1 não termina, então podem ocorrer três situações distintas:

- (i) um ciclo infinito entre os Passos 1 e 3;
- (ii) são gerados finitos passos sérios, seguidos de infinitos passos nulos;
- (iii) são gerados infinitos passos sérios.

O resultado seguinte considera o item (i).

Lema 3.3 *Considere o Algoritmo 3.1. Se houver um ciclo infinito entre os Passos 1 e 3, então $V'_k \rightarrow 0$.*

Prova. Suponha que para alguma iteração \bar{k} , um ciclo infinito entre os Passos 1 e 3 ocorre. Então, tanto o centro de estabilidade x^k quanto o modelo \check{f}_k não se alteram para $k \geq \bar{k}$. Pela relação (3.24) vale $0 \leq V'_k \leq V_k \leq \sqrt{4\varepsilon/t_k}$ para todo $k \geq \bar{k}$. Como $t_k \rightarrow \infty$ e $\varepsilon < \infty$, então $V'_k \rightarrow 0$. ■

Número finitos de passos sérios seguidos de infinitos passos nulos

O caso de um último passo sério seguido de infinitos passos nulos pode ser subdividido em dois itens mutuamente excludentes:

- (a) o processo de *atenuação do ruído* é realizado infinitas vezes, i.e., a sequência $\{\mathbf{a}^k\} \rightarrow \infty$ (ver Algoritmo 3.1);
(b) para alguma iteração \bar{k} , e todo $k > \bar{k}$, somente passos nulos são gerados pelo Algoritmo 3.1 sem que t_k aumente (neste caso $\bar{k} \geq \{\mathbf{a}^k\}_{k > \bar{k}}$).

Para mostrar que $\lim V'_k = 0$ considerando o item (a), basta tomar o conjunto $\mathcal{K} := \{k : \mathbf{a}^k = k\}$ (iterações em que o procedimento de atenuação do ruído foi executado), e aplicar o Lema 3.3 com $t_k \xrightarrow{\mathcal{K}} \infty$. Para o item (b) considera-se o Lema 3.4 a seguir.

Lema 3.4 *Considere o Algoritmo 3.1. Suponha que para alguma iteração \bar{k} , vale que $\{\mathbf{a}^k\}_{k > \bar{k}} \leq \bar{k}$, i.e., o Passo 3 não aumenta t_k , e somente passos nulos ocorrem com $t_{k+1} \leq t_k$ determinado pelo Passo 6. Então*

$$f(x^{\bar{k}}) + \varepsilon_f + 2\varepsilon_g \geq \phi_{k+1}(z^{k+2}) \geq \phi_k(z^{k+1}) + \frac{1}{2t_k} \|z^{k+2} - z^{k+1}\|^2, \quad (3.27)$$

para cada z^{k+1} e z^{k+2} gerados pelo método, e $\lim V'_k = 0$.

Prova. Inicialmente será mostrado que $\check{f}_{\mathcal{X}}^{k+1}(\cdot) \geq \bar{f}_{\mathcal{X}}^k(\cdot)$.

Da definição do conjunto de índices \hat{J}^k , segue que

$$\hat{f}_k(\cdot) := \max_{j \in \hat{J}^k} \{f_z^j + (\cdot - z^j)^\top g_z^j\} \leq \check{f}_k(\cdot).$$

O Lema 3.1 garante que $\hat{f}_k(z^{k+1}) = \check{f}_k(z^{k+1})$ e $p_f^k \in \partial \hat{f}_k(z^{k+1})$. Por (3.13), $\bar{f}_k(\cdot)$ é a linearização em torno de $\check{f}_k(z^{k+1})$, então (3.13) pode ser alternativamente reescrita por

$$\bar{f}_k(\cdot) = \hat{f}_k(z^{k+1}) + (\cdot - z^{k+1})^\top p_f^k.$$

Pela convexidade da função $\hat{f}_k(\cdot)$, tem-se que $\bar{f}_k(\cdot) \leq \hat{f}_k(\cdot)$. Sendo \hat{J}^k um subconjunto de J^{k+1} , então $\hat{f}_k(\cdot) \leq \check{f}_{k+1}(\cdot)$. Deste modo, a relação $\bar{f}_k(\cdot) \leq \hat{f}_k(\cdot) \leq \check{f}_{k+1}(\cdot)$ implica

$$\bar{f}_{\mathcal{X}}^k(\cdot) := \bar{f}_k(\cdot) + \bar{i}_{\mathcal{X}}^k(\cdot) \leq \check{f}_{k+1}(\cdot) + \bar{i}_{\mathcal{X}}^k(\cdot) \leq \check{f}_{\mathcal{X}}^{k+1}(\cdot) := \check{f}_{k+1}(\cdot) + i_{\mathcal{X}}(\cdot).$$

Dada a relação acima, a seguir é mostrado que o decréscimo predito v_k vai para zero e, em consequência, $V'_k \rightarrow 0$.

Seja a função

$$\bar{\phi}_k(\cdot) := \bar{f}_{\mathcal{X}}^k(\cdot) + \frac{1}{2t_k} \|\cdot - x^k\|^2. \quad (3.28)$$

A função $\bar{\phi}_k(z)$ é diferenciável em todo ponto $z \in \mathcal{X}$. Segue diretamente da otimalidade do ponto $z^{k+1} \in \mathcal{X}$ que $\nabla \bar{\phi}_k(z^{k+1}) = \nabla \bar{f}_{\mathcal{X}}^k(z^{k+1}) + (z^{k+1} - x^k)/t_k =$

$p^k + (z^{k+1} - x^k)/t_k = 0$. Por (3.16) tem-se que $\bar{f}_{\mathcal{X}}^k(z^{k+1}) = \check{f}_k(z^{k+1})$, então

$$\phi_k(z^{k+1}) := \check{f}_k(z^{k+1}) + \frac{1}{2t_k} \|z^{k+1} - x^k\|^2 = \bar{f}_{\mathcal{X}}^k(z^{k+1}) + \frac{1}{2t_k} \|z^{k+1} - x^k\|^2 =: \bar{\phi}_k(z^{k+1}).$$

Utilizando a expansão de Taylor em torno de z^{k+1} ,

$$\bar{\phi}_k(\cdot) = \phi_k(z^{k+1}) + \frac{1}{2t_k} \|\cdot - z^{k+1}\|^2. \quad (3.29)$$

Segue de (3.15) e (3.28) que $\bar{\phi}_k(x^k) = \bar{f}_{\mathcal{X}}^k(x^k) \leq f(x^k) + \varepsilon_f + 2\varepsilon_g$ (usando que $x^k \in \mathcal{X}$); portanto, por (3.29),

$$\phi_k(z^{k+1}) + \frac{1}{2t_k} \|z^{k+1} - x^k\|^2 = \bar{\phi}_k(x^k) \leq f(x^k) + \varepsilon_f + 2\varepsilon_g. \quad (3.30)$$

Dadas as hipóteses do enunciado, para todo $k > \bar{k}$ tem-se que $x^k = x^{\bar{k}} \in \mathcal{X}$ e $t_{k+1} \leq t_k$. Da desigualdade $\check{f}_{\mathcal{X}}^{k+1}(\cdot) \geq \bar{f}_{\mathcal{X}}^k(\cdot)$ segue que

$$\begin{aligned} \bar{\phi}_k(\cdot) &= \bar{f}_{\mathcal{X}}^k(\cdot) + \frac{1}{2t_k} \|\cdot - x^k\|^2 \\ &\leq \check{f}_{\mathcal{X}}^{k+1}(\cdot) + \frac{1}{2t_k} \|\cdot - x^k\|^2 \\ &\leq \check{f}_{\mathcal{X}}^{k+1}(\cdot) + \frac{1}{2t_{k+1}} \|\cdot - x^{k+1}\|^2 \\ &= \phi_{k+1}(\cdot). \end{aligned}$$

Desta forma, a desigualdade (3.27) resulta de (3.29):

$$\bar{\phi}_k(z^{k+2}) = \phi_k(z^{k+1}) + \frac{1}{2t_k} \|z^{k+2} - z^{k+1}\|^2 \leq \phi_{k+1}(z^{k+2}) \leq f(x^{\bar{k}}) + \varepsilon_f + 2\varepsilon_g,$$

i.e., a sequência $\{\phi_k(z^{k+1})\}_{k \geq \bar{k}}$ é monótona e limitada superiormente por $f(x^{\bar{k}}) + \varepsilon_f + 2\varepsilon_g$. Assim,

$$\lim_k \phi_k(z^{k+1}) =: \phi_\infty \leq f(x^{\bar{k}}) + \varepsilon_f + 2\varepsilon_g, \quad \text{e} \quad z^{k+2} - z^{k+1} \rightarrow 0.$$

De (3.30)

$$\|z^{k+1} - x^{\bar{k}}\| \leq t_k (f(x^{\bar{k}}) + \varepsilon_f + 2\varepsilon_g - \phi_k(z^{k+1})).$$

Como t_k é limitado, \bar{k} é fixado, e $\phi_k(z^{k+1})$ é limitada (por $f(x^{\bar{k}}) + \varepsilon_f + 2\varepsilon_g$), resulta que a sequência $\{z^k\}$ é limitada.

A seguir será mostrado que o erro $\check{\epsilon}_k := f_z^{k+1} - \check{f}_k(z^{k+1})$ tende a zero. Para isto, utilizam-se as seguintes relações:

$$\begin{aligned}
\check{f}_{k+1}(z^{k+2}) &= \phi_{k+1}(z^{k+2}) - \frac{1}{2t_{k+1}} \left\| z^{k+2} - x^{\bar{k}} \right\|^2 \\
&= \phi_{k+1}(z^{k+2}) - \frac{1}{2t_{k+1}} \left[\left\| z^{k+2} - z^{k+1} \right\|^2 + \left\| z^{k+1} - x^{\bar{k}} \right\|^2 \right. \\
&\quad \left. - (z^{k+2} - z^{k+1})^\top (x^{\bar{k}} - z^{k+1}) \right] \\
&\leq \phi_{k+1}(z^{k+2}) - \frac{1}{2t_{k+1}} \left[\left\| z^{k+2} - z^{k+1} \right\|^2 - (z^{k+2} - z^{k+1})^\top (x^{\bar{k}} - z^{k+1}) \right] \\
&\quad - \frac{1}{2t_k} \left\| z^{k+1} - x^{\bar{k}} \right\|^2.
\end{aligned} \tag{3.31}$$

Por (3.4), a linearização $f_{k+1}(\cdot) := f_z^{k+1} + (\cdot - z^{k+1})^\top g_z^{k+1} \leq \check{f}_{k+1}(\cdot)$. Então, utilizando (3.31)

$$\begin{aligned}
\check{\epsilon}_k &= f_z^{k+1} - \check{f}_k(z^{k+1}) \\
&= f_{k+1}(z^{k+2}) + (z^{k+1} - z^{k+2})^\top g_z^{k+1} - \check{f}_k(z^{k+1}) \\
&\leq \check{f}_{k+1}(z^{k+2}) + (z^{k+1} - z^{k+2})^\top g_z^{k+1} - \check{f}_k(z^{k+1}) \\
&\leq \phi_{k+1}(z^{k+2}) + (z^{k+1} - z^{k+2})^\top g_z^{k+1} - \check{f}_k(z^{k+1}) \\
&\quad - \frac{1}{2t_{k+1}} \left[\left\| z^{k+2} - z^{k+1} \right\|^2 - (z^{k+2} - z^{k+1})^\top (x^{\bar{k}} - z^{k+1}) \right] - \frac{1}{2t_k} \left\| z^{k+1} - x^{\bar{k}} \right\|^2 \\
&= \phi_{k+1}(z^{k+2}) - \phi_k(z^{k+1}) + (z^{k+1} - z^{k+2})^\top g_z^{k+1} \\
&\quad - \frac{1}{2t_{k+1}} \left[\left\| z^{k+2} - z^{k+1} \right\|^2 - (z^{k+2} - z^{k+1})^\top (x^{\bar{k}} - z^{k+1}) \right].
\end{aligned} \tag{3.32}$$

O subdiferencial $\partial_\varepsilon f$ é localmente limitado [6, XI.4.1], e $g_z^k \in \partial_\varepsilon f(z^k)$ para todo k . Sendo a sequência $\{z^k\}$ limitada, segue que $\{g_z^k\}$ também o é. Portanto, pela desigualdade (3.32) tem-se que $\limsup_k \check{\epsilon}_k \leq 0$, porque $\{g_z^k\}$ é limitada, $\{z^k\}$ e $\{\phi_k(z^{k+1})\}$ são convergentes, \bar{k} é fixo, e $t_k \geq \tau_{\min} > 0$. Por outro lado, pela condição de passo nulo $f_z^{k+1} > f_x^{\bar{k}} - \kappa v_k$, para todo $k \geq \bar{k}$

$$\check{\epsilon}_k = [f_z^{k+1} - f_x^{\bar{k}}] + [f_x^{\bar{k}} - \check{f}_k(z^{k+1})] > -\kappa v_k + v_k = (1 - \kappa)v_k \geq 0,$$

onde $\kappa \in (0, 1)$. Deste modo, $\check{\epsilon}_k \rightarrow 0$ e $v_k \rightarrow 0$. Como $v_k \geq -\alpha_k$ e $t_k \geq \tau_{\min}$, então por (3.24),

$$0 \leq V'_k \leq V_k \leq \max\{\sqrt{2v_k/t_k}, v_k\} \rightarrow 0,$$

como se queria mostrar. ■

Falta analisar o caso (iii), em que ocorrem infinitos passos sérios.

Infinitos passos sérios

Seja $\mathcal{K} := \{k \geq 0 : k(\ell) = k, \text{ para } \ell \geq 0\}$ o conjunto de índices das iterações em que o ponto z^k gerado pelo Algoritmo 3.1 é aceito como um passo sério.

Lema 3.5 *Considere o Algoritmo 3.1, e suponha que a hipótese A1 do Lema 3.2 é satisfeita. Suponha também que \mathcal{K} possui infinitos elementos. Então $\lim V'_k = 0$.*

Prova. Dada a hipótese A1, a sequência de passos sérios $\{x^{k(\ell)}\}$ é limitada, a sequência $\{f_x^{k(\ell)}\}$ possui um ponto de acumulação finito f_x^∞ , e pela condição de passo sério segue que

$$\infty > f_x^{k(0)} - f_x^\infty = \sum_{\ell=0}^{\infty} (f_x^{k(\ell)} - f_x^{k(\ell+1)}) \geq \kappa \sum_{\ell=0}^{\infty} v_{k(\ell)},$$

i.e., $\lim_{\ell \rightarrow \infty} v_{k(\ell)} = \lim_{k \in \mathcal{K}} v_k = 0$. A relação (3.24) resulta que $0 \leq V'_k \leq V_k \leq \max\{\sqrt{2v_k/t_k}, v_k\} \xrightarrow{\mathcal{K}} 0$, completando deste modo a demonstração. ■

Como demonstrado, a relação $\lim V'_k = 0$ (logo, $\liminf V_k = 0$) vale para todas as possibilidades (i)-(iii) do Algoritmo 3.1. A seguir é considerado o principal resultado de convergência.

Convergência

Teorema 3.1 *Considere o Algoritmo 3.1, e suponha que a hipótese A1 do Lema 3.2 é satisfeita. Então, todo ponto de acumulação $\bar{x} \in \mathcal{X}$ da sequência $\{x^{k(\ell)}\}$ gerada pelo método é uma 2ε -solução do problema (3.1), para $\varepsilon = \varepsilon_f + \varepsilon_g$.*

Prova. Suponha que o procedimento do Algoritmo 3.1 não termina (caso contrário, $V_k = 0$ para alguma iteração k). Então, como mencionado, podem ocorrer três situações possíveis: (i) um ciclo infinito entre os Passos 1 e 3; (ii) o algoritmo gera finitos passos sérios, seguidos de infinitos passos nulos; e, finalmente, (iii) infinitos passos sérios são obtidos. Para mostrar que $\liminf V_k = 0$, basta recorrer ao Lema 3.3 na situação (i), ao Lema 3.4 para (ii), e, juntamente com a hipótese A1, ao Lema 3.5 na situação (iii). Então, a desigualdade $f_x^\infty \leq f_* + \varepsilon_f + 2\varepsilon_g$ é assegurada pelo Lema 3.2. Da relação $f_x^k \in [f(x^k) - \varepsilon_f, f(x^k) + \varepsilon_g]$ segue que $f(\bar{x}) - \varepsilon_f \leq f_x^\infty$, e deste modo, $f(\bar{x}) \leq f_* + 2(\varepsilon_f + \varepsilon_g)$, como queira se mostrar. ■

A seguir o problema de otimização que define o novo iterado z^{k+1} do método feixes proximal inexato é caracterizado a partir das condições do Lema 3.1, quando o conjunto viável \mathcal{X} é poliedral.

3.3.3 Determinação dos Iterados

Esta seção se restringe aos problemas do tipo (3.1), com $\mathcal{X} := \{z \in \mathcal{R}_+^n : Az = b\}$. Deste modo, o problema que se deseja resolver a cada iteração é (ver (3.6))

$$\begin{cases} \min_z & \check{f}_k(z) + \frac{1}{2t_k} \|z - x^k\|^2 \\ \text{s.a} & Az = b \\ & z \geq 0, \end{cases} \quad (3.33)$$

ou equivalentemente,

$$\begin{cases} \min_{z,r} & r + \frac{1}{2t_k} \|z - x^k\|^2 \\ \text{s.a} & Az = b, \\ & f_z^j + (z - z^j)^\top g_z^j \leq r, \quad j \in J^k \\ & z \geq 0. \end{cases} \quad (3.34)$$

Sejam z^{k+1} a solução ótima do problema (3.34), e $\lambda, \nu_j \geq 0, \mu \geq 0$ os multiplicadores ótimos de Lagrange associados, respectivamente, às restrições $Az = b, f_z^j + (z - z^j)^\top g_z^j \leq r$, e $-z \leq 0$. Seja também z um ponto viável para (3.34). Então, pela identidade

$$(A^\top \lambda - \mu)^\top (z - z^{k+1}) = \lambda^\top A(z - z^{k+1}) - \mu^\top (z - z^{k+1}),$$

e usando que $A(z - z^{k+1}) = b - b = 0$ e $\mu^\top z^{k+1} = 0$, pela condição de complementariedade, resulta que $(A^\top \lambda - \mu)^\top (z - z^{k+1}) \leq 0$. Deste modo,

$$p_{\mathcal{X}}^k := (A^\top \lambda - \mu) \in \mathcal{N}_{\mathcal{X}}(z^{k+1}).$$

Como consequência do Lema 3.1: $\nu, \mu \geq 0, \sum_{i \in J^k} \nu_i = 1, p_f^k := \sum_{j \in J^k} \nu_j g_z^j \in \partial \check{f}_k(z^{k+1})$ e, $z^{k+1} = x^k - t_k(p_f^k + p_{\mathcal{X}}^k) = x^k - t_k p^k$. Deste modo, para obter o novo iterado do método basta calcular os multiplicadores de Lagrange do problema (3.34).

Dado que $z^{k+1} \in \mathcal{X}$, resulta

$$\begin{aligned} \mathcal{L}(z^{k+1}, \lambda, \nu, \mu) &= \frac{1}{2t_k} \|z^{k+1} - x^k\|^2 + (A^\top \lambda + \sum_{i \in J^k} \nu_i p^{j_i} - \mu)^\top z^{k+1} \\ &\quad - \lambda b + \sum_{i \in J^k} \nu_i (f_z^{j_i} - g_z^{j_i \top} z^{j_i}) \\ &= \frac{1}{2t_k} \|-t_k p^k\|^2 + p^{k \top} (x^k - t_k p^k) - \lambda b + \sum_{i \in J^k} \nu_i (f_z^{j_i} - g_z^{j_i \top} z^{j_i}) \\ &= \frac{t_k}{2} \|p^k\|^2 + p^{k \top} x^k - t_k \|p^k\|^2 - \lambda b + \sum_{i \in J^k} \nu_i (f_z^{j_i} - g_z^{j_i \top} z^{j_i}) \\ &= -\frac{t_k}{2} \|p^k\|^2 + p^{k \top} x^k - \lambda b + \sum_{i \in J^k} \nu_i (f_z^{j_i} - g_z^{j_i \top} z^{j_i}). \end{aligned}$$

Utilizando a notação matriz-vetor a seguir:

$$R := \begin{bmatrix} A^\top & g_z^{j_1} \cdots g_z^{j_{|J^k|}} & -I^{n \times n} \end{bmatrix},$$

$$\Lambda := \begin{bmatrix} \lambda \\ \nu \\ \mu \end{bmatrix}, \quad H := \begin{bmatrix} -b \\ f_z^{j_1} - g_z^{j_1} z^{j_1} \\ \vdots \\ f_z^{j_{|J^k|}} - g_z^{j_{|J^k|}} z^{j_{|J^k|}} \\ 0^{n \times 1} \end{bmatrix},$$

$$G := \begin{bmatrix} 0^{|J^k|+n \times m_1} & -I^{|J^k|+n \times |J^k|+n} \end{bmatrix} \text{ e } G_{eq} := \begin{bmatrix} 0^{1 \times m_1} & 1^{1 \times |J^k|} & 0^{1 \times n} \end{bmatrix},$$

as relações primais-duais ótimas se escrevem da forma

$$R\Lambda = A^\top \lambda + \sum_{i=1}^{|J^k|} \nu_i g_z^{j_i} - \mu = p^k,$$

$$H\Lambda = -\lambda b + \sum_{i=1}^{|J^k|} \nu_i (f_x^{j_i} - g_z^{j_i \top} x^{j_i}),$$

$$G\Lambda \leq 0 \Rightarrow \nu, \mu \geq 0, \text{ e } G_{eq}\Lambda = 1 \Rightarrow \sum_{i \in J^k} \nu_i = 1.$$

Logo, o problema dual

$$\max_{\lambda, \nu, \mu} \left\{ \mathcal{L}(z^{k+1}, \lambda, \nu, \mu) : \sum_{i \in J^k} \nu_i = 1, \nu, \mu \geq 0 \right\}$$

é equivalente ao problema dual quadrático

$$\begin{cases} \max_{\Lambda} & -\frac{t_k}{2} \Lambda^\top (R^\top R) \Lambda + (R^\top x^k + H)^\top \Lambda \\ \text{s.a} & G\Lambda \leq 0 \\ & G_{eq}\Lambda = 1. \end{cases} \quad (3.35)$$

Assim sendo, a cada iteração do Algoritmo 3.1 deve-se resolver o programa quadrático (3.35) para obter as variáveis duais que definem o novo iterado, solução do problema (3.34), como definido no Lema 3.1. Se R não tem posto completo, a matriz quadrada $R^\top R$ é semidefinida positiva, e não há unicidade dos multiplicadores. Logo, precisa-se de um bom pacote de programação quadrática semidefinida para resolver (3.35), que em geral procuram a solução de norma mínima.

A seguir é dada uma regra para aprimorar a eficiência do Algoritmo 3.1.

3.4 Atualização do Passo Proximal

O passo proximal t_k exerce um papel importante na eficiência do método de feixes proximal. Para os casos em que calcular o valor da função f envolve um custo computacional elevado, um processo hábil para atualizar t_k é, em termos práticos, de grande relevância. Quando a atualização do passo proximal é realizada de maneira eficaz, o número de iterações (principalmente os passos nulos) do algoritmo diminui, e deste modo, o tempo computacional naturalmente decresce. Fundamentado nos trabalhos [23, 50] é apresentado a seguir um procedimento para atualizar t_k . Afim de manter a nomenclatura utilizada em [50], sejam:

$$u^k := 1/t_k, \quad u_{int}^{k+1} := 2u^k(1 - [f_z^{k+1} - f_x^k]/v_k), \quad u_{min} := 1/\tau_k, \quad (3.36)$$

$$\tilde{\alpha}_p^k := f_x^k - [\check{f}_k(z^{k+1}) + (x^k - z^{k+1})^\top p^k]. \quad (3.37)$$

Assim sendo, o seguinte subalgoritmo pode ser utilizado no Passo 6 do Algoritmo 3.1.

Subalgoritmo 3.1 ATUALIZAÇÃO DO PARÂMETRO PROXIMAL

Dados u^k , u_{int}^{k+1} , u_{min} , $\tilde{\alpha}_p^k$ como em (3.36) e (3.37), i_u^k um contador que introduz alguma inércia no passo proximal u^k , e ϵ_v^k uma estimativa da variação do decréscimo predito v_k (defina-se $i_u^k = 0$ e $\epsilon_v^k = \infty$ quando $k = 1$).

Passo a. Faça $u = u^k$.

Passo b. Se $x^{k+1} = x^k$ (passo nulo) vá ao Passo f.

Passo c. Se a desigualdade (3.26) é satisfeita (passo sério) e $i_u^k > 0$, faça $u = u_{int}^{k+1}$ e vá ao Passo e.

Passo d. Se $i_u^k > 3$ faça $u = u^k/2$.

Passo e. Faça $u^{k+1} = \max\{u, u^k/10, u_{min}\}$, $\epsilon_v^{k+1} = \max\{\epsilon_v^k, -2v_k\}$ e $i_u^{k+1} = \max\{i_u^k + 1, 1\}$. Se $u^{k+1} \neq u^k$ faça $i_u^{k+1} = 1$. Vá ao Passo g.

Passo f. Faça $\epsilon_v^{k+1} = \min\{\epsilon_v^k, \|p^k\| + \tilde{\alpha}_p^k\}$. Se a desigualdade $f_x^k - f_{k+1}(x^k) \geq \|p^k\| + \alpha_k$ é satisfeita e $i_u^k < -3$, faça $u = u_{int}^{k+1}$. Faça $u^{k+1} = \min\{u, 10u^k\}$ e $i_u^{k+1} = \min\{i_u^k - 1, -1\}$. Se $u^{k+1} \neq u^k$, faça $i_u^{k+1} = -1$.

Passo g. Faça $t_k = \max\{1/u^{k+1}, \tau_{min}\}$ e pare.

□

É importante ressaltar que o esquema do Subalgoritmo 3.1 preserva o comportamento da atualização do passo proximal realizado pelo Passo 6 do Algoritmo 3.1.

Portanto, a análise de convergência da Seção 3.3 continua valendo se o Passo 6 invocar o Subalgoritmo 3.1.

Uma análise detalhada do Subalgoritmo 3.1 aplicado ao método de feixes proximal (exato) é apresentada em [50, Seção 2].

A seguir o método de feixes proximal inexato é adaptado para lidar com linearizações parcialmente inexatas.

Capítulo 4

Método de Feixes Proximal Parcialmente Inexato

Assim como o MFI apresentado no Capítulo 3, o *método de feixes proximal parcialmente inexato* - MFPI - introduzido em [23] é aplicável aos problemas de otimização convexa, exigindo somente o cálculo da função e de seu subgradiente com uma imprecisão positiva limitada, mas possivelmente desconhecida. Diferentemente do MFI, o método apresentado neste capítulo usa linearizações exatas para alguns iterados. Como apresentado a seguir, a principal característica do MFPI é que todo ponto de acumulação da sequência de iterados sérios é a uma solução ótima do problema considerado.

4.1 Considerações Iniciais

Utilizando a mesma notação do Capítulo 3, seja o problema de minimização $\inf_{z \in \mathcal{X}} f(z)$, onde a função $f : \mathcal{R}^n \rightarrow \mathcal{R} \cup \{\infty\}$ é fechada e convexa, e o conjunto $\mathcal{X} \subseteq \mathcal{R}^n$ é fechado e convexo, tal que $\emptyset \neq \mathcal{X} \subseteq \text{dom } f$. Este problema pode ser reescrito como um programa convexo irrestrito

$$f_* = \inf f_{\mathcal{X}}(z), \quad \text{com} \quad f_{\mathcal{X}}(z) := f(z) + i_{\mathcal{X}}(z). \quad (4.1)$$

Seja $z \in \mathcal{X}$ um ponto dado. Ao longo deste capítulo é assumido que calcular o valor da função $f(z)$ e um subgradiente $g(z) \in \partial f(z)$ envolve um esforço computacional elevado, mas não impossível. Por este motivo, dispõe-se de um oráculo parcialmente inexato que fornece valores exatos ou inexatos, dependendo de um parâmetro $\zeta_{inex} \in \{0, 1\}$ informado, juntamente com z . Mais especificamente,

para todo $z \in \mathcal{X}$ dado, existe uma cota superior $0 \leq \varepsilon_f < \infty$ (independente de z , mas possivelmente desconhecida), tal que

$$\begin{aligned} \text{se } \zeta_{inex} = 1 \text{ calcula-se } & \begin{cases} \text{uma estimativa da função} & f_z \leq f(z) \\ \text{uma estimativa do subgradiente} & g_z \in \partial_{\varepsilon_z} f(z), \\ \text{com } \varepsilon_z := f(z) - f_z \leq \varepsilon_f, \end{cases} \\ \text{se } \zeta_{inex} = 0 \text{ calcula-se } & \begin{cases} \text{o valor exato da função} & f(z) \\ \text{um subgradiente} & g(z) \in \partial f(z). \end{cases} \end{aligned} \quad (4.2)$$

Serão apresentados nos Capítulos 7, 8 e 9 oráculos que satisfazem (4.2), para a programação estocástica.

Após chamar o oráculo para $z^k \in \mathcal{X}$ um ponto dado, o valor da função

$$f_z^k := \begin{cases} f_{z^k} & \text{se } \zeta_{inex} = 1 \\ f(z^k) & \text{se } \zeta_{inex} = 0 \end{cases}$$

e o subgradiente

$$g_z^k := \begin{cases} g_{z^k} & \text{se } \zeta_{inex} = 1 \\ g(z^k) & \text{se } \zeta_{inex} = 0 \end{cases}$$

forneem a linearização f_k , que será exata ou inexata segundo o valor de ζ_{inex} :

$$f_k(\cdot) := f_z^k + (\cdot - z^k)^\top g_z^k \leq f(\cdot) \quad \text{com} \quad f_k(z) = f_z^k \geq f(z^k) - \varepsilon_f. \quad (4.3)$$

Quando $\zeta_{inex} = 0$, o valor exato do par $(f(z^k), g(z^k))$ é exigido, e o erro de aproximação é nulo; já quando $\zeta_{inex} = 1$ o erro $\varepsilon_z^k := \varepsilon_{z^k}$ é desconhecido, porque o valor $f(z^k)$ é desconhecido.

A diferença principal entre o método parcialmente inexato e o método inexato apresentado no Capítulo 3 é a condição de que o erro $\varepsilon_z^{k(\ell)}$ satisfaz

$$\varepsilon_z^{k(\ell)} = 0,$$

para toda iteração $k(\ell)$ tal que o ponto $z^{k(\ell)} \in \mathcal{X}$ é declarado um passo sério pelo algoritmo. Mais especificamente, sejam k uma iteração, $\{z^i\}_{i=1}^k \subset \mathcal{X}$ uma sequência de pontos gerada pelo método, e $J^k \subset \{1, \dots, k\}$ um conjunto de índices de iterações passadas. O modelo de planos cortantes \check{f}_k do MFPI é dado por

$$\check{f}_k(\cdot) := \max_{j \in J^k} f_j(\cdot),$$

mas pela relação (4.3), tem-se que $\check{f}_k(\cdot) \leq f(\cdot)$. O novo iterado z^{k+1} do método é obtido resolvendo o problema quadrático (3.6). Novamente, $\{x^k\}$ denota a sequência de passos sérios gerada pelo MFPI.

Dado um iterado $z^{k+1} \in \mathcal{X}$, o oráculo (4.2) é chamado pela primeira vez com $\zeta_{inex} = 1$ para obter o par inexato (f_z^{k+1}, g_z^{k+1}) . Se o valor inexato f_z^{k+1} atingir um nível satisfatório de decréscimo da função objetivo $f_z^{k+1} \leq f_x^k - \kappa v_k$, onde v_k é o decréscimo predito definido em (3.23), o oráculo é chamado uma segunda vez, mas com $\zeta_{inex} = 0$, para obter o par exato $(f(z^{k+1}), g(z^{k+1}))$. Assim sendo, a cada iteração se satisfaz

$$\varepsilon_z^{k+1} := \begin{cases} 0 & \text{se } f_z^{k+1} \leq f_x^k - \kappa v_k, \\ f(z^{k+1}) - f_z^{k+1} > 0 & \text{é desconhecido, se } f_z^{k+1} > f_x^k - \kappa v_k. \end{cases} \quad (4.4)$$

É importante ressaltar que:

- (i) para todo passo sério x^{k+1} , o oráculo (4.2) é chamado duas vezes (a primeira com $\zeta_{inex} = 1$, e a segunda vez com $\zeta_{inex} = 0$), logo $f_x^{k+1} = f(x^{k+1})$;
- (ii) o oráculo (4.2) também pode ser chamado duas vezes para um iterado z^{k+1} que resultará em um passo nulo. Isto acontece quando

$$f_z^{k+1} \leq f_x^k - \kappa v_k, \quad \text{porém,} \quad f(z^{k+1}) > f_x^k - \kappa v_k.$$

A situação do item (ii) é indesejada, pois o esforço computacional para calcular o par exato $(f(z^{k+1}), g(z^{k+1}))$ é, em certo sentido, inútil. Para evitar que tal situação ocorra com frequência é importante que se empreguem oráculos acurados, com erros ε_z pequenos; mas isto nem sempre é possível, pois depende da função a minimizar.

A seguir são apresentadas as principais diferenças entre o método de feixes proximal inexato apresentado no Capítulo 3, e o método de feixes proximal considerado neste capítulo.

4.1.1 Comparação entre os Métodos Inexato e Parcialmente Inexato

Inicia-se a comparação entre os métodos com uma síntese de seus respectivos algoritmos:

<ul style="list-style-type: none"> • Método de Feixes Proximal Inexato - MFI, [10] <ol style="list-style-type: none"> 1. $J^k \subset \{1, \dots, k\}$, $\{(z^i, f_z^i, g_z^i)\}_{i \in J^k}$ 2. $\check{f}_k(\cdot) = \max_{j \in J^k} \{f_z^j + g_z^{j\top}(\cdot - z^j)\}$ 3. $\phi_k(z) = \check{f}_k(z) + i_{\mathcal{X}}(z) + \frac{1}{2t_k} \ z - x^k\ ^2$ 4. $z^{k+1} = \arg \min \phi_k(z)$, 5. Se a imprecisão é excessiva, aumente t_k e volte ao passo 4 6. cálculo de $f_z^{k+1} \in [f(z^{k+1}) - \varepsilon_f, f(z^{k+1}) + \varepsilon_g]$ e $g_z^{k+1} \in \partial_{\varepsilon_f + \varepsilon_g} f(z^{k+1})$ 7. Se $f_z^{k+1} \leq f_x^k - \kappa v_k$, então $x^{k+1} = z^{k+1}$; caso contrário, $x^{k+1} = x^k$ 8. $k = k + 1$ e volte ao passo 1. <ul style="list-style-type: none"> • todo ponto de acumulação de $\{x^k\}$ é uma $2(\varepsilon_f + \varepsilon_g)$-solução. 	<ul style="list-style-type: none"> • Método de Feixes Proximal Parcialmente Inexato - MFPI, [23] <ol style="list-style-type: none"> 1. $J^k \subset \{1, \dots, k\}$, $\{(z^i, f_z^i, g_z^i)\}_{i \in J^k}$ 2. $\check{f}_k(\cdot) = \max_{j \in J^k} \{f_z^j + g_z^{j\top}(\cdot - z^j)\}$ 3. $\phi_k(z) = \check{f}_k(z) + i_{\mathcal{X}}(z) + \frac{1}{2t_k} \ z - x^k\ ^2$ 4. $z^{k+1} = \arg \min \phi_k(z)$, 5. cálculo de $f_z^{k+1} \in [f(z^{k+1}) - \varepsilon_f, f(z^{k+1})]$ e $g_z^{k+1} \in \partial_{\varepsilon_f} f(z^{k+1})$ 6. Se $f_z^{k+1} \leq f_x^k - \kappa v_k$, então calcule $(f_z^{k+1}, g_z^{k+1}) = (f(z^{k+1}), g(z^{k+1}))$; caso contrário, faça $x^{k+1} = x^k$ e vá ao passo 8 7. Se $f(z^{k+1}) \leq f_x^k - \kappa v_k$, então $x^{k+1} = z^{k+1}$; caso contrário, $x^{k+1} = x^k$ 8. $k = k + 1$ e volte ao passo 1. <ul style="list-style-type: none"> • todo ponto de acumulação de $\{x^k\}$ é uma solução ótima.
---	--

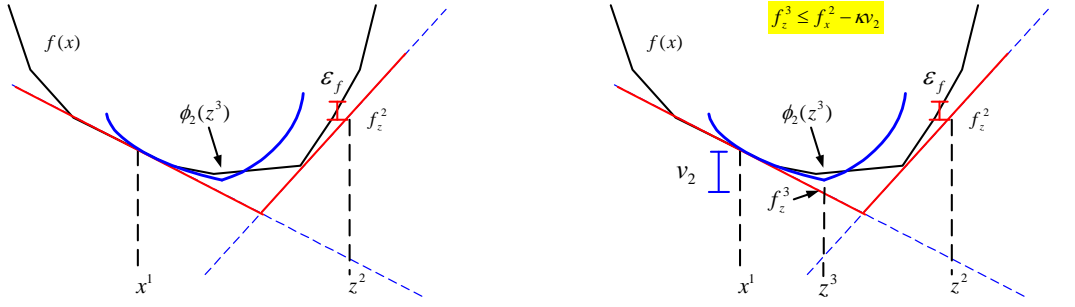
O algoritmo do MFPI é apresentado com detalhes na Seção 4.2.1. Apesar de se utilizar a mesma notação $\{(z^i, f_z^i, g_z^i)\}_{i \in J^k}$ para o feixe de informações no item 1, é importante ressaltar que, para o método parcialmente inexato, alguns dos pares (f_z^i, g_z^i) são calculados de forma exata.

O MFPI procura combinar a velocidade do MFI com a convergência exata do método de feixes proximal exato. Para que este propósito seja alcançado, é necessário empregar oráculos parcialmente inexatos rápidos e de boa qualidade. Este assunto é tratado nos Capítulos 7, 8 e 9.

A Figura 4.1 contém uma ilustração gráfica para o MFPI, quando o conjunto viável é a reta real, i.e., $\mathcal{X} = \mathcal{R}$.

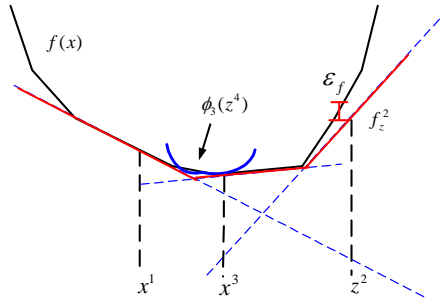
Como apresentado na Figura 4.1, o oráculo é chamado uma única vez para $k = 1$, com $\zeta_{inex} = 0$. Diferentemente, o oráculo é chamado uma única vez com $\zeta_{inex} = 1$ na iteração $k = 2$. É ilustrada na Figura 4.1(b) a situação em que a função deve ser avaliada de forma exata, i.e., o oráculo é chamado duas vezes: a primeira com $\zeta_{inex} = 1$, e a segunda com $\zeta_{inex} = 0$.

A seguir é dado o desenvolvimento teórico do método.



(a) Método parcialmente inexato (iteração 3)

(b) Método parcialmente inexato (decr. satisfatório)



(c) Método parcialmente inexato (iteração 4)

Figura 4.1: Método de feixes parcialmente inexato.

4.2 Desenvolvimento Teórico

A relação (3.11) permite obter uma estimativa da otimalidade a partir das seguintes *linearizações agregadas* de \check{f}_k (e f) e $i_{\mathcal{X}}$, respectivamente:

$$\bar{f}_k(\cdot) := \check{f}_k(z^{k+1}) + (\cdot - z^{k+1})^\top p_f^k \leq \check{f}_k(\cdot) \leq f(\cdot) \quad (4.5)$$

$$\bar{i}_{\mathcal{X}}^k(\cdot) := (\cdot - z^{k+1})^\top p_{\mathcal{X}}^k \leq i_{\mathcal{X}}(\cdot). \quad (4.6)$$

Dado que $\check{f}_k \leq f$ e que $f_{\mathcal{X}} := f + i_{\mathcal{X}}$, de (4.5) e (4.6) resulta em

$$\bar{f}_{\mathcal{X}}^k(\cdot) := \bar{f}_k(\cdot) + \bar{i}_{\mathcal{X}}^k(\cdot) \leq \check{f}_{\mathcal{X}}^k(\cdot) := \check{f}_k(\cdot) + i_{\mathcal{X}}(\cdot) \leq f_{\mathcal{X}}(\cdot). \quad (4.7)$$

Assim sendo, a desigualdade seguinte é obtida a partir de (3.16)-(3.18)

$$f_x^k + (\cdot - x^k)^\top p^k - \alpha_k = \bar{f}_{\mathcal{X}}^k(\cdot) \leq f_{\mathcal{X}}(\cdot), \quad (4.8)$$

e com isso,

$$\begin{aligned}
f_x^k &\leq f_{\mathcal{X}}(\cdot) - (\cdot - x^k)^\top p^k + \alpha_k \\
&\leq f_{\mathcal{X}}(z) + \|p^k\| \|\cdot - x^k\| + \alpha_k \\
&\leq f_{\mathcal{X}}(\cdot) + |V_k| \left(\|\cdot - x^k\| + 1 \right),
\end{aligned} \tag{4.9}$$

para $V_k := \max \{ \|p^k\|, \alpha_k \}$, como no Capítulo 3. Além disso, todo passo sério $x^k \in \mathcal{X}$ é avaliado de forma exata, i.e., $f_x^k = f(x^k)$. Desta forma, se $V_k = 0$ tem-se

$$f_{\mathcal{X}}(x^k) = f_x^k \leq \inf f_{\mathcal{X}}(z) \Rightarrow f_{\mathcal{X}}(x^k) - f_* \leq 0, \tag{4.10}$$

i.e., o último centro de estabilidade x^k é uma solução para o problema (4.1).

Diferentemente do MFI, a situação ilustrada na Figura 3.2 não pode acontecer para o MFPI, como mostrado pelo seguinte resultado.

Proposição 4.1 *O erro de linearização α_k definido em (3.17) é sempre não negativo.*

Prova. Pelo Lema 3.1, todo iterado é um ponto viável. Logo, vale (4.7) e a desigualdade $\bar{f}_{\mathcal{X}}^k(x^k) \leq f(x^k)$. Por (4.4), para todo passo sério o valor $f_x^k = f(x^k)$ é calculado de maneira exata, logo $\alpha_k = f_x^k - \bar{f}_{\mathcal{X}}^k(x^k) \geq f_x^k - f(x^k) = 0$. ■

Pela Proposição 3.1, $v_k := \alpha_k + t_k \|p^k\|^2$. Então, pela Proposição 4.1, tem-se que $v_k \geq \alpha_k \geq 0$, e vale a seguinte cota superior para a medida de otimalidade:

$$V_k := \max \{ \|p^k\|, \alpha_k \} = \max \{ \sqrt{(v_k - \alpha_k)/t_k}, \alpha_k \} \leq \max \{ \sqrt{v_k/t_k}, v_k \}. \tag{4.11}$$

Portanto, se a sequência de passos $\{t_k\}$ é limitada inferiormente por alguma constante $\tau_1 > 0$, para obter uma solução assintoticamente exata do problema (4.1), basta fazer com que $v_k \xrightarrow{k \in \mathcal{K}} 0$, para $\mathcal{K} \subset \mathcal{N}$ um conjunto de contadores de iteração.

Este é o objetivo do Algoritmo 4.1 apresentado a seguir.

4.2.1 Algoritmo Parcialmente Inexato

O algoritmo do método de feixes proximal parcialmente inexato é baseado no Algoritmo 3.1, incorporando etapas adicionais de modo que um oráculo parcialmente inexato possa ser utilizado.

Algoritmo 4.1 MÉTODO DE FEIXES PROXIMAL PARCIALMENTE INEXATO

Passo 0 (*inicialização*). *Selecione $x^1 \in \mathcal{X}$, um parâmetro de descida $\kappa \in (0, 1)$, e um passo proximal $t_1 \in [\tau_{min}, \tau_{max}]$. Faça $z^1 := x^1$, $\zeta_{inex} = 0$, e calcule $f_x^1 = f_z^1 := f(z^1)$, $g_z^1 := g(z^1)$. Defina $J^1 := \{1\}$, $k = k(0) := 1$, $\ell := 0$ ($k(\ell) - 1$ determina a iteração do ℓ -ésimo passo sério).*

Passo 1 (*ponto teste*). Obtenha z^{k+1} e os multiplicadores de Lagrange ν_j^k tais como no Lema 3.1.

Passo 2 (*teste de parada*). Se $V_k = 0$ pare!

Passo 3 (*teste de descida*).

Passo 3.1 (*cálculos inexatos*). Fixe $\zeta_{inex} = 1$ e chame o oráculo (4.2) para calcular os valores inexatos f_z^{k+1} e g_z^{k+1} .

Passo 3.2 (*teste de descida*). Se o teste de descida

$$f_z^{k+1} \leq f_x^k - \kappa v_k, \quad (4.12)$$

não é satisfeito (*passo nulo inexato*), faça $x^{k+1} = x^k$ e vá ao Passo 4.

Passo 3.3 (*cálculos exatos*). Fixe $\zeta_{inex} = 0$ e chame novamente o oráculo para calcular $(f_z^{k+1}, g_z^{k+1}) = (f(z^{k+1}), g(z^{k+1}))$ de forma exata.

Se $f(z^{k+1}) \leq f_x^k - \kappa v_k$ (*passo sério*), faça $x^{k+1} := z^{k+1}$, $k(\ell+1) := k+1$, e $\ell := \ell+1$. Caso contrário, (*passo nulo exato*) faça $x^{k+1} = x^k$ e vá ao Passo 4.

Passo 4 (*seleção do feixe*). Escolha $J^{k+1} \supset \hat{J}^k \cup \{k+1\}$, com $\hat{J}^k := \{j \in J^k : \nu_j^k \neq 0\}$.

Passo 5 (*atualização do passo proximal*). Se $k(\ell) = k+1$ (*i.e.*, após um passo sério), selecione $t_{k+1} \in [t_k, \tau_{max}]$; caso contrário, faça $t_{k+1} := t_k$, ou escolha $t_{k+1} \in [\max\{\tau_{min}, 0.1t_k\}, t_k]$.

Passo 6 (*ciclo*). Faça $k := k+1$ e volte ao Passo 1.

A atualização do passo proximal no Passo 5 pode ser realizada pelo Subalgoritmo 3.1.

Como o valor funcional e um subgradiente são calculados de forma exata para o ponto inicial $x^1 \in \mathcal{X}$, a Proposição 4.1 garante que o erro de linearização (3.17) é sempre não negativo. Portanto, o procedimento de atenuação do ruído apresentado em [23, Algoritmo 2.3], [47, Algoritmo 1], e no Passo 3 do Algoritmo 3.1 é desnecessário para o Algoritmo 4.1 (veja também comentários em [23, Seção 4]).

4.2.2 Análise de Convergência

Como para o MFI, seja $f_x^\infty := \lim_{k \rightarrow \infty} f_x^k$ e seja $V'_k := \min_{j \in \{1, \dots, k\}} V_j$. O lema seguinte é importante para analisar a convergência do método.

Lema 4.1 *Considere o Algoritmo 4.1, e dado o problema (4.1), suponha que:*

A1 - existe $c \in \mathcal{R}$ tal que o conjunto de nível

$$L_{\mathcal{X}}(c) := \{z \in \mathcal{X} : f(z) \leq c\}$$

é não vazio e limitado.

Então, $f_x^\infty \leq f_*$ quando $\lim V'_k = 0$.

Prova. De forma análoga ao Lema 3.2, pode-se mostrar que a sequência de passos sérios $\{x^{k(\ell)}\}$ gerada pelo Algoritmo 4.1 é limitada, sob a hipótese A1. Então, se $\lim V'_k = 0$ a relação (4.9) fornece $f_x^\infty \leq f_{\mathcal{X}}(x)$. Logo, $f_x^\infty \leq \inf f_{\mathcal{X}}(x) = f_*$, como se queria demonstrar. ■

Novamente, quando $\mathcal{X} \neq \emptyset$ é compacto ou a função f é coerciva, a hipótese A1 é satisfeita. Uma versão nova do Teorema 3.1 adaptada para o MFPI é apresentada a seguir.

Teorema 4.1 *Considere o Algoritmo 4.1, e suponha que a hipótese A1 do Lema 4.1 é satisfeita. Então, todo ponto de acumulação $\bar{x} \in \mathcal{X}$ da sequência $\{x^{k(\ell)}\}$ gerada pelo método é uma solução do problema (4.1).*

Prova. Suponha que o procedimento do Algoritmo 4.1 não termina (caso contrário, $V_k = 0$ para alguma iteração k). Então, podem ocorrer duas situações possíveis: (i) o algoritmo gera um número finito de passos sérios, seguidos de infinitos passos nulos; ou (ii) infinitos passos sérios são obtidos. Para mostrar que $\lim V'_k = 0$, basta recorrer ao Lema 3.4 na situação (i), e, juntamente com a hipótese A1, ao Lema 3.5 na situação (ii). Então, a desigualdade $f_x^\infty \leq f_*$ é assegurada pelo Lema 4.1. Da relação $f_x^{k(\ell)} = f(x^{k(\ell)})$ e da semicontinuidade de f segue que $f(\bar{x}) \leq \liminf f_x^{k(\ell)} = f_x^\infty$, e deste modo, $f(\bar{x}) \leq f_*$. Portanto, \bar{x} é uma solução ótima do problema (4.1), como se queria demonstrar. ■

A seguir será mostrado que quando a função f e o conjunto \mathcal{X} são poliedrais, o Algoritmo 4.1 (utilizando hipóteses adicionais) encontra um minimizador do problema (4.1) em um número finito de iterações.

4.3 Terminação Finita

Para $i = 0, \dots, n_f$, dados os conjuntos finitos de índices \mathcal{K}_i , vetores $a_j^i \in \mathcal{R}^n$, e escalares $b_j^i \in \mathcal{R}$, com $j \in \mathcal{K}_i$, considere que a função objetivo do problema (4.1) tem a forma

$$f(x) = \sum_{i=0}^{n_f} f_i(x), \quad \text{com} \quad f_i(x) = \max_{j \in \mathcal{K}_i} \{a_j^{i\top} x + b_j^i\}, \quad \text{para } i = 0, \dots, n_f. \quad (4.13)$$

Este tipo de função, que é poliedral porque n_f é finito, aparece com frequência em problemas de otimização não diferenciável, como função dual de problemas separáveis em blocos, [2, Capítulo 11]. Este é, por exemplo, o caso da programação linear estocástica em dois estágios. A saber, sob a hipótese de linearidade das funções de primeiro e segundo estágios, e quando o número N de cenários $\xi^i := (q_i, h_i, T_i)$ é finito, o problema (2.3) pode ser escrito como

$$\min_{x \in \mathcal{X}} \left\{ c^\top x + \sum_{i=1}^N p_i \min \{ q_i^\top y : T_i x + W y = h_i, y \geq 0 \} \right\}.$$

Como $p_i \geq 0$, a função objetivo dual do problema acima pode ser escrita por

$$f(x) = c^\top x + \sum_{i=1}^N \Psi_i(x), \quad \text{com} \quad \Psi_i(x) = \max \{ (h_i - T_i x)^\top u : W^\top u \leq p_i q_i \}.$$

O conjunto poliedral $\Pi(p_i q_i) := \{ u : W^\top u \leq p_i q_i \}$ possui apenas um número finito k_i de vértices u_{i_j} . Deste modo, $\Psi_i(x) = \max_{j \in \{1, \dots, k_i\}} (h_i - T_i x)^\top u_{i_j}$ e, tomando $a_j^i := -T_i^\top u_{i_j}$, $b_j^i := h_i^\top u_{i_j}$ e $\mathcal{K}_i := \{u_{i_1}, \dots, u_{i_{k_i}}\}$, tem-se a relação (4.13) com $f_0(x) := c^\top x$.

A seguir são dados dois resultados importantes para mostrar a terminação finita do Algoritmo 4.1, no caso em que f e \mathcal{X} são poliedrais.

Lema 4.2 *Seja f uma função real, convexa, e poliedral. Seja $\mathcal{X} \subset \mathcal{R}^n$ um conjunto convexo, poliedral, e não vazio. Dada a função indicadora $i_{\mathcal{X}}$ de \mathcal{X} e a função $f_{\mathcal{X}} := f + i_{\mathcal{X}}$, existem apenas um número finito de subdiferenciais $\partial f_{\mathcal{X}}$ diferentes. Além disso, cada subdiferencial $\partial f_{\mathcal{X}}$ é um poliedro convexo.*

Prova. Um resultado conhecido em análise convexa é o seguinte [29, p. 215]:

$$\partial i_{\mathcal{X}}(x) := \begin{cases} \mathcal{N}_{\mathcal{X}}(x) & \text{se } x \in \mathcal{X} \\ \emptyset & \text{caso contrário.} \end{cases}$$

Na relação acima, $\mathcal{N}_{\mathcal{X}}(x)$ é o cone normal do conjunto \mathcal{X} , em um ponto $x \in \mathcal{X}$. Como \mathcal{X} é um poliedro convexo, existem m vetores n -dimensionais B^1, \dots, B^m e $b \in \mathcal{R}^m$, tais que

$$\mathcal{X} := \{ x \in \mathcal{R}^n : B^j x \leq b_j, \text{ para } j = 1, \dots, m \}.$$

Para cada $x \in \mathcal{X}$, seja $J(x) := \{ j : B^j x = b_j \}$ o conjunto dos índices ativos em \mathcal{X} . Então,

$$\mathcal{N}_{\mathcal{X}}(x) := \left\{ \sum_{j \in J(x)} \alpha_j B^j : \alpha_j \geq 0 \right\},$$

[6, Ex. 5.2.6, Cap. III, vol. I]. Como o número de subconjuntos de $\{B^1, \dots, B^m\}$ é igual a 2^m , existem no máximo 2^m diferentes subdiferenciais $\partial i_{\mathcal{X}}(\cdot)$.

Por simplicidade de notação e sem perda de generalidade, suponha que $n_f = 0$ em (4.13), i.e.,

$$f(x) := \max_{j \in \{1, 2, \dots, k_0\}} \{a_j^\top x + b_j\}.$$

Seja $I(x)$ o conjunto dos índices ativos em x , i.e., índices tais que $f(x) = a_j^\top x + b_j$ para todo $j \in I(x)$. Com esta notação,

$$\forall x \in \mathcal{X}, \quad \partial f(x) := \text{conv}\{a_j : j \in I(x)\}.$$

Como existem apenas $2^k - 1$ conjuntos diferentes $I(\cdot)$ não vazios, então existem no máximo $2^k - 1$ subdiferenciais diferentes $\partial f(\cdot)$.

Como $\text{dom } f = \mathcal{R}^n$ por hipótese e \mathcal{X} é não vazio, o Teorema de Moreau-Rockafellar [29, Teorema 23.8] proporciona a igualdade seguinte

$$\partial f_{\mathcal{X}}(x) = \partial f(x) + \partial i_{\mathcal{X}}(x), \quad \text{para todo } x \in \mathcal{X}.$$

Como resultado, devem haver um número finito de diferentes subdiferenciais $\partial f_{\mathcal{X}}$ e, pelas definições de $\mathcal{N}_{\mathcal{X}}$ e ∂f , o conjunto $\partial f_{\mathcal{X}}$ é um poliedro convexo, [29, Corolário 19.3.2]. ■

Como consequência do Lema 4.2, o subdiferencial $\partial f_{\mathcal{X}}$ é um conjunto convexo e fechado. Estas características são importantes para o lema seguinte.

Lema 4.3 *Seja $\mathcal{X} \subset \mathcal{R}^n$ um conjunto poliedral convexo não vazio, e seja f uma função real, convexa e poliedral. Então existe uma constante $\eta > 0$ tal que*

$$\text{se } \partial f_{\mathcal{X}}(x) \cap B(0, \eta) \neq \emptyset, \quad \text{então } x \in \arg \min_{z \in \mathcal{R}^n} f_{\mathcal{X}}(z),$$

onde $B(0, \eta)$ é uma bola aberta centrada em zero e com raio η , e $f_{\mathcal{X}} = f + i_{\mathcal{X}}$.

Prova. Seja $\hat{g} = \hat{g}(x)$ a projeção do ponto zero sobre o subdiferencial $\partial f_{\mathcal{X}}(x)$, i.e.,

$$\hat{g}(x) := \arg \min_{s \in \partial f_{\mathcal{X}}(x)} \|s\|.$$

O vetor \hat{g} está bem definido para todo $x \in \mathcal{X}$, e é único porque $\partial f_{\mathcal{X}}$ é um conjunto convexo, fechado, e não vazio. Pelo Lema 4.2, existem apenas um número finito n_g de diferentes subgradientes \hat{g} . Seja $\eta > 0$ definido por

$$\eta := \min_{i=1, \dots, n_g} \{\|\hat{g}_i\| : \hat{g}_i \neq 0\},$$

e seja \bar{x} um ponto em \mathcal{X} que satisfaça $g(\bar{x}) \in \partial f_{\mathcal{X}}(\bar{x}) \cap B(0, \eta)$. Deste modo, $\hat{g}(\bar{x}) = 0 \in \partial f_{\mathcal{X}}(\bar{x})$, i.e., \bar{x} é um minimizador de $f_{\mathcal{X}}$, como enunciado. ■

Seja $J_0^k := \{j \leq k : \varepsilon_z^j = 0\}$ o conjunto de índices das linearizações exatas, e seja \check{f}_k^E o modelo de planos cortantes construído com tais linearizações:

$$\check{f}_k^E(\cdot) := \max_{j \in J_0^k} \{f(z^j) + g(z^j)^\top (\cdot - z^j)\}. \quad (4.14)$$

Como $\varepsilon_z^1 = 0$ no Algoritmo 4.1, então $J_0^1 \subset J_0^k$ e $J_0^k \neq \emptyset$ para todo k . Seja também a medida de estabilização do modelo de planos cortantes dada por

$$\wp_k := \frac{f_x^k - \phi_k(z^{k+1})}{f_x^k - \phi_{k-1}(z^k)},$$

com $\phi_k(\cdot)$ definida em (3.6), e considere a seguinte alteração do Passo 3.2 do Algoritmo 4.1:

Passo 3.2' (teste de descida). Se ao menos uma das relações

$$f_z^{k+1} \leq f_x^k - \kappa v_k, \quad \text{ou} \quad 0 \leq 1 - \wp_k \leq \delta_\wp,$$

(para $\delta_\wp > 0$ uma tolerância fixa) é satisfeita, vá ao Passo 3.3. Caso contrário, (passo nulo inexato) faça $x^{k+1} = x^k$ e vá ao Passo 4.

Fundamentado no Teorema 6.2 de Burachik, Scheimberg e Sagastizábal [51], o Lema 4.4 a seguir garante que, sob hipóteses adicionais (introduzidas pelo fato de se ter um oráculo inexato), o Algoritmo 4.1 sempre obtém um passo sério após realizar um número finito de passos nulos.

Lema 4.4 *Considere o Algoritmo 4.1, e suponha que a função objetivo do problema (4.1) é dada por (4.13), com $\mathcal{X} \neq \emptyset$ um conjunto convexo e poliedral. Além disso, suponha que:*

- (i) o passo proximal $t_k = t > 0$ é mantido fixo no Passo 5;
- (ii) o conjunto de índices das linearizações é atualizado pela regra

$$J^{k+1} \supset J_0^k \cup \hat{J}^k \cup \{k+1\},$$

onde J_0^k é o conjunto de índices das linearizações exatas, e \hat{J}^k é o conjunto de índices ativos definido no Passo 4 do Algoritmo 4.1;

- (iii) o Passo 3.2 do Algoritmo 4.1 é substituído pelo Passo 3.2' acima.

Então um passo sério é determinado após um número finito de passos nulos.

Prova. Pela hipótese (ii), o conjunto de índices das iterações exatas J_0^k está contido em J^{k+1} . Se o $(k+1)$ -ésimo iterado resulta em passo inexato (i.e., $\varepsilon_z^{k+1} > 0$), então $J_0^{k+1} = J_0^k$, e vale a inclusão $J_0^{k+1} \subset J^{k+1}$. Por outro lado, se o $(k+1)$ -ésimo iterado resulta em passo exato (i.e., $\varepsilon_z^{k+1} = 0$), então $J_0^{k+1} = J_0^k \cup \{k+1\}$, e mais uma vez, $J_0^{k+1} \subseteq J^{k+1}$ por (ii). Assim sendo, pela hipótese (ii) tem-se que $J_0^k \subseteq J^k$ para toda iteração k , e como resultado vale a desigualdade

$$\check{f}_k^E(\cdot) \leq \check{f}_k(\cdot), \quad (4.15)$$

com $\check{f}_k^E(\cdot)$ definido em (4.14). Sendo a função f dada por (4.13), então existe no máximo um número finito de diferentes modelos de planos cortantes $\check{f}_k^E(\cdot)$ ($\leq \check{f}_k(\cdot) \leq f(\cdot)$). Desde que \mathcal{X} é um conjunto convexo e poliedral, a função $\check{f}_k^E + i_{\mathcal{X}}$, é poliedral e, como $t_k = t$ por (i), para $\tilde{x} \in \mathcal{X}$ fixo, existem somente um número finito de valores diferentes

$$\phi_m^E(z^{E,m+1}) := \min \phi_m^E(z), \quad \text{com} \quad \phi_m^E(z) := \check{f}_m^E(z) + i_{\mathcal{X}}(z) + \frac{1}{2t} \|z - \tilde{x}\|^2, \quad (4.16)$$

mesmo quando $m \rightarrow \infty$. Para mostrar que um passo sério é determinado após realizar um número finito de passos nulos, suponha por contradição que o Algoritmo 4.1, considerando as hipóteses (i)-(iii), gera uma sequência infinita de passos nulos após um último passo sério gerado na iteração \tilde{k} . Por (3.27) (com $\varepsilon_g = 0$), a sequência $\{\phi_m(z^{m+1})\}_{m > \tilde{k}}$ é convergente; então $\varrho_k \rightarrow 1$. Sendo $\delta_\varphi > 0$, após um número finito k_1 de passos nulos, o par (f_z, g_z) é calculado de maneira exata por (iii). Seja $\mathcal{K}^E := \{k \geq k_1 + \tilde{k}\}$ o conjunto dos índices de iterações correspondentes aos passos nulos calculados de forma exata, obtidos após o último passo sério $\tilde{x} = x^{\tilde{k}}$. Portanto,

$$m \in \mathcal{K}^E \quad \Rightarrow \quad (f_z^m, g_z^m) = (f(z^m), g(z^m)),$$

e o conjunto \mathcal{K}^E possui um número infinito de elementos, por hipótese. Suponha que exista $m \in \mathcal{K}^E$ tal que $z^{m+1} = z^m$, então,

$$\begin{aligned} f(z^{m+1}) &\geq \check{f}_m(z^{m+1}) = \check{f}_m(z^m) \\ &= \max_{j \in J^m} \{f_z^j + g_z^{j^\top}(z^m - z^j)\} \\ &\geq f_z^m = f(z^m) = f(z^{m+1}), \end{aligned}$$

porque $m \in J^m$ pela hipótese (ii). Então, as igualdades $\check{f}_m(z^{m+1}) = f(z^{m+1})$ e $f_x^{\tilde{k}} = f(x^{\tilde{k}}) = f(\tilde{x})$, juntamente com a condição de passo nulo, implicam que

$$f(z^{m+1}) > f(\tilde{x}) - \kappa(f(\tilde{x}) - \check{f}_m(z^{m+1})) = f(\tilde{x}) - \kappa(f(\tilde{x}) - f(z^{m+1})).$$

Mas como $\kappa \in (0, 1)$, resulta que

$$f(z^{m+1}) > f(\tilde{x}). \quad (4.17)$$

No entanto, a relação

$$\check{f}_m(z^{m+1}) \leq \check{f}_m(z^{m+1}) + \frac{1}{2t} \|z^{m+1} - \tilde{x}\|^2 = \phi_m(z^{m+1}) \leq \phi_m(\tilde{x}) = \check{f}_m(\tilde{x}) \leq f(\tilde{x}),$$

contraria a desigualdade (4.17), porque $f(z^{m+1}) = \check{f}_m(z^{m+1})$. Assim sendo, os iterados nulos são todos diferentes, $z^{m+1} \neq z^m$ para todo $m \in \mathcal{K}^E$ e, deste modo, a desigualdade estrita $\phi_m(z^{m+1}) > \phi_{m-1}(z^m)$ vale por (3.27). Logo, a sequência infinita $\{\phi_m(z^{m+1})\}_{m \in \mathcal{K}^E}$ é monótona crescente, e é uniformemente limitada inferiormente por $\{\phi_m^E(z^{E,m+1})\}_{m \in \mathcal{K}^E}$, devidos às relações (4.15) e (4.16). Como resultado, a sequência $\{\phi_m^E(z^{E,m+1})\}_{m \in \mathcal{K}^E}$ é também infinita e monótona crescente (porque $J_0^k \subseteq J^k$ para todo k , e as linearizações exatas são mantidas no feixe), contrariando o fato de que existem somente um número finito de valores diferentes $\phi_m^E(z^{E,m+1})$. Portanto, o conjunto \mathcal{K}^E é finito, e um passo sério deve ser determinado após um número finito de passos nulos consecutivos. ■

O Passo 3.2 é substituído pelo Passo 3.2' com o objetivo de obter mais avaliações exatas da função. Deste modo, tanto as desigualdades em (3.27) quanto o resultado do Teorema 4.1 permanecem válidos.

O seguinte resultado é baseado no Lema 15 apresentado em [52, p. 173].

Lema 4.5 *Considere o Algoritmo 4.1 com as hipóteses (i)-(iii) do Lema 4.4, e suponha que o problema (4.1), com função objetivo dada por (4.13), tem ao menos uma solução. Se o algoritmo não para após realizar um número finito de iterações, então o algoritmo gera infinitos passos sérios, e existe uma iteração \tilde{k}_1 tal que, para todo $k \geq \tilde{k}_1$ o iterado obtido no Passo 1 satisfaz*

$$z^{k+1} \in \arg \min_{x \in \mathcal{X}} \check{f}_k(x), \quad (4.18)$$

e

$$\check{f}_k(z^{k+1}) = f_*, \quad (4.19)$$

onde f_* é o valor ótimo de f .

Prova. Pelo Teorema 4.1, se o Algoritmo 4.1 termina com $V_k = 0$, então o último passo sério do método é uma solução do problema (4.1). Contudo, se o algoritmo não termina, o número de passos sérios deve ser infinito, pelo Lema 4.4. Então, o conjunto $\mathcal{K}^s = \{k \geq 1 : k(\ell) = k, \text{ para algum } \ell \geq 0\}$, de índices das iterações em que um passo sério foi determinado pelo Algoritmo 4.1, é infinito. Ordenando

os elementos de \mathcal{K}^s de forma que k_1 seja o índice da iteração em que o primeiro passo sério foi declarado, k_2 o índice da iteração em que o segundo passo sério foi declarado, e assim sucessivamente, o Lema 3.1 define cada passo sério por

$$x^{k_{i+1}} = x^{k_i} - t_l p^l, \quad \text{com } l := k_{i+1} - 1.$$

Por hipótese, $p^k \neq 0$ para todo k ; pois do contrário, $p^k = 0 \Rightarrow \alpha_k = 0 \Rightarrow V_k = 0$ (ver (3.17)) e o algoritmo terminaria. Além disso, $x^{k_i} \neq x^{k_j}$ para todo $i \neq j$, pois do contrário ter-se-ia que $f(x^{k_i}) = f(x^{k_j})$, contrariando o fato de que a sequência de valores funcionais $\{f(x^k)\}_{k \in \mathcal{K}^s}$ seja monótona decrescente.

Como \mathcal{K}^s é um conjunto infinito, o número de linearizações exatas é infinito pela hipótese (ii) do Lema 4.4. Sendo a função f dada por (4.13), existem somente um número finito de diferentes modelos de planos cortantes \check{f}_k^E ($\leq \check{f}_k \leq f$), definido em (4.14). Então, existe um índice \tilde{k}_0 tal que, para todo $k \geq \tilde{k}_0$ tem-se a igualdade $\check{f}_k^E(\cdot) = \check{f}_k(\cdot)$, porque existem infinitas avaliações exatas (em infinitos pontos distintos), e todas as linearizações exatas são mantidas no feixe J^k .

Pelo Lema 3.1, vale a relação

$$p^k = \frac{1}{t_k}(x^k - z^{k+1}) \in \partial\{\check{f}_k(z^{k+1}) + i_{\mathcal{X}}(z^{k+1})\}.$$

Então, para todo $k \geq \tilde{k}_0$,

$$p^k \in \partial\{\check{f}_k^E(z^{k+1}) + i_{\mathcal{X}}(z^{k+1})\}, \quad \text{porque } \check{f}_k^E(\cdot) = \check{f}_k(\cdot).$$

Desde que \mathcal{X} é um conjunto convexo e poliedral, o Lema 4.2 assegura que existem apenas um número finito de subdiferenciais $\partial\{\check{f}_k^E + i_{\mathcal{X}}\}$, para k fixo. De fato, porque existem somente um número finito de modelos diferentes \check{f}_k^E , vai existir apenas um número finito de subdiferenciais $\partial\{\check{f}_k^E + i_{\mathcal{X}}\}$, mesmo quando $k \rightarrow \infty$. O Teorema 4.1 garante que $V_k = \max\{\|p^k\|, \alpha_k\} \rightarrow 0$. Então, pelo Lema 4.3, existe um índice $\tilde{k}_1 \geq \tilde{k}_0$ tal que, para todo $k \geq \tilde{k}_1$,

$$0 \in \partial\{\check{f}_k^E(z^{k+1}) + i_{\mathcal{X}}(z^{k+1})\},$$

i.e., $0 \in \{\check{f}_k(z^{k+1}) + i_{\mathcal{X}}(z^{k+1})\}$. Esta inclusão demonstra a relação (4.18).

Novamente, pelo Teorema 4.1, tem-se que $f(x^k) - \check{f}_k(z^{k+1}) \rightarrow 0$. Como somente um número finito de valores diferentes $\check{f}_k^E(z^{k+1})$ ($= \check{f}_k(z^{k+1})$, para $k \geq \tilde{k}_1$) podem ocorrer, tem-se a relação (4.19). ■

Como resultado dos Lemas 4.4 e 4.5 para o caso de infinitas iterações, os passos sérios do Algoritmo 4.1, modificado com as hipóteses (ii)-(iii), se comportam de maneira similar aos iterados do método de planos cortantes. A única função que o

termo quadrático em (3.6) desempenha nestes iterados é a de selecionar dentre as múltiplas soluções do programa linear mestre aquela que está mais próxima do passo sério x^k . Por (4.19), os valores ótimos dessas soluções não se alteram, e são iguais ao valor ótimo do problema (4.1). O resultado a seguir mostra que não podem haver infinitas iterações.

Teorema 4.2 *Considere o Algoritmo 4.1, e suponha que o problema (4.1), com função objetivo dada por (4.13), tenha ao menos uma solução ótima. Além disso, suponha que as hipóteses (i)-(iii) do Lema 4.4 sejam satisfeitas, e que a partir de uma certa iteração \tilde{k} , o conjunto de índices das linearizações seja atualizado pela regra $J^{k+1} = J^k \cup \{k+1\}$ (não há seleção nem compressão do feixe). Então, o Algoritmo 4.1 determina uma solução ótima de (4.1) após um número finito de iterações.*

Prova. Pelo Lema 4.5, após um número finito de iterações, diga-se \tilde{k} , a sequência de pontos gerada pelo Algoritmo 4.1 é idêntica aquela gerada pelo método de planos cortantes. Desde que para todo $k \geq \tilde{k}$ nenhum corte é eliminado, o método de planos cortantes tem terminação finita quando f é dada por (4.13), ver [4] e [2, Teorema 9.6, e comentários na página 134]. A medida de otimalidade de um método de planos cortantes pode ser, por exemplo, $\Delta_k = f(z^{k+1}) - \check{f}_k(z^{k+1})$. Neste sentido, para todo $m \geq \tilde{k}$ tal que $x^{m+1} = z^{m+1}$ é um passo sério gerado pelo Algoritmo 4.1, o decréscimo predito pelo modelo $v_m = f(z^{m+1}) - \check{f}_m(z^{m+1})$ coincide com Δ_m , o qual vale zero após um número finito de iterações. Mas por (4.11), $v_m = 0$ resulta em $V_m = 0$, i.e., o Algoritmo 4.1 termina após realizar um número finito de iterações.

■

Segundo as hipóteses do Teorema 4.2, todos os cortes (linearizações) calculados pelo Algoritmo 4.1 devem ser mantidos no modelo de planos cortantes \check{f}_k , a partir de uma certa iteração \tilde{k} (por exemplo, quando ocorre (4.19)). Contudo, a possibilidade de eliminar os cortes inativos é uma das principais vantagens dos métodos de feixes proximais. O processo de eliminação de cortes se torna particularmente importante quando a dimensão n da variável x é elevada (assim sendo, muitos cortes são necessários para aproximar f). Outro ponto prático importante é o fato que, para assegurar a convergência finita do algoritmo, o passo proximal deve permanecer constante. Esta exigência, juntamente com a condição de manter no feixe os cortes inativos, reduz a eficiência do Algoritmo 4.1.

Apesar de eliminar os cortes inativos, Kiwiel [53] mostra que o método de feixes proximal exato tem terminação finita no caso poliedral, se o passo t permanece constante durante os passos sérios, a partir de uma certa iteração \tilde{k} . Durante os passos nulos, t é permitido variar somente um número limitado de vezes. Para que o

método termine após fazer um número finito de iterações, Kiwiel adota um critério alternativo para a escolha de um passo sério: z^{k+1} é declarado um passo sério se

$$f(z^{k+1}) < f(x^k) \quad \text{e} \quad f(z^{k+1}) = \check{f}_k(z^{k+1}). \quad (4.20)$$

O critério acima pode ser empregado de forma periódica, considerando um número $n_s \geq 0$ de passos sérios escolhidos pelo critério convencional: após n_s passos sérios serem escolhidos pelo critério (3.26), o próximo passo sério deve ser definido por (4.20). A partir de então, outros n_s passos sérios devem ser determinado por (3.26), e novamente o critério (4.20) deve ser empregado, e assim sucessivamente.

O critério adotado por Kiwiel não assegura a terminação finita do Algoritmo 4.1, a não ser que seja empregado a partir da iteração em que vale a desigualdade do item (ii) do Lema 4.4.

Encerra-se este capítulo com um comentário acerca da qualidade do oráculo parcialmente inexato. Como mencionado no início deste capítulo, se o erro $\varepsilon_z = f(z) - f_z$ do oráculo utilizado no Passo 3.1 do Algoritmo 4.1 for considerável, a relação seguinte pode ocorrer com frequência:

$$f_z^{k+1} \leq f_x^k - \kappa v_k, \quad \text{porém,} \quad f(z^{k+1}) > f_x^k - \kappa v_k.$$

Isto faz com que o algoritmo exija (com frequência) o cálculo dos valores inexato e exato pelo oráculo, para passos nulos. Logo, deve-se definir um oráculo parcialmente inexato com bastante cautela, de modo que o esforço computacional do método não acabe sendo maior do que o esforço computacional do método de feixes proximal exato.

Com o objetivo de definir um método parcialmente inexato menos susceptível à qualidade do oráculo, é apresentado no Capítulo 6 uma extensão do *método de nível*, capaz de lidar com oráculos parcialmente inexatos. Para isto, primeiro é considerado um método de nível adaptado para oráculos inexatos.

Capítulo 5

Método de Nível Proximal Inexato

Como já mencionado, os métodos de feixes estabilizam o método de planos cortantes. É bem conhecido que iterações sucessivas do método de planos cortantes podem não proporcionar um progresso no processo de otimização, no sentido que o decréscimo no valor da função objetivo é pequeno (ou mesmo negativo), enquanto pode haver uma grande diferença entre os iterados da variável de decisão. Este fato é ilustrado em [2, Exemplo 9.7].

Além de estáveis, os métodos de feixes proximais fornecem um mecanismo eficiente para manter o número de linearizações (cortes) limitado, chamado de compressão do feixe, [2, Algoritmo 10.11]. Outro método importante da família dos métodos de feixes é o *método de nível*, desenvolvido por Lemaréchal, Nemirovskii e Nesterov [19]. Como o próprio nome sugere, este método determina o novo candidato à solução projetando o último iterado em um conjunto de nível, que é atualizado a cada iteração. O conjunto de nível é uma alternativa ao termo quadrático do método de feixes proximal, para estabilizar o processo de otimização.

Um método de nível capaz de lidar com oráculos inexatos foi desenvolvido por Fábíán [17]. O problema de otimização convexa considerado em [17] é bastante geral, no sentido que a função objetivo e as funções que definem as restrições são supostas serem convexas e Lipschitz contínuas na variável de decisão. No método de nível inexato proposto por Fábíán a imprecisão do oráculo é assintoticamente nula, fazendo com que o método convirja para uma solução exata do problema de otimização considerado. Dadas estas características, o método de nível inexato desenvolvido por Fábíán pode ser entendido como um método incremental, [47]. A aplicação do método de nível (resp. inexato) à programação estocástica é denominada *decomposição de nível* (resp. inexata), e desenvolvida por Fábíán e Szőke [18]. Os autores consideram um oráculo para os programas lineares estocásticos em dois estágios da forma (2.1)-(2.2), que é inexato em dois sentidos distintos: (i) todos os programas lineares de segundo estágio são resolvidos aproximadamente; e (ii) uma estrutura em células é construída no espaço Ξ das incertezas, de maneira que os

cenários pertencentes a mesma célula são agregados no baricentro. Desta forma, o oráculo inexato proposto em [18] pode utilizar (i) ou (ii), ou simultaneamente (i) e (ii). Diferentemente, o método de nível inexato proposto neste capítulo utiliza um oráculo que é inexato porque apenas alguns problemas de segundo estágio são resolvidos. Os demais problemas são aproximados por um procedimento rápido, que não envolve a resolução de problemas de otimização.

Uma outra variante do método de nível é o *método de nível proximal*, desenvolvido por Kiwiel [9]. Ao invés de projetar o último iterado z^k no conjunto de nível para obter z^{k+1} , o método de nível proximal projeta um centro de estabilidade $x^k \in \mathcal{X}$. Diferentemente do método de feixes proximal, o centro de estabilidade do método de nível proximal não precisa ser, necessariamente, o melhor candidato à solução, encontrado durante o processo de otimização. Assim como o método de feixes proximal, e diferentemente dos métodos de nível propostos em [17–19], o método de nível proximal é capaz de manter o tamanho do feixe (valores funcionais, subgradientes e iterados) limitado.

Neste capítulo é proposta uma abordagem inexata para o método de nível proximal. O método resultante é denominado *método de nível proximal inexato* - MNI. Assim como o MFI apresentado no Capítulo 3, o MNI utiliza linearizações inexatas construídas a partir de valores funcionais e subgradientes aproximados, retornado por um oráculo inexato. Como apresentado a seguir, o MNI converge a uma solução aproximada do problema de otimização convexa considerado.

5.1 Considerações Iniciais

Neste capítulo o problema de interesse é escrito por

$$f_* := \min_{z \in \mathcal{X}} f(z), \quad (5.1)$$

com a função fechada e convexa $f : \mathcal{R}^n \rightarrow \mathcal{R} \cup \{\infty\}$. O conjunto $\mathcal{X} \subset \mathcal{R}^n$ é suposto ser convexo e compacto, com diâmetro D não necessariamente conhecido. Além disso, é assumido que $\emptyset \neq \mathcal{X} \subseteq \text{ri}(\text{dom } f)$. Dadas estas hipóteses, então a função f é Lipschitziana em \mathcal{X} . Para os programas lineares estocásticos em dois estágios, tais hipóteses acerca da função f são asseguradas pela Proposição 2.2, quando \mathcal{X} é compacto.

Para resolver o problema (5.1) de forma aproximada e com um custo computacional aceitável, é assumida a disponibilidade de um oráculo inexato que:

para todo ponto $z \in \mathcal{X}$ dado, e dois erros $\varepsilon_f, \varepsilon_g \geq 0$, independentes de z , limitados, mas possivelmente desconhecidos, são fornecidos

$$\left[\begin{array}{ll} \text{uma estimativa da função} & f_z \in [f(z) - \varepsilon_f, f(z) + \varepsilon_g] \\ \text{uma estimativa do subgradiente} & g_z \in \partial_{\varepsilon_f + \varepsilon_g} f(z). \end{array} \right. \quad (5.2)$$

Como f é convexa e $\varepsilon_f + \varepsilon_g$ é limitado, o subdiferencial $\partial_{\varepsilon_f + \varepsilon_g} f(\cdot)$ é localmente limitado, [6, XI.4.1]. Sendo \mathcal{X} compacto, existe uma constante independente de $z \in \mathcal{X}$ e $\varepsilon_f + \varepsilon_g$, que limita superiormente $\|g_z\|$ para todo $z \in \mathcal{X}$. Sem perda de generalidade, é assumido neste capítulo que esta constante coincide com a constante de Lipschitz Λ de f , i.e.,

$$\|g_z\| \leq \Lambda \quad \text{para todo } z \in \mathcal{X}.$$

Seja k um contador de iteração. O MNI proposto neste trabalho, quando empregado com um oráculo inexato satisfazendo (5.2), gera uma sequência $\{z^k\} \subset \mathcal{X}$ de candidatos à solução do problema (5.1), utilizando os valores aproximados para a função $f_z^k := f_{z^k}$, e para o subgradiente $g_z^k := g_{z^k}$, cumprindo a relação seguinte:

$$f_k(\cdot) = f_z^k + (\cdot - z^k)^\top g^k \leq f(\cdot) + \varepsilon_f + 2\varepsilon_g \quad \text{com} \quad f_k(z^k) = f_z^k \geq f(z^k) - \varepsilon_f. \quad (5.3)$$

Além do mais, a condição de Lipschitz com constante uniforme Λ é satisfeita pela linearização f_k :

$$|f_k(\tilde{z}) - f_k(\bar{z})| \leq \Lambda \|\tilde{z} - \bar{z}\|, \quad \text{para todo } \tilde{z}, \bar{z} \in \mathcal{X}. \quad (5.4)$$

Serão apresentados nos Capítulos 7 e 8 diversos oráculos que satisfazem as exigências acima, para a programação estocástica. A seguir são descritas as principais diferenças entre o método de nível proximal exato apresentado em [9], e o método de nível proximal inexato considerado neste capítulo.

5.1.1 Comparação entre os Métodos Exato e Inexato

Inicia-se a comparação entre os métodos com uma síntese de seus respectivos algoritmos:

<ul style="list-style-type: none"> • Método de Nível Proximal Exato [9] <ol style="list-style-type: none"> 1. $J^k \subset \{1, \dots, k\}, \{(z^i, f(z^i), g(z^i))\}_{i \in J^k}$ 2. $\check{f}_k(\cdot) = \max_{j \in J^k} \{f(z^j) + g(z^j)^\top (\cdot - z^j)\}$ 3. $X^k = \{z \in \mathcal{X} : \check{f}_k(z) \leq f_{niv}^k\}$ e $f_{niv}^k = \kappa f_{inf}^k + (1 - \kappa) f_{sup}^k$ 4. Se $X^k = \emptyset$ atualize x_k e f_{inf}^k e volte ao passo 3 5. $z^{k+1} = \arg \min_{z \in X^k} \ z - x^k\ ^2$, 6. cálculo de $f(z^{k+1})$ e $g(z^{k+1}) \in \partial f(z^{k+1})$ 7. $f_{sup}^{k+1} = \min\{f_{sup}^k, f(z^{k+1})\}$, $k = k + 1$ e volte ao passo 1. <ul style="list-style-type: none"> • $\mathcal{X} \subseteq \text{ri}(\text{dom } f)$ deve ser compacto. • todo ponto de acumulação de $\{x^k\}$ é uma solução ótima. 	<ul style="list-style-type: none"> • Método de Nível Proximal Inexato <ol style="list-style-type: none"> 1. $J^k \subset \{1, \dots, k\}, \{(z^i, f_z^i, g_z^i)\}_{i \in J^k}$ 2. $\check{f}_k(\cdot) = \max_{j \in J^k} \{f_z^j + g_z^{j\top} (\cdot - z^j)\}$ 3. $X^k = \{z \in \mathcal{X} : \check{f}_k(z) \leq f_{niv}^k\}$ e $f_{niv}^k = \kappa f_{inf}^k + (1 - \kappa) f_{sup}^k$ 4. Se $X^k = \emptyset$ atualize x_k e f_{inf}^k e volte ao passo 3 5. $z^{k+1} = \arg \min_{z \in X^k} \ z - x^k\ ^2$, 6. cálculo de $f_z^{k+1} \in [f(z^{k+1}) - \varepsilon_f, f(z^{k+1}) + \varepsilon_g]$ e $g_z^{k+1} \in \partial_{\varepsilon_f + \varepsilon_g} f(z^{k+1})$ 7. $f_{sup}^{k+1} = \min\{f_{sup}^k, f_z^{k+1}\}$, $k = k + 1$ e volte ao passo 1. <ul style="list-style-type: none"> • $\mathcal{X} \subseteq \text{ri}(\text{dom } f)$ deve ser compacto. • todo ponto de acumulação de $\{x^k\}$ é uma $2(\varepsilon_f + \varepsilon_g)$-solução.
---	--

Como pode ser verificado acima, a única diferença entre os algoritmos dos métodos de nível proximal exato e inexato consiste no oráculo (exato ou inexato) utilizado para definir o feixe de informações, no item 1.

Para o método exato, o valor f_{inf}^k é uma cota inferior e f_{sup}^k é uma cota superior para o valor ótimo f_* . Já no algoritmo inexato, estes valores são apenas estimativas destas cotas.

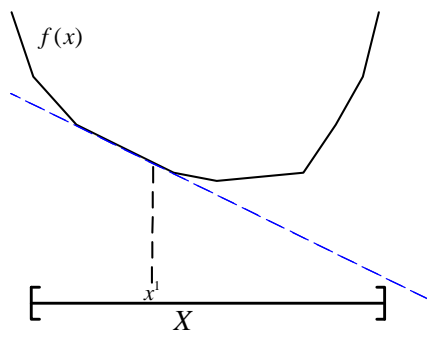
A comparação continua com os métodos inexatos MFI e MNI, cuja síntese é descrita a seguir.

<ul style="list-style-type: none"> • Método de Feixes Proximal Inexato [10] <ol style="list-style-type: none"> 1. $J^k \subset \{1, \dots, k\}, \{(z^i, f_z^i, g_z^i)\}_{i \in J^k}$ 2. $\check{f}_k(\cdot) = \max_{j \in J^k} \{f_z^j + g_z^{j\top}(\cdot - z^j)\}$ 3. $\phi_k(z) = \check{f}_k(z) + i_{\mathcal{X}}(z) + \frac{1}{2t_k} \ z - x^k\ ^2$ 4. $z^{k+1} = \arg \min \phi_k(z),$ 5. Se a imprecisão é excessiva, aumente t_k e volte ao passo 4 6. cálculo de $f_z^{k+1} \in [f(z^{k+1}) - \varepsilon_f, f(z^{k+1}) + \varepsilon_g]$ e $g_z^{k+1} \in \partial_{\varepsilon_f + \varepsilon_g} f(z^{k+1})$ 7. Se $f_z^{k+1} \leq f_x^k - \kappa v_k$, então $x^{k+1} = z^{k+1}$; caso contrário, $x^{k+1} = x^k$ 8. $k = k + 1$ e volte ao passo 1. <ul style="list-style-type: none"> • todo ponto de acumulação de $\{x^k\}$ é uma $2(\varepsilon_f + \varepsilon_g)$-solução. 	<ul style="list-style-type: none"> • Método de Nível Proximal Inexato <ol style="list-style-type: none"> 1. $J^k \subset \{1, \dots, k\}, \{(z^i, f_z^i, g_z^i)\}_{i \in J^k}$ 2. $\check{f}_k(\cdot) = \max_{j \in J^k} \{f_z^j + g_z^{j\top}(\cdot - z^j)\}$ 3. $X^k = \{z \in \mathcal{X} : \check{f}_k(z) \leq f_{niv}^k\}$ e $f_{niv}^k = \kappa f_{inf}^k + (1 - \kappa) f_{sup}^k$ 4. Se $X^k = \emptyset$ atualize x_k e f_{inf}^k e volte ao passo 3 5. $z^{k+1} = \arg \min_{z \in X^k} \ z - x^k\ ^2,$ 6. cálculo de $f_z^{k+1} \in [f(z^{k+1}) - \varepsilon_f, f(z^{k+1}) + \varepsilon_g]$ e $g_z^{k+1} \in \partial_{\varepsilon_f + \varepsilon_g} f(z^{k+1})$ 7. $f_{sup}^{k+1} = \min\{f_{sup}^k, f_z^{k+1}\}, k = k + 1$ e volte ao passo 1. <ul style="list-style-type: none"> • $\mathcal{X} \subseteq \text{ri}(\text{dom } f)$ deve ser compacto. • todo ponto de acumulação de $\{x^k\}$ é uma $2(\varepsilon_f + \varepsilon_g)$-solução.
---	--

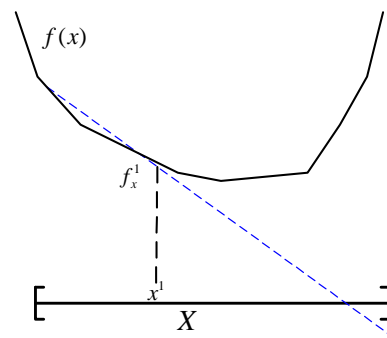
O algoritmo do MNI é apresentado com detalhes na Seção 5.3. A diferença principal entre o método de feixes proximal e o método de nível proximal são os mecanismos empregados para estabilizar os planos cortantes. O método de feixes proximal utiliza um termo quadrático, enquanto o método de nível proximal considera um conjunto de nível.

Ambos os métodos permitem manter o feixe de informações limitado, como comentado na Seção 5.3.

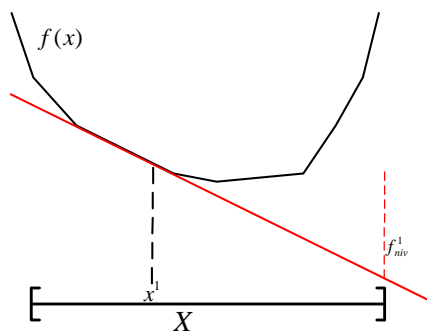
Com o intuito de ajudar o entendimento dos métodos, é apresentada nas Figuras 5.1-5.3 uma ilustração gráfica para os métodos de nível proximal exato e inexato, quando o conjunto viável é um intervalo (compacto) da reta real.



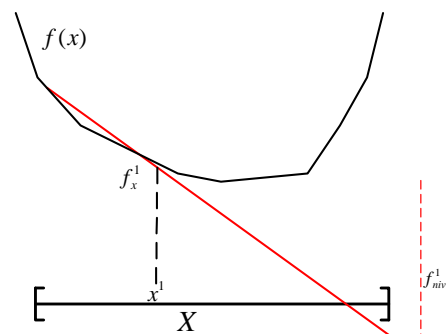
(a) Método exato ($k=1$)



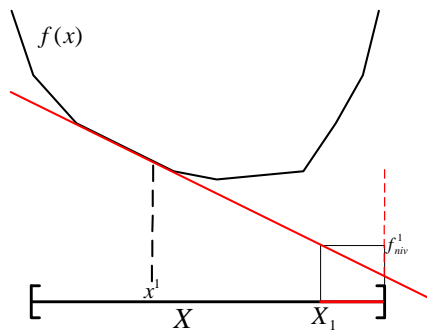
(b) Método inexacto ($k=1$)



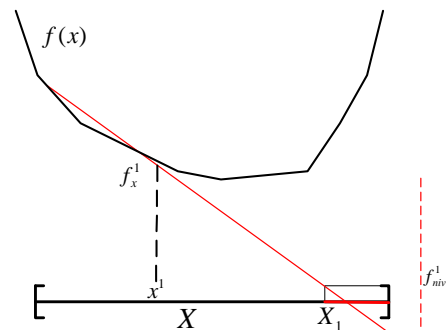
(c) Método exato (valor de nível)



(d) Método inexacto (valor de nível)

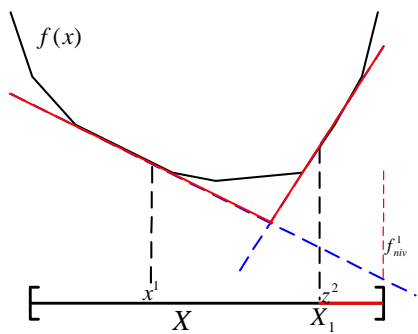


(e) Método exato (conj. de nível)

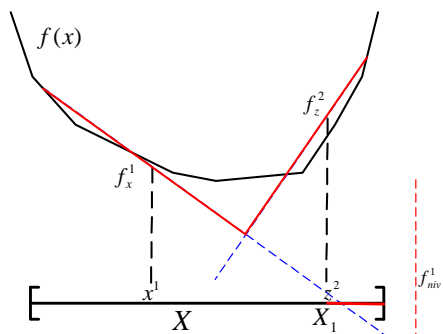


(f) Método inexacto (conj. de nível)

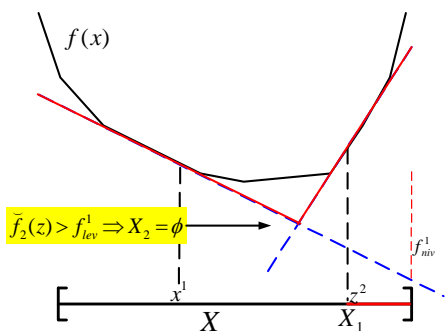
Figura 5.1: Métodos de nível proximal exato (esquerda) e inexacto (direita). Iteração 1.



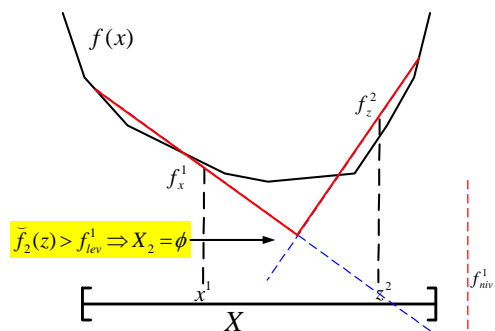
(a) Método exato ($k=2$)



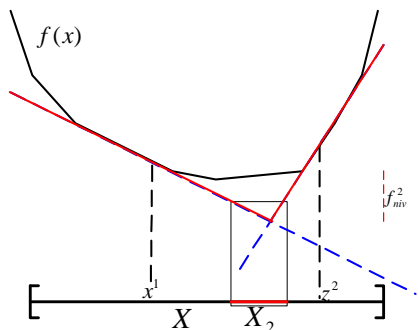
(b) Método inexato ($k=2$)



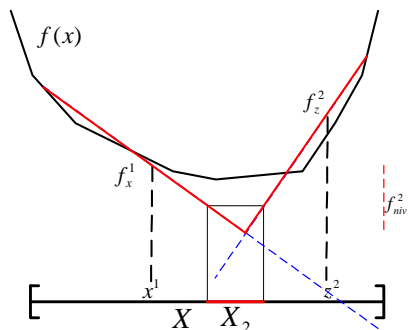
(c) Método exato (atualiza o valor de nível)



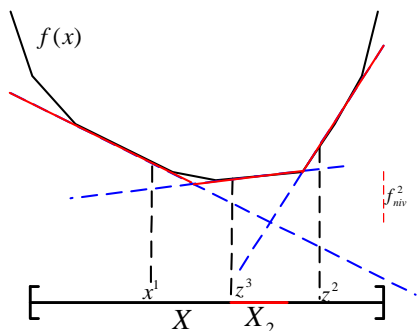
(d) Método inexato (atualiza o valor de nível)



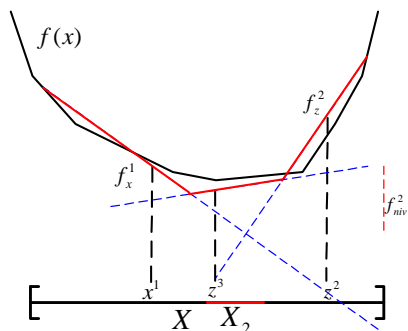
(e) Método exato (conj. de nível)



(f) Método inexato (conj. de nível)

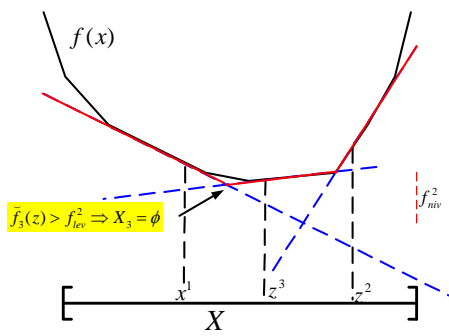


(g) Método exato ($k=3$)

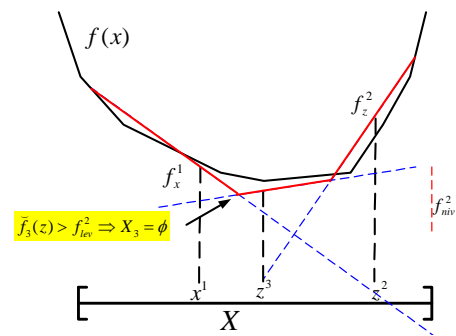


(h) Método inexato ($k=3$)

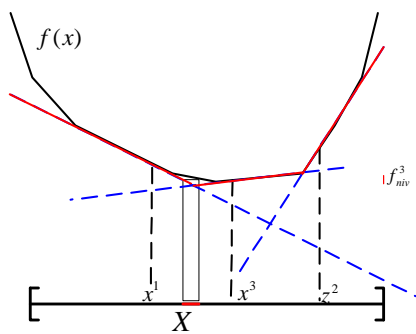
Figura 5.2: Métodos de nível proximal exato (esquerda) e inexato (direita). Iterações 2 e 3.



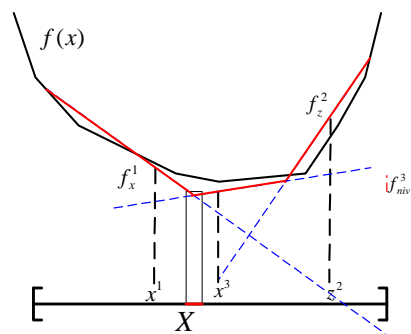
(a) Método exato (atualiza o valor de nível)



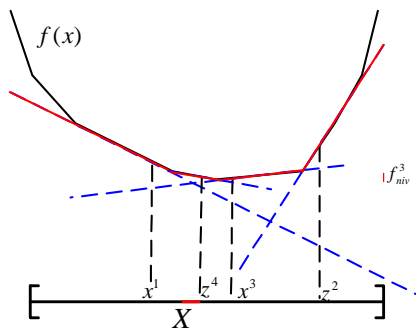
(b) Método inexacto (atualiza o valor de nível)



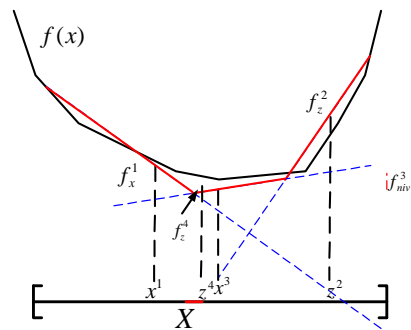
(c) Método exato (conj. de nível)



(d) Método inexacto (conj. de nível)



(e) Método exato (k=4)



(f) Método inexacto (k=4)

Figura 5.3: Métodos de nível proximal exato (esquerda) e inexacto (direita). Iteração 4.

O desenvolvimento teórico do método de nível proximal inexacto é detalhado a seguir.

5.2 Desenvolvimento Teórico

Dado um oráculo inexato satisfazendo (5.2), seja $\{z^i\}_{i=1}^k \subset \mathcal{X}$ uma sequência de pontos gerada pelo método, e seja o k -ésimo modelo de planos cortantes dado por

$$\check{f}_k(\cdot) := \max_{j \in J^k} f_j(\cdot), \quad \text{com } J^k \subset \{1, \dots, k\}. \quad (5.5)$$

Pela hipótese de convexidade (e por ser \mathcal{X} limitado), o valor

$$\check{f}_{min}^k := \min_{z \in \mathcal{X}} \check{f}_k(z) \quad (5.6)$$

é uma cota inferior finita para $f_* + \varepsilon_f + 2\varepsilon_g$ (porque o oráculo inexato (5.2) proporciona $\check{f}_k(\cdot) \leq f(\cdot) + \varepsilon_f + 2\varepsilon_g$, para todo k). Deste modo, o valor

$$f_{inf}^k := \max_{j \in \{1, \dots, k\}} \check{f}_{min}^j = \max_{j \in \{1, \dots, k\}} \left\{ \min_{z \in \mathcal{X}} \check{f}_j(z) \right\}, \quad (5.7)$$

é uma cota inferior do valor ótimo aproximado

$$f_{inf}^k \leq f_* + \varepsilon_f + 2\varepsilon_g. \quad (5.8)$$

Já o valor

$$f_{sup}^k := \min_{j \in \{1, \dots, k\}} f_z^j, \quad (5.9)$$

fornece uma cota superior para o valor ótimo aproximado. Seja $\bar{k} \in \{1, \dots, k\}$ o índice de iteração satisfazendo $f_z^{\bar{k}} = f_{sup}^k$. Então, pelo oráculo (5.2), as desigualdades

$$f_{sup}^k \geq f(z^{\bar{k}}) - \varepsilon_f \geq f_* - \varepsilon_f, \quad (5.10)$$

são satisfeitas. Cabe ressaltar que f_{sup}^k (resp. f_{inf}^k) é apenas uma estimativa do valor ótimo f_* , pois devido à imprecisão do oráculo, f_{sup}^k pode ser menor do que f_* (resp. $f_{inf}^k > f_*$).

A diferença entre as estimativas superior e inferior é representada pela *brecha de otimalidade inexata*

$$\Delta_k := f_{sup}^k - f_{inf}^k.$$

Pode-se notar por (5.9) e (5.7) (e comentários entre (5.16) e (5.17)) que a sequência $\{\Delta_k\}$ é monótona não crescente e, diferentemente do método de nível proximal exato proposto em [9], pode assumir valores negativos. No entanto, usando (5.10) e (5.8), tem-se que

$$\Delta_k = f_{sup}^k - f_{inf}^k \geq (f_* - \varepsilon_f) - (f_* + \varepsilon_f + 2\varepsilon_g) = -2(\varepsilon_f + \varepsilon_g). \quad (5.11)$$

Logo, se ε_f e ε_g são limitados, a brecha de otimalidade inexata é limitada inferiormente por -2ε , com $\varepsilon := \varepsilon_f + \varepsilon_g$.

Sejam $\kappa \in (0, 1)$ um parâmetro que define o valor do nível

$$f_{niv}^k := (1 - \kappa)f_{inf}^k + \kappa f_{sup}^k = f_{inf}^k + \kappa \Delta_k, \quad (5.12)$$

e o k -ésimo conjunto de nível definido por

$$X^k := \{z \in \mathcal{X} : \check{f}_k(z) \leq f_{niv}^k\}. \quad (5.13)$$

Dado $x^k \in \{z^j : j \in J^k\}$ o centro de estabilidade do método na iteração k , o novo iterado z^{k+1} do MNI resulta de projetar o centro de estabilidade no k -ésimo conjunto de nível

$$z^{k+1} := \arg \min_{z \in X^k} \|z - x^k\|^2, \quad (5.14)$$

com $\|\cdot\|$ a norma euclidiana. Deste modo, quando as avaliações do oráculo são exatas, o método coincide com o método de nível proximal exato desenvolvido em [9]. Adicionalmente, se $x^k = z^k$ e $J^k = \{1, \dots, k\}$ para toda iteração k , o método é idêntico ao método de nível introduzido em [19]. Finalmente, se κ fosse fixado igual a zero, o MNI coincidiria com o método de planos cortantes.

Seja \bar{k} o iterado satisfazendo $f_{sup}^{\bar{k}} = f_z^{\bar{k}}$. Então, por (5.8) tem-se que

$$\Delta_k = f_{sup}^k - f_{inf}^k \geq f_z^{\bar{k}} - (f_* + \varepsilon_f + 2\varepsilon_g),$$

que, junto com (5.2), implica a relação

$$f(z^{\bar{k}}) - \varepsilon_f \leq f_z^{\bar{k}} \leq f_* + \eta_k, \quad \text{com } \eta_k := \varepsilon_f + 2\varepsilon_g + \Delta_k. \quad (5.15)$$

Pela equação (5.11), $\eta_k \geq -\varepsilon_f$, e, se $\Delta_k \leq \delta_{\text{To1}}$ para δ_{To1} uma tolerância positiva, $\eta_k \in [-\varepsilon_f, \varepsilon_f + 2\varepsilon_g + \delta_{\text{To1}}]$. Assim sendo, por (5.15), $z^{\bar{k}}$ é uma $\varepsilon_f + \eta_k$ -solução do problema (5.1), com $|\eta_k|$ uniformemente limitado. Portanto, o critério $\Delta_k \leq \delta_{\text{To1}}$ pode ser usado para parar as iterações. Em particular, como a sequência $\{\Delta_k\}$ é monótona decrescente, o método para na primeira iteração para qual Δ_k torna-se negativo.

Segundo está apresentado, a cada iteração k o método resolve dois subproblemas: o programa linear (5.6) para definir \check{f}_{min}^k , e o problema quadrático (5.14) para definir o novo iterado z^{k+1} . No entanto, para o método de nível proximal (exato ou inexato), o programa linear (5.6) não precisa ser resolvido a cada iteração. Para isto, pode se manter o valor f_{inf}^k fixo enquanto o conjunto de nível X^k em (5.13) for não vazio. De fato, o conjunto de nível X^k resulta vazio quando $f_{niv}^k < \check{f}_k(\cdot)$. Neste caso, para definir um conjunto X^k não vazio, basta aumentar f_{inf}^k e atualizar o valor f_{niv}^k . Para

aumentar f_{inf}^k , pode-se simplesmente tomar $f_{inf}^k = f_{niv}^k$, ou resolver (5.6) para obter \check{f}_{min}^k , e então fixar $f_{inf}^k = \check{f}_{min}^k$, e atualizar f_{niv}^k por (5.12). Com este artifício, o esforço computacional requerido a cada iteração dos métodos de níveis proximais fica mais próximo dos métodos de feixes proximais.

Tem-se agora as condições necessárias para apresentar o algoritmo do MNI em detalhes.

5.3 Algoritmo Inexato

O algoritmo do método de nível proximal inexato - MNI - é essencialmente o Algoritmo 1 apresentado em [9], com a diferença que o oráculo utilizado é inexato.

Algoritmo 5.1 MÉTODO DE NÍVEL PROXIMAL INEXATO

Passo 0 (*inicialização*). Selecione $z^1 \in \mathcal{X}$, um parâmetro para a combinação convexa $\kappa \in (0, 1)$, uma tolerância $\delta_{\text{ToI}} \geq 0$ para o teste de parada, e defina $x^1 = z^1$. Calcule o par inexato (f_z^1, g_z^1) , faça $J^1 = \{1\}$, e defina o modelo \check{f}_1 . Faça $f_{sup}^0 = \infty$, $f_{inf}^1 = \check{f}_{min}^1 := \min_{z \in \mathcal{X}} \check{f}_1(z)$, $k = 1$, $\ell = 0$, e $k(0) = 1$ ($k(\ell)$ corresponde à iteração em que o valor f_{inf}^k é aumentado).

Passo 1 (*atualização do nível*). Determine $f_{sup}^k = \min\{f_z^k, f_{sup}^{k-1}\}$ e $\Delta_k = f_{sup}^k - f_{inf}^k$. Seja \bar{k} um índice de iteração tal que $f_{sup}^{\bar{k}} = f_z^{\bar{k}}$.

Passo 2 (*teste de parada*). Se $\Delta_k \leq \delta_{\text{ToI}}$, pare. O ponto $z^{\bar{k}}$ é uma solução aproximada para o problema (5.1).

Passo 3 (*viabilidade do conjunto de nível*). Determine f_{niv}^k por (5.12). Se o conjunto de nível X^k definido em (5.13) é não vazio, vá ao Passo 5; caso contrário, continue.

Passo 4 (*programa linear*). Resolva (5.6) e obtenha \check{f}_{min}^k . Faça $\hat{f}_{inf}^k = f_{inf}^k = \check{f}_{min}^k$, e escolha um centro de estabilidade $x^k \in \{z^j : j \in J^k\}$, de modo que $f_x^k = \min_{j \in J^k} f_z^j$. Defina $\hat{\Delta}_k = f_{sup}^k - \hat{f}_{inf}^k$, e faça $\ell = \ell + 1$, $k(\ell) = k$, e volte ao Passo 1.

Passo 5 (*programa quadrático*). Obtenha z^{k+1} resolvendo o programa (5.14), e seja ν^k o multiplicador ótimo associado.

Passo 6 (*cálculos inexatos*). Calcule f_z^{k+1} e g_z^{k+1} que satisfaçam (5.2).

Passo 7 (*seleção do feixe*). Escolha $J^{k+1} \supset \hat{J}^k \cup \{k+1\}$, com $\hat{J}^k := \{j \in J^k : \nu_j^k \neq 0\}$.

Passo 8 (*ciclo*). Se $k > k(\ell)$, defina $\hat{\Delta}_k = \Delta_k$ e $\hat{f}_{inf}^k = f_{inf}^k$. Faça $x^{k+1} = x^k$, $f_{inf}^{k+1} = f_{inf}^k$, $k = k + 1$, e volte ao Passo 1.

□

Dada a iteração \bar{k} determinada pelo Algoritmo 5.1, o ponto $z^{\bar{k}}$ é, por (5.15), uma $2(\varepsilon_f + \varepsilon_g) + \delta_{\text{Tot}}$ -solução do problema (5.1).

Segundo comentado por Kiwiel [9, p. 92], muitos programas computacionais especializados em problemas quadráticos fornecem automaticamente $|\hat{J}^k| \leq n$, com $n = \dim(z)$. Portanto, no Passo 7 do Algoritmo 5.1 pode-se sempre escolher J^{k+1} tal que $|J^{k+1}| \leq n + 1$. Se n for grande, o tamanho do feixe $|J^{k+1}|$ do MNI pode ser mantido menor do que $n + 1$, efetuando a compressão do feixe, [2, Algoritmo 10.11] como nos métodos de feixes. No entanto, é importante notar que, diferentemente dos métodos de feixes proximais, o multiplicador ν^k obtido no Passo 5 tem componentes positivas, mas não é simplicial. Deste modo, para efetuar a compressão do feixe é necessário normalizar ν^k , i.e., dividir ν^k pelo somatório $\sum_{j \in J^k} \nu_j^k$.

Para descobrir que o conjunto X^k é vazio, no Passo 3 tenta-se resolver o problema quadrático (5.14). Caso seja possível encontrar z^{k+1} o Passo 5 é, por consequência, realizado. Caso contrário, o Passo 4 deve ser acessado. Quando $f_{inf}^k = \min_{z \in \mathcal{X}} \check{f}_k(z)$, a desigualdade $f_{inf}^k \leq f_{niv}^k$ é sempre verificada por (5.12). Então, imediatamente após a execução do Passo 4, o conjunto de nível X^k é sempre não vazio. Como resultado, o Passo 4 é acessado, no máximo, uma única vez a cada iteração k .

Pelo Passo 4, as iterações geradas pelo Algoritmo 5.1 podem ser divididas em ciclos, definindo a ℓ -ésima subsequência por

$$K^\ell := \{k(\ell), \dots, k(\ell + 1) - 1\}, \quad \text{para } \ell \geq 0.$$

Cada subsequência com índices em K^ℓ termina na iteração $k = k(\ell + 1)$, ao acessar o Passo 4. Portanto, durante o ℓ -ésimo ciclo o valor f_{inf}^j permanece constante, e f_{niv}^j é não crescente. Para todo $j \in K^\ell$

$$f_{inf}^{k(\ell)} = f_{inf}^j \quad \text{e} \quad f_{niv}^{k(\ell)} \geq f_{niv}^j \geq f_{niv}^k, \quad \text{se } k(\ell) < j \leq k \leq k(\ell + 1) - 1, \quad (5.16)$$

com $\ell \geq 0$. Além disso, o valor f_{inf}^k aumenta a cada passagem pela Passo 4 e, deste modo, a sequência $\{\Delta_k\}$ é monótona não crescente.

É importante ressaltar que, imediatamente após o Passo 4 ser acessado, a desigualdade $\hat{f}_{inf}^k \geq f_{niv}^k = f_{inf}^{k(\ell-1)} + \kappa \Delta_k$ é válida para todo $\ell \geq 1$ (porque $X^k = \emptyset$). Então,

$$\hat{f}_{inf}^k - f_{sup}^k \geq f_{inf}^{k(\ell-1)} - f_{sup}^k + \kappa \Delta_k = -\Delta_k + \kappa \Delta_k = \Delta_k(\kappa - 1),$$

que implica a desigualdade

$$\Delta_k \geq \frac{\hat{\Delta}_k}{1 - \kappa},$$

onde $\hat{\Delta}_k = f_{sup}^k - \check{f}_{inf}^k$ foi definido no Passo 4. Para todo $\ell \geq 0$, o Passo 4 define $k(\ell + 1) = k$. Logo, usando a definição de $\hat{\Delta}_k$ e a monotocidade de Δ_k , resulta que a sequência $\{\hat{\Delta}_k\}$ também é monótona decrescente e

$$(1 - \kappa)\Delta_k \geq \hat{\Delta}_{k(\ell+1)}, \quad \text{para todo } k \in K^\ell \text{ e } \ell \geq 0.$$

Em particular, como $k(\ell) \in K^\ell$ quando $K^\ell \neq \emptyset$,

$$(1 - \kappa)\Delta_{k(\ell)} \geq \Delta_{k(\ell+1)}, \quad \text{para todo } \ell \geq 0, \quad (5.17)$$

porque pelo Passo 8, $\hat{\Delta}_k = \Delta_k$ para todo k .

As relações acima são importantes para análise de convergência do Algoritmo 5.1, dada a seguir.

5.3.1 Análise de Convergência

Ao longo desta seção, sejam $\kappa \in (0, 1)$ um parâmetro do Algoritmo 5.1, Λ uma constante de Lipschitz para f em \mathcal{X} , e D o diâmetro do conjunto compacto $\mathcal{X} \neq \emptyset$. A análise de convergência do Algoritmo 5.1 apresentada nesta seção é baseada em [9, Seção 3], mas considerando o oráculo inexato (5.2).

É suposto para os resultados seguintes que $\delta_{\text{To1}} = 0$, e o Algoritmo 5.1 não termina. É importante mencionar que se $\delta_{\text{To1}} = 0$ e o Algoritmo 5.1 para, então o ponto $z^{\bar{k}}$ é uma 2ε -solução do problema (5.1), por (5.15).

Dado $x \in \mathcal{R}^n$, a projeção de x sobre o conjunto convexo $\mathcal{X} \subset \mathcal{R}^n$ não vazio, denominada $P_{\mathcal{X}}(x)$, satisfaz

$$\|P_{\mathcal{X}}(x) - z\|^2 \leq \|x - z\|^2 - \|P_{\mathcal{X}}(x) - x\|^2 \quad \text{para todo } z \in \mathcal{X}. \quad (5.18)$$

Lema 5.1 *Os iterados do Algoritmo 5.1 satisfazem as relações*

$$\begin{aligned} \|z^{k+1} - z^k\| &\geq \frac{(1 - \kappa)\Delta_k}{\Lambda} \quad \text{se } k > k(\ell), \text{ e} \\ \|z^{k+1} - x^k\| &\geq \frac{(1 - \kappa)\Delta_k}{\Lambda} \quad \text{se } k = k(\ell). \end{aligned}$$

Prova. Para todo $j \in J^k$, tem-se que $\check{f}_k(z^j) = f_z^j \geq f_{sup}^k$ por construção. Além do mais, pelo Passo 5, $z^{k+1} \in X^k$ e, portanto, $\check{f}_k(z^{k+1}) \leq f_{niv}^k$. O modelo de planos cortantes \check{f}_k é uniformemente Lipschitz contínuo com constante Λ , por (5.4). Então,

para todo $j \in J^k$

$$\Lambda \|z^j - z^{k+1}\| \geq \check{f}_k(z^j) - \check{f}_k(z^{k+1}) \geq f_{sup}^k - f_{niv}^k = f_{sup}^k - (f_{inf}^k + \kappa \Delta_k) = (1 - \kappa) \Delta_k.$$

Como $k \in J^k$ pelo Passo 7, as relações valem em particular para $j = k$.

Além disso, se $k = k(\ell)$, pelo Passo 4 tem-se que $x^k = z^j$, para algum índice $j \in J^k$. Então, a relação também é verificada com x^k em vez de z^j . ■

Lema 5.2 *Considere o Algoritmo 5.1 e seja $\ell \geq 0$. Se $k \in K^\ell$ é tal que $k > k(\ell)$ no Passo 5 do Algoritmo 5.1, então $x^k = x^{k-1}$, e*

$$\|z^{k+1} - x^k\|^2 \geq \|z^k - x^k\|^2 + \|z^{k+1} - z^k\|^2.$$

Prova. Pelo Passo 4, $k > k(\ell)$ implica que $x^k = x^{k-1}$ para todo $k \in K^\ell$.

O iterado z^k é obtido projetando x^{k-1} em X^{k-1} , i.e., $z^k = P_{X^{k-1}}(x^{k-1})$. Este problema de projeção pode ser escrito como

$$\begin{cases} \min & \|z - x^{k-1}\|^2 \\ \text{s.a} & z \in \mathcal{X} \\ & f_j(z) \leq f_{niv}^{k-1}, \quad \forall j \in J^{k-1}. \end{cases}$$

Excluindo as restrições inativas, o problema acima é equivalente, em termos de solução, ao problema (possivelmente de porte menor)

$$\begin{cases} \min & \|z - x^{k-1}\|^2 \\ \text{s.a} & z \in \mathcal{X} \\ & f_j(z) \leq f_{niv}^{k-1}, \quad \forall j \in \hat{J}^{k-1}. \end{cases}$$

Assim sendo, definindo o conjunto de nível

$$\hat{X}^{k-1} := \{z \in \mathcal{X} : \hat{f}_{k-1}(z) \leq f_{niv}^{k-1}\}, \quad \text{com} \quad \hat{f}_{k-1}(\cdot) := \max_{j \in \hat{J}^{k-1}} f_j(\cdot),$$

tem-se que $z^k = P_{X^{k-1}}(x^{k-1}) = P_{\hat{X}^{k-1}}(x^{k-1})$.

Pelo Passo 7, $\hat{J}^{k-1} \subset J^k$. Então, $\check{f}_k \geq \hat{f}_{k-1}$, e $X^k \subset \hat{X}^{k-1}$, porque f_{niv}^k é não crescente para $k \in K^\ell$ tal que $k > k(\ell)$. Como $z^{k+1} \in X^k$, então $z^{k+1} \in \hat{X}^{k-1}$. O resultado desejado é obtido aplicando (5.18) com $\mathcal{X} = \hat{X}^{k-1}$, $x = x^{k-1} (= x^k)$, $z^k = P_{\hat{X}^{k-1}}(x^{k-1})$, e $z = z^{k+1} \in \hat{X}^{k-1}$. ■

O resultado seguinte determina um limite superior para o número de iterações em K^ℓ .

Lema 5.3 *Considere o Algoritmo 5.1, e suponha que $\Delta_k > \delta_{\text{ToI}}$ para $k \in K^\ell$ com $\ell \geq 0$. O número de iterações $k - k(\ell) + 1$, realizadas pelo Algoritmo 5.1 entre as iterações $k(\ell)$ e k , não excede*

$$\left(\frac{D\Lambda}{(1 - \kappa)\Delta_k} \right)^2.$$

Prova. Considere $k \in K^\ell$ com $k \geq k(\ell)$. Como $\Delta_k > \delta_{\text{ToI}}$, o Algoritmo 5.1 não para, sendo $x^k = x^{k(\ell)} \in \mathcal{X}$. Pelo Lema 5.2,

$$\begin{aligned} \|z^{k+1} - x^k\|^2 &\geq \|z^k - x^k\|^2 + \|z^{k+1} - z^k\|^2 \\ &= \|z^k - x^{k-1}\|^2 + \|z^{k+1} - z^k\|^2 \\ &\geq \|z^{k-1} - x^{k-1}\|^2 + \|z^k - z^{k-1}\|^2 + \|z^{k+1} - z^k\|^2 \\ &\vdots \\ &\geq \|z^{k(\ell)+1} - x^{k(\ell)}\|^2 + \sum_{j=k(\ell)+1}^k \|z^{j+1} - z^j\|^2, \quad \text{para todo } k \in K^\ell, \ell \geq 0. \quad (*) \end{aligned}$$

Além disso, o diâmetro do conjunto X^k é menor ou igual a D , porque $X^k \subset \mathcal{X}$. Então, usando o Lema 5.1, a desigualdade (*) acima resulta na relação

$$\begin{aligned} D^2 &\geq \left(\frac{(1 - \kappa)\Delta_{k(\ell)}}{\Lambda} \right)^2 + \sum_{j=k(\ell)+1}^k \left(\frac{(1 - \kappa)\Delta_j}{\Lambda} \right)^2 \\ &\geq \left(\frac{(1 - \kappa)\Delta_k}{\Lambda} \right)^2 (k - k(\ell) + 1), \end{aligned}$$

porque $\Delta_k \leq \Delta_j$ para todo $j \leq k$, pela monotonia da sequência $\{\Delta_k\}$. Então,

$$\left(\frac{D\Lambda}{(1 - \kappa)\Delta_k} \right)^2 \geq (k - k(\ell) + 1)$$

como desejado. ■

A convergência do Algoritmo 5.1 segue do seguinte teorema.

Teorema 5.1 *Para que o Algoritmo 5.1 determine uma brecha de otimalidade menor do que $\delta_{\text{ToI}} > 0$ é suficiente realizar, no máximo,*

$$c(\kappa) \left(\frac{D\Lambda}{\delta_{\text{ToI}}} \right)^2$$

iteraões, onde $c(\kappa)$ é uma constante que depende somente do parâmetro κ escolhido no Passo 0 do Algoritmo 3.1.

Prova. Seja o conjunto de iterações $K(\delta_{\text{To1}}) := \{1, \dots, k_{\delta_{\text{To1}}}\} \subset \cup_{\ell=0}^m K^\ell$, tal que

$$\Delta_k \geq \delta_{\text{To1}} \quad \text{para todo } k \in K(\delta_{\text{To1}}).$$

Então, por (5.17): $(1 - \kappa)\Delta_{k(\ell)} \geq \Delta_{k(\ell+1)}$ para $\ell = 0, \dots, m - 1$, e $\Delta_{k(m)} \geq \delta_{\text{To1}}$. Como $\Delta_{k(\ell)} \geq 0$,

$$(1 - \kappa)^{m-\ell} \Delta_k \geq \delta_{\text{To1}}, \quad \text{para todo } k \in K^\ell \cap K(\delta_{\text{To1}}), \quad \text{e } \ell = 0, \dots, m.$$

Segue do Lema 5.3 que

$$|K^\ell \cap K(\delta_{\text{To1}})| \leq \left(\frac{D\Lambda}{(1 - \kappa)\Delta_k} \right)^2 \leq \left(\frac{D\Lambda(1 - \kappa)^{m-\ell}}{(1 - \kappa)\delta_{\text{To1}}} \right)^2, \quad \text{para todo } \ell = 0, \dots, m.$$

Logo,

$$k_{\delta_{\text{To1}}} = \sum_{\ell=0}^m |K^\ell \cap K(\delta_{\text{To1}})| \leq \sum_{\ell=0}^m \left(\frac{D\Lambda(1 - \kappa)^{m-\ell}}{(1 - \kappa)\delta_{\text{To1}}} \right)^2 \leq \left(\frac{D\Lambda}{\delta_{\text{To1}}} \right)^2 \frac{1}{(1 - (1 - \kappa)^2)(1 - \kappa)^2},$$

e o número máximo de iterações $k_{\delta_{\text{To1}}}$, tal que $\Delta_{k_{\delta_{\text{To1}}}} \geq \delta_{\text{To1}}$, não pode ser maior do que

$$c(\kappa) \left(\frac{D\Lambda}{\delta_{\text{To1}}} \right)^2, \quad \text{com } c(\kappa) = \frac{1}{(1 - (1 - \kappa)^2)(1 - \kappa)^2}.$$

■

Em termos de número de iterações, o método de nível proximal inexato proposto no Algoritmo 5.1 tem a mesma complexidade do método de nível proximal exato desenvolvido em [9]. Para verificar isto, basta tomar $\lambda = 1 - \kappa$ na função $c(\cdot)$ definida acima, e comparar com [9, Corolário 3.6].

Finaliza-se este capítulo mencionando que o Algoritmo 5.1 não realiza, necessariamente, as escolhas em princípio mais naturais, a saber:

- (i) $f_{inf}^k = \check{f}_{min}^k$ para toda iteração k ;
- (ii) $x^k = z^{\bar{k}}$, com $\bar{k} := \arg \min_{j \in \{1, \dots, k\}} f_z^j$, para toda iteração k .

Quando o oráculo é exato, o *método de nível* desenvolvido em [19] emprega a regra (i), mas não utiliza a regra (ii). Já o *método de planos cortantes com estabilização por nível* apresentado em [6, Capítulo XV], utiliza a regra (ii), mas não o princípio (i). A versão analisada neste capítulo, desenvolvida a partir de [9], modifica (i) e (ii) incorporando o conceito de ciclos.

A seguir o método de nível inexato é estendido a oráculos parcialmente inexatos.

Capítulo 6

Método de Nível Parcialmente Inexato

Neste capítulo é proposto um *método de nível parcialmente inexato* - MNPI - que, assim como o MFPI apresentado no Capítulo 4, utiliza linearizações da função objetivo que podem ser exatas ou inexatas.

O MFPI exige a avaliação exata da função e de um subgradiente para todo candidato a um passo sério. Como o oráculo parcialmente inexato deve fornecer estimativas inferiores do valor exato da função, a eficiência do MFPI é dependente da qualidade do oráculo parcialmente inexato disponível.

Com o intuito de se desenvolver um método parcialmente inexato que seja menos sensível à qualidade do oráculo, o MNPI proposto neste capítulo não é um método proximal. Porém, pertence à família dos métodos de feixes e é fundamentado no *método de nível*, desenvolvido por Lemaréchal, Nemirovskii e Nesterov [19].

Diferentemente dos métodos de feixes proximais, o método de nível não possui um mecanismo de compressão do feixe, e por isso o subproblema que define o iterado tem um número crescente de restrições. No entanto, pode ser considerado um procedimento que permite eliminar os cortes inativos sem que a convergência do método seja comprometida, como apresentado a seguir.

6.1 Considerações Iniciais

Assim como no Capítulo 5, são considerados neste capítulo problemas da forma

$$f_* := \min_{z \in \mathcal{X}} f(z), \quad (6.1)$$

com a função fechada e convexa $f : \mathcal{R}^n \rightarrow \mathcal{R} \cup \infty$. O conjunto $\mathcal{X} \subset \mathcal{R}^n$ é suposto ser convexo e compacto, com diâmetro D não necessariamente conhecido. Além disso, é assumido que $\emptyset \neq \mathcal{X} \subseteq \text{ri}(\text{dom } f)$. Dadas estas hipóteses, então a função f

é Lipschitziana em \mathcal{X} .

Ao longo deste capítulo é suposto que calcular o valor da função $f(\cdot)$ e um subgradiente $g(\cdot) \in \partial f(\cdot)$ envolve um esforço computacional elevado, mas não impossível. Por este motivo, o método proposto considera um oráculo parcialmente inexato que fornece valores exatos ou inexatos, dependendo de um parâmetro $\zeta_{inex} \in \{0, 1\}$ informado. Mais especificamente,

para todo $z \in \mathcal{X}$ dado, existe uma cota superior $0 \leq \varepsilon_f < \infty$ (independente de z , mas possivelmente desconhecida), tal que

$$\begin{aligned} \text{se } \zeta_{inex} = 1 \text{ calcula-se } & \begin{cases} \text{uma estimativa da função} & f_z \leq f(z) \\ \text{uma estimativa do subgradiente} & g_z \in \partial_{\varepsilon_z} f(z), \\ \text{com } \varepsilon_z := f(z) - f_z \leq \varepsilon_f, \end{cases} \\ \text{se } \zeta_{inex} = 0 \text{ calcula-se } & \begin{cases} \text{o valor exato da função} & f(z) \\ \text{um subgradiente} & g(z) \in \partial f(z). \end{cases} \end{aligned} \quad (6.2)$$

Após chamar o oráculo para $z^k \in \mathcal{X}$ um ponto dado, o valor da função

$$f_z^k := \begin{cases} f_{z^k} & \text{se } \zeta_{inex} = 1 \\ f(z^k) & \text{se } \zeta_{inex} = 0 \end{cases}$$

e o subgradiente

$$g_z^k := \begin{cases} g_{z^k} & \text{se } \zeta_{inex} = 1 \\ g(z^k) & \text{se } \zeta_{inex} = 0 \end{cases}$$

forneem a linearização f_k , que será exata ou inexata segundo o valor de ζ_{inex} :

$$f_k(\cdot) := f_z^k + (\cdot - z^k)^\top g_z^k \leq f(\cdot). \quad (6.3)$$

A linearização satisfaz a condição de Lipschitz com constante uniforme Λ , i.e.,

$$|f_k(\tilde{z}) - f_k(\bar{z})| \leq \Lambda \|\tilde{z} - \bar{z}\|, \quad \text{para todo } \tilde{z}, \bar{z} \in \mathcal{X}. \quad (6.4)$$

Serão apresentados nos Capítulos 7 e 8 oráculos que satisfazem as exigências de (6.2). A seguir são descritas as principais diferenças entre o método de nível proximal inexato do Capítulo 5, e o método de nível parcialmente considerado neste capítulo.

6.1.1 Comparação entre os Métodos Inexato e Parcialmente Inexato

Novamente, analisam-se as diferenças entre os métodos com uma síntese de seus respectivos algoritmos:

<ul style="list-style-type: none"> • Método de Nível Proximal Inexato <ol style="list-style-type: none"> 1. $J^k \subset \{1, \dots, k\}, \{(z^i, f_z^i, g_z^i)\}_{i \in J^k}$ 2. $\check{f}_k(\cdot) = \max_{j \in J^k} \{f_z^j + g_z^{j\top}(\cdot - z^j)\}$ 3. $X^k = \{z \in \mathcal{X} : \check{f}_k(z) \leq f_{niv}^k\}$ e $f_{niv}^k = \kappa f_{inf}^k + (1 - \kappa) f_{sup}^k$ 4. Se $X^k = \emptyset$ atualize x_k e f_{inf}^k e volte ao passo 3 5. $z^{k+1} = \arg \min_{z \in X^k} \ z - x^k\ ^2$, 6. cálculo de $f_z^{k+1} \in [f(z^{k+1}) - \varepsilon_f, f(z^{k+1}) + \varepsilon_g]$ e $g_z^{k+1} \in \partial_{\varepsilon_f + \varepsilon_g} f(z^{k+1})$ 7. $f_{sup}^{k+1} = \min\{f_{sup}^k, f_z^{k+1}\}$, $k = k + 1$ e volte ao passo 1. <ul style="list-style-type: none"> • $\mathcal{X} \subseteq \text{ri}(\text{dom } f)$ deve ser compacto. • todo ponto de acumulação de $\{x^k\}$ é uma $2(\varepsilon_f + \varepsilon_g)$-solução. 	<ul style="list-style-type: none"> • Método de Nível Parcialmente Inexato <ol style="list-style-type: none"> 1. $J^k \subset \{1, \dots, k\}, \{(z^i, f_z^i, g_z^i)\}_{i \in J^k}$ 2. $\check{f}_k(\cdot) = \max_{j \in J^k} \{f_z^j + g_z^{j\top}(\cdot - z^j)\}$ 3. $X^k = \{z \in \mathcal{X} : \check{f}_k(z) \leq f_{niv}^k\}$ e $f_{inf}^k = \min_{z \in \mathcal{X}} \check{f}_k(z)$, $f_{niv}^k = \kappa f_{inf}^k + (1 - \kappa) f_{sup}^k$ 4. $z^{k+1} = \arg \min_{z \in X^k} \ z - z^k\ ^2$, 5. cálculo de $f_z^{k+1} \in [f(z^{k+1}) - \varepsilon_f, f(z^{k+1})]$ e $g_z^{k+1} \in \partial_{\varepsilon_f} f(z^{k+1})$ 6. Se $f_z^{k+1} < f_{inf}^k$, então calcule $(f_z^{k+1}, g_z^{k+1}) = (f(z^{k+1}), g(z^{k+1}))$, $f_{sup}^{k+1} = \min\{f_{sup}^k, f(z^{k+1})\}$; caso contrário, $f_{sup}^{k+1} = f_{sup}^k$ 7. $k = k + 1$ e volte ao passo 1. <ul style="list-style-type: none"> • $\mathcal{X} \subseteq \text{ri}(\text{dom } f)$ deve ser compacto. • todo ponto de acumulação de $\{x^k\}$ é uma solução ótima.
---	---

O algoritmo do MNPI é apresentado com detalhes na Seção 6.3. Apesar de se utilizar a mesma notação $\{(z^i, f_z^i, g_z^i)\}_{i \in J^k}$ para o feixe de informações no item 1, é importante ressaltar que, para o método parcialmente inexato alguns dos pares (f_z^i, g_z^i) são calculados de forma exata, quando a imprecisão do oráculo é excessiva, e o método fixa $\zeta_{inex} = 0$.

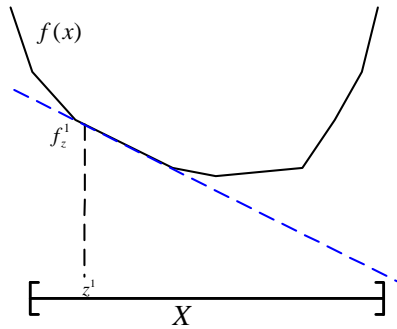
Por (6.3), pode-se assumir que a imprecisão do oráculo é excessiva quando $f_z^{k+1} \leq \check{f}_k(z^{k+1})$. No entanto, uma condição mais fraca para a identificação da imprecisão excessiva é verificar a desigualdade $f_z^{k+1} \leq f_{inf}^k$, como considerado no item 6 na comparação acima.

Para o MNPI, como no item 3 o valor f_{inf}^k é atualizado a cada iteração, o conjunto de nível X^k é sempre não vazio. Outra diferença importante entre os dois métodos considerados consiste na determinação do iterado z^{k+1} : enquanto o MNI projeta um centro de estabilidade no conjunto de nível, o MNPI projeta o último iterado (conferir com os comentários finais do Capítulo 5).

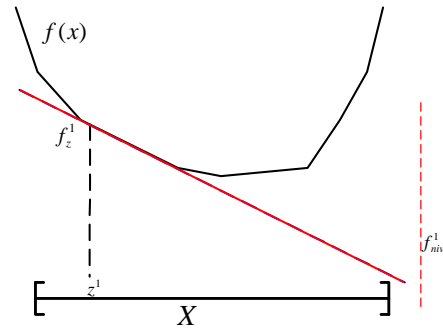
O MNPI procura combinar a velocidade do MNI com a convergência exata do

método de nível exato. Para que este propósito seja alcançado, é necessário empregar oráculos parcialmente inexatos rápidos e de boa qualidade.

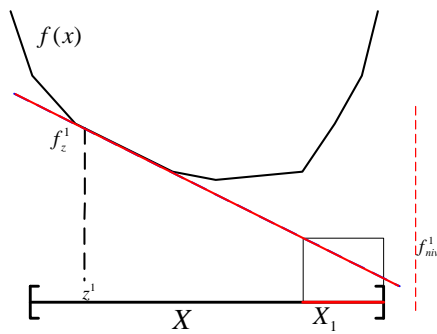
As Figuras 6.1 e 6.2 contem uma ilustração gráfica para o MNPI, quando o conjunto viável é um intervalo (compacto) da reta real.



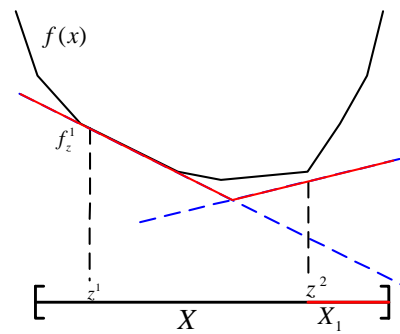
(a) Método parcialmente inexato (iteração 1)



(b) Método parcialmente inexato (valor de nível)



(c) Método parcialmente inexato (conj. de nível)



(d) Método parcialmente inexato (iteração 2)

Figura 6.1: Método de nível parcialmente inexato. Iterações 1 e 2.

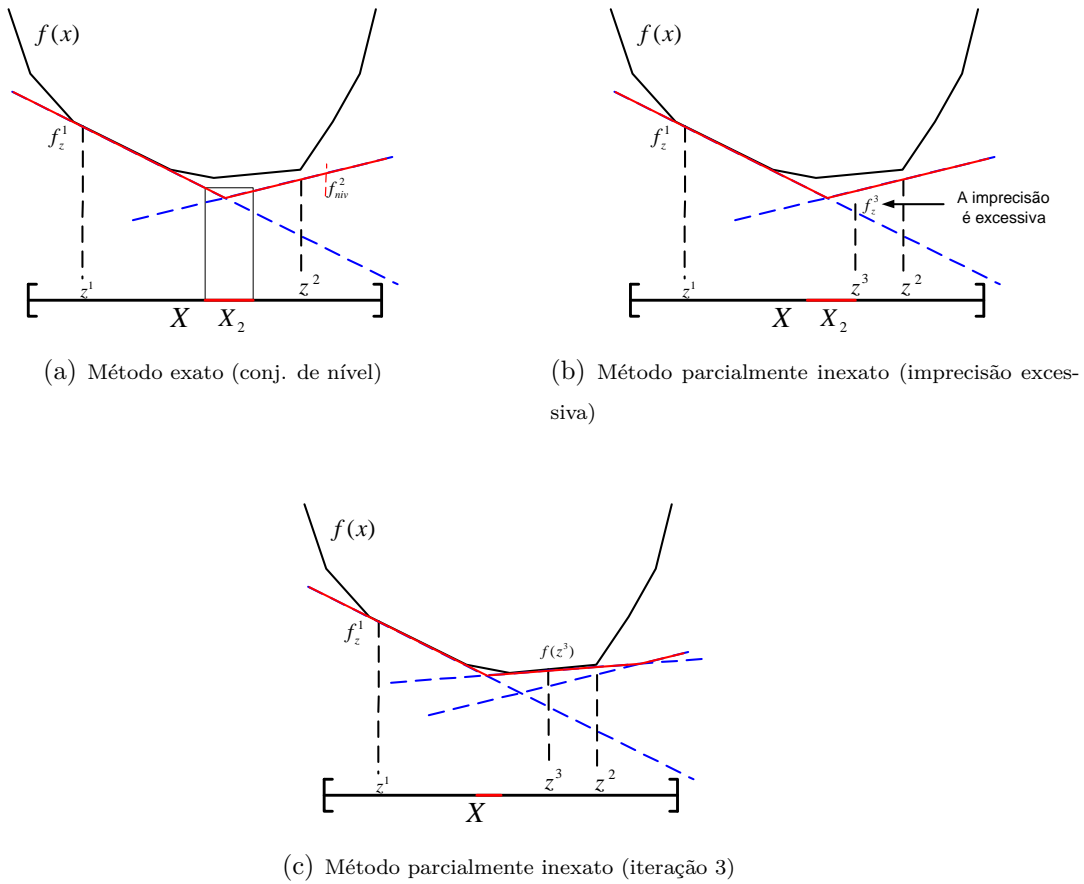


Figura 6.2: Método de nível parcialmente inexato. Iteração 3.

A seguir é dada a descrição do método de nível parcialmente inexato.

6.2 Desenvolvimento Teórico

Seja k um contador de iteração. Dado um oráculo inexato satisfazendo (6.2), e dada uma sequência de pontos $\{z^i\}_{i=1}^k \subset \mathcal{X}$ gerada pelo algoritmo até a iteração k , o seguinte modelo de planos cortantes

$$\check{f}_k(\cdot) := \max_{j \in J^k} f_j(\cdot), \quad \text{com } J^k \subset \{1, \dots, k\}, \quad (6.5)$$

satisfaz a identidade $\check{f}_k(z^j) = f_z^j$ e, por (6.3), $\check{f}_j(\cdot) \leq f(\cdot)$ para todo $j \in J^k$.

O método de nível parcialmente inexato define o melhor limite superior do valor ótimo como sendo

$$f_{sup}^k := \min_{j \in \{j \leq k: f_z^j = f(z^j)\}} f_z^j, \quad (6.6)$$

enquanto que o melhor limite inferior (até a iteração k) é dado por

$$\check{f}_{min}^k := \min_{z \in \mathcal{X}} \check{f}_k(z) \quad \text{com} \quad f_{inf}^k = \check{f}_{min}^k. \quad (6.7)$$

Diferentemente do MNI do Capítulo 5, para identificar os cortes inativos e então possivelmente descartá-los, o programa linear (6.7) precisa ser resolvido a cada iteração do MNPI.

A diferença entre os limites superior e inferior define a *brecha de otimalidade* $\Delta_k = f_{sup}^k - f_{inf}^k$, que é sempre não negativa por (6.2), (6.6), e (6.7). Seja x^* uma solução do problema (6.1). A partir das definições acima obtém-se que

$$f_{inf}^k \leq \check{f}_k(x^*) \leq f(x^*) \leq f_{sup}^k, \quad \text{para todo } k \geq 1.$$

Assim sendo, $\Delta_k = 0$ implica em particular que $f_{sup}^k = f(x^*)$, com o qual o ponto $z^{\bar{k}}$ tal que $f_{sup}^k = f_{z^{\bar{k}}}^k = f(z^{\bar{k}})$ é uma solução ótima de (6.1). Então, um bom critério de parada para o MNPI é tomar $\Delta_k \leq \delta_{\text{To1}}$, para $\delta_{\text{To1}} \geq 0$ uma tolerância dada.

Sejam $\kappa \in (0, 1)$ um parâmetro que define o valor do nível

$$f_{niv}^k := (1 - \kappa)f_{inf}^k + \kappa f_{sup}^k = f_{inf}^k + \kappa \Delta_k, \quad (6.8)$$

e o k -ésimo conjunto de nível definido por

$$X^k := \{z \in \mathcal{X} : \check{f}_k(z) \leq f_{niv}^k\}. \quad (6.9)$$

O novo iterado z^{k+1} do método de nível parcialmente inexato resulta em projetar o último iterado z^k (ao invés de projetar o centro de estabilidade x^k , como no MNI) no k -ésimo conjunto de nível

$$z^{k+1} := \arg \min_{z \in X^k} \|z - z^k\|^2. \quad (6.10)$$

Uma questão fundamental para o MNPI é quando fixar ζ_{inex} para calcular os valores exatos da função $f(z^{k+1})$ e de um subgradiente $g(z^{k+1}) \in \partial f(z^{k+1})$. Uma ideia natural é calcular uma linearização exata $f_k(\cdot) = f(z^{k+1}) + (\cdot - z^{k+1})^\top g(z^{k+1})$ quando a imprecisão $\varepsilon_z^{k+1} = f(z^{k+1}) - f_z^{k+1}$ for inaceitável. Mas esta imprecisão é desconhecida, porque o valor $f(z^{k+1})$ é desconhecido. No entanto, é possível verificar quando a imprecisão é inaceitável usando a propriedade de convexidade: suponha que um oráculo satisfazendo (6.2) seja empregado pelo método; então $\check{f}_k(\cdot) \leq f(\cdot)$ pela convexidade de f , e a imprecisão ε_z^{k+1} pode ser declarada inaceitável quando

$$f_z^{k+1} < \check{f}_k(z^{k+1}). \quad (6.11)$$

Quando o modelo \check{f}_k estiver bastante próximo de f , a desigualdade acima não necessariamente implica que a imprecisão do oráculo seja grande. Em outras palavras, admitir uma linearização inexata pode ainda ser uma boa opção para aprimorar o modelo de planos cortantes \check{f}_k , com um custo computacional relativamente baixo. Assim sendo, para determinar a necessidade de construir uma linearização exata será usado um teste menos restritivo:

$$f_z^{k+1} < \check{f}_{min}^k, \quad (6.12)$$

com $\check{f}_{min}^k = f_{inf}^k$ definido em (6.7). Cabe ressaltar que os critérios (6.11) ou (6.12) para identificar a imprecisão excessiva não são suficientes para garantir que o teste de parada $\Delta_k \leq \delta_{\text{To1}}$ seja satisfeito. Por este motivo, o MNPI calcula uma linearização exata da função f quando

$$f_z^{k+1} < \check{f}_{min}^k, \quad \text{ou} \quad \left(1 - \frac{\Delta_k}{\Delta_{k-1}}\right) \leq \epsilon_\Delta, \quad (6.13)$$

para $\epsilon_\Delta > 0$ uma tolerância dada para a imprecisão do oráculo inexato. Mais especificamente, como a sequência positiva $\{\Delta_k\}$ é monótona não crescente por (6.6) e (6.7), tem um limite $\Delta_* \geq 0$. Se $\Delta_* > 0$, então a razão $\left(1 - \frac{\Delta_k}{\Delta_{k-1}}\right) \rightarrow 0$ e pelo critério em (6.13), uma linearização exata é construída e adicionada ao modelo \check{f}_k , evitando deste modo que a imprecisão do oráculo prejudique o processo de convergência.

Tem-se agora as condições necessárias para apresentar o algoritmo do MNPI.

6.3 Algoritmo Parcialmente Inexato

O algoritmo do método de nível parcialmente inexato é baseado em [17, Seção 2.1.1], incorporando etapas adicionais para lidar com a imprecisão do oráculo (Passos 4.1, 5, e 6).

Algoritmo 6.1 MÉTODO DE NÍVEL PARCIALMENTE INEXATO

Passo 0 (*inicialização*). *Selecione $z^1 \in \mathcal{X}$, um parâmetro para a combinação convexa $\kappa \in (0, 1)$, uma tolerância $\delta_{\text{To1}} > 0$ para o teste de parada, e um parâmetro para a escolha de um corte exato $\epsilon_\Delta \in (0, 1)$. Defina $\zeta_{inex} = 0$ e calcule os valores exatos para o par $(f(z^1), g(z^1))$. Faça $f_{sup}^1 = f_z^1 := f(z^1)$, $g_z^1 := g(z^1)$, $\check{f}_1(\cdot) := f_z^1 + (\cdot - z^1)^\top g_z^1$, $\Delta_0 = \Delta_{Ref} = \infty$, $J^1 = \{1\}$, $k = \bar{k} = 1$, $\ell = 0$, e $k(\ell) = 1$ (a iteração $k(\ell)$ é chamada de iteração crítica).*

Passo 1 (*programa linear*). *Determine $f_{inf}^k = \check{f}_{min}^k$ resolvendo (6.7), e faça $\Delta_k = f_{sup}^k - f_{inf}^k$. Seja \check{x}^k uma solução do problema (6.7).*

Passo 2 (teste de parada). Se $\Delta_k \leq \delta_{Tot}$ pare. O ponto $z^{\bar{k}}$ é uma δ_{Tot} -solução para o problema (6.1).

Passo 3 (programa quadrático). Atualize o valor do nível f_{niv}^k por (6.8), e o conjunto de nível X^k por (6.9). Obtenha z^{k+1} resolvendo o problema (6.10).

Passo 4 (tamanho do passo). Se $\|z^{k+1} - z^k\| \leq \epsilon_\Delta \Delta_k$ e $\zeta_{inex} = 1$, continue ao Passo 4.1; caso contrário, vá ao Passo 4.2.

Passo 4.1 (cálculos exatos para z^k). Fixe $\zeta_{inex} = 0$ e chame o oráculo (6.2) para calcular o par $(f(z^k), g(z^k))$ de forma exata. Substitua o par inexato definindo $(f_z^k, g_z^k) = (f(z^k), g(z^k))$. Atualize o modelo de planos cortantes, e o limite superior por $f_{sup}^k := \min\{f_{sup}^k, f(z^k)\}$. Escolha $\bar{k} \in \{1, \dots, k\}$ tal que $f_{sup}^k = f_z^{\bar{k}}$, e volte ao Passo 1.

Passo 4.2 (cálculos inexatos para z^{k+1}). Faça $\zeta_{inex} = 1$, e invoque o oráculo (6.2) para calcular os valores inexatos f_z^{k+1} e g_z^{k+1} .

Passo 5 (verificação da imprecisão). Se ao menos uma das desigualdades dadas em (6.13) é verificada, defina $\zeta_{inex} = 0$ e compute $(f_z^{k+1}, g_z^{k+1}) = (f(z^{k+1}), g(z^{k+1}))$ de forma exata. Atualize o limite superior por $f_{sup}^k := \min\{f_{sup}^k, f(z^{k+1})\}$, e escolha $\bar{k} \in \{1, \dots, k\}$ tal que $f_{sup}^k = f_z^{\bar{k}}$.

Passo 6 (seleção do feixe) Se $(1 - \kappa)\Delta_{Ref} \leq \Delta_k$, continue; caso contrário, vá ao Passo 6.2.

Passo 6.1 Faça $J^{k+1} = J^k \cup \{k+1\}$, e vá ao Passo 7.

Passo 6.2 Defina J^{k+1} de tal modo que $J^{k+1} \supset \hat{J}^k \cup \{k+1\}$, com $\hat{J}^k := \{j \in J^k : f_j(\tilde{x}^k) = f_{inf}^k\}$ é o conjunto dos cortes ativos. Faça $k(\ell+1) = k$, $\ell = \ell + 1$, e atualize o valor de referência por $\Delta_{Ref} = \Delta_k$.

Passo 7 (ciclo). Faça $f_{sup}^{k+1} = f_{sup}^k$, $k = k + 1$, e volte ao Passo 1.

□

O limite superior f_{sup}^k é atualizado somente quando o oráculo (5.2) é chamado com $\zeta_{inex} = 0$. Suponha que, como no MNI, o valor f_{inf}^k permaneça fixo para algumas iterações. Então, a brecha de dualidade Δ_k fica constante, e o Passo 5 determina uma linearização exata da função. Assim sendo, fixar o valor f_{inf}^k resulta em muitas avaliações exatas da função, e deste modo, em um esforço computacional excessivo. Para evitar esta situação, resolve-se o programa linear (6.7) a cada iteração.

Conhecer uma cota superior para o valor ótimo é fundamental para definir o conjunto de nível. Para obter tal cota, o Algoritmo 6.1 é inicializado considerando

os valores exatos do par $(f(z^1), g(z^1))$ (a primeira linearização é exata). Caso uma cota superior finita, diga-se f_{sup}^1 , for conhecida, o Algoritmo 6.1 pode ser inicializado considerando um corte inexato, diminuindo deste modo o esforço computacional na primeira iteração.

Diferentemente do Algoritmo 4.1, o Algoritmo 6.1 é menos exigente em relação ao cálculo de um corte exato, porque em vez de analisar o decréscimo da função (que pode ser um decréscimo falso, dependendo da qualidade do oráculo inexato), o algoritmo analisa a estabilidade da sequência $\{\Delta_k\}$, e a qualidade de f_z^{k+1} em função do limite inferior f_{inf}^k .

O Passo 4 do algoritmo mede a distância entre os iterados consecutivos. Quando a razão entre esta distância e o valor Δ_k é pequena, isto significa que o progresso no processo de otimização é insatisfatório (em termos da tolerância ϵ_Δ), e deste modo, investir esforços computacionais para calcular uma linearização inexata em torno do valor (inexato) f_z^{k+1} pode não ser relevante.

Em comparação ao Algoritmo 5.1, a seleção do feixe realizada no Passo 6.2 do Algoritmo 6.1 não utiliza os multiplicadores de Lagrange do problema de projeção (6.10), mas sim considera os índices ativos do modelo de planos cortantes (6.5). Este procedimento é realizado para assegurar que a sequência $\{f_{inf}^k\}$ (resp. $\{\Delta_k\}$) seja monótona não decrescente (resp. não crescente), mesmo quando algumas linearizações são eliminadas do modelo de planos cortantes. Cabe ressaltar que o Algoritmo 5.1 sempre gera uma sequência $\{f_{inf}^k\}$ monótona não decrescente.

De acordo com o Passo 6.2, as iterações geradas pelo Algoritmo 6.1 podem ser divididas em ciclos, definindo a ℓ -ésima subsequência por

$$K^\ell := \{k(\ell), \dots, k(\ell + 1) - 1\}, \quad \text{para } \ell \geq 0.$$

Então, pelo Passo 6 do Algoritmo 6.1 vale a relação seguinte

$$\Delta_k \geq \frac{\Delta_{k(\ell+1)}}{1 - \kappa}, \quad \text{para todo } k \in K^\ell \text{ e } \ell \geq 0.$$

Em particular, $k(\ell) \in K^\ell$ quando $K^\ell \neq \emptyset$; então,

$$(1 - \kappa)\Delta_{k(\ell)} \geq \Delta_{k(\ell+1)}, \quad \text{para todo } \ell \geq 0. \quad (6.14)$$

A iteração $k(\ell)$ é chamada de *iteração crítica*. Para diminuir o tamanho do feixe, o Algoritmo 6.1 elimina os cortes inativos durante cada iteração crítica $k(\ell)$, com $\ell \geq 0$. Porém, ao contrário dos métodos de feixes proximais, o MNPI é incapaz de manter limitado o número de cortes do modelo (6.5). É importante mencionar que, tanto a estratégia do Passo 4, como a estratégia do Passo 6, são fundamentais para a análise de convergência do algoritmo do método de nível parcialmente inexato,

realizada a seguir.

6.3.1 Análise de Convergência

Ao longo desta seção, sejam $\kappa \in (0, 1)$ um parâmetro do Algoritmo 6.1, Λ uma constante de Lipschitz para f em \mathcal{X} , e D o diâmetro do conjunto compacto $\mathcal{X} \neq \emptyset$.

Dada uma iteração $k(\ell)$, a proposição a seguir estabelece o número máximo de iterações que o Algoritmo 6.1 deve efetuar de modo que se obtenha um decréscimo suficiente da brecha de otimalidade.

Proposição 6.1 *Considere o Algoritmo 6.1. Suponha que $\Delta_k > \delta_{\text{Tot}}$ para $k \in K^\ell$ com $\ell \geq 0$. O número de iterações $k - k(\ell) + 1$, realizadas pelo algoritmo entre as iterações $k(\ell)$ e k , não excede*

$$\left(\frac{D\Lambda}{\eta(\epsilon_\Delta)\Delta_k} \right)^2, \quad \text{com } \eta(\epsilon_\Delta) := \min \{ (1 - \kappa), \Lambda\epsilon_\Delta \}.$$

Prova. Primeiramente será mostrado que $\|z^{j+1} - z^j\| \geq \eta(\epsilon_\Delta)\Delta_k$ para todo $k(\ell) \leq j \leq k - 1$. Como $k \in K^\ell$, o Passo 6 do Algoritmo 6.1 garante que

$$\{k(\ell), k(\ell) + 1, \dots, k - 1, k\} \subset J^k.$$

Suponha primeiro que o corte $f_j(\cdot) = f_z^j + (\cdot - z^j)^\top g^j$ seja inexato. Então, segue do Passo 4 que

$$\|z^{j+1} - z^j\| > \epsilon_\Delta \Delta_j = \epsilon_\Delta \Lambda \frac{\Delta_j}{\Lambda}. \quad (6.15)$$

Suponha agora que o corte $f_z^j(\cdot)$ seja exato. Então, $\check{f}_j(z^j) = f_z^j = f(z^j) \geq f_{sup}^j$ por (6.2) e (6.6). Segue do Passo 3 que $\check{f}_j(z^{j+1}) \leq f_{niv}^j = f_{inf}^j + \kappa\Delta_j$, por (6.8). Estas duas relações, juntamente com a propriedade de Lipschitz (6.4) e definição de Δ_j , fornecem as relações

$$\Lambda \|z^j - z^{j+1}\| \geq \check{f}_j(z^j) - \check{f}_j(z^{j+1}) \geq (f_{sup}^j - f_{inf}^j) - \kappa\Delta_j \geq (1 - \kappa)\Delta_j.$$

Resulta então que

$$\|z^j - z^{j+1}\| \geq (1 - \kappa) \frac{\Delta_j}{\Lambda}. \quad (6.16)$$

Como $\Delta_k \leq \Delta_j$ para $j \in \{k(\ell), \dots, k\}$, de (6.15) e (6.16) obtém-se que

$$\|z^{j+1} - z^j\| \geq \eta(\epsilon_\Delta) \frac{\Delta_k}{\Lambda}, \quad \text{para todo } k(\ell) \leq j \leq k - 1, \quad (6.17)$$

como se queria mostrar.

O seguinte passo consiste em mostrar que $\bigcap_{j=k(\ell)}^k X^j \neq \emptyset$. Para isto, consideram-

se os intervalos $[f_{inf}^k, f_{sup}^k] \subseteq [f_{inf}^{k(\ell)}, f_{sup}^{k(\ell)}]$, com comprimentos Δ_k e $\Delta_{k(\ell)}$, respectivamente. O ponto $f_{inf}^{k(\ell)} + \kappa\Delta_{k(\ell)}$ divide o segundo intervalo em dois subintervalos, tal que o subintervalo $[f_{inf}^{k(\ell)} + \kappa\Delta_{k(\ell)}, f_{sup}^{k(\ell)}]$ tem comprimento $(1-\kappa)\Delta_{k(\ell)}$. Como $k \in K^\ell$, $(1-\kappa)\Delta_{k(\ell)} < \Delta_k$ por (6.14), e o subintervalo $[f_{inf}^{k(\ell)} + \kappa\Delta_{k(\ell)}, f_{sup}^{k(\ell)}] \not\subseteq [f_{inf}^k, f_{sup}^k]$. Assim sendo, $f_{inf}^k \leq f_{inf}^{k(\ell)} + \kappa\Delta_{k(\ell)}$. Seja \check{x}_k um ponto no conjunto

$$\{z \in \mathcal{X} : \check{f}_k(z) \leq f_{inf}^k\}.$$

Tal ponto \check{x}_k sempre existe, porque o conjunto acima é não vazio devido a definição (6.7) de f_{inf}^k . Então $\check{x}_k \in X^{k(\ell)}$ e, similarmente, $\check{x}_k \in X^j$ para todo $k(\ell) \leq j \leq k-1$.

Por definição, o iterado z^{j+1} é a projeção de z^j sobre o conjunto de nível X^j . Será mostrado a seguir que cada iterado (com $k(\ell) \leq j \leq k-1$) está cada vez mais próximo de \check{x}_k . A relação (5.18) para $z^j = x$, $z^{j+1} = P_{X^j}(z^j)$ e $\check{x}_k \in X^j$, garante a desigualdade

$$\|z^{j+1} - \check{x}_k\|^2 \leq \|\check{x}_k - z^j\|^2 - \|z^{j+1} - z^j\|^2, \quad \text{para } k(\ell) \leq j \leq k-1.$$

Então, segue de (6.17) que

$$\|z^{j+1} - \check{x}_k\|^2 \leq \|\check{x}_k - z^j\|^2 - \left(\eta(\epsilon_\Delta) \frac{\Delta_k}{\Lambda}\right)^2, \quad \text{para } k(\ell) \leq j \leq k-1,$$

ou seja, a cada iteração $j \in \{k(\ell), \dots, k-1\}$ o ponto z^{j+1} está mais próximo de \check{x}_k por um fator de, no mínimo, $(\eta(\epsilon_\Delta)\Delta_k/\Lambda)^2$. Além disso, os diâmetros dos conjuntos $X^k \subseteq \mathcal{X}$ são menores ou iguais a D , e assim sendo, $\|z^{k(\ell)} - \check{x}_k\| \leq D$. Portanto, o número de iterações realizadas pelo Algoritmo 6.1 entre as iterações $k(\ell)$ e k não pode exceder

$$\left(\frac{D\Lambda}{\eta(\epsilon_\Delta)\Delta_k}\right)^2,$$

como se queria demonstrar. ■

O resultado de complexidade a seguir é semelhante ao Teorema 5.1.

Teorema 6.1 *Para que o Algoritmo 6.1 determine uma brecha de otimalidade menor do que $\delta_{\text{ToI}} > 0$ é suficiente realizar, no máximo,*

$$c(\kappa, \epsilon_\Delta) \left(\frac{D}{\delta_{\text{ToI}}}\right)^2$$

iteraões, onde $c(\kappa, \epsilon_\Delta)$ é uma constante que depende somente de κ e ϵ_Δ .

Prova. Seja o conjunto de iterações $K(\delta_{\text{ToI}}) := \{1, \dots, k_{\delta_{\text{ToI}}}\} \subset \cup_{\ell=0}^m K^\ell$, tal que

$$\Delta_k \geq \delta_{\text{ToI}} \quad \text{para todo } k \in K(\delta_{\text{ToI}}).$$

Então, por (6.14): $(1 - \kappa)\Delta_{k(\ell)} \geq \Delta_{k(\ell+1)}$ para $\ell = 0, \dots, m - 1$, e $\Delta_{k(m)} \geq \delta_{\text{To1}}$. Como $\Delta_{k(\ell)} \geq 0$,

$$(1 - \kappa)^{m-\ell}\Delta_k \geq \delta_{\text{To1}}, \quad \text{para todo } k \in K^\ell \cap K(\delta_{\text{To1}}), \text{ e } \ell = 0, \dots, m.$$

A Proposição 6.1 fornece um limite superior para a cardinalidade de $K^\ell \cap K(\delta_{\text{To1}})$:

$$|K^\ell \cap K(\delta_{\text{To1}})| \leq \left(\frac{D\Lambda}{\eta(\epsilon_\Delta)\Delta_k} \right)^2 \leq \left(\frac{D\Lambda(1 - \kappa)^{m-\ell}}{\eta(\epsilon_\Delta)\delta_{\text{To1}}} \right)^2, \quad \text{para todo } \ell = 0, \dots, m.$$

Somando a desigualdade acima em ℓ , tem-se que

$$k_{\delta_{\text{To1}}} = \sum_{\ell=0}^m |K^\ell \cap K(\delta_{\text{To1}})| \leq \sum_{\ell=0}^m \left(\frac{D\Lambda(1 - \kappa)^{m-\ell}}{\eta(\epsilon_\Delta)\delta_{\text{To1}}} \right)^2 \leq \left(\frac{D\Lambda}{\delta_{\text{To1}}} \right)^2 \frac{1}{(1 - (1 - \kappa)^2)\eta(\epsilon_\Delta)^2},$$

e, deste modo, o número máximo de iterações $k_{\delta_{\text{To1}}}$, tal que $\Delta_{k_{\delta_{\text{To1}}}} \geq \delta_{\text{To1}}$, não pode ser maior do que

$$c(\kappa, \epsilon_\Delta) \left(\frac{D\Lambda}{\delta_{\text{To1}}} \right)^2, \quad \text{com } c(\kappa, \epsilon_\Delta) = \frac{1}{(1 - (1 - \kappa)^2)\eta(\epsilon_\Delta)^2}.$$

■

Se $\Lambda\epsilon_\Delta > (1 - \kappa)$, a complexidade do MNPI coincide com a do método de nível exato (e a do método de nível proximal exato ou inexato). Cabe ressaltar que $\epsilon_\Delta > 0$ é uma condição essencial para que o Algoritmo 6.1 termine, encontrando uma δ_{To1} -solução para o problema (6.1).

O Teorema 6.1 usa o fato que as linearizações inativas são mantidas no feixe ao menos durante as iterações não críticas.

Ao contrário do método de feixes proximal, o método de nível (exato ou parcialmente inexato) não possui convergência finita mesmo quando a função f é poliedral, e todos os cortes são preservados ao longo das iterações. Para verificar esta afirmação, considere o seguinte exemplo.

Exemplo 6.1 *Dado o conjunto viável representado pelo intervalo fechado $\mathcal{X} := [0, 1]$, seja a função real $f : \mathcal{R} \rightarrow \mathcal{R}$ dada por $f(x) = x$. Iniciando o processo de otimização com $z^1 = 1$ no método de nível, tem-se que $f_{inf}^k = 0$ e $\Delta_k = \kappa^{k-1} > 0$ para todo $k \geq 1$. Portanto, não há convergência finita.*

A seguir são propostos alguns oráculos inexatos para aplicação dos métodos de feixes inexatos e parcialmente inexatos, apresentados nos Capítulos 3, 4, 5 e 6, aos problemas de otimização estocástica em dois estágios.

Capítulo 7

Métodos de Feixes Inexatos Aplicados à Programação Linear Estocástica em Dois Estágios

Neste capítulo, a aplicação dos métodos de feixes inexatos e parcialmente inexatos aos problemas lineares estocásticos em dois estágios considera dois oráculos distintos, porém, fundamentados no *critério de colinearidade* das funções objetivo dos problemas duais de segundo estágio. O primeiro deles é um oráculo parcialmente inexato, e pode ser utilizado tanto pelos métodos de feixes inexatos apresentados nos Capítulos 3 e 5, quanto pelos métodos de feixes parcialmente inexatos, introduzidos nos Capítulos 4 e 6. O segundo oráculo é puramente inexato, e é somente utilizado pelos métodos de feixes inexatos.

7.1 Programação Linear Estocástica em Dois Estágios

Seja o programa linear estocástico em dois estágios da forma

$$\min_{x \in \mathcal{X}} f(x), \quad \text{com} \quad f(x) := c^\top x + \mathbb{E}[Q(x, \xi)], \quad (7.1a)$$

onde $\mathcal{X} := \{x \in \mathcal{R}_+^n : Ax = b\}$, e $Q(x, \xi)$ é o valor ótimo do problema de segundo estágio

$$\min_{y \in \mathcal{R}^{n_2}} q^\top y \quad \text{s.a.} \quad Tx + Wy = h, \quad y \geq 0. \quad (7.1b)$$

As matrizes $A \in \mathcal{R}^{m_1 \times n}$ e $W \in \mathcal{R}^{m_2 \times n_2}$, e os vetores $c \in \mathcal{R}^n$ e $b \in \mathcal{R}^{m_1}$ são determinísticos e, deste modo, não envolvem incertezas. Ao longo deste capítulo é suposto que a variável aleatória ξ (que define o cenário $\xi := (q, h, T)$) tem variância finita, e pertence a algum espaço de probabilidades (Ξ, \mathcal{F}, P) .

Esta seção se restringe aos programas estocásticos com *recurso fixo e relativamente completo*. No entanto, as técnicas aqui apresentadas podem ser estendidas ao caso em que o recurso é aleatório. Para isto, basta assumir que o problema considerado possui uma solução viável. Quando o problema de otimização não satisfaz a hipótese de recurso relativamente completo, as estratégias comumente adotadas são: a inclusão de variáveis de folga (penalizadas no custo), ou a consideração de *cortes de inviabilidade*. Para maiores esclarecimentos, ver [2, p. 170] e [11, Seção 7.1].

Dado este desenvolvimento, a seguir é realizada uma síntese das hipóteses assumidas para o problema (7.1):

- L1** o conjunto viável de primeiro estágio $\mathcal{X} = \{x \in \mathcal{R}_+^n : Ax = b\}$ é não vazio;
- L2** o problema (7.1) possui recurso fixo, e relativamente completo;
- L3** a variável aleatória ξ que define o cenário $\xi = (q, h, T)$ tem variância finita, e possui (ou é discretizada assumindo) um número finito N de cenários $\{\xi^1, \dots, \xi^N\}$, com probabilidade associada $p_i = P(\xi = \xi^i)$, para todo $i = 1, \dots, N$.

Dada a hipótese L3, o problema (7.1) com finitos cenários pode ser escrito da forma

$$\min_{x \in \mathcal{X}} f^N(x), \quad \text{com} \quad f^N(x) := c^\top x + \sum_{i=1}^N p_i Q(x, \xi^i). \quad (7.2)$$

Sob as hipóteses L1-L3 valem as Proposições 2.1 e 2.2, e deste modo, a função f^N é convexa, sci, com $f^N(x) > -\infty$ para todo $x \in \mathcal{R}^n$, e o conjunto $\mathcal{X} \cap \text{dom } f^N \neq \emptyset$ é convexo e fechado. Além disso, a função f^N é Lipschitz contínua no seu domínio, $\text{dom } f^N = \{x \in \mathcal{R}_+^n : h - Tx = Wy\}$.

Para calcular o valor funcional e um subgradiente da função objetivo f^N é necessário resolver N subproblemas, como em (7.1b). Quando N é um número elevado, a função $f^N(\cdot)$ é difícil de ser calculada. Neste sentido, para moderar o esforço computacional torna-se interessante aproximar a função utilizando estimativas f_x do valor $f^N(x)$, para $x \in \mathcal{X}$. É desejado que tais estimativas estejam próximas do valor $f^N(x)$, i.e., que $f_x \in [f^N(x) - \varepsilon_f, f^N(x) + \varepsilon_g]$, para $\varepsilon_f, \varepsilon_g \geq 0$ dois erros de precisão tão pequenos quanto possível.

O desenvolvimento acima motiva a aplicação dos método de feixes inexatos e parcialmente inexatos apresentados nos capítulos anteriores. A seguir são dados os detalhes desta aplicação.

Pelas Proposições 2.3 e 2.5, a função $Q_i(x) := Q(x, \xi^i)$ é convexa e subdiferenciável, com $-T_i^\top u_i$ um elemento do subdiferencial $\partial Q_i(x)$, para $\xi^i = (q_i, h_i, T_i) \in \Xi$,

e $u_i := u(x, \xi^i)$ uma solução do problema dual de (7.1b):

$$\begin{aligned} Q_i(x) &= \max_u (h_i - T_i x)^\top u \text{ s.a. } W^\top u \leq q_i \\ &= (h_i - T_i x)^\top u_i. \end{aligned} \quad (7.3)$$

Assim sendo, o vetor $g(x) := c - \sum_{i=1}^N p_i T_i^\top u_i$ é um subgradiente da função f^N no ponto $x \in \mathcal{X}$. A aplicação de um método (exato) de otimização não diferenciável ao problema (7.2) usa um oráculo que fornece

$$\left[\begin{array}{ll} \text{a função} & f^N(x) = c^\top x + \sum_{i=1}^N p_i Q_i(x) \\ \text{um subgradiente} & g(x) = c - \sum_{i=1}^N p_i T_i^\top u_i. \end{array} \right. \quad (7.4)$$

Para a aplicação dos métodos de feixes inexatos, em vez de (7.4) o oráculo pode ser inexato e fornecer

$$\left[\begin{array}{ll} \text{uma estimativa da função} & f_x \in [f^N(x) - \varepsilon_f, f^N(x) + \varepsilon_g] \\ \text{uma estimativa do subgradiente} & g_x \in \partial_{\varepsilon_f + \varepsilon_g} f^N(x), \end{array} \right. \quad (7.5)$$

com $\varepsilon_f, \varepsilon_g \geq 0$ limitados.

A seguir são considerados dois oráculos inexatos satisfazendo (7.5), baseado no *critério de colinearidade* das funções objetivo dos problemas duais (7.3), para diferentes cenários ξ^i e ξ^j .

7.2 Critério de Colinearidade

Seja $x \in \mathcal{X}$ um ponto fixo. Dado algum cenário $\xi^i \in \Xi$, uma solução u_i do problema dual (7.3) aproxima uma solução u_j do problema dual no cenário $\xi^j \in \Xi$, quando as funções objetivo (7.3) (definidas nos cenários ξ^i e ξ^j) são semelhantes.

Podem-se distinguir duas situações, segundo as incertezas:

Caso 1 *Programas lineares com recurso e custo fixos, i.e., $\xi^i := (h_i, T_i)$ com $q_i = q$ e $W_i = W$ para todo $i = 1, \dots, N$.*

Caso 2 *Programas lineares estocásticos gerais, com $\xi^i := (q_i, h_i, T_i)$.*

7.2.1 Caso 1: Programas com Recurso e Custo Fixos

Seja $x \in \mathcal{X}$ um ponto dado. Se dois vetores $h_i - T_i x$ e $h_j - T_j x$ são colineares, existe um escalar positivo ρ tal que $h_i - T_i x = \rho(h_j - T_j x)$, i.e., $\cos(\theta_{ij}) = 1$, para

$$\cos(\theta_{ij}) := \frac{(h_i - T_i x)^\top (h_j - T_j x)}{\|h_i - T_i x\| \|h_j - T_j x\|}. \quad (7.6)$$

Logo, as soluções duais u_i (do subproblema $Q_i(x)$) e u_j (do subproblema $Q_j(x)$) são equivalentes, por (7.3). Neste caso, é necessário resolver apenas um dos problemas $Q_i(x)$ ou $Q_j(x)$. Partindo desta observação, o seguinte oráculo considera os vetores “quase” colineares:

Oráculo Inexato 1 (ESTRATÉGIA DE COLINEARIDADE - CUSTO FIXO)

Passo 0 (inicialização). Sejam $\varepsilon_{\cos} \in (0, 1)$ um parâmetro de colinearidade, e

\mathcal{D}^E um conjunto de variáveis duais (podendo ser $\mathcal{D}^E = \emptyset$),

e $\zeta_{inex} \in \{0, 1\}$ um parâmetro.

Dado um ponto x fixo, vá ao Passo 1.

Passo 1 (seleção).

Se $\zeta_{inex} = 0$, defina $I^E := \{1, \dots, N\}$.

Se $\zeta_{inex} = 1$, selecione um conjunto não vazio $I^E \subseteq \{1, \dots, N\}$ tal que

$i, l \in I^E \Rightarrow \cos(\theta_{il}) \leq 1 - \varepsilon_{\cos}$.

Passo 2 (cálculos exatos). Para cada $i \in I^E$ encontre u_i

resolvendo (7.3), e adicione u_i ao conjunto \mathcal{D}^E .

Passo 3 (cálculos inexatos). Para cada $j \notin I^E$:

compute $\vartheta_j := \arg \max_{u \in \mathcal{D}^E} u^\top (h_j - T_j x)$.

□

O Oráculo Inexato 1 fornece as estimativas:

$$\begin{cases} f_x = c^\top x + \sum_{i \in I^E} p_i u_i^\top (h_i - T_i x) + \sum_{j \notin I^E} p_j \vartheta_j^\top (h_j - T_j x) \\ g_x = c - \sum_{i \in I^E} p_i T_i^\top u_i - \sum_{j \notin I^E} p_j T_j^\top \vartheta_j. \end{cases}$$

Note que, para $\zeta_{inex} = 0$, o Oráculo Inexato 1 satisfaz (7.4).

Para mostrar que a imprecisão deste oráculo é limitada, a seguir é dada uma cota superior para as estimativas da função, que será útil subsequentemente.

Lema 7.1 *Dado o conjunto convexo e não vazio $\Pi(q) := \{u : W^\top u \leq q\}$, considere a função suporte*

$$s_q(d) := \max \{u^\top d : u \in \Pi(q)\} .$$

Então, existe uma constante $K = K(q, W)$ tal que, para todo vértice $v \in \Pi(q)$, as desigualdades

$$s_q(d) - K(q, W) \|d\| \leq v^\top d \leq s_q(d)$$

valem para qualquer $d \in \Pi^\circ := \{d \in \mathbb{R}^{m_2} : d^\top u \leq 0 \ \forall u \in \Pi(0)\}$.

Prova. O domínio da função suporte é Π° , o cone normal de $\Pi(0)$; veja por exemplo [1, pp. 28-29]. Para $d \in \Pi^\circ$, a segunda desigualdade vale, porque $v \in \Pi(q)$. Seja

$u_d \in \Pi(q)$ uma solução básica ótima tal que $s_q(d) = u_d^\top d$. Então,

$$0 \leq s_q(d) - v^\top d \leq \|u_d - v\| \|d\|.$$

Desde que ambos v e u_d são viáveis, existem bases primais viáveis B_v e B_u de $\Pi(q)$ tais que $v = B_v^{-1}q$ e $u_d = B_u^{-1}q$. Como o conjunto $\Pi(q)$ é não vazio, existe um número finito de tais bases, e o resultado se segue. ■

A seguir será mostrado que a imprecisão do Oráculo Inexato 1 satisfaz (7.5).

Proposição 7.1 *Seja o problema (7.1) tal que o vetor de custos q é determinístico e as hipóteses L1-L3 são satisfeitas. Suponha que o algoritmo de programação linear utilizado para resolver os programas lineares no Passo 1 do Oráculo Inexato 1 determina soluções básicas (vértices do poliedro).*

Se \mathcal{X} é limitado, então a imprecisão do oráculo inexato satisfaz (7.5) com $\varepsilon_g = 0$.

Prova. Sejam $x \in \mathcal{X}$ e $j \in \{1, \dots, N\} \setminus I^E$ dados. Desde que o recurso é relativamente completo, o valor $Q_j(x) = s_q(h_j - T_j x)$ é finito. A hipótese acerca do algoritmo de programação linear implica que $v = \vartheta_j$ é uma solução básica viável de $\Pi(q)$. Além disso, definindo $d = h_j - T_j x$ no Lema 7.1, tem-se que

$$(0 \leq) \quad \varepsilon_j := Q_j(x) - \vartheta_j^\top (h_j - T_j x) \leq K(W, q) \|h_j - T_j x\|.$$

A hipótese de recurso e custo fixos implica que $K = K(W, q)$ não depende de j , e sendo a variância de ξ finita (hipótese L3), a existência de uma cota superior para $\|h_j - T_j x\|$ é garantida. Como N é finito e \mathcal{X} é limitado, então $\varepsilon_j \leq \varepsilon_f$ para alguma constante $\varepsilon_f < \infty$. Para mostrar que $g_x \in \partial_{\varepsilon_f} f^N(x)$ e que (7.5) vale com $\varepsilon_g = 0$, considere $z \in \mathcal{X}$ e $Q_j(z)$ o valor ótimo de (7.3), substituindo x por z . Seja u'_j uma solução ótima deste problema. Desde que ambos recurso e vetor de custos dos problemas de segundo estágio são fixos, tem-se que

$$\begin{aligned} Q_j(z) &= (h_j - T_j z)^\top u'_j \geq (h_j - T_j z)^\top \vartheta_j \\ &= Q_j(x) - z^\top T_j^\top \vartheta_j + x^\top T_j^\top \vartheta_j - [Q_j(x) + x^\top T_j^\top \vartheta_j - h_j^\top \vartheta_j] \\ &= Q_j(x) - (z - x)^\top T_j^\top \vartheta_j - [Q_j(x) - (h_j - T_j x)^\top \vartheta_j] \\ &= Q_j(x) - (z - x)^\top T_j^\top \vartheta_j - \varepsilon_j. \end{aligned} \tag{7.7}$$

Pela Proposição 2.2, $-u_i^\top T_i \in \partial Q_i(z)$ para $i \in I^E$ e, portanto,

$$\begin{aligned}
f^N(z) &= c^\top z + \sum_{i=1}^N p_i Q_i(z) \\
&= c^\top z + \sum_{i \in I^E} p_i Q_i(z) + \sum_{j \notin I^E} p_j Q_j(z) \\
&\geq c^\top z + \sum_{i \in I^E} p_i [Q_i(x) - u_i^\top T_i(z-x)] + \sum_{j \notin I^E} p_j [Q_j(x) - \vartheta_j^\top T_j(z-x) - \epsilon_j] \\
&= c^\top x + \sum_{i=1}^N p_i Q_i(x) - \sum_{j \notin I^E} p_j \epsilon_j + [c^\top - \sum_{i \in I^E} p_i u_i^\top T_i - \sum_{j \notin I^E} p_j \vartheta_j^\top T_j](z-x) \\
&= f^N(x) - \sum_{j \notin I^E} p_j \epsilon_j + g_x^\top(z-x).
\end{aligned}$$

Além disso, $f^N(z) \geq f^N(x) - \varepsilon_f + g_x^\top(z-x)$, como desejado. Finalmente, as seguintes manipulações algébricas

$$\begin{aligned}
f^N(x) &= c^\top x + \sum_{i \in I^E} p_i Q_i(x) + \sum_{j \notin I^E} p_j Q_j(x) \\
&= c^\top x + \sum_{i \in I^E} p_i Q_i(x) + \sum_{j \notin I^E} p_j \vartheta_j^\top (h_j - T_j x) + \sum_{j \notin I^E} p_j [Q_j(x) - \vartheta_j^\top (h_j - T_j x)] \\
&= f_x + \sum_{j \notin I^E} p_j \epsilon_j
\end{aligned}$$

implicam que $f_x \in [f^N(x) - \varepsilon_f, f^N(x)]$, e a prova está completa. ■

Como sugerido em [8], a linearização inexata do Oráculo Inexato 1 satisfaz (3.2). Para verificar este resultado, basta substituir ϵ_j por $Q_j(x) - (h_j - T_j x)^\top \vartheta_j$ na desigualdade seguinte

$$f^N(z) \geq c^\top x + \sum_{i=1}^N p_i Q_i(x) - \sum_{j \notin I^E} p_j \epsilon_j + g_x^\top(z-x),$$

obtida na demonstração da Proposição 7.1. Deste modo,

$$\begin{aligned}
f^N(z) &\geq c^\top x + \sum_{i=1}^N p_i Q_i(x) - \sum_{j \notin I^E} p_j \epsilon_j + g_x^\top(z-x) \\
&= c^\top x + \sum_{i=1}^N p_i Q_i(x) - \sum_{j \notin I^E} p_j [Q_j(x) - (h_j - T_j x)^\top \vartheta_j] + g_x^\top(z-x) \\
&= c^\top x + \sum_{i \in I^E} p_i Q_i(x) + \sum_{j \notin I^E} p_j (h_j - T_j x)^\top \vartheta_j + g_x^\top(z-x) \\
&= f_x + g_x^\top(z-x).
\end{aligned}$$

Portanto, $\check{f}_k(\cdot) \leq f^N(\cdot)$, e pode-se mostrar que o resultado do Teorema 3.1 vale com $\varepsilon = \varepsilon_f/2$, i.e., a solução dada pelo Algoritmo 3.1 é ε_f -ótima, ver [8, Teorema 3.9].

O resultado a seguir é consequência do desenvolvimento apresentado acima.

Corolário 7.1 *Sob as mesmas hipóteses da Proposição 7.1, o Oráculo Inexato 1 satisfaz as condições de um oráculo parcialmente inexato, dadas em (4.2).*

A seguir será mostrado que o Oráculo Inexato 1 também satisfaz as condições dadas em (5.4) e (6.4).

Proposição 7.2 *Seja o problema (7.1) tal que o vetor de custos q é determinístico e as hipóteses L1-L3 são satisfeitas. Suponha que o algoritmo de programação linear utilizado para resolver os programas lineares no Passo 1 do Oráculo Inexato 1 encontra soluções básicas (vértices do poliedro).*

Se \mathcal{X} é limitado, então o oráculo inexato satisfaz (5.4) e (6.4), para alguma constante de Lipschitz $\Lambda > 0$.

Prova. Dado o resultado do Corolário 7.1, resta mostrar que $\|g_z\| \leq \Lambda$ e $\|g(z)\| \leq \Lambda$, para todo $z \in \mathcal{X}$. Desde que cada u_i é um vértice do poliedro $\Pi(q)$, tem-se que $u_i := B_i^{-1}q$, para alguma base B_i . Logo, o subgradiente g_z pode ser escrito por

$$g_z = c - \left(\sum_{i \in I^E} p_i T_i^\top B_i - \sum_{j \notin I^E} p_j T_j^\top B_{i(j)} \right) q.$$

Dado que a variância de ξ é finita, e existem apenas um número finito de cenários e bases, a existência de uma constante $\Lambda < \infty$ tal que

$$\|g_z\| \leq \Lambda, \quad \text{para todo } z \in \mathcal{X},$$

é assegurada. Se $I^E = \{1 \dots, N\}$, então $g(z) = g_z$, e resultado enunciado está provado. ■

Como resultado das Proposições 7.1, 7.2 e do Corolário 7.1, o Oráculo Inexato 1 pode ser utilizado tanto pelos métodos de feixes inexatos, quanto pelos métodos de feixes parcialmente inexatos. No entanto, este não é o caso quando o vetor de custos q é aleatório, como mostrado a seguir.

7.2.2 Caso 2: Programas Lineares Estocásticos Gerais

Quando o vetor q_i é dependente dos cenários, dois vetores $h_i - T_i x$ e $h_j - T_j x$ podem ser colineares sem que uma solução dual u_i do subproblema $Q_i(x)$, seja também uma solução do subproblema $Q_j(x)$. Os respectivos conjuntos viáveis de (7.3), $\Pi(q_i)$ e $\Pi(q_j)$, são diferentes. Com o objetivo de economizar esforço computacional, todos os problemas de segundo estágio são aproximados. A abordagem proposta consiste em, para cada $i \in I^E$, agrupar os vetores $h_j - T_j x$ quase colineares em um conjunto \mathcal{J}_i , e considerar um conjunto viável comum, definido pela substituição de q_j por \bar{q}_i , a

média amostral dos custos considerando o conjunto de índices \mathcal{J}_i . Desde que $i \in \mathcal{J}_i$, resolve-se o programa linear com vetor de custo $h_i - T_i x$ sujeito ao conjunto viável médio, e usa-se a correspondente solução com uma aproximação para os demais cenários do grupo.

Oráculo Inexato 2 (ESTRATÉGIA DE COLINEARIDADE - CUSTO ALEATÓRIO)

Passo 0 (inicialização). Seja $\varepsilon_{\cos} \in (0, 1)$ um parâmetro de colinearidade.

Para x fixo, selecione um conjunto $I^E \subseteq \{1, \dots, N\}$ não vazio, tal que $i, l \in I^E \Rightarrow \cos(\theta_{il}) \leq 1 - \varepsilon_{\cos}$.

Passo 1 (agrupamento pela colinearidade). Para cada $i \in I^E$, faça $\mathcal{J}_i := \{i\} \cup \{j \notin I^E : \cos(\theta_{ij}) > 1 - \varepsilon_{\cos}\}$.

Passo 2 (Cálculos inexatos). Para todo $i \in I^E$: determine \bar{u}_i resolvendo

$$\begin{cases} \max_u & u^\top (h_i - T_i x) \\ \text{s.a} & W^\top u \leq \bar{q}_i := \frac{\sum_{j \in \mathcal{J}_i} p_j q_j}{\sum_{j \in \mathcal{J}_i} p_j}. \end{cases}$$

Para cada $j \in \mathcal{J}_i$, faça $\vartheta_j = \bar{u}_i$.

□

O Oráculo Inexato 2 proporciona as seguintes estimativas:

$$\begin{cases} f_x = & c^\top x & + \sum_{i \in I^E} p_i \bar{u}_i^\top (h_i - T_i x) & + \sum_{j \notin I^E} p_j \vartheta_j^\top (h_j - T_j x) \\ g_x = & c & - \sum_{i \in I^E} p_i T_i^\top \bar{u}_i & - \sum_{j \notin I^E} p_j T_j^\top \vartheta_j. \end{cases}$$

Quando os custos são determinísticos ($q = q_i$), as estimativas acima para $i \in I^E$ são exatas, e coincidem com as soluções calculadas no Passo 3 do Oráculo Inexato 1. Este não é o caso para $j \notin I^E$, porque não há uma busca no conjunto de vértices determinado previamente, \mathcal{D}^E . Por este motivo, mesmo quando $q = q_i$, este oráculo inexato não coincide com o Oráculo Inexato 1.

A seguir será mostrado que a condição (7.5) é satisfeita.

Proposição 7.3 *Seja o problema (7.1) satisfazendo as hipóteses L1-L3 e suponha que o Oráculo Inexato 2 é empregado. Suponha também que o algoritmo de programação linear utilizado para resolver os programas lineares no Passo 1 do Oráculo Inexato 2 determina soluções básicas.*

Se \mathcal{X} é limitado, então a condição (7.5) vale com $\varepsilon_f = \varepsilon_g > 0$.

Prova. De maneira similar ao Lema 7.1, será mostrado que as diferenças entre os valores funcionais $Q_j(x)$ e suas estimativas são limitadas. Mais precisamente, definindo $d_j := h_j - T_j x$:

$$\begin{aligned} \text{se } j \in I^E & & Q_j(x) \text{ é substituído por } s_{\bar{q}_j}(d_j) \\ \text{se } j \notin I^E \text{ então } j \in \mathcal{J}_i & & Q_j(x) \text{ é substituído por } \bar{u}_i^\top d_j, \end{aligned}$$

para \bar{u}_i tal que $s_{\bar{q}_i}(d_i) = \bar{u}_i^\top d_i$. Como $Q_j(x) = s_{q_j}(d_j)$, a imprecisão na estimativa da função é dada por

$$\epsilon_j = \begin{cases} s_{q_j}(d_j) - s_{\bar{q}_j}(d_j) & \text{se } j \in I^E \\ s_{q_j}(d_j) - \bar{u}_i^\top d_j & \text{se } j \notin I^E, j \in \mathcal{J}_i. \end{cases}$$

Os termos acima são todos finitos, porque o recurso é relativamente completo. Por [54, Teorema 2.4], o conjunto viável do programa linear é Lipschitziano com respeito às perturbações do lado direito das restrições, então existe uma constante $L(d_j, W)$ tal que

$$\epsilon_j \leq |s_{q_j}(d_j) - s_{\bar{q}_j}(d_j)| \leq L(d_j, W) \|q_j - \bar{q}_j\| \quad \text{se } j \in I^E. \quad (7.8)$$

Quando $j \notin I^E, j \in \mathcal{J}_i$, pode-se escrever

$$\epsilon_j = s_{q_j}(d_j) - \bar{u}_i^\top d_j = s_{q_j}(d_j) - s_{\bar{q}_i}(d_j) + s_{\bar{q}_i}(d_j) - \bar{u}_i^\top d_j =: \Delta_1 + \Delta_2.$$

Como em (7.8), $|\Delta_1| \leq L(d_j, W) \|q_j - \bar{q}_i\|$. Para limitar o termo Δ_2 , note que $d_j \in \Pi^\circ$ porque o recurso é relativamente completo, e que $u_i \in \Pi(\bar{q}_i)$ é um vértice, por hipótese. Como resultado, aplicando o Lema 7.1, escrito com $(q, d, v) = (\bar{q}_i, d_j, \bar{u}_i)$, conjuntamente com (7.8), tem-se que

$$\epsilon_j \leq L(d_j, W) \|q_j - \bar{q}_i\| + K(\bar{q}_i, W) \|d_j\| \quad \text{se } j \notin I^E, j \in \mathcal{J}_i. \quad (7.9)$$

Sendo \mathcal{X} limitado, e tanto N quanto a variância de $\boldsymbol{\xi}$ são finitos, existem constantes L, K , e M_d para $L(d_j, W)$, $K(\bar{q}_i, W)$, e $\|d_j\|$, respectivamente. Seja M_q uma cota superior para $\|q_j - \bar{q}_j\|$ e $\|q_j - \bar{q}_i\|$, então o resultado enunciado é verificado considerando $\varepsilon_f = \varepsilon_g = 2LM_q + KM_d$. ■

Como o número de cenários N e a variância de $\boldsymbol{\xi}$ são finitos, a existência de uma constante $\Lambda > 0$ tal que $\|g_z\| \leq \Lambda$ é garantida. Para verificar esta afirmação, basta proceder de forma análoga à demonstração da Proposição 7.2, substituído q por q_i , com $i = 1 \dots, N$. Deste modo, o Oráculo Inexato 2 também satisfaz (5.2).

Embora esteja-se trabalhando com dois erros de precisão (ε_f e ε_g), os mesmos são controlados facilmente por uma única tolerância $\varepsilon_{\cos} \geq 0$, que determina se o

vetor $h_i - T_i x$ é quase colinear ao vetor $h_j - T_j x$, i.e., vale a equivalência

$$j \in \mathcal{J}_i \iff 1 - \cos(\theta_{ij}) \leq \varepsilon_{\cos}.$$

Logo, se $\varepsilon_{\cos} = 0$ o problema (7.2) é resolvido sem aproximações.

Outra abordagem possível é variar a tolerância ε_{\cos} ao longo das iterações. Por exemplo, considerando (para $k \geq 1$ um contador de iteração) $\varepsilon_{\cos}^k \geq 0$ tal que $\lim_{k \rightarrow \infty} \varepsilon_{\cos}^k = 0$. Neste caso, pode ser melhor considerar $\varepsilon_{\cos}^1 = 0$, $\varepsilon_{\cos}^2 > 0$ com $\varepsilon_{\cos}^{k+1} \leq \varepsilon_{\cos}^k$ para $k \geq 2$, e (ver definição (3.5))

$$f_z^k := \max \left\{ c^\top z^k + \sum_{i \in I^E} p_i \bar{u}_i^\top (h_i - T_i z^k) + \sum_{j \notin I^E} p_j \vartheta_j^\top (h_j - T_j z^k), \check{f}_{k-1}(z^k) \right\},$$

para evitar que erros ε_f e ε_g se tornem excessivamente grandes. Este tipo de abordagem juntamente com o método de feixes inexato é denominado *método de feixes incremental*, [47].

O critério de colinearidade apresentado neste capítulo não altera a distribuição de probabilidades P de ξ . A seguir são apresentadas três técnicas para aproximar f^N e um subgradiente de f^N , redistribuindo P .

Capítulo 8

Métodos de Feixes Inexatos Aplicados à Programação Não Linear Estocástica em Dois Estágios

A aplicação dos métodos de feixes inexatos e parcialmente inexatos aos problemas não lineares estocásticos em dois estágios considera três técnicas distintas para os oráculos inexatos. A primeira e a segunda técnicas são fundamentadas na estratégia de *redução ótima de cenários* proposta em [35]. A terceira técnica é baseada na classificação de cenários em grupos, e é motivada pela *desigualdade de Jensen* [24, Seção 3.4.1]. Os oráculos baseados em reduções/seleções de cenários podem ser aplicados tanto no caso linear quanto no caso não linear. Estes oráculos são desenvolvidos especialmente para o métodos de feixes inexatos. Já o oráculo baseado na desigualdade de Jensen é utilizado pelos métodos de feixes parcialmente inexatos. Apesar de ser desenvolvido para os programas não lineares estocásticos em dois estágios, este oráculo pode ser, naturalmente, aplicado ao caso linear.

8.1 Programação Não Linear Estocástica em Dois Estágios

São considerados nesta seção os problemas de otimização estocástica em dois estágios da forma

$$\min_{x \in \mathcal{X}} \mathbb{E}[f(x, \xi)], \quad (8.1a)$$

com

$$f(x, \xi) := f_1(x) + \inf_{y \in \mathcal{X}(x, \xi)} f_2(y, \xi), \quad (8.1b)$$

sendo que f_1 e f_2 são funções convexas nas respectivas variáveis de decisão x e y . Conforme o desenvolvimento do Capítulo 2, a função objetivo do problema (8.1a) pode ser escrita por $f(x) = f_1(x) + \mathcal{Q}(x)$, com \mathcal{Q} o valor esperado dos problemas de segundo estágio (cf. (2.3)). No entanto, nesta seção é conveniente utilizar a notação empregada em (8.1).

Ao longo desta seção são assumidas as seguintes hipóteses acerca do problema (8.1):

- NL1** o conjunto viável de primeiro estágio \mathcal{X} é não vazio, convexo e compacto;
- NL2** o problema (8.1) possui recurso relativamente completo, sendo o conjunto de segundo estágio dado por $\mathcal{X}(x, \xi) := \{y \in \mathcal{R}_+^{n_2} : Tx + Wy = h\}$;
- NL3** o conjunto $K_2^P = \{x \in \mathcal{X} : \cap_{\xi \in \Xi} \mathcal{X}(x, \xi) \neq \emptyset\}$ é não vazio;
- NL4** as funções $f_1 : \mathcal{R}^n \rightarrow \mathcal{R}$ e $f_2 : \mathcal{R}^{n_2} \times \Xi \rightarrow \mathcal{R}$ são convexas nas respectivas variáveis de decisão, e f_2 é *asci* no sentido da definição (2.4).
- NL5** a variável aleatória $\boldsymbol{\xi}$ que define o cenário $\xi = (q, h, T)$ tem variância finita, e possui (ou é discretizada assumindo) um número finito N de cenários $\{\xi^1, \dots, \xi^N\}$, com probabilidade associada $p_i = P(\boldsymbol{\xi} = \xi^i)$, para todo $i = 1, \dots, N$.
- NL6** existe uma métrica $d_\Xi : \Xi \times \Xi \rightarrow \mathcal{R}_+$ induzida por uma norma (ou pseudo-norma) tal que a seguinte desigualdade é satisfeita

$$|f(x, \tilde{\xi}) - f(x, \xi)| \leq d_\Xi(\tilde{\xi}, \xi), \text{ para todo } x \in \mathcal{X} \text{ e } \tilde{\xi}, \xi \in \Xi.$$

As condições NL1-NL4 garantem que o problema (8.1) seja bem definido, e possua uma solução ótima. É importante notar que sob NL3, qualquer que seja o conjunto de cenários $\{\xi^1, \dots, \xi^N\}$ o problema (8.1), com $f(x)$ substituída por $f^N(x) := \sum_{i=1}^N p_i f(x, \xi^i)$, está bem definido e tem uma solução ótima (ver Proposição (2.1)).

Diferentemente do critério de colinearidade, as técnicas apresentadas neste capítulo não utilizam soluções duais de alguns subproblemas para aproximar a soluções duais de outros subproblemas. O objetivo destas propostas é selecionar um subconjunto com relativamente poucos cenários para aproximar o problema (8.1). Estas propostas são fundamentas na técnica de *redução ótima de cenários* - ROC - desenvolvida por Dupačová, Gröwe-Kuska e Römisch [35], e apresentada a seguir.

8.1.1 Redução Ótima de Cenários

Dado um conjunto de cenários que define um programa estocástico, chamado subsequentemente de “problema original”, a ROC seleciona um subconjunto dos cenários mais representativos para definir um programa estocástico de porte menor, chamado de “problema reduzido”. Além disso, a ROC determina um subconjunto de cenários que proporciona estabilidade e aderência entre os dois problemas, original e reduzido. A seleção dos cenários é realizada utilizando uma métrica probabilística, como apresentado a seguir.

Sejam $P := \{p_1, \dots, p_N\}$ as probabilidades dos N cenários $\{\xi^1, \dots, \xi^N\}$, e

$$f^N(\cdot) := \sum_{i=1}^N p_i f(\cdot, \xi^i) = \mathbb{E}_P[f(\cdot, \xi)]$$

a função objetivo de (8.1) obtida a partir da discretização de Ξ . Como já mencionado, quando o número N é muito grande, a solução numérica de (8.1) se torna muito complexa computacionalmente. Nesta situação, é conveniente escolher um subconjunto relativamente pequeno de cenários representativos $\{\xi^{j_1}, \xi^{j_2}, \dots, \xi^{j_{n_{LP}}}\}$, com $n_{LP} \ll N$, e uma nova distribuição de probabilidades $\tilde{P} = \{\tilde{p}_1, \dots, \tilde{p}_N\}$, com $\tilde{p}_i := 0$ se $i \notin I^E := \{j_1, \dots, j_{n_{LP}}\}$, para aproximar a função $f^N(\cdot)$ por

$$f_{I^E}(x) := \mathbb{E}_{I^E, \tilde{P}}[f(x, \xi)] = \sum_{i \in I^E} \tilde{p}_i f(x, \xi^i). \quad (8.2)$$

Esta nova função objetivo define o problema reduzido

$$\min_{x \in \mathcal{X}} f_{I^E}(x), \quad (8.3)$$

que será efetivamente resolvido. Quando comparado ao problema (8.1), fica claro que a escolha de I^E e \tilde{P} determina o nível de qualidade do problema reduzido (8.3). A técnica ROC introduzida em [35] para programas estocásticos em dois estágios é uma ferramenta eficiente, tanto do ponto de vista teórico quanto computacional, para manter próximos os valores ótimos e os respectivos conjuntos de soluções dos problemas (8.1) e (8.3).

Sob a hipótese NL6, existe uma métrica $d_{\Xi} : \Xi \times \Xi \rightarrow \mathcal{R}_+$ tal que a seguinte desigualdade é satisfeita

$$|f(x, \xi^i) - f(x, \xi^j)| \leq d_{\Xi}(\xi^i, \xi^j), \text{ para todo } x \in \mathcal{X} \text{ e } i, j \in \{1, \dots, N\}.$$

Em programação linear estocástica, a hipótese de recurso fixo e relativamente completo é suficiente para assegurar a hipótese NL6. Isto se deve ao fato que a função \mathcal{Q} é Lipschitz contínua no seu domínio $\text{dom } \mathcal{Q}$, como apresentado na Proposição 2.2.

Dada a existência de uma métrica d_{Ξ} satisfazendo NL6, pode-se utilizá-la para definir o subconjunto de índices

$$\mathbb{A}_i := \left\{ j \in \{1, \dots, N\} \setminus I^E : i = i(j) \in \arg \min_{k \in I^E} d_{\Xi}(\xi^k, \xi^j) \right\},$$

dos cenários mais próximos ao cenário ξ^i , que tem probabilidade $P_{\mathbb{A}_i} = \sum_{j \in \mathbb{A}_i} p_j$. Se \mathbb{A}_i é vazio para algum $i \in I^E$, define-se $P_{\mathbb{A}_i} = 0$. Definindo o conjunto de índices dos cenários “menos representativos” por $\mathbb{J} := \{1, \dots, N\} \setminus I^E$, e a nova medida de probabilidade \tilde{P} por

$$\tilde{p}_i = \tilde{p}(\xi^i) := p_i + P_{\mathbb{A}_i} \quad \text{para todo } i \in I^E, \quad (8.4)$$

segue que (para $x \in \mathcal{X}$)

$$\begin{aligned} f_{I^E}(x) &= \sum_{i \in I^E} f(x, \xi^i) \tilde{p}_i \\ &= \sum_{i \in I^E} f(x, \xi^i) p_i + \sum_{i \in I^E} f(x, \xi^i) P_{\mathbb{A}_i} \\ &= \sum_{i \in I^E \cup \mathbb{J}} f(x, \xi^i) p_i + \sum_{i \in I^E} f(x, \xi^i) P_{\mathbb{A}_i} - \sum_{j \in \mathbb{J}} f(x, \xi^j) p_j \\ &= f_N(x) + \sum_{i \in I^E} f(x, \xi^i) P_{\mathbb{A}_i} - \sum_{j \in \mathbb{J}} f(x, \xi^j) p_j \\ &= f_N(x) + \sum_{j \in \mathbb{J}} \left(f(x, \xi^{i(j)}) - f(x, \xi^j) \right) p_j. \end{aligned}$$

Então,

$$|f_{I^E}(x) - f_N(x)| \leq \sum_{j \in \mathbb{J}} p_j |f(x, \xi^{i(j)}) - f(x, \xi^j)| \leq \sum_{j \in \mathbb{J}} p_j d_{\Xi}(\xi^{i(j)}, \xi^j) = c(\mathbb{J}), \quad (8.5)$$

onde

$$c(\mathbb{J}) := \sum_{j \in \mathbb{J}} p_j \min_{i \in I^E} d_{\Xi}(\xi^i, \xi^j) \quad (8.6)$$

é uma representação do problema de transporte de massas

$$\left\{ \begin{array}{l} \inf_{\eta} \int_{\Xi \times \Xi} d_{\Xi}(\xi, \tilde{\xi}) \eta(dP(\xi), d\tilde{P}(\tilde{\xi})) \\ \text{s.a. : } \eta \in \mathcal{P}(\Xi \times \Xi) \\ \eta(B \times \Xi) = P(B) \\ \eta(\Xi \times B) = \tilde{P}(B), \text{ para todo } B \subset \Xi, \end{array} \right. \quad (8.7)$$

no caso em que a medida de probabilidade P é discreta e finita, e \tilde{P} é uma redistribuição de P pela regra (8.4), ver [35, Teorema 2].

O problema (8.7) possui nomes distintos dependendo da métrica $d_{\Xi}(\cdot, \cdot)$ empregada, [39]:

- se $d_{\Xi}(\xi, \tilde{\xi}) := \|\xi - \tilde{\xi}\| \max \left\{ 1, \|\xi - \xi_0\|^{r-1}, \|\tilde{\xi} - \xi_0\|^{r-1} \right\}$ (para $r \geq 1$) o problema (8.7) é chamado de funcional de Monge-Kantorovich (em geral, para qualquer pseudo-norma d_{Ξ} que não satisfaz a desigualdade triangular, (8.7) leva este nome). Adicionalmente, se as restrições de (8.7) forem substituídas por

$$\eta \in \mathcal{P}(\Xi \times \Xi), \quad \eta(B \times \Xi) - \eta(\Xi \times B) = P(B) - \tilde{P}(B), \text{ para todo } B \subset \Xi,$$

o problema (8.7) é chamado de Kantorovich-Rubinstein, [55].

- se $d_{\Xi}(\xi, \tilde{\xi}) := |\xi - \tilde{\xi}|^r$, o problema dual de (8.7) é chamado de métrica de Wasserstein de ordem r .

Quando $r = 1$, a métrica de Wasserstein coincide com a métrica de Fortet-Mourier, dada por

$$d_{FM}(P, \tilde{P}) := \sup_{g \in \mathcal{F}_d} \left\{ \left| \int_{\Xi} g(\cdot, \xi) dP(\xi) - \int_{\Xi} g(\cdot, \xi) d\tilde{P}(\xi) \right| \right\}, \quad (8.8)$$

com $\mathcal{F}_d := \{g : \mathcal{X} \times \Xi \rightarrow \mathcal{R} : |g(\cdot, \xi) - g(\cdot, \tilde{\xi})| \leq d_{\Xi}(\xi, \tilde{\xi})\}$. Com este desenvolvimento, para cada $x \in \mathcal{X}$ uma variável de primeiro estágio, o funcional de Monge-Kantorovich (que pode ser representado por $c(\mathbb{J})$) é uma cota superior para a diferença absoluta entre os valores funcionais $f_{IE}(x)$ e $f^N(x)$. Dado um conjunto \mathbb{J} , é demonstrado em [35, Teorema 2] que a redistribuição de probabilidades pela regra (8.4) é ótima para o problema

$$\min_{\tilde{P}} |f_{IE}(x) - f^N(x)| \quad \text{s.a.} \quad \tilde{P} = \{\tilde{p}_1, \dots, \tilde{p}_N\} \geq 0, \quad \sum_{i \in IE} \tilde{p}_i = 1, \quad \sum_{j \notin IE} \tilde{p}_j = 0,$$

onde $f_{IE}(\cdot)$ é definida em (8.2). Neste sentido, para garantir a proximidade entre as duas funções é necessário escolher os cenários de forma que o valor $c(\mathbb{J})$ seja tão pequeno quanto possível. Matematicamente, procura-se o conjunto ótimo \mathbb{J}^* tal que

$$\mathbb{J}^* := \arg \min_{\mathbb{J}} c(\mathbb{J}) \quad \text{s.a.} \quad |\mathbb{J}| = N - n_{LP}. \quad (8.9)$$

Uma maneira alternativa consiste em definir $\varepsilon_{roc} > 0$ e encontrar \mathbb{J}^* com a maior cardinalidade possível, de forma que $c(\mathbb{J}) \leq \varepsilon_{roc}$.

Por se tratar de um caso típico de otimização combinatória, é exigido um esforço computacional elevado para resolver o problema (8.9). No entanto, uma solução aproximada pode ser obtida por algoritmos eficientes, baseados em heurísticas, ver [38]. Dentre eles se destaca o algoritmo de seleção progressiva rápida (*fast forward selection*) apresentado em [32, 38].

Com base no desenvolvimento apresentado nesta subseção, tem-se um algoritmo para selecionar os cenários mais representativos, uma regra para redistribuí-los, e uma representação explícita do funcional de Monge-Kantorovich que permite calcular um limite superior para a diferença $|f^N(\cdot) - f_{IE}(\cdot)|$. Mais informações acerca da técnica de redução ótima de cenários podem ser obtidas em [35, 38, 56] para a programação estocástica em dois estágios, e em [37, 57, 58] para a programação estocástica em multiestágios. A seguir a ROC é empregada conjuntamente com os métodos de feixes inexatos para os programas estocásticos em dois estágios.

8.2 Redução Ótima de Cenários Sequencial

Dada a desigualdade (8.5), seja z um ponto arbitrário em \mathcal{X} . Então,

$$f_{IE}(z) \in [f^N(z) - c(\mathbb{J}), f^N(z) + c(\mathbb{J})].$$

Alterando (decrecendo) o subconjunto de índices \mathbb{J} , e definindo

$$f_z := f_{IE}(z), \quad e \quad g_z := \sum_{i \in I^E} \tilde{p}_i \mathfrak{g}_i \quad (8.10)$$

para $\mathfrak{g}_i := \mathfrak{g}(z, \xi^i) \in \partial f(z, \xi^i)$ um elemento no subdiferencial de $f(z, \xi^i)$, pode-se tomar $\varepsilon_f = \varepsilon_g = c(\mathbb{J})$ para aplicar o método de feixes proximal inexato ao problema (8.1).

Se as hipóteses NL1-NL6 são verificadas, $c(\mathbb{J})$ é finito por (8.10). Além disso, tem-se o resultado a seguir.

Proposição 8.1 *Seja $z \in \mathcal{X}$ um ponto fixo. Sob as hipóteses NL1-NL6, sejam $c(\mathbb{J})$ e g_z definidos em (8.6) e (8.10), respectivamente. Então*

$$g_z \in \partial_{2c(\mathbb{J})} f^N(z).$$

Prova. Dadas NL1, NL2 e NL4, a função $f^N(\cdot)$ e a sua aproximação f são convexas. Então, para qualquer $z \in \mathcal{X}$ fixo, tem-se por (8.10) a seguinte desigualdade

$$f_x \geq f_z + (x - z)^\top g_z, \quad \text{para todo } x \in \mathcal{X}. \quad (8.11)$$

A seguinte relação é devido a (8.5), juntamente com as hipóteses NL2, NL5 e NL6

$$f^N(x) + c(\mathbb{J}) \geq f_x \geq f^N(z) - c(\mathbb{J}) + (x - z)^\top g_z, \quad \text{para todo } x \in \mathcal{X},$$

completando deste modo a demonstração. ■

Se a métrica d_{Ξ} em (8.6) é perfeitamente conhecida, então os erros ε_f e ε_g também os são. Em prática, a métrica d_{Ξ} é da forma $d_{\Xi}(\xi, \tilde{\xi}) := M\|\xi - \tilde{\xi}\|$, para $\|\cdot\|$ alguma norma apropriada, e M uma constante desconhecida, mas dependente dos dados do problema (8.1).

Dado um contador de iteração k , o procedimento de *redução ótima de cenários sequencial* - ROCSeq - considera uma tolerância $\varepsilon_{roc}^k \geq 0$ para determinar, pelo Passo 1 do Oráculo Inexato 3 a seguir, o maior subconjunto de índices \mathbb{J}^k satisfazendo a desigualdade $c(\mathbb{J}^k) \leq \varepsilon_{roc}^k$. Para qualquer $z \in \mathcal{X}$, se $\varepsilon_{roc}^k \xrightarrow{k} 0$, então por (8.5) $f_z \xrightarrow{k} f^N(z)$. Fazer com que ε_{roc}^k tenda a zero significa que a qualidade da representação do processo estocástico vai sendo cada vez mais acurada, no entanto, o oráculo fica mais e mais demorado. Deste modo, torna-se interessante nas iterações iniciais do Algoritmo 4.1 considerar poucos cenários para aproximar a função f^N , e ao longo das iterações aumentar n_{LP}^1 sucessivamente (equivalentemente, decrescer ε_{roc}^k). É ilustrado na Figura 8.1 o esquema ROCSeq.

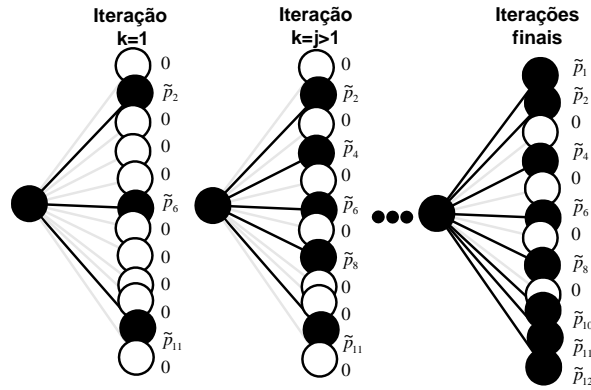


Figura 8.1: Aprimoramento sucessivo do processo estocástico.

Para considerar a técnica ROCSeq conjuntamente com os métodos de feixes inexatos apresentados nos Capítulos 4 e 5, define-se o seguinte oráculo inexato.

Oráculo Inexato 3 (REDUÇÕES ÓTIMAS DE CENÁRIOS SEQUENCIAL)

Passo 0 (inicialização). Seja x um ponto dado,

um conjunto de índices $I^E \subseteq \{1, \dots, N\}$, um parâmetro $\varepsilon_{roc} > 0$

(ou alternativamente $n_{LP} \leq N$), e uma pseudo-norma $d_{\Xi}(\cdot, \cdot)$.

Se $I^E = \emptyset$, calcule as pseudo-distâncias $d_{jl} = d_{\Xi}(\xi^j, \xi^l)$, para $j, l \in \{1, \dots, N\}$.

Faça $s_{l_m} = +\infty$.

Passo 1 (seleção de cenários). Enquanto $s_{l_m} > \varepsilon_{roc}$

¹ n_{LP} é o número de cenários representativos

(ou alternativamente $|I^E| < n_{LP}$) calcule para $l \notin I^E$

$$s_l = \sum_{j \in I^E, j \neq l} p_j d_{jl} \quad \text{e selecione} \quad l_m \in \arg \min_{l \notin I^E} s_l.$$

Faça $I^E = I^E \cup \{l_m\}$.

Para cada $j, l \notin I^E$, atualize as distancias $d_{jl} = \min(d_{jl}, d_{jl_m})$.

Passo 2 (cenários representativos e cálculos exatos).

Para todo $i \in I^E$, determine $f(x, \xi^i)$ e $g_i \in \partial f(x, \xi^i)$ resolvendo (8.1b).

Passo 3 (redistribuição) Redistribua

as probabilidades pela regra (8.4).

□

Como N é finito, o melhor conjunto I^E é sempre encontrado após um número finito de iterações no Passo 1. O Oráculo Inexato 3 fornece as estimativas abaixo:

$$\left[\begin{array}{l} f_x = \sum_{i \in I^E} \tilde{p}_i f(x, \xi^i) \\ g_x = \sum_{i \in I^E} \tilde{p}_i g_i. \end{array} \right.$$

Pelo Teorema 6 em [32], o valor s_l equivale a $c(I^{E^c})$, com $I^{E^c} \subset \{1, \dots, N\}$ o conjunto complementar de I^E .

Segue da Proposição 8.1 e da desigualdade (8.11) que o Oráculo Inexato 3 garante a relação $\check{f}^{\bar{k}}(\cdot) \leq f^N(\cdot) + c(\mathbb{J}^{\bar{k}})$, para \bar{k} a iteração do último passo sério do Algoritmo 4.1. Deste modo, pode-se mostrar que o resultado do Teorema 3.1 vale com $\varepsilon = c(\mathbb{J}^{\bar{k}})$, ver [8, Teorema 3.9].

A seguir são enumeradas algumas vantagens do emprego do Oráculo Inexato 3, conjuntamente com os métodos de feixes inexatos.

1. em relação ao critério de colinearidade definido nos Oráculos 1 e 2:
 - (a) a função convexa $f(\cdot, \xi^i)$ não precisa ser linear (no entanto, deve satisfazer a hipótese LN6);
2. em relação à ROC:
 - (a) a representação do processo estocástico é aprimorada sucessivamente;
 - (b) devido ao número reduzido de subproblemas resolvidos nas iterações iniciais, dada a mesma acurácia para a representação do processo estocástico, os métodos de feixes inexatos empregando o Oráculo Inexato 3 exigem esforços computacionais menores do que exigido pela ROC.

É importante ressaltar que a escolha de ε_{roc}^1 desempenha um papel fundamental na qualidade da ε -solução obtida quando o Oráculo Inexato 3 é aplicado. Como exemplo, suponha que nas iterações iniciais do método o valor $c(\mathbb{J}^k)$ seja significativamente grande. Então, o valor aproximado f_z^{k+1} pode subestimar $f^N(z^{k+1})$ com uma imprecisão de aproximadamente $\varepsilon_f^k = c(\mathbb{J}^k)$. Assim sendo, é provável que o iterado z^{k+1} seja classificado como um passo sério e, por causa da atualização do processo estocástico no Passo 4.1, o método seja incapaz de gerar um novo passo sério. Nesta situação, o candidato z^{k+1} será uma solução de baixa qualidade para o programa estocástico considerado.

Para $z \in \mathcal{X}$ e cada $i \in \{1, \dots, N\}$ o oráculo a seguir utiliza informações dos valores da função $f(z, \xi^i)$ para selecionar os cenários representativos.

8.3 Seleção Sucessiva de Cenários

O critério de *seleção sucessiva de cenários* - SSC - é bastante similar ao esquema ROCSeq, porém, com duas importantes diferenças:

- o número de cenários representativos n_{LP} não necessariamente é aumentado ao longo das iterações do método, como apresentado na Figura 8.2;
- a métrica d_{Ξ} em (8.6) é substituída pela métrica d_{λ} , dependente de um parâmetro $\lambda \in (0, 1]$, que calcula a distância entre os cenários e considera alguma informação acerca dos valores $f(z, \xi^i)$, para $z \in \mathcal{X}$ e $i \in \{1, \dots, N\}$.

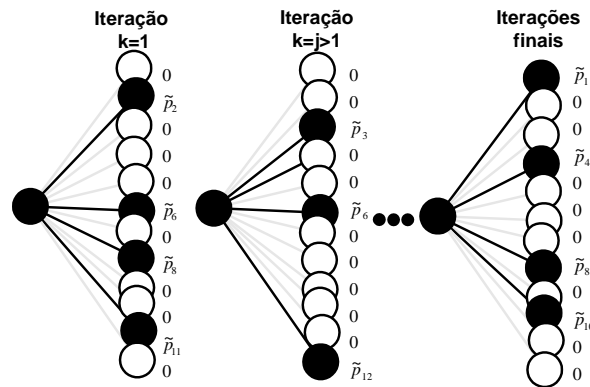


Figura 8.2: Seleção sucessiva de cenários.

Seja a métrica $d_{\lambda} : \Xi \times \Xi \rightarrow \mathcal{R}_+$ dada por

$$d_{\lambda}(\xi, \tilde{\xi}) := \lambda d_{\Xi}(\xi, \tilde{\xi}) + (1 - \lambda) d_f(\xi, \tilde{\xi}), \quad (8.12)$$

com $\lambda \in (0, 1]$ e $d_f(\xi, \tilde{\xi})$ uma função não negativa que *estima* a distância entre os valores $f(z, \xi)$ e $f(z, \tilde{\xi})$, para $z \in \mathcal{X}$ um ponto fixo, como apresentado a seguir.

Oráculo Inexato 4 (SELEÇÃO SUCESSIVA DE CENÁRIOS)

Passo 0 (inicialização).

Sejam $x \in \mathcal{R}^n$ um ponto dado, dois parâmetros $\lambda \in (0, 1]$ e $\varepsilon_{roc} \geq 0$ (ou alternativamente $n_{LP} \leq N$), e uma pseudo-norma $d_{\Xi}(\cdot, \cdot)$.
 Se ainda não foram calculadas, calcule as pseudo-distâncias $d_{\Xi}(\xi^j, \xi^l)$.
 Defina $I^E = \emptyset$ e $d_{jl} = d_{\Xi}(\xi^j, \xi^l)$, para $j, l \in \{1, \dots, N\}$.
 Faça $s_{l_m} = +\infty$.

Passo 1 (seleção de cenários). Enquanto $s_{l_m} > \varepsilon_{roc}$ (ou alternativamente $|I^E| < n_{LP}$) calcule para cada $l \notin I^E$

$$z_l = \sum_{j \notin I^E, j \neq l} p_j d_{jl} \quad \text{e selecione } l_m \in \arg \min_{l \notin I^E} z_l.$$

Faça $I^E = I^E \cup \{l_m\}$.

Para cada $j, l \notin I^E$, atualize as distancias $d_{jl} = \min(d_{jl}, d_{jl_m})$.

Passo 2 (cenários representativos e cálculos exatos).

Para todo $i \in I^E$, determine $f(x, \xi^i)$ e $\mathfrak{g}_i \in \partial f(x, \xi^i)$ resolvendo (8.1b).

Passo 3 (cenários não representativos e cálculos inexatos). Para $j \notin I^E$,

determine a aproximação $f_x(\xi^j)$ de $f(x, \xi^i)$ resolvendo aproximadamente o problema (8.1b).

Passo 4 (atualização das distâncias). Calcule para todo $i, l \in \{1, \dots, N\}$

$$d_f(\xi^i, \xi^l) := |\mathfrak{f}_i - \mathfrak{f}_l|, \quad \text{para } \mathfrak{f}_i := \begin{cases} f(x, \xi^i) & \text{se } i \in I^E \\ f_x(\xi^i) & \text{se } i \notin I^E, \end{cases} \quad (8.13)$$

e defina $d_{jl} = \lambda d_{\Xi}(\xi^j, \xi^l) + (1 - \lambda) d_f(\xi^i, \xi^l)$.

Step 5 (redistribuição) Redistribua as probabilidades pela regra (8.4).

□

O Oráculo Inexato 4 fornece as seguintes estimativas:

$$\begin{cases} f_x = \sum_{i \in I^E} \tilde{p}_i f(x, \xi^i) \\ g_x = \sum_{i \in I^E} \tilde{p}_i \mathfrak{g}_i. \end{cases}$$

Desde que $\mathfrak{g}_i \in \partial f(x, \xi^i)$ tem norma limitada por alguma constante $\Lambda > 0$, tanto o Oráculo Inexato 3, quanto o Oráculo Inexato 4, satisfazem (5.2).

Em particular, quando o problema de segundo estágio (8.1b) é linear, as aproximações no Passo 3 podem ser obtidas como explicado a seguir.

Para cada $j \notin I^E$, defina

$$f_x(\xi^j) = f_1(x) + (b_j - T_j x)^\top u_{i(j)},$$

com $u_{i(j)}$ uma solução dual do problema (8.1b) definido no cenário $\xi^{i(j)}$, onde

$$i(j) := \arg \min_{l \in I^E} d_\lambda(\xi^j, \xi^l)$$

é o índice do cenário “representativo” mais próximo do cenário “não representativo” $j \notin I^E$.

A consideração dos valores $f(\cdot, \xi)$ para calcular a distâncias entre os cenários foi originalmente proposta em [59]. Os autores consideram um programa linear estocástico em dois estágios, e utilizam a função não negativa $d_0(\xi^i, \xi^j) = d_f(\xi^i, \xi^j) = |f(z^1, \xi^i) - f(z^1, \xi^j)|$ para calcular a “distância” entre os cenários. A função d_0 assim definida possui duas deficiências:

- (i) o ponto $z^1 \in \mathcal{X}$ utilizado para calcular $f(z^1, \xi^i)$, com $i \in \{1, \dots, N\}$, deve ser escolhido com cautela para não proporcionar um viés no processo de seleção de cenários, que é realizado uma única vez;
- (ii) a função d_f não satisfaz as condições de uma métrica, pois cenários diferentes podem proporcionar o mesmo custo $f(z^1, \cdot)$, i.e., $d_f(\xi^i, \xi^j) = 0$ para $\xi^i \neq \xi^j$. Sem esta condição importante, a desigualdade

$$|f_{I^E}(z) - f^N(z)| \leq \sum_{j \in \mathbb{J}} \tilde{p}_j \min_{i \in I^E} d_f(\xi^j, \xi^i)$$

pode não ser verificada para $z \in \mathcal{X} \setminus \{z^1\}$. Isto inviabiliza, em termos teóricos, a aplicação da ROC.

Desde que $\lambda > 0$ e d_Ξ seja uma métrica, então a função d_λ é uma métrica. Deste modo, a deficiência do item (ii) é reparada. Como as distâncias $d_\lambda(\cdot, \cdot)$ são atualizadas a cada iteração do método, os cenários que eram pouco representativos para a decisão viável z^j podem ser selecionados para definir o subconjunto de índices representativos I^E em uma iteração $k > j$, fazendo com que a deficiência do item (i) também seja sanada.

A seguir é desenvolvido um oráculo inexato aplicável tanto ao método de feixes parcialmente inexatos, quanto ao método de nível parcialmente inexato.

8.4 Classificação em Grupos

O oráculo parcialmente inexato proposto nesta seção satisfaz as condições dadas em (4.2) e (6.2). Para este fim, é suposto que o problema (8.1) satisfaça as hipóteses

NL1-NL6, e além disso, a função objetivo de segundo estágio é determinística, i.e.,

$$f_2(z, \xi) = f_2(z), \quad \text{para todo } z \in \mathcal{X} \text{ e } \xi \in \Xi.$$

Esta hipótese coincide com a hipótese de custo q fixo para PLs, como considerado na Seção (7.2.1). Assim sendo, pelas Proposições (2.3) e (2.4), a função de recurso

$$Q(z, \xi) := \inf_{y \in \mathcal{X}(z, \xi)} f_2(y, \xi)$$

é convexa tanto na variável z , quanto na variável ξ . Esta propriedade é, pela hipótese NL4, transferida para a função

$$f(z, \xi) := f_1(z) + \inf_{y \in \mathcal{X}(z, \xi)} f_2(y, \xi),$$

definida em (8.1b).

Em especial, a convexidade em relação a ξ assegura a desigualdade seguinte,

$$\mathbb{E}[f(z, \xi)] \geq f(z, \mathbb{E}[\xi]) \quad \text{para todo } z \in \mathcal{X}, \quad (8.14)$$

denominada *desigualdade de Jensen*.

Suponha que o conjunto $\{\xi^1, \xi^2, \dots, \xi^N\}$ com N cenários seja subdividido em $ng < N$ grupos \mathcal{I}_j , $j = 1, \dots, ng$, satisfazendo:

$$\mathcal{I}_j \cap \mathcal{I}_\ell = \emptyset \quad \text{se } j \neq \ell, \quad \text{e } \mathcal{I}_1 \cup \dots \cup \mathcal{I}_{ng} = \{1, 2, \dots, N\}. \quad (8.15a)$$

Para $j = 1, \dots, ng$ sejam, respectivamente,

$$\bar{p}_j := \sum_{i \in \mathcal{I}_j} p_i, \quad \text{e } \bar{\xi}^j := \frac{1}{\bar{p}_j} \sum_{i \in \mathcal{I}_j} p_i \xi^i \quad (8.15b)$$

a probabilidade de ocorrência, e o cenário médio do grupo \mathcal{I}_j . Então, o resultado seguinte é devido a desigualdade de Jensen.

Proposição 8.2 *Dados N cenários com probabilidades associadas p_i , $i = 1, \dots, N$, seja \mathcal{I}_j (para $j = 1, \dots, ng$) uma classificação dos cenários satisfazendo (8.15a). Suponha que os cenários médios $\bar{\xi}^j$ e as respectivas probabilidades \bar{p}_j satisfazem (8.15b). Então, se a função $f(z, \xi)$ é convexa em ξ para todo $z \in \mathcal{X}$ fixo, vale a relação*

$$\sum_{i=1}^N p_i f(z, \xi^i) \geq \sum_{j=1}^{ng} \bar{p}_j f(z, \bar{\xi}^j) \geq f(z, \sum_{i=1}^N p_i \xi^i). \quad (8.16)$$

Prova. Dadas as hipóteses, pela desigualdade de Jensen, tem-se que

$$\sum_{i \in \mathcal{I}_j} \frac{p_i}{\bar{p}_j} f(z, \xi^i) \geq f(z, \bar{\xi}^j) \quad \text{para todo } z \in \mathcal{X} \text{ e } j = 1, \dots, ng.$$

Então, para verificar a primeira desigualdade em (8.16) basta multiplicar a desigualdade acima por \bar{p}_j e somar em j . A segunda desigualdade em (8.16) segue novamente da desigualdade de Jensen, pois $\sum_{j=1}^{ng} \bar{p}_j \bar{\xi}^j = \sum_{i=1}^N p_i \xi^i$. ■

O resultado acima é útil no sentido que, qualquer que seja o agrupamento dos N cenários satisfazendo (8.15), a função $f_z = f^{ng}(z) := \sum_{j=1}^{ng} \bar{p}_j f(z, \bar{\xi}^j)$ é uma aproximação inferior para $f(z)$, para $z \in \mathcal{X}$. Para explorar esta relação, é dado a seguir um oráculo inexato aplicável aos métodos de feixes parcialmente inexatos.

Oráculo Inexato 5 (CLASSIFICAÇÃO EM GRUPOS)

Passo 0 (inicialização).

Dado $z \in \mathcal{X}$, seja ng um parâmetro referente ao número de grupos.

Se são informados os ng grupos \mathcal{I}_j , vá ao Passo 2;

caso contrário, vá ao Passo 1.

Passo 1 (classificação).

Classifique os N cenários em ng grupos \mathcal{I}_j satisfazendo (8.15a).

Calcule os cenários médios e as respectivas probabilidades pela regra (8.15b).

Passo 2 (cálculos exatos para os cenários médios).

Para cada $j = 1, \dots, ng$, determine $f(z, \bar{\xi}^j)$ e $\mathfrak{g}_j \in \partial f(z, \bar{\xi}^j)$ resolvendo (8.1b).

□

O Oráculo Inexato 5 fornece as seguintes estimativas:

$$\left[\begin{array}{l} f_z = \sum_{j=1}^{ng} \bar{p}_j f(z, \bar{\xi}^j) \\ g_z = \sum_{j=1}^{ng} \bar{p}_j \mathfrak{g}_j, \end{array} \right.$$

que satisfazem as condições dadas em (4.2).

Proposição 8.3 *Seja o problema (8.1), e suponha que as hipóteses NL1-NL6 sejam satisfeitas. Além disso, suponha que a função objetivo de segundo estágio f_2 é determinística. Então, se \mathcal{X} é limitado o Oráculo Inexato 5 satisfaz (4.2).*

Prova. Sejam $z \in \mathcal{X}$ um ponto fixo, e $j(i)$ o índice do grupo a que o cenário ξ^i

pertence, i.e., $i \in \mathcal{I}_{j(i)}$. Considere o seguinte desenvolvimento

$$\begin{aligned}
f^N(z) - f_z &= \sum_{i=1}^N p_i f(z, \xi^i) - \sum_{j=1}^{ng} \bar{p}_j f(z, \bar{\xi}^j) \\
&= \sum_{i=1}^N p_i [f(z, \xi^i) - f(z, \bar{\xi}^{j(i)})] \\
&\leq \sum_{i=1}^N p_i \|f(z, \xi^i) - f(z, \bar{\xi}^{j(i)})\| \\
&\leq \sum_{i=1}^N p_i d_{\Xi}(\xi^i, \bar{\xi}^{j(i)}).
\end{aligned}$$

A primeira desigualdade é devida às hipóteses NL2 e NL3, e última desigualdade é devida a hipótese NL6. Como o processo estocástico ξ tem variância finita (e N é finito), o termo $\sum_{i=1}^N p_i d_{\Xi}(\xi^i, \bar{\xi}^{j(i)})$ é finito e, deste modo, é limitado por alguma constante $\varepsilon_f < \infty$. Logo, pela Proposição 8.2 tem-se que

$$f_z \in [f^N(z) - \varepsilon_f, f^N(z)], \quad \text{para todo } z \in \mathcal{X}.$$

Segue do Oráculo Inexato 5 que $g_z \in \partial f_z$ para todo $z \in \mathcal{X}$. Então, para qualquer x tem-se que

$$\begin{aligned}
f^N(x) \geq f_x &\geq f_z + g_z^\top(x - z) \\
&\geq f^N(z) - \varepsilon_f + g_z^\top(x - z),
\end{aligned}$$

que proporciona as relações desejadas dadas em (4.2). ■

Para que o Oráculo Inexato 5 possa ser empregado no método de nível parcialmente inexato é necessário assumir que o subdiferencial $\partial f(\cdot, \xi)$ seja limitado por alguma constante $\Lambda > 0$.

Proposição 8.4 *Seja o problema (8.1), e suponha que as hipóteses NL1-NL6 sejam satisfeitas. Além disso, suponha que a função de segundo estágio f_2 é determinística, e que o subdiferencial $\partial f(\cdot, \xi)$ seja limitado por alguma constante $\Lambda > 0$, para todo $\xi \in \Xi$. Então, se \mathcal{X} é limitado o Oráculo Inexato 5 satisfaz (6.2).*

Prova. Dada a Proposição 8.3, o resultado enunciado é imediato a partir da definição de f_z e g_z . ■

Pela Proposição 8.3, a imprecisão do oráculo é limitada superiormente por $\sum_{i=1}^N p_i d_{\Xi}(\xi^i, \bar{\xi}^{j(i)})$. Assim sendo, deve-se classificar os cenários em ng grupos de modo que esta soma seja minimizada. Quando a métrica d_{Ξ} é da forma $d_{\Xi}(\xi, \tilde{\xi}) = M \|\xi - \tilde{\xi}\|^2$, com $M > 0$ uma constante e $\|\cdot\|$ uma norma proveniente de algum produto interno, várias técnicas são possíveis para a classificação dos ce-

nários. Dentre elas se destaca o bem conhecido *Algoritmo K-means*, desenvolvido em [60]. Outra possibilidade é utilizar o algoritmo proposto em [61].

É importante mencionar que o Oráculo Inexato 5 pode ser aplicado de maneira “estática”, efetuando o Passo 2 uma única vez; ou “dinâmica”, selecionando grupos de cenários a cada iteração. Suponha que a função f_2 seja linear, i.e., $f_2(y, \xi) = q$, com q fixo, para garantir a desigualdade de Jensen. Então, definindo $\mathfrak{q}_i^k := h_i - T_i z^k$, o processo de agrupamento dos cenários pode ser realizado (dinamicamente) no conjunto $\{\mathfrak{q}_1^k, \mathfrak{q}_2^k, \dots, \mathfrak{q}_N^k\}$, de modo que as informações do iterado $z^k \in \mathcal{X}$ sejam utilizadas.

A seguir são considerados os programas estocásticos em multiestágios.

Capítulo 9

Métodos de Feixes Inexatos Aplicados à Programação Linear Estocástica em Multiestágios

São considerados neste capítulo os programas lineares estocásticos em multiestágios da forma

$$\min_{\substack{A_1 x_1 = b_1 \\ x_1 \geq 0}} c_1^\top x_1 + \mathbb{E}_{|\xi_{[1]}} \left[\min_{\substack{B_2 x_1 + A_2 x_2 = b_2 \\ x_2 \geq 0}} c_2^\top x_2 + \mathbb{E}_{|\xi_{[2]}} \left[\cdots + \mathbb{E}_{|\xi_{[T-1]}} \left[\min_{\substack{B_T x_{T-1} + A_T x_T = b_T \\ x_T \geq 0}} c_T^\top x_T \right] \right] \right]. \quad (9.1)$$

A representação acima é conhecida como *formulação aninhada*, [25]. Utilizando a notação da Seção 2.3, algumas (ou todas) as componentes dos vetores c_t, b_t e matrizes B_t são aleatórias, formando o processo estocástico $\xi_t := (c_t, B_t, b_t)$, $t = 2, \dots, T$, com $\xi_1 = \xi_1 = (c_1, A_1, b_1)$ determinístico. Deste modo, o operador *valor esperado* $\mathbb{E}_{|\xi_{[t]}}$ é aplicado com respeito a distribuição de probabilidades de ξ_{t+1} condicionada à trajetória realizada $\xi_{[t]} = (\xi_1, \dots, \xi_t)$. Denotando o conjunto suporte de ξ_t por Ξ_t , então $\xi_{[t]}$ pertence ao conjunto suporte dado por $\Xi_{[t]} = (\Xi_1 \times \dots \times \Xi_t)$, para $t = 1, \dots, T$.

Como mencionado no Capítulo 2, os métodos de otimização para resolver problemas do tipo (9.1) são divididos em duas classes, [15]: aqueles que definem todo o conjunto de incertezas mediante uma *árvore de cenários*, denominados *métodos baseados em árvore*; e aqueles que consideram amostragem durante o processo de otimização, denominados *métodos baseados em amostragem*.

Dentre os métodos baseados em árvores, destacam-se o *método dual de planos cortantes* apresentado em [52, Capítulo 3], o *progressive hedging* desenvolvido em [62], o *método dual de barreira lagrangiana* apresentado em [63], e a *decomposição aninhada de Benders - DAB* [25], que é uma extensão ao caso multiestágios do método *L-shaped* dado em [3]. A DAB é o principal método da classe de métodos

baseados em árvores, e é o pilar dos métodos baseados em amostragem.

A diferença principal entre estas duas classes de métodos se deve ao fato que os métodos com amostragem percorrem a árvore de cenários (que é em geral bem maior do que as árvores utilizadas nos métodos baseados em árvore) através de apenas *algumas de suas ramificações*, escolhidas aleatoriamente. Esta é, por exemplo, a estratégia adotada pela *programação dinâmica dual estocástica* - SDDP - desenvolvida em [12]; pela *decomposição aninhada abreviada* - AND [13]; pela *decomposição com amostragem reduzida* - ReSa [14]; ou ainda, pelo *método de planos cortantes com amostragem parcial* - CUPPS - desenvolvido em [15].

É importante ressaltar que os métodos baseados em amostragem são bem definidos para problemas do tipo (9.1) que satisfazem a hipótese de *independência temporal* do processo estocástico, i.e., o valor esperado $\mathbb{E}_{|\xi_{[t]}}$ independe da trajetória $\xi_{[t]}$, e pode ser escrito por \mathbb{E}_{P_t} , onde P_t é a distribuição de probabilidades definida no espaço amostral (Ξ_t, \mathcal{F}_t) . Esta característica faz com que os métodos baseados em amostragem sejam menos gerais do que os métodos baseados em árvores. Quando o processo estocástico é definido por algum modelo autorregressivo, é possível reformular o programa estocástico de forma que atenda à hipótese de independência temporal, [27] (estendendo o vetor de estado x_t com a trajetória $\xi_{[t-1]}$).

É apresentada na seção 9.1 a seguir uma revisão da técnica de decomposição aninhada, incluindo alguns comentários sobre como combinar os métodos de feixes com esta decomposição. Com o intuito de aplicar os métodos de feixes inexato e parcialmente inexatos, uma aproximação por dois estágios e uma abordagem dual do problema (9.1) são consideradas nas Seções 9.2 e 9.3, respectivamente.

9.1 Decomposição Aninhada

Novamente, seja $t \in \{1, \dots, T\}$ um índice de tempo no horizonte de T estágios, (Ξ, \mathcal{F}, P) o espaço de probabilidades onde $\xi := (\xi_1, \dots, \xi_T) \in \Xi = \Xi_{[T]}$ é um parâmetro do problema chamado de *cenário*, e \mathcal{F} é a σ -álgebra gerada por Ξ .

Definindo $\mathcal{X}_1 := \{x_1 \in \mathcal{R}_+^{n_1} : A_1 x_1 = b_1\}$, seja para $t = 2, \dots, T$ e cada $\xi \in \Xi$, $x_t(\xi) \in \mathcal{X}_t(x_{t-1}, \xi_{[t]}) := \{x_t \in \mathcal{R}_+^{n_t} : B_t x_{t-1} + A_t x_t = b_t\}$ uma função de ξ . Definindo $\eta_t := \sum_{i=1}^t n_i$, uma *estratégia* $x(\xi_{[t]}) := (x_1(\xi), \dots, x_t(\xi)) \in \mathcal{R}^{\eta_t}$ é função dos cenários $\xi_{[t]} \in \Xi_{[t]}$. Com esta notação, a formulação dinâmica do programa linear estocástico em multiestágios (9.1) é dada por

$$\min_{x_1 \in \mathcal{X}_1} f(x_1), \quad \text{com} \quad f(x_1) := c_1^\top x_1 + \mathcal{Q}_2(x_1, \xi_{[1]}), \quad (9.2a)$$

onde

$$\mathcal{Q}_{t+1}(x_t, \xi_{[t]}) := \mathbb{E}_{|\xi_{[t]}}[\mathcal{Q}_{t+1}(x_t, \xi_{[t+1]})] \text{ para } t = 1, \dots, T-1, \quad (9.2b)$$

e $\mathcal{Q}_{T+1}(x_T, \xi_{[T]}) := 0$. A função de recurso médio \mathcal{Q} usa as funções de recurso

$$\mathcal{Q}_{t+1}(x_t, \xi_{[t+1]}) := \min c_{t+1}^\top x_{t+1} + \mathcal{Q}_{t+2}(x_{t+1}, \xi_{[t+1]}) \text{ s.a. } x_{t+1} \in \mathcal{X}_{t+1}(x_t, \xi_{[t+1]}). \quad (9.2c)$$

A função \mathcal{Q}_T é convexa, pela Proposição 2.3. Utilizando o argumento de indução, desde o T -ésimo até o primeiro estágio, conclui-se que f é uma função convexa e, portanto, (9.2a) é um programa convexo.

Ao longo desta seção a função de recurso $\mathcal{Q}_{t+1}(x_t, \xi_{[t]})$ será representada por $\mathcal{Q}_{t+1}(x_t)$, enquanto que cada problema $\mathcal{Q}_{t+1}(x_t, \xi_{[t+1]})$ será escrito por $Q_{t+1}(x_t, \xi_{t+1})$. Esta é a representação utilizada quando a hipótese de independência temporal do processo estocástico se faz presente. No entanto, esta nomenclatura foi escolhida apenas para não sobrecarregar a notação e, deste modo, não implica necessariamente a independência temporal.

Quando o processo estocástico é contínuo e não há uma representação analítica para o valor esperado que define a função objetivo em (9.1), o problema é intratável computacionalmente devido às limitações numéricas para o cálculo da integral multidimensional. Desta forma, o processo estocástico $\{\xi_t\}_{t=1}^T$ é discretizado em um número finito N de cenários, e para $p_t^i = P(\xi_t = \xi_t^i | \xi_{[t-1]})$ a probabilidade de ocorrência do evento ξ_t^i condicionada à trajetória $\xi_{[t-1]}$, a função de recurso $\mathcal{Q}_t(x_{t-1})$ é representada pelo somatório

$$\mathbb{E}_{|\xi_{[t-1]}}[\mathcal{Q}_t(x_{t-1}, \xi_t)] = \sum_{i=1}^{S(\xi_{[t-1]})} p_t^i [\mathcal{Q}_t(x_{t-1}, \xi_t^i)],$$

onde $S(\xi_{[t-1]})$ é o número de nós sucessores (no estágio t) da trajetória $\xi_{[t-1]}$. Novamente, por conveniência notacional, será considerado que $N_t := S(\xi_{[t-1]})$ para $t = 1, \dots, T$, ou seja, o processo estocástico discretizado é representado por árvores *equilibradas*, como ilustrado pela Figura 2.1 ($N_t = 2$). Deste modo, o problema (9.2a) pode ser aproximado pela representação

$$\min_{x_1 \in \mathcal{X}_1} f^N(x_1), \quad \text{com } f^N(x_1) := c_1^\top x_1 + \sum_{i=1}^{N_2} p_2^i [\mathcal{Q}_2(x_1, \xi_2^i)], \quad (9.3)$$

onde $N := \prod_{t=2}^T N_t$. Se $p_t^i = N_t^{-1}$ para $i = 1, \dots, N_t$ e $t = 2, \dots, T$, o problema (9.3) é denominado *SAA (Sample Average Approximation)*, [1]. Naturalmente, quanto mais fina for a discretização de $\{\xi_t\}_{t=1}^T$, i.e., quanto maior for o valor de N , melhor será a aproximação do processo estocástico e mais próximo ficará (9.3) de (9.2a).

No entanto, maior será o esforço computacional para resolver (9.3).

Ao longo desta seção são assumidas as seguintes hipóteses acerca do problema (9.1).

H1 - o processo estocástico possui (ou é discretizado assumindo) um número finito N de eventos $\xi^i = \xi_{[T]}^i$;

H2 - o programa linear (9.2c) possui valor ótimo finito para qualquer realização de ξ_t c.p.1, com $t = 1, \dots, T$.

A hipótese H1 permite a manipulação computacional do problema. Já a hipótese H2 assegura que todos os programas (9.2c) no estágio t são viáveis, independentemente da realização ξ_t e da decisão no estágio anterior x_{t-1} , e que possuem solução.

9.1.1 Decomposição Aninhada de Benders

Dado um conjunto fixo de eventos incertos e suas respectivas probabilidades, o procedimento para resolver (9.2) pela decomposição aninhada de Benders - DAB - consiste em aproximar as funções de recurso $Q_t(x_{t-1})$ por seus hiperplanos suporte, ou “cortes”. Cada iteração tem duas etapas: uma *regressiva*, para definir os cortes, e outra *progressiva*, para definir as novas variáveis de estado.

ETAPA REGRESSIVA:

Seja $x = (x_1, x_2(\xi), \dots, x_T(\xi))$ uma estratégia para o cenário $\xi \in \Xi$. Para $t = T$ e k uma iteração do método, o modelo de planos cortantes $\check{Q}_T^k(\cdot)$ é definido como

$$\check{Q}_T^k(\cdot) := \max_{j=1, \dots, k} \ell_T^j(\cdot), \quad \text{para} \quad \ell_T^j(\cdot) := Q_T(x_{T-1}^j) + (\cdot - x_{T-1}^j)^\top g_T^j. \quad (9.4a)$$

onde $g_T^j \in \partial Q_T(x_{T-1}^j)$. A função ℓ_T^j é chamada de *plano cortante*, ou simplesmente, corte¹. Note que, para calcular $\check{Q}_T^k(x_{T-1}^j)$, o problema (9.2c) deve ser resolvido para todos os N_T cenários. Pela convexidade da função $Q_T(\cdot)$, vale a relação $\check{Q}_T^k(\cdot) \leq Q_T(\cdot)$. Então o problema

$$\underline{Q}_{T-1}^k(x_{T-2}, \xi_{T-1}) := \min c_{T-1}^\top x_{T-1} + \check{Q}_T^k(x_{T-1}) \quad \text{s.a.} \quad x_{T-1} \in \mathcal{X}_{T-1}(x_{T-2}, \xi_{T-1}) \quad (9.4b)$$

é uma aproximação (inferior) do problema (9.2c) para $t = T - 2$. Novamente, o modelo de planos cortantes para $t = T - 1$ é definido por

$$\check{Q}_{T-1}^k(\cdot) := \max_{j=1, \dots, k} \ell_{T-1}^j(\cdot), \quad (9.4c)$$

¹Porém, o corte não necessariamente é exato, e deste modo, o plano não é “cortante”.

para $\ell_{T-1}^j(\cdot) := \mathbb{E}[Q_{T-1}^j(x_{T-2}^j, \xi_{T-1}) + (\cdot - x_{T-2}^j)^\top g_{T-1}^j(\xi_{T-1})]$, com $g_{T-1}^j(\xi_{T-1}) \in \partial Q_{T-1}^j(x_{T-2}^j, \xi_{T-1})$. Utilizando a convexidade tem-se que $\check{Q}_{T-1}^k(\cdot) \leq Q_{T-1}(\cdot)$. Este procedimento é repetido até $t = 1$, que fornece o problema

$$\min_{x_1 \in \mathcal{X}_1} \check{f}_k(x_1), \quad \text{com} \quad \check{f}_k(x_1) := c_1^\top x_1 + \check{Q}_2^k(x_1). \quad (9.4d)$$

□

O resultado seguinte é consequência direta da etapa regressiva, acima desenvolvida.

Proposição 9.1 *Se as hipóteses H1 e H2 são verificadas, então*

$$\check{Q}_t^k(\cdot) \leq Q_t(\cdot) \quad \text{para } t = 2, \dots, T. \quad (9.5)$$

□

Como consequência, a etapa regressiva define uma cota inferior

$$\underline{z}^{k+1} = \check{f}_k(x_1^{k+1}), \quad (9.6)$$

do valor ótimo de (9.3). A fim de obter uma cota superior para o valor ótimo de (9.3), a DAB considera a etapa progressiva a seguir.

ETAPA PROGRESSIVA:

A etapa progressiva para a DAB consiste em resolver, para $t = 2$ até $t = T$ e todos os eventos ξ_{t-1}^i , os programas lineares (PLs)

$$\underline{Q}_{t-1}^k(x_{t-2}, \xi_{t-1}) := \min c_{t-1}^\top x_{t-1} + \check{Q}_t^k(x_{t-1}) \quad \text{s.a. } x_{t-1} \in \mathcal{X}_{t-1}(x_{t-2}, \xi_{t-1}). \quad (9.7)$$

Esta etapa define $x^k(\xi^i)$, uma solução \mathcal{F} -mensurável, i.e., um ponto viável para (9.2), que fornece uma cota superior para o valor ótimo de (9.3):

$$\bar{z}^i(x_1^i) := c_1^\top x_1^i + \mathbb{E} \left[\sum_{t=2}^T c_t^\top x_t^i(\xi_t) \right]. \quad (9.8)$$

□

Seja a sequência $\{x^i\}_{i=1}^k$ de pontos \mathcal{F} -mensuráveis obtidos até a iteração k , e $\bar{z}^k := \min_{i=1, \dots, k} \{\bar{z}^i(x_1^i)\}$. O teste de parada empregado pela DAB termina as iterações do algoritmo quando $\bar{z}^k - \underline{z}^k \leq \delta_{\text{To1}}$, para $\delta_{\text{To1}} > 0$ uma tolerância dada. Nesta situação, $x_1^k \in \mathcal{X}_1$ é uma δ_{To1} -solução do problema (9.3).

É apresentada na Figura 9.1 uma estrutura em árvore de cenários típica, considerada pela DAB, com $N_t = S(\xi_{[t-1]}) = 3$ para $t = 2, 3, 4, 5$. Cada nó (evento ξ_t) da árvore representa um problema do tipo (9.2c).

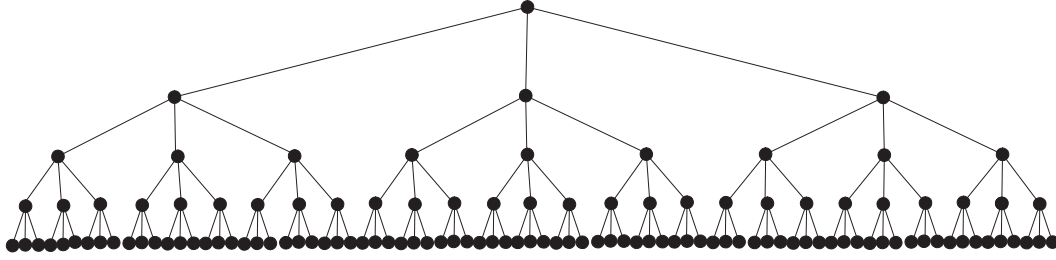


Figura 9.1: Árvore com 5 períodos, 81 cenários, e 121 nós.

Apesar de aproveitar da estrutura de decomposição e possuir um teste de parada bastante eficiente, o algoritmo da DAB é vulnerável à denominada “maldição da dimensionalidade”, devido à necessidade de resolver um subproblema para cada nó da árvore de cenários. Por exemplo, para a árvore de cenários da Figura 9.1, com apenas três discretizações por estágio, a DAB necessita resolver 160 PLs a cada iteração k : 121 na etapa progressiva e 39 na etapa regressiva (para $t = T$, a etapa regressiva pode ser realizada utilizando o multiplicador de Lagrange obtido na etapa progressiva).

Como na maioria das aplicações a árvore de cenários é muito mais densa, percorrer todos os cenários, resolvendo cada subproblema, envolve um esforço computacional excessivo. Dada esta dificuldade, foram desenvolvidos os métodos de resolução inexata baseados em amostragens, semelhantes à DAB, que escolhem aleatoriamente alguns cenários da árvore durante a etapa progressiva.

A seguir considera-se a aplicação dos métodos de feixes conjuntamente com a decomposição aninhada de Benders.

9.1.2 Decomposição Aninhada e Métodos de Feixes

Assim como a decomposição aninhada de Benders, o método feixes proximal pode ser empregado para resolver o problema (9.1). O termo quadrático do método pode ser adicionado no primeiro estágio, definindo a função objetivo

$$\phi_k(\cdot) := c^\top(\cdot) + \check{Q}_2^k(\cdot) + \frac{1}{2t_k} \|\cdot - x_1^k\|^2.$$

Neste sentido, o procedimento do método é quase idêntico aquele do caso em dois estágios. No entanto, cada subproblema de segundo estágio é agora definido a partir da subárvore determinada no cenário ξ_2^j , com $j = 1, \dots, N_2$. É importante mencionar que, ao contrário do método de feixes proximal exato, a cada iteração do programa mestre é incorporada uma linearização da função

$$\check{f}_k(\cdot) := c^\top(\cdot) + \check{Q}_2^k(\cdot),$$

em vez da função

$$f^N(\cdot) := c^\top(\cdot) + Q_2(\cdot),$$

como nos programas estocásticos em dois estágios. Portanto, a escolha de um *passo sério* para o método difere da regra apresentada no Capítulo 3, a menos que valha a igualdade $\check{Q}_2^k(z_1^{k+1}) = Q_2(z_1^{k+1})$, para o atual iterado $z_1^{k+1} \in \mathcal{X}_1$. Esta última relação é satisfeita se a DAB é aplicada em cada uma das N_2 subárvores, com z_1^{k+1} fixo. Só então (após cada programa estocástico definido em cada subárvore ser resolvido até a otimalidade) pode-se determinar se o ponto z_1^{k+1} será ou não um novo centro de estabilidade, x_1^{k+1} . Fica assim evidenciada a principal dificuldade de aplicar o método de feixes proximal exato (resp. inexato) ao problema (9.1): a ausência de um critério de descida para determinar os passos sérios e nulos, devido a falta de conhecimento do valor $f^N(z_1^k)$ (resp. f_z^k , uma aproximação razoável de $f^N(z_1^k)$). Pelo mesmo motivo, a determinação dos conjuntos de níveis nos métodos de níveis fica comprometida, quando combinada com a decomposição aninhada.

Como mencionado em [52, p. 186], incluir um termo quadrático (ou um conjunto de nível) em cada subproblema (9.7) apresenta ainda mais complicações. Com efeito, os valores ótimos de tais subproblemas modificados definidos no estágio t não proporcionam aproximações inferiores para as funções de recurso Q_{t-1} .

Estas implicações conduzem à seguinte conclusão: a aplicação dos métodos de feixes conjuntamente com a decomposição aninhada pode não ser uma boa opção para resolver o programa estocástico em multiestágio (9.1). Porém, a conclusão é diferente se ao invés da decomposição aninhada, a *decomposição dual*, ou simplesmente uma *aproximação por dois estágios*, for empregada. Estes são os assuntos tratados a seguir.

9.2 Aproximação por Dois Estágios de Programas em Multiestágios

Uma abordagem em dois estágios de um programa estocástico em multiestágios é obtida desprezando, a partir do segundo estágio, as restrições de mensurabilidade (ou implementabilidade) do problema. Afim de explicitar as restrições de mensura-

bilidade, considere a seguir a formulação estática do problema (9.1),

$$\left\{ \begin{array}{l} \min \quad \mathbb{E}[c_1^\top x_1(\xi) + c_2^\top x_2(\xi) + c_3^\top x_3(\xi) + \dots + c_T^\top x_T(\xi)] \\ \text{s. a} \quad A_1 x_1(\xi) = b_1 \\ \quad B_2 x_1(\xi) + A_2 x_2(\xi) = b_2 \\ \quad \quad B_3 x_2(\xi) + A_3 x_3(\xi) = b_3 \\ \quad \quad \quad \ddots \quad \quad \quad \ddots \quad \quad \quad \ddots \quad \quad \quad \vdots \\ \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad B_{T-1} x_{T-1}(\xi) + A_T x_T(\xi) = b_T \\ \quad \quad x_t(\xi) \geq 0, \quad t = 1, \dots, T \\ \quad \quad x_t(\xi) \in \mathcal{F}_t, \quad t = 1, \dots, T, \end{array} \right. \quad (9.9)$$

onde a incerteza é representada pelo parâmetro $\xi_t = (c_t, B_t, b_t)$, para $t = 2, \dots, T$. Há, entretanto, um abuso da notação:

$$x_t(\xi) \in \mathcal{F}_t \Rightarrow x_t(\xi) \text{ é } \mathcal{F}_t \text{ - mensurável,}$$

e não um subconjunto de Ξ_t , como sugeri o símbolo \in . São apresentados no Exemplo 2.3 maiores esclarecimentos sobre as restrições de mensurabilidade. Com esta notação, a variável $x_t(\xi)$ é uma função da trajetória $\xi_{[t]}$, para $t = 1, \dots, T$. Isto significa que se dois cenários $\bar{\xi}$ e $\tilde{\xi}$ compartilham a mesma trajetória até um estágio t dado, as decisões associadas $x_j(\bar{\xi})$ e $x_j(\tilde{\xi})$ para $j = 1, \dots, t$ devem ser idênticas, i.e.,

$$\bar{\xi}_{[t]} = \tilde{\xi}_{[t]} \Rightarrow x_j(\bar{\xi}) = x_j(\tilde{\xi}), \quad \text{para todo } j = 1, \dots, t.$$

Uma simplificação do problema (9.9) consiste em desconsiderar as restrições de mensurabilidade $x_t(\xi) \in \mathcal{F}_t$, para $t = 2, \dots, T$. Com esta abordagem, o problema (9.9) pode ser escrito por

$$\left\{ \begin{array}{l} \min \quad c_1^\top x_1 + \mathbb{E}[c_2^\top x_2(\xi) + c_3^\top x_3(\xi) + \dots + c_T^\top x_T(\xi)] \\ \text{s. a} \quad A_1 x_1 = b_1 \\ \quad B_2 x_1 + A_2 x_2(\xi) = b_2 \\ \quad \quad B_3 x_2(\xi) + A_3 x_3(\xi) = b_3 \\ \quad \quad \quad \ddots \quad \quad \quad \ddots \quad \quad \quad \ddots \quad \quad \quad \vdots \\ \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad B_{T-1} x_{T-1}(\xi) + A_T x_T(\xi) = b_T \\ \quad \quad x_t(\xi) \geq 0, \quad t = 1, \dots, T. \end{array} \right. \quad (9.10)$$

Adotando a seguinte notação:

$$q := \begin{bmatrix} c_2 \\ c_3 \\ \vdots \\ c_T \end{bmatrix}, \quad T := \begin{bmatrix} B_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad h := \begin{bmatrix} b_2 \\ b_3 \\ \vdots \\ b_T \end{bmatrix}, \quad \text{e } W := \begin{bmatrix} A_2 & & & & & \\ B_3 & A_3 & & & & \\ & & \ddots & & & \\ & & & & & \\ & & & & & B_T & A_T \end{bmatrix},$$

o problema (9.10) é essencialmente um programa linear estocástico em dois estágios, e pode ser escrito como em (7.1). Claramente, o programa em dois estágios resultante possui um valor ótimo menor ou igual ao valor ótimo do problema (9.9). A proximidade entre estes valores depende da natureza do processo estocástico.

É apresentada na parte superior da Figura 9.2 uma árvore de cenários típica para a formulação estática (9.9), para $T = 3$. Cada nó da árvore representa um PL como em (9.2c). A árvore da parte inferior da Figura 9.2 é obtida a partir da árvore da parte superior, adotando a aproximação por dois estágios do problema (9.9). Cada retângulo representa um PL, com mais variáveis e restrições.

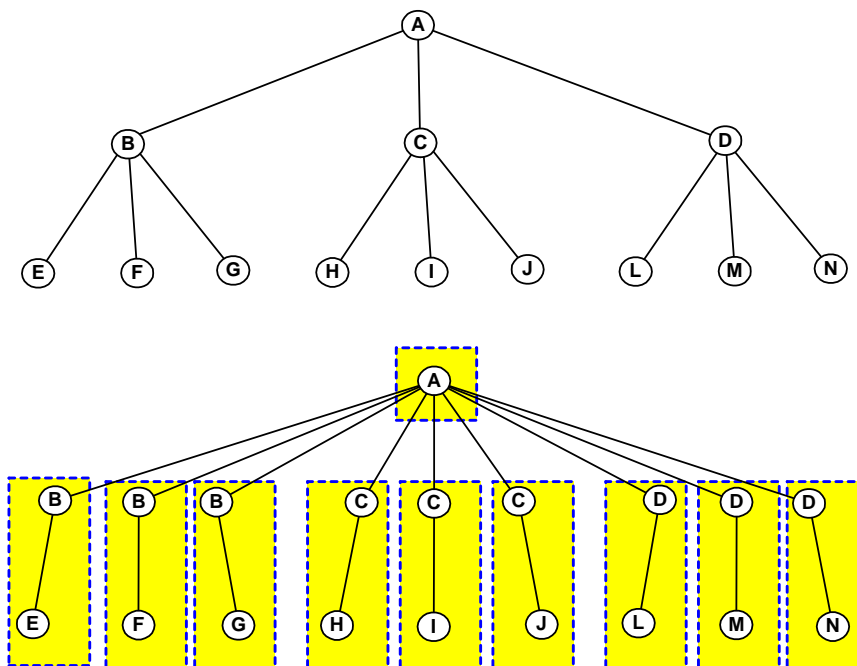


Figura 9.2: Aproximação por dois estágios de programas em multiestágios.

A abordagem em dois estágios de programas estocásticos em multiestágios é bastante conhecida, e é empregada, por exemplo, em [16] para a aplicação da decomposição inexata de Benders, e em [64] para a decomposição de nível. É apresentada em [65] uma análise deste tipo de aproximação em problemas de investimento, quando o horizonte temporal é de $T = 3$ estágios. Como explicado pelos autores, para o

tipo de problema estudado, a aproximação em dois estágios fornece quase sempre as mesmas decisões ótimas e valores ótimos do problema em multiestágios ($T = 3$).

Sem a necessidade de considerações adicionais, este tipo de simplificação permite que os oráculos inexatos e parcialmente inexatos apresentados no Capítulo 7 sejam empregados com os métodos de feixes (parcialmente) inexatos para resolver o programa resultante.

Como na programação em dois estágios, o principal interesse em um programa em multiestágios é definir uma decisão de primeiro estágio x_1 viável, que atenda algum critério de otimalidade. Neste sentido, apesar de ser uma aproximação trivial, a abordagem em dois estágios de programas estocásticos em multiestágios pode ser bastante útil, pois proporciona uma decisão x_1 viável para o primeiro estágio, e um limite inferior para o valor ótimo do problema (com N cenários). Um limite superior para o valor ótimo pode ser obtido através de simulações com a decisão x_1 , ou por exemplo, através da estratégia proposta a seguir.

9.2.1 Aproximação por Multiproblemas em Dois Estágios

Fundamentada na *técnica de contaminação* de uma árvore de cenários recentemente introduzida em [65], é empregada nesta seção uma abordagem para determinar candidatos à solução, e um intervalo (c.p.1) para o valor ótimo de um programa linear estocástico em multiestágios, baseado em uma árvore com um número finito cenários. Ao contrário de [65], a *distribuição de probabilidades contaminada* P_λ considerada nesta seção é obtida pela combinação de várias distribuições π^j , e não de apenas duas, i.e.,

$$P_\lambda = \sum_{i=1}^{n_d} \lambda_i \pi^i, \quad \text{para } \lambda \in \mathcal{R}_+^{n_d} \text{ com } \sum_{i=1}^{n_d} \lambda_i = 1. \quad (9.11)$$

Fixando um cenário ξ , seja $\mathcal{X}(\xi)$ o conjunto de pontos satisfazendo as restrições do problema (9.9), exceto, possivelmente, as restrições de mensurabilidade. Tal conjunto é dado por

$$\mathcal{X}(\xi) := \{x \geq 0 : A_1 x_1 = b_1, \text{ e } B_t x_{t-1} + A_t x_t = b_t, \quad t = 2, \dots, T\}, \quad (9.12)$$

onde novamente ξ_t é identificado com (c_t, B_t, b_t) . É suposto que o conjunto $\mathcal{X}(\xi)$ é q.c. não vazio e limitado. Dada a função indicadora $i_{\mathcal{X}(\xi)}(\cdot)$ do conjunto $\mathcal{X}(\xi)$, considere as seguintes funções reais estendidas

$$f_{\mathcal{X}(\xi)}(x, \xi) := \sum_{t=1}^T c_t^{j^\top} x_t + i_{\mathcal{X}(\xi)}(x).$$

Dada a distribuição de probabilidades P_λ , seja $v(P_\lambda)$ o valor ótimo do seguinte

programa estocástico

$$\min_{x \in \mathcal{F}} f_{P_\lambda}(x), \quad \text{com} \quad f_{P_\lambda}(x) := \int_{\Xi} f_{\mathcal{X}_\xi}(x, \xi) dP_\lambda(\boldsymbol{\xi}). \quad (9.13)$$

Para todo $x \mathcal{F}$ -mensurável, pela linearidade da integral, tem-se que

$$f_{P_\lambda}(x) = \int_{\Xi} f_{\mathcal{X}_\xi}(x, \xi) dP_\lambda(\boldsymbol{\xi}) = \sum_{j=1}^{n_d} \int_{\Xi} f_{\mathcal{X}_\xi}(x, \xi) d\pi^j(\boldsymbol{\xi}) = \sum_{j=1}^{n_d} \lambda_j f_{\pi^j}(x).$$

Então,

$$\sum_{j=1}^{n_d} \lambda_j f_{\pi^j}(x) = f_{P_\lambda}(x) \geq v(P_\lambda) = \min_{x \in \mathcal{F}} \left\{ \sum_{j=1}^{n_d} \lambda_j f_{\pi^j}(x) \right\} \geq \sum_{j=1}^{n_d} \lambda_j v(\pi^j).$$

A relação acima é importante no sentido que, se as distribuições π^j para $j = 1, \dots, n_d$ forem selecionadas apropriadamente, o programa em multiestágios na distribuição P_λ pode ser aproximado por n_d programas em dois estágios com distribuição π^j .

Mais especificamente, seja $P_\lambda = P = \{p^1, \dots, p^N\}$ as probabilidades de N cenários. Utilizando novamente a formulação estática, o problema (9.1) pode ser escrito por

$$\min_{x \in \mathcal{F}} f_P^N(x), \quad \text{com} \quad f_P^N(x) := \sum_{i=1}^N p^i f_{\mathcal{X}_{\xi^i}}(x, \xi^i), \quad (9.14)$$

como apresentando na Seção 2.3. Pela definição de $P_\lambda = P$ em (9.11), cada distribuição π^j possui alguma coordenada nula quando $n_d > 1$. Definindo o subconjunto de cenários $\Xi^j := \{\xi \in \Xi : \pi^j(\xi) > 0\}$, seja \mathcal{G}^j a σ -álgebra associada, i.e., $\mathcal{G}^j := \sigma(\Xi^j)$. Para todo $j = 1, \dots, n_d$, é suposto que a distribuição π^j seja escolhidas de modo que \mathcal{G}^j seja uma *filtração grosseira* de \mathcal{F} , i.e. $\mathcal{G}_t^j \subset \mathcal{F}_t$ para $t = 1, \dots, T$. Deste modo, se x é \mathcal{F} -mensurável então x também é \mathcal{G}^j -mensurável, e o programa estocástico com distribuição π^j ,

$$\min_{x \in \mathcal{G}^j} f_{\pi^j}(x)$$

possui menos cenários do que o problema (9.14), e por isso é mais fácil de ser resolvido. Tal problema se torna mais simples se é considerada uma aproximação em dois estágios, como em (9.10). Esta estratégia é referenciada a seguir por *aproximação por multiproblemas em dois estágios*.

Com este desenvolvimento, ao invés de resolver uma única aproximação em dois estágios (de porte maior) do problema (9.1), propõe-se resolver n_d aproximações em dois estágios que determinam n_d valores $v(\pi^j)$, e n_d soluções aproximadas $x_1(\pi^j) \in \mathcal{X}_1$ do problema (9.1). Além disso, para obter um limite superior para o valor ótimo, basta avaliar as funções em dois estágios $f_{\pi^j}(\bar{x})$, para $j = 1, \dots, n_d$, em uma solução aproximada $\bar{x} \mathcal{F}$ -mensurável.

Supondo que a árvore de cenários representada na parte inferior da Figura 9.2 seja equiprovável, com $p_2^j = 1/3$ e $p_3^j = 1/9$ para $j = 1, \dots, 9$, a distribuição $P = \{1/9, \dots, 1/9\}$ pode ser representada por

$$P = P_\lambda = \lambda_1 \pi^1 + \lambda_2 \pi^2 + \lambda_3 \pi^3, \quad (9.15)$$

com, por exemplo²,

$$\begin{aligned} \pi^1 &= \{1/3, 0, 0, 1/3, 0, 0, 1/3, 0, 0\} \\ \pi^2 &= \{0, 1/3, 0, 0, 1/3, 0, 0, 1/3, 0\} \\ \pi^3 &= \{0, 0, 1/3, 0, 0, 1/3, 0, 0, 1/3\}, \end{aligned}$$

e $\lambda = \{1/3, 1/3, 1/3\}$. É importante ressaltar que existem outras escolhas de distribuições π^1 , π^2 e π^3 que:

- (i) satisfazem a relação (9.15) e;
- (ii) definem árvores que proporcionam $\mathcal{G}_t^j = \{\emptyset, \Xi_t^j\}$, para $t = 1, \dots, T$.

Ressalta-se que tão importante quanto a satisfação da condição (i), é a satisfação de (ii), que possibilita a aplicação de métodos de otimização estocástica em dois estágios e, sobretudo, permite que seja calculado um limite superior para o valor ótimo $v(P_\lambda)$, como explicado a seguir.

Uma vez que todos os n_d programas em dois estágios são resolvidos, a análise de robustez da modelagem do processo estocástico fica facilitada. Com efeito, para o exemplo em $T = 3$ estágios considerado, dada uma solução $\bar{x}(\xi_{[2]}) \in \mathcal{G}_2^j = \mathcal{F}_2$ de algum programa em dois estágios, uma solução \mathcal{F}_3 -mensurável pode ser obtida resolvendo os PLs

$$\bar{x}_3(\xi) \in \arg \min_{x_3 \geq 0} c_3^\top x_3 \quad \text{s.a.} \quad A_3 x_3 = b_3 - B_3 \bar{x}_2, \quad \text{com} \quad \xi \in \Xi.$$

Assim sendo, o limite superior

$$U_\lambda := \sum_{j=1}^{n_d} \lambda_j f_{\pi^j}(\bar{x}),$$

e o limite inferior

$$L_\lambda := \sum_{j=1}^{n_d} \lambda_j v(\pi^j)$$

do valor ótimo $v(P_\lambda)$ é dependente do vetor λ . Portanto, uma estimativa do valor ótimo de outro programa estocástico em mustiestágios com distribuição P_λ é obtido apenas substituindo o vetor λ por $\tilde{\lambda}$ nas definições de U_λ e L_λ , sem ter a necessidade

²A distribuição π^1 define a árvore composta pelos três cenários ABE, ACH, e ADN

de resolver nenhum problema de otimização adicional. Esta estratégia é interessante para analisar a variabilidade dos custos envolvidos no problema de otimização considerado, quando a distribuição de probabilidades é alterada.

A aproximação por multiproblemas em dois estágios é particularmente interessante quando o problema em multiestágios considerado possui como parâmetros aleatórios apenas $\xi_t := b_t$, para $t = 2, \dots, T$, com (c_t, A_t, B_t) fixos para todo $t \in \{1, \dots, T\}$. Esta característica permite que os métodos de feixes parcialmente inexatos sejam aplicados aos subprogramas em dois estágios, utilizando o Oráculo Parcialmente Inexato 1. Deste modo, a resolução de cada programa aproximado se torna mais rápida. É importante ressaltar que uma vez que o primeiro programa em dois estágios seja resolvido, os pontos $\{x_1^k\} \in \mathcal{X}_1$ do feixe do MFPI ou MNPI podem ser utilizados para obter um modelo de planos cortantes aproximados para a próxima função a ser minimizada. Para isto, basta trocar de subárvore de cenários e invocar os oráculos 4.2 ou 6.2 com $\zeta_{inex} = 1$ para estes pontos. Além disso, a solução do programa resolvido previamente pode ser utilizada como ponto inicial para o programa seguinte. Neste sentido, por ser um método proximal, o MFPI pode ser preferível ao MNPI.

A seguir é apresentada uma abordagem baseada na *decomposição dual* do problema (9.9), que leva em conta (de forma dual) todas as restrições de mensurabilidade do problema.

9.3 Decomposição Dual

As abordagens apresentadas nas duas seções anteriores para os programas estocásticos em multiestágios correspondem a dois casos extremos: a *decomposição aninhada* exige que a cada iteração k e estágio $t = 1, \dots, T$, as restrições de mensurabilidade $x_t^k \in \mathcal{F}_t$ sejam rigorosamente satisfeitas, enquanto a *aproximação por dois estágios* exige apenas que as restrições de mensurabilidade sejam satisfeitas para $t = 1$. Nesta seção é considerada uma abordagem não extrema, em que as restrições de mensurabilidade para $t = 1, \dots, T$ são satisfeitas assintoticamente.

Mais precisamente, ao invés de desconsiderar as restrições de mensurabilidade para $t = 2, \dots, T$, como apresentado na seção anterior, a decomposição dual, ou decomposição por cenários, para programas estocásticos é obtida dualizando as restrições de mensurabilidade do problema (9.9), para $t = 1, \dots, T$.

9.3.1 Considerações Iniciais

A menos que se tenha um número finito N de cenários, o problema (9.9) corresponde a um problema de otimização infinita sob o conjunto de funções \mathcal{F} -mensuráveis. As-

sim sendo, é suposto que o processo estocástico possua N cenários $\xi^i = (\xi_1^i, \dots, \xi_T^i)$, cada um com probabilidade p_i , para $i = 1, \dots, N$. Seja $\chi := (x(\xi^1), x(\xi^2), \dots, x(\xi^N))$ o vetor composto pelas variáveis de decisão associadas a todos os N cenários. Então, seguindo o desenvolvimento de [52, Capítulo 3], as restrições de mensurabilidade podem ser reescritas como um sistema linear da forma $G\chi = 0$. Neste sentido,

$$x(\xi^i) \text{ é } \mathcal{F} - \text{ mensurável para todo } i = 1, \dots, N \quad \Leftrightarrow \quad G\chi = 0,$$

onde a matriz G possui apenas dois elementos diferentes de zero em cada coluna, a saber, 1 e -1. Retornando ao Exemplo 2.1, afim de ajudar o entendimento, as restrições de mensurabilidades da árvore de cenários da Figura 2.1 são dadas por

$$\begin{aligned} x_1(\xi^1) - x_1(\xi^2) &= 0, \quad x_1(\xi^2) - x_1(\xi^3) = 0, \quad x_1(\xi^3) - x_1(\xi^4) = 0 \\ x_2(\xi^1) - x_2(\xi^2) &= 0 \quad \text{e} \quad x_2(\xi^3) - x_2(\xi^4) = 0. \end{aligned}$$

Assim sendo, a matriz G tem a forma

$$G = \left[\begin{array}{c|c|c|c} I & -I & -I & \\ & I & I & -I \\ & & -I & \\ & I & & -I \end{array} \right],$$

onde I é uma matriz identidade, de dimensão apropriada (1×1 para o Exemplo 2.1, porque $x_1, x_2 \in \mathcal{R}$). Cada bloco da matriz G está associado a um cenário. Deste modo, a matriz pode ser decomposta em N blocos:

$$G = [G^1, G^2, \dots, G^N].$$

Fixando um cenário $\xi^i := (c_t^i, B_t^i, b_t^i)$, seja o conjunto $\mathcal{X}_i := \mathcal{X}(\xi^i)$, com $\mathcal{X}(\xi^i)$ definido em (9.12). Ao longo desta seção é assumida a hipótese H1 e, adicionalmente, a hipótese seguinte:

H2' - o conjunto \mathcal{X}_i é não vazio e limitado para todo $i = 1, \dots, N$.

Seguindo o desenvolvimento apresentado na Seção 6 de [52, Capítulo 3], a seguir é realizada a dualização do problema (9.9), de modo que uma estrutura favorável à decomposição em subproblemas menores seja introduzida.

9.3.2 Relaxação das Restrições de Mensurabilidade

Dada a função indicadora $i_{\mathcal{X}_j}(\cdot)$ do conjunto \mathcal{X}_j , considere para $j = 1, \dots, N$ as seguintes funções reais estendidas

$$f_{\mathcal{X}_j}(x) := \sum_{t=1}^T c_t^{j\top} x_t + i_{\mathcal{X}_j}(x).$$

Então, o problema (9.9) pode ser alternativamente representado por

$$\min f_T^N(\chi) \quad \text{s.a} \quad G\chi = 0, \quad (9.16)$$

com $f_T^N(\chi) := \sum_{i=1}^N p_i f_{\mathcal{X}_i}(x(\xi^i))$.

O problema acima é chamado de *equivalente determinístico*, e como pode ser verificado, não pode ser decomposto em subproblemas menores, devido a restrição de mensurabilidade $G\chi = 0$. Esta característica é particularmente indesejável, pois para valores moderados de N , o equivalente determinístico pode atingir dimensões colossais, impossibilitando deste modo a sua resolução numérica direta (pelo menos em tempos computacionais aceitáveis).

Afim de obter uma estrutura separável para o problema (9.16), considere a função Lagrangiana

$$\begin{aligned} \mathcal{L}(\chi, u) &:= f_T^N(\chi) + u^\top G\chi \\ &= \sum_{i=1}^N p_i f_{\mathcal{X}_i}(x(\xi^i)) + \sum_{i=1}^N u^\top G^i x(\xi^i) \\ &= \sum_{i=1}^N [p_i f_{\mathcal{X}_i}(x(\xi^i)) + u^\top G^i x(\xi^i)]. \end{aligned}$$

Sob a hipótese H2', a dualidade em programação linear assegura a existência de um par primal-dual (χ^*, u^*) ótimo, tal que χ^* é uma solução do problema (9.16). Além do mais, não há brecha de dualidade:

$$\min_{\chi} \max_u \mathcal{L}(\chi, u) = \mathcal{L}(\chi^*, u^*) = \max_u D(u), \quad (9.17)$$

onde a função dual

$$D(u) := \min_{\chi} \mathcal{L}(\chi, u)$$

pode ser decomposta em subproblemas:

$$D(u) = \sum_{i=1}^N D^i(u),$$

com $D^i(u)$ o valor ótimo do problema de porte menor

$$\min_x \{p_i f_{\mathcal{X}_i}(x) + u^\top G^i x\}. \quad (9.18)$$

Note que para $p_i > 0$,

$$D^i(u) = -\sup_x \{-(G^{i\top} u)^\top x - p_i f_{\mathcal{X}_i}(x)\} = -p_i f_{\mathcal{X}_i}^*(-G^{i\top} u/p_i),$$

onde $f_{\mathcal{X}_i}^*(\cdot)$ é a função conjugada de $f_{\mathcal{X}_i}(\cdot)$. Segue da convexidade de $f_{\mathcal{X}_i}^*(\cdot)$ que $D^i(\cdot)$ é uma função côncava e, pelo Teorema 23.5 em [29],

$$G^i x_u(\xi^i) \in \partial D^i(u),$$

com $x_u(\xi^i)$ uma solução de (9.18). A existência de soluções é assegurada pela hipótese H2'. Então, pelo Teorema de Moreau-Rockafellar [1, Teorema 7.4], conclui-se que

$$\sum_{i=1}^N G^i x_u(\xi^i) \in \partial D(u).$$

Com este desenvolvimento, os métodos de feixes podem ser aplicados ao problema convexo, sem restrições, e separável:

$$\min_u -D(u). \quad (9.19)$$

A dimensão da variável dual é bastante importante, e pode dificultar o processo de otimização (ver comentários finais deste capítulo).

Apesar da solução dual ser importante (pois determina o “preço da decisão”), em muitas aplicações o principal interesse está na variável primal. Para obter uma solução primal, considere $\check{D}(\cdot)$ o modelo de planos cortantes de $D(\cdot)$. Seja \hat{J} o conjunto de índices ativos na solução ótima dual u^* , i.e.,

$$D(u^*) = D(u^j) + (G\chi^j)^\top (u^* - u^j), \quad \text{para todo } j \in \hat{J}.$$

A inclusão $0 \in \partial \check{D}(u^*)$ é verificada pela otimalidade de u^* . Então, existe um multiplicador simplicial ν ($\nu_j \geq 0$ tais que $\sum_{j \in J} \nu_j = 1$), tal que a igualdade seguinte é satisfeita:

$$\sum_{j \in J} \nu_j G\chi^j = 0.$$

Cada componente ν_j corresponde ao multiplicador de Lagrange associado ao corte

$D(u^j) + (G\chi^j)^\top(u^* - u^j)$ (ver Lema 3.1). Assim sendo, o ponto

$$\hat{\chi} := \sum_{j \in \hat{J}} \nu_j \chi^j$$

é viável para o problema (9.16). Portanto, pela dualidade forte $f_T^N(\chi^*) \geq f_T^N(\hat{\chi})$, i.e., $\hat{\chi}$ é uma solução do problema (9.16), [52, p. 194].

Para obter uma solução ε_f -ótima $\hat{\chi}$, a seguir é apresentado um oráculo inexato para a função $-D(\cdot)$, seguindo as linhas gerais do critério de colinearidade apresentado para os programas estocásticos em dois estágios.

9.3.3 Critério de Colinearidade

Esta seção busca definir um oráculo inexato que resolve o problema (9.18) para um subconjunto pequeno de todos os cenários $i = 1, \dots, N$, quando há pequenas variações na variável dual u . Afim de descrever formalmente esta concepção, seja o seguinte programa linear

$$D^i(u) = \min_{x \in \mathcal{X}_i} \left\{ \sum_{t=1}^T p_t c_t^{i\top} x_t + u^\top G^i x \right\}, \quad (9.20)$$

equivalente (em termos de valor ótimo e soluções) ao problema (9.18). Note que para dois pontos $u \neq \tilde{u}$ tais que $G^i u = G^i \tilde{u}$, os respectivos programas lineares $D^i(u)$ e $D^i(\tilde{u})$ são indistinguíveis. Outra observação importante a ser feita refere-se à identidade dos respectivos conjuntos soluções de $D^i(u)$ e $D^i(\tilde{u})$, quando os vetores $G^i u$ e $G^i \tilde{u}$ são colineares. Partindo destas observações, o seguinte oráculo considera quando dois vetores são “quase” colineares, ou “quase” indistinguíveis:

Oráculo Inexato 6 (ESTRATÉGIA DE COLINEARIDADE - MULTIESTÁGIO)

Passo 0 (inicialização). Sejam $\epsilon_{cos} \in (0, 1)$ um parâmetro de colinearidade, e $\epsilon > 0$ uma tolerância para a variação dual.

Dados um par primal-viável (χ^k, u^k) , e o atual iterado u^{k+1} , vá ao Passo 1.

Passo 1 (seleção de cenários). Se $k = 1$, faça $I^E = \{1, \dots, N\}$ e vá ao Passo 2.

Caso contrário, calcule para cada $i = 1, \dots, N$, os valores

$$\cos(\theta_i) := \frac{u^{k\top} [G^i G^{i\top}] u^{k+1}}{\|G^{i\top} u^k\| \|G^{i\top} u^{k+1}\|} \quad \text{e} \quad d_i := \|G^{i\top} (u^k - u^{k+1})\|.$$

Defina um subconjunto não vazio $I^E \subset \{1, \dots, N\}$, tal que

$$\{i : \cos(\theta_i) \leq 1 - \epsilon_{cos} \quad \text{e} \quad d_i \geq \epsilon\} \subset I^E.$$

Passo 2 (cálculos exatos). Para cada $i \in I^E$, determine uma solução $x(\xi^i)$ e o valor ótimo $D^i(u^{k+1})$ resolvendo (9.20).

Passo 3 (cálculos inexatos). Para cada $j \notin I^E$:

seja $\tilde{x}(\xi^j)$ igual ao j -ésimo vetor que compõem a variável χ^k .

Passo 4 (atualização). Defina o j -ésimo vetor componente de χ^{k+1} por

$$\chi_j^{k+1} := \begin{cases} x(\xi^j) & \text{se } j \in I^E \\ \tilde{x}(\xi^j) & \text{se } j \notin I^E. \end{cases}$$

□

O Oráculo Inexato 6 fornece as estimativas:

$$\begin{cases} D_u^{k+1} &= \sum_{i \in I^E} D^i(u^{k+1}) + \sum_{j \notin I^E} \left(p_j \sum_{t=1}^T c_t^{j\top} \tilde{x}_t(\xi^j) + u^{k+1\top} G^j \tilde{x}(\xi^j) \right) \\ g_u^{k+1} &= \sum_{i \in I^E} G^i x(\xi^i) + \sum_{j \notin I^E} G^j \tilde{x}(\xi^j). \end{cases}$$

O resultado seguinte é útil para o desenvolvimento subsequente.

Proposição 9.2 *Seja $j \in \{1, \dots, N\}$ um índice fixo, e seja $z \in \mathcal{X}_j$ um ponto viável. Considere o vetor $c_{[T]}^j = p_j(c_1^{j\top}, \dots, c_T^{j\top})^\top$. Então*

$$\epsilon_j := (c_{[T]}^j + G^{j\top} u)^\top z - D^j(u),$$

é não negativo, e a inclusão $G^j z \in \partial_{\epsilon_j} D^j(u)$ é satisfeita.

Prova. Dado um ponto u' , seja $x(\xi^j)$ a solução do subproblema (9.20) com u substituído por u' . Então, pela definição da função dual D^j e usando que $z \in \mathcal{X}_j$, segue que $\epsilon_j \geq 0$, e

$$\begin{aligned} D^j(u') &= (c_{[T]}^j + G^{j\top} u')^\top x(\xi^j) \\ &\leq (c_{[T]}^j + G^{j\top} u')^\top z \\ &= D^j(u) + (G^{j\top} u')^\top z - (G^{j\top} u)^\top z - [D^j(u) - (c_{[T]}^j + G^{j\top} u)^\top z] \\ &= D^j(u) + (G^j z)^\top (u' - u) + \epsilon_j. \end{aligned}$$

Como u' é um ponto arbitrário, então $G^j z \in \partial_{\epsilon_j} D^j(u)$. ■

A seguir será mostrado que o Oráculo Inexato 6 pode ser utilizado pelos Algoritmos 3.1, 4.1 e 6.1.

Proposição 9.3 *Dadas as hipóteses H1 e H2', suponha que o problema sem restrições (9.19) seja substituído pelo problema restrito*

$$\min_{\|u\| \leq M} -D(u),$$

para $M > 0$ alguma constante suficientemente grande. Então o Oráculo Inexato 6 satisfaz as relações (6.2) escritas com $(f_x, g_x) = -(D_u, g_u)$, para uma constante $0 < \varepsilon_f < \infty$, dependente dos dados de (9.19) e de M .

Prova. Dados um ponto u e um índice $j \in \{1, \dots, N\}$ fixo, sejam, respectivamente, $x(\xi^j)$ e $\tilde{x}(\xi^j)$ a solução ótima e aproximada (obtida pelo Oráculo Inexato 6) para o subproblema (9.20), definido para u^{k+1} . O vetor coordenada χ_j proporcionado pelo oráculo inexato pode assumir uma, e somente uma, das opções $x(\xi^j)$ ou $\tilde{x}(\xi^j)$. Então,

$$\begin{aligned} D^j(u) &= (c_{[T]}^j + G^{j\top} u)^\top x(\xi^j) \\ &\leq (c_{[T]}^j + G^{j\top} u)^\top \chi_j \\ &= (c_{[T]}^j + G^{j\top} u^{k+1})^\top \chi_j + (G^j \chi_j)^\top (u - u^{k+1}). \end{aligned}$$

Somando a relação acima para $j = 1, \dots, N$, segue que

$$-D(u) \geq \ell^{k+1}(u) := -D_u^{k+1} - g_u^{k+1\top}(u - u^{k+1}),$$

e portanto, $D(u) \leq D_u$. Desde que a norma de u é limitado por M e vale H2', então existe uma constante finita $\varepsilon_f > 0$ tal que $\sum_{j=1}^N \varepsilon_j \leq \varepsilon_f$, para ε_j definido na Proposição 9.2. Então,

$$\varepsilon_f \geq \sum_{j=1}^N \varepsilon_j = \sum_{j=1}^N (c_{[T]}^j + G^{j\top} u)^\top \chi_j - D^j(u) = D_u - D(u).$$

Portanto, para qualquer ponto arbitrário u , o oráculo inexato fornece

$$D_u \in [D(u), D(u) + \varepsilon_f].$$

Finalmente, a existência de uma constante finita Λ tal que $\|g_u^{k+1}\| \leq \Lambda$ é assegurada por H1 e H2'. Logo, o oráculo inexato satisfaz as relações dadas em (6.2). ■

Se M é suficientemente grande, a hipótese de que a variável dual u é limitada por M não compromete a relação de dualidade do problema (9.16), [52, p. 192].

O seguinte resultado é consequência do Teorema 5.2 demonstrado em [8].

Proposição 9.4 *Seja o modelo de planos cortantes inexato*

$$\check{D}_k(\cdot) := \max_{j \in J^k} \{D_u^j + g_u^j(\cdot - u^j)\},$$

construído a partir dos valores aproximados retornados pelo Oráculo Inexato 6. Suponha que o ponto \bar{u} obtido pelo MFI com $\delta_{\text{ToI}} = 0$ seja uma ε_f -solução para o

problema (9.19), e que ν seja o multiplicador (simplicial) ótimo associado. Seja $\hat{J} \subset J^k$ o conjunto dos índices ativos:

$$\check{D}_k(\bar{u}) = D_u^j + g_u^j(\bar{u} - u^j), \quad \text{para todo } j \in \hat{J}.$$

Seja também o vetor χ^j retornado pelo Oráculo Inexato 6, na iteração j . Então o ponto

$$\hat{\chi} := \sum_{j \in \hat{J}} \nu_j \chi^j$$

é uma ε_f -solução para o problema (9.16).

Finaliza-se este capítulo mencionando que a dimensão da variável dual u do problema (9.16) é dependente da árvore de cenários considerada. Mais especificamente, a dimensão $\dim(u)$ é função do número de sucessores $S(\xi_{[t]}^i)$ de cada trajetória $\xi_{[t]}^i$, com $t = 1, \dots, T$ e $i = 1, \dots, N$. Quando possível, para evitar um crescimento exponencial de $\dim(u)$, torna-se interessante utilizar árvores que possuam $S(\xi_{[t]}^i) = 1$, para determinados i e t . Este tipo de árvore reduz consideravelmente o número de restrições de mensurabilidade e, deste modo, a dimensão de u como ilustrado na Figura 9.3.

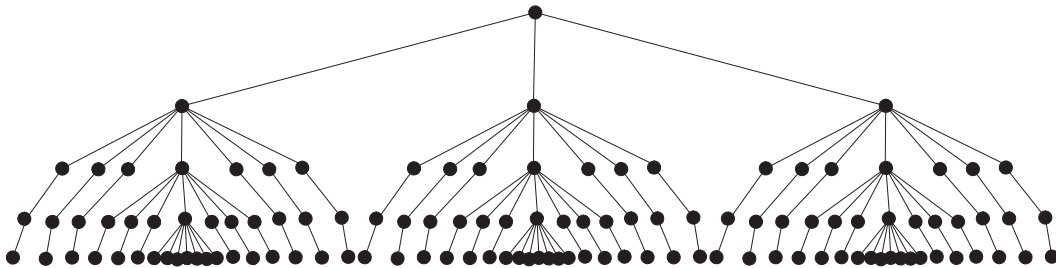


Figura 9.3: Árvore que permite um número menor de restrições de mensurabilidade.

A seguir são apresentados alguns resultados numéricos que validam o desenvolvimento teórico formulado ao longo deste trabalho de tese.

Capítulo 10

Resultados Numéricos

Neste capítulo são apresentados alguns resultados numéricos obtidos com vários programas lineares estocásticos. A Seção 10.1 é dedicada aos programas em dois estágios. Os programas estocásticos em multiestágios são considerados na Seção 10.2

10.1 Programas Lineares em Dois Estágios

É considerada nesta seção uma família de 10 problemas, com a dimensão da variável aleatória ξ variando entre 2 até 200. Cada família possui 11 casos diferentes, correspondentes à variação do número

$$N \in \{100, 200, 300, 500, 800, 1000, 1200, 1500, 1800, 2000, 2500\}$$

de cenários, para os quais foram considerados 9 técnicas diferentes para resolver o problema de otimização resultante. As comparações são realizadas tanto em termos de acurácia e tempo de CPU, quanto em termos de qualidade das soluções primais e duais, quando possível. Todos os testes foram realizados usando Matlab, versão 7.8.0 (R2009a), em um computador AMD Athlon II X2 240 com 2800 MHz, 2 GB RAM, e sistema operacional Ubuntu, utilizando as rotinas de programação linear e quadrática do pacote de otimização do MOSEK¹, para Matlab.

10.1.1 Principais Características dos Casos

Métodos de Otimização

Adicionalmente ao método de feixes proximal inexato, e ao método de nível proximal inexato, ambos aplicados com os três oráculos inexatos (colinearidade, ROCSeq,

¹<http://www.mosek.com>

SSC) apresentados nos Capítulos 7 e 8, com efeito de comparação também é considerada uma abordagem oposta. Mais precisamente, foram selecionadas árvore reduzidas com n_{LP} cenários usando a pseudo-norma d_λ , e o problema de otimização resultante foi resolvido aplicando o método de planos cortantes, o método de feixes proximal, e o método de nível proximal, todos utilizando oráculos exatos. A seguir são dados os mnemônicos das técnicas empregadas:

- d_λ -MPCE- (ou simplesmente d_λ -MPC, por conveniência notacional) método de planos cortantes exato aplicado à uma árvore reduzida e fixa, com n_{LP} cenários;
- d_λ -MFE- método de feixes proximal exato aplicado à uma árvore reduzida e fixa, com n_{LP} cenários;
- d_λ -MNE- método de nível proximal exato aplicado à uma árvore reduzida e fixa, com n_{LP} cenários;
- MFI-cos- método de feixes proximal inexato com um dos Oráculos Inexatos 1 ou 2, baseados na estratégia de colinearidade;
- MNI-cos- método de nível proximal inexato com um dos Oráculos Inexatos 1 ou 2;
- MFI- d_1 - método de feixes proximal inexato com o Oráculo Inexato 3, usando $\lambda = 1$ para a distância d_λ definida em (8.12), como em [35];
- MNI- d_1 - método de nível proximal inexato com o Oráculo Inexato 3, usando $\lambda = 1$;
- MFI- d_λ - método de feixes proximal inexato com o Oráculo Inexato 4, e $\lambda \in (0, 1)$;
- MNI- d_λ - método de feixes proximal inexato com o Oráculo Inexato 4, e $\lambda \in (0, 1)$.

Os três primeiros métodos são denotados “estáticos”, ao contrário dos demais seis métodos que alteram o conjunto de cenários considerado, ao longo das iterações. Em particular, mesmo com a tolerância $\epsilon_{cos} \in (0, 1)$ fixa para o MFI-cos (ou MNI-cos), o conjunto I^E pode mudar ao longo das iterações, porque os valores $\cos(\theta_{ij})$ dependem de cada iterado z^k .

Teste de Parada e Parâmetros

Como proposto em [23] para o método de feixes proximal, todos os algoritmos, exato quanto os inexatos, param quando $\|g^k\| \leq 5 \cdot 10^{-4} \sqrt{n}$ (com n a dimensão de $g^k \in \mathcal{R}^n$) e alguma das seguintes condições $\delta_k \leq 10^{-5}(1 + |f_x^k|)$ com $-\alpha_k \leq \delta_k$, ou $\|g^k\| + \alpha_k \leq 10^{-5}(1 + |f_x^k|)$ é verificada. Para escolha dos passos sérios foi fixado $\kappa = 10^{-1}$, e o passo proximal inicial foi definido como $t_1 = \min\{\|g^k\|^{-1}, \tau_1\}$, para $\tau_1 = 10$. Para os métodos de nível, κ corresponde ao parâmetro de nível, e é dado por $\kappa = 1/2$. Tanto o algoritmo do método de planos cortantes, quanto os algoritmos dos métodos de nível param quando a brecha de otimalidade Δ_k é menor do que a tolerância $\delta_{Tot} = 10^{-4}$.

Medida da Qualidade

Para comparar a qualidade das ε -soluções obtidas pelas várias técnicas é utilizado o erro relativo:

$$e\% := 100 \frac{|f^N(x_N^*) - f^N(x_N^\varepsilon)|}{1 + |f^N(x_N^*)|},$$

com $x_N^* := \arg \min_{z \in \mathcal{X}} f^N(z)$ para $f^N(z) := c^\top z + \sum_{i=1}^N p_i Q(z, \xi_i)$, e x_N^ε uma solução aproximada encontrada pelo técnica analisada. Para encontrar um ponto x_N^* , é utilizado o *equivalente determinístico* - EquiDet. Se a complexidade do problema faz com que o EquiDet não seja resolvível, o problema é decomposto e resolvido pelo método de feixes proximal exato, denotado MFE. Analogamente à métrica $e\%$, CPU% mede o decréscimo relativo do tempo computacional, quando comparado com um tempo de referência. Por este motivo, não são considerados os valores absolutos, pois valores negativos (positivos) para CPU% são entendidos como acarretando redução (aumento) do tempo de CPU.

Métricas e Ponto Inicial

O ponto inicial z^1 é a solução \bar{x} do *problema de valor esperado*, [11, Seção 4.2],

$$EV := \min_{z \in \mathcal{X}} \bar{f}^N(z), \quad \text{com} \quad \bar{f}^N(z) := c^\top z + Q(z, \mathbb{E}[\xi]).$$

Para a métrica d_λ dada em (8.12) é utilizada a norma Euclidiana, e o valor para λ é determinado como explicado a seguir.

Desde que $Q(\cdot, \xi)$ é uma função convexa em $\xi = (h, T)$ (Proposição (2.4)), tem-se que $EV \leq f^N(\bar{x})$. Se $EV = f^N(\bar{x})$, então o parâmetro aleatório ξ não proporciona nenhuma informação adicional acerca do problema de otimização, e o processo de seleção de cenários pode ser considerado tomando $\lambda = 0$ na função d_λ dada em (8.12). Com esta observação, pode-se considerar a relação entre EV e $f^N(\bar{x})$ para definir o parâmetro λ de d_λ , tomando

$$\lambda := 1 - \frac{EV}{f^N(\bar{x})}.$$

Para as técnicas MFI- d_λ e MNI- d_λ o parâmetro λ é atualizado ao longo das iterações, substituindo $f^N(\bar{x})$ por f_z^k na relação acima. Finalmente, a salvaguarda $\lambda \in [0.05, 0.95]$ assegura que d_λ seja uma norma, diferente da norma Euclidiana empregada em d_1 .

Problemas Teste

Os problemas considerados são chamados de SH10, InvestmentRisk, SH10Risk, SH31, SH25, AirCraft, ProductMix, ProjectSchedule, ExpTerm3, ExpTerm23. A

seguir as suas descrições e resultados obtidos.

10.1.2 Resultados para SH10

O primeiro teste numérico corresponde ao problema artificial apresentado em [66], com $\xi = h$ uma variável aleatória normalmente distribuída com suporte $\Xi \subset \mathcal{R}^{10}$. Este problema é chamado de SH10, e a sua formulação é dada em [66]; em [67] são apontadas duas correções tipográficas para o problema SH10. A versão corrigida é apresentada na Seção C. São apresentados na Tabela 10.1 alguns parâmetros para as técnicas empregadas, onde n_{LP}^k representa a cardinalidade máxima permitida para o conjunto I^E , quando aplicados os Oráculos Inexatos 3 e 4.

Tabela 10.1: Parâmetros

Métodos	n_{LP}^1	n_{LP}^k , com $k > 1$	ϵ_{cos}
MFE	-	-	-
d_λ -MPC	$\min\{0.5N, 350\}$	-	-
d_λ -MFE	$\min\{0.5N, 350\}$	-	-
MFI- d_1	$\min\{0.4N, 300\}$	$\min\{1.1n_{LP}^{k-1}, 0.6N\}$	-
MFI- d_λ	$\min\{0.4N, 300\}$	$\min\{0.4N, 300\}$	-
MFI-cos	-	-	$2 \cdot 10^{-3}$

Os valores ótimos e os tempos de CPU (segundos) são dados na Tabela 10.2. As duas colunas mais à direita correspondem aos números de variáveis e restrições do equivalente determinístico de SH10.

Tabela 10.2: Valor ótimo e tempo de CPU para SH10.

N	EquiDet		MFE		tamanho	
	$f^N(x_N^*)$	CPU	$f^N(x_N^*)$	CPU	Nvar	Nrest
100	15.064	1.443	15.064	7.02	1510	1005
200	15.216	5.960	15.216	16.00	3010	2005
300	15.129	18.868	15.129	27.82	4510	3005
500	15.140	85.106	15.140	39.46	7510	5005
800	15.155	357.551	15.155	68.85	12010	8005
1000	-	-	15.129	73.30	15010	10005
1200	-	-	15.130	88.34	18010	12005
1500	-	-	15.164	129.85	22510	15005
1800	-	-	15.160	167.52	27010	18005
2000	-	-	15.159	158.37	30010	20005
2500	-	-	15.149	210.93	37510	25005

São apresentados nas Tabelas 10.3 e 10.4 os resultados das 9 técnicas aplicadas aos 11 casos, correspondentes à variação de N . Observa-se uma boa qualidade das ε -soluções, com uma redução significativa do tempo de CPU. Como esperado, quando o número de cenários N aumenta, as variantes MFI- d_λ e MNI- d_λ se tornam menos rápidas, devido à complexidade do processo de seleção de cenários. Este não é o caso para as técnicas MFI-cos e MNI-cos, as quais se mantêm rápidas mesmo quando N grande, e são as mais acuradas dentre as 9 técnicas estudadas para o SH10.

Aplicando a técnica SAA apresentada em [1, Seção 5.1.2] em 20 casos independentes, cada um com $N = 200$ cenários, foi possível calcular um limite inferior

Tabela 10.3: Qualidade da solução - SH10.

N	d_λ -MPC	d_λ -MFE	d_λ -MNE	MFI- d_1	MNI- d_1	MFI- d_λ	MNI- d_λ	MFI-cos	MNI-cos
100	0.05	0.05	0.05	0.01	0.16	0.06	0.16	0.00	0.00
200	0.01	0.01	0.01	0.07	0.03	0.07	0.03	0.00	0.00
300	0.01	0.01	0.01	0.01	0.10	0.01	0.10	0.00	0.00
500	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.00	0.00
800	0.01	0.01	0.01	0.00	0.02	0.00	0.02	0.00	0.00
1000	0.00	0.00	0.00	0.01	0.04	0.01	0.04	0.00	0.00
1200	0.01	0.01	0.01	0.03	0.03	0.03	0.09	0.00	0.00
1500	0.03	0.02	0.03	0.01	0.08	0.01	0.08	0.00	0.00
1800	0.06	0.05	0.05	0.00	0.09	0.00	0.09	0.00	0.00
2000	0.03	0.03	0.02	0.01	0.01	0.01	0.06	0.00	0.00
2500	0.02	0.01	0.01	0.01	0.06	0.01	0.06	0.00	0.00
min.	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00
max.	0.06	0.05	0.05	0.07	0.16	0.07	0.16	0.00	0.00
med.	0.02	0.02	0.02	0.02	0.06	0.02	0.07	0.00	0.00

Tabela 10.4: Redução de tempo de CPU - SH10.

N	d_λ -MPC	d_λ -MFE	d_λ -MNE	MFI- d_1	MNI- d_1	MFI- d_λ	MNI- d_λ	MFI-cos	MNI-cos
100	-18.83	-24.52	-24.27	-60.84	-49.83	-77.17	-69.91	-44.18	-36.03
200	-25.86	-32.93	-39.89	-66.22	-70.45	-80.12	-80.04	-50.27	-45.86
300	-34.97	-44.37	-35.62	-74.84	-74.99	-78.42	-77.34	-49.14	-57.79
500	-10.91	-46.01	-45.85	-71.37	-65.27	-68.66	-64.28	-36.40	-59.63
800	-22.87	-38.57	-38.56	-49.28	-65.60	-59.84	-51.47	-71.41	-77.97
1000	-11.57	-25.42	-33.35	-62.28	-60.19	-48.06	-30.44	-68.32	-67.94
1200	-24.90	-36.63	-33.71	-59.96	-45.41	-40.41	4.46	-70.09	-66.57
1500	-40.07	-33.99	-42.57	-61.88	-59.67	-42.98	-43.63	-69.77	-73.08
1800	-43.57	-42.41	-45.05	-58.44	-46.77	-39.61	-9.71	-73.63	-74.25
2000	-27.53	-30.57	-30.10	-45.67	-31.14	-21.79	-20.09	-72.19	-65.12
2500	-25.48	-30.02	-23.95	-42.15	-29.30	-9.22	36.61	-74.95	-74.43
min.	-43.57	-46.01	-45.85	-74.84	-74.99	-80.12	-80.04	-74.95	-77.97
max.	-10.91	-24.52	-23.95	-42.15	-29.30	-9.22	36.61	-36.40	-36.03
med.	-26.05	-35.04	-35.72	-59.36	-54.42	-51.48	-36.89	-61.85	-63.52

$L = 15.12496$ (com um intervalo de confiança de 95%) para o problema SH10. Para calcular um limite superior foi utilizado o ponto x_{2500}^ϵ , que é uma solução aproximada obtida pelos métodos inexatos para o caso com $N = 2500$ cenários:

$$U := f_{8000}(x_{2500}^\epsilon) + 1.96 \frac{\sigma_x}{\sqrt{8000}}, \quad \text{com} \quad \sigma_x^2 := \frac{1}{7999} \sum_{i=1}^{8000} (Q(x_{2500}^\epsilon, \xi^i) - \mathbb{E}[Q(x_{2500}^\epsilon, \xi)])^2.$$

São apresentados na Tabela 10.5 os limites obtidos usando x_{2500}^* (uma solução obtida com MFE para o caso com $N = 2500$ cenários) e x_{2500}^ϵ para as várias técnicas. Além disso, são dados os limites obtidos usando as três soluções diferentes dadas em [67, Tabela 3]. Observa-se uma boa aderência para todos os métodos empregados, com uma leve vantagem dos métodos inexatos baseados no critério de colinearidade sobre os métodos estáticos.

Os limites superiores encontrados com as soluções aproximadas são bastante próximos daqueles obtidos com x_{2500}^* . Um comportamento similar é observado na Figura 10.1, para as variáveis primais. Mais precisamente, são mostradas na Figura 10.1 as 10 componentes da solução aproximada de primeiro estágio, com um intervalo de confiança dado em [66].

Para algumas aplicações a variável dual de segundo estágio u é bastante importante, devido a sua interpretação econômica. São apresentados na Tabela 10.6 os

Tabela 10.5: Limites para o valor ótimo - SH10.

Método	U	$U - L$	$100(U - L)/U$	$f_{8000}(x^\epsilon)$	$\sigma_x/\sqrt{8000}$
MFE	15.18253	0.05757	0.37918	15.16754	0.00911
d_λ -MPC	15.18569	0.06073	0.39993	15.17018	0.00943
d_λ -MFE	15.18512	0.06016	0.39618	15.16965	0.00940
d_λ -MNE	15.18523	0.06027	0.39691	15.16976	0.00941
MFI- d_1	15.18394	0.05898	0.38846	15.16862	0.00932
MNI- d_1	15.19150	0.06654	0.43804	15.17734	0.00861
MFI- d_λ	15.18394	0.05898	0.38846	15.16862	0.00932
MNI- d_λ	15.19150	0.06654	0.43804	15.17734	0.00861
MFI-cos	15.18254	0.05758	0.37927	15.16755	0.00911
MNI-cos	15.18248	0.05752	0.37886	15.16755	0.00907
[67] ¹	15.18500	0.06004	0.39537	15.16996	0.00914
[67] ²	15.18531	0.06035	0.39740	15.17039	0.00907
[67] ³	15.18495	0.05999	0.39505	15.17008	0.00904

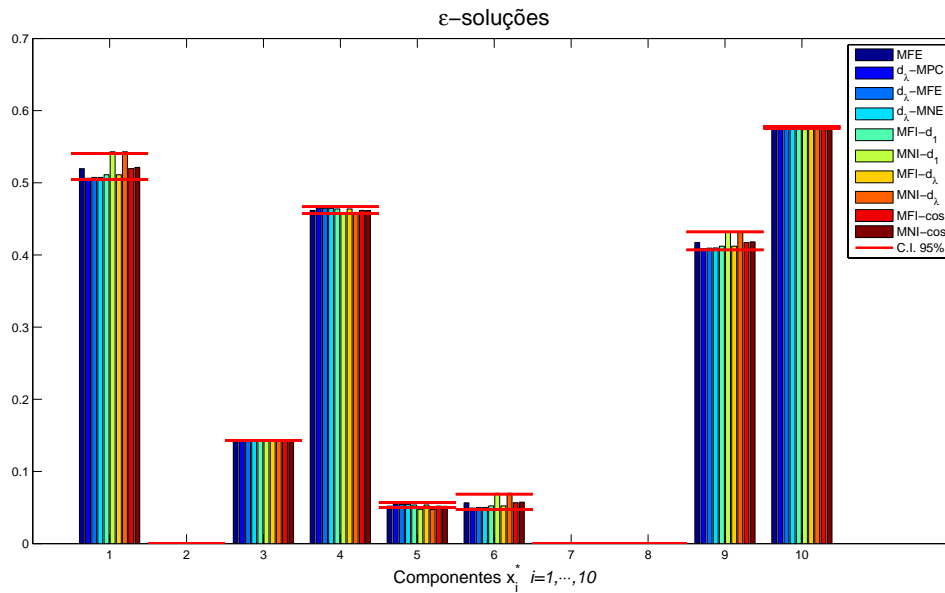


Figura 10.1: Intervalo de confiança para as variáveis de primeiro estágio - SH10.

valores esperados de cada componente da variável $u = u(x_{2500}^\epsilon)$.

Tabela 10.6: Variáveis duais de segundo estágio - SH10.

Method	$\mathbb{E}[u_1]$	$\mathbb{E}[u_2]$	$\mathbb{E}[u_3]$	$\mathbb{E}[u_4]$	$\mathbb{E}[u_5]$	$\mathbb{E}[u_6]$	$\mathbb{E}[u_7]$	$\mathbb{E}[u_8]$	$\mathbb{E}[u_9]$	$\mathbb{E}[u_{10}]$
MFE	-0.1573	-0.0494	-0.1955	0	-0.1897	0.0138	-0.6330	1.1349	0.6133	0.1704
d_λ -MPC	-0.2024	-0.0877	-0.1955	0	-0.0979	-0.0138	-0.6626	1.2162	0.6435	0.2193
d_λ -MFE	-0.1981	-0.0833	-0.1955	0	-0.1073	-0.0109	-0.6595	1.2074	0.6447	0.2146
d_λ -MNE	-0.1982	-0.0849	-0.1955	0	-0.1064	-0.0116	-0.6602	1.2098	0.6357	0.2147
MFI- d_1	-0.1857	-0.0748	-0.1955	0	-0.1315	-0.0040	-0.6521	1.1879	0.6245	0.2011
MNI- d_1	-0.0769	0.0165	-0.1955	0	-0.3507	0.0625	-0.5809	0.9930	0.5459	0.0833
MFI- d_λ	-0.1857	-0.0748	-0.1955	0	-0.1315	-0.0040	-0.6521	1.1879	0.6245	0.2011
MNI- d_λ	-0.0769	0.0165	-0.1955	0	-0.3507	0.0625	-0.5809	0.9930	0.5459	0.0833
MFI-cos	-0.1571	-0.0494	-0.1955	0	-0.1900	0.0139	-0.6329	1.1348	0.6121	0.1702
MNI-cos	-0.1527	-0.0446	-0.1955	0	-0.1992	0.0170	-0.6296	1.1253	0.6158	0.1655
[67] ¹	-0.1672	-0.0475	-0.1955	0	-0.1736	0.0117	-0.6352	1.1371	0.6826	0.1811
[67] ²	-0.1496	-0.0455	-0.1955	0	-0.2040	0.0176	-0.6290	1.1249	0.5921	0.1620
[67] ³	-0.1479	-0.0375	-0.1955	0	-0.2099	0.0211	-0.6252	1.1119	0.6311	0.1603

Como apresentado na Tabela 10.6, todas as 9 técnicas empregadas fornecem variáveis duais bastante semelhantes. Verifica-se que o valor $\mathbb{E}[u_6]$ tem sinal oposto para as técnicas que utilizam seleção de cenários. No entanto, esta variação é pequena, visto que $\mathbb{E}[u_6]$ está próximo de zero para todas as técnicas.

10.1.3 Resultados para os Problemas com Medida de Risco

A consideração da medida de risco *Conditional Value-at-Risk* - $CV@R$ - dada em [27] faz com que a matriz de recurso W seja aleatória, quando o vetor de custos q do segundo estágio é aleatório. Deste modo, um teste interessante para analisar o comportamento dos oráculos inexatos consiste em utilizar programas estocásticos em dois estágios, nos quais esta medida de risco seja empregada. Como, para esta situação, a estratégia de colinearidade calcula um conjunto viável médio, espera-se que a performance do Oráculo Inexato 2 não seja tão boa.

A seguir é considerado o problema *InvestmentRisk* dado em [68], com 200 variáveis aleatórias normalmente distribuídas ($\xi = (q, h)$, com $h, q \in \mathcal{R}^{100}$). Para se adequar ao tipo de abordagem deste trabalho, a medida de risco deve ser aplicada somente no segundo estágio. Por este motivo, a medida de risco apresentada em [68] foi substituída pela medida $CV@R$ dada em [27].

A mesma medida de risco foi empregada no problema SH10, resultando no problema denominado SH10Risk. Como o vetor de custos q de segundo estágio é determinístico em SH10, a matriz de recurso W resultante para o problema SH10Risk é, deste modo, determinística. Exceto por $\epsilon_{cos} = 10^{-3}$, os parâmetros dos métodos coincidem com aqueles dados na Tabela 10.1.

São apresentados nas Tabelas 10.7, 10.8, 10.9 e 10.10 os resultados obtidos para os dois problemas, com todas as 9 técnicas. Em ambos os casos pode-se observar uma boa performance em tempo de CPU para MFI-cos e MN-cos, com uma leve redução na qualidade da solução para o problema *InvestmentRisk*.

Tabela 10.7: Qualidade da solução - InvestmentRisk.

N	d_λ -MPC	d_λ -MFE	d_λ -MNE	MFI- d_1	MNI- d_1	MFI- d_λ	MNI- d_λ	MFI-cos	MNI-cos
100	0.65	0.67	0.67	0.65	0.64	1.88	0.65	1.89	5.12
200	1.01	1.01	1.01	0.10	0.11	0.43	0.56	1.46	1.34
300	0.47	0.47	0.57	0.03	0.03	0.28	0.16	1.78	0.19
500	0.03	0.04	0.03	0.02	0.02	0.39	0.04	1.80	0.71
800	0.14	0.14	0.12	0.05	0.09	0.03	0.07	2.18	2.19
1000	0.23	0.23	0.21	0.10	0.10	0.18	0.12	1.11	2.24
1200	0.06	0.06	0.06	0.36	0.33	0.08	0.18	2.25	1.61
1500	0.01	0.01	0.01	0.05	0.16	0.08	0.09	1.60	2.20
1800	0.07	0.06	0.07	0.14	0.57	0.14	0.37	1.73	2.27
2000	0.06	0.07	0.06	0.02	0.07	0.54	0.82	1.81	1.83
2500	0.06	0.06	0.07	0.14	0.25	0.08	0.14	1.62	1.50
min.	0.01	0.01	0.01	0.02	0.02	0.03	0.04	1.11	0.19
max.	1.01	1.01	1.01	0.65	0.64	1.88	0.82	2.25	5.12
med.	0.26	0.26	0.26	0.15	0.22	0.37	0.29	1.75	1.93

Tabela 10.8: Redução de tempo de CPU - InvestmentRisk.

N	d_λ -MPC	d_λ -MFE	d_λ -MNE	MFI- d_1	MNI- d_1	MFI- d_λ	MNI- d_λ	MFI-cos	MNI-cos
100	-49.46	-52.84	-51.47	-49.19	-56.92	-58.32	-61.32	-62.08	-76.78
200	-54.13	-63.91	-74.07	-60.79	-67.66	-74.89	-80.15	-69.09	-78.65
300	-46.76	-57.51	-68.06	-50.52	-62.35	-59.64	-72.00	-65.30	-80.10
500	-55.06	-55.37	-64.10	-44.32	-65.43	-65.56	-72.12	-69.74	-82.72
800	-54.37	-63.60	-71.49	-57.27	-69.31	-62.82	-75.71	-78.34	-84.49
1000	-52.61	-58.61	-66.90	-37.65	-61.59	-60.23	-66.84	-78.12	-78.17
1200	-64.56	-61.06	-70.20	-55.96	-72.98	-62.29	-72.70	-75.60	-77.36
1500	-64.47	-70.17	-75.07	-58.89	-74.42	-69.15	-74.32	-72.51	-81.84
1800	-68.30	-74.68	-79.34	-63.47	-72.79	-66.66	-69.48	-77.65	-78.86
2000	-69.35	-71.62	-80.24	-63.90	-73.81	-66.13	-70.09	-80.88	-82.42
2500	-76.96	-78.37	-79.22	-65.53	-75.43	-66.98	-70.27	-85.36	-85.03
min.	-76.96	-78.37	-80.24	-65.53	-75.43	-74.89	-80.15	-85.36	-85.03
max.	-46.76	-52.84	-51.47	-37.65	-56.92	-58.32	-61.32	-62.08	-76.78
med.	-59.64	-64.34	-70.92	-55.23	-68.43	-64.79	-71.36	-74.06	-80.58

Tabela 10.9: Qualidade da solução - SH10Risk.

N	d_λ -MPC	d_λ -MFE	d_λ -MNE	MFI- d_1	MNI- d_1	MFI- d_λ	MNI- d_λ	MFI-cos	MNI-cos
100	0.30	0.30	0.30	0.03	0.05	0.71	0.67	0.00	0.00
200	0.28	0.28	0.29	0.28	0.48	0.29	0.29	0.02	0.10
300	0.06	0.06	0.06	0.06	0.05	0.08	0.08	0.00	0.00
500	0.01	0.01	0.01	0.10	0.11	0.09	0.06	0.07	0.10
800	0.15	0.16	0.15	0.03	0.16	0.14	0.05	0.00	0.00
1000	0.03	0.03	0.03	0.11	0.26	0.47	0.23	0.01	0.00
1200	0.03	0.04	0.03	0.04	0.25	0.03	0.06	0.00	0.00
1500	0.17	0.17	0.18	0.16	0.19	0.18	0.46	0.00	0.00
1800	0.10	0.09	0.09	0.11	0.26	0.49	0.05	0.00	0.00
2000	0.32	0.32	0.32	0.40	0.41	0.44	0.09	0.00	0.00
2500	0.26	0.26	0.26	0.31	0.37	0.29	0.56	0.00	0.00
min.	0.01	0.01	0.01	0.03	0.05	0.03	0.05	0.00	0.00
max.	0.32	0.32	0.32	0.40	0.48	0.71	0.67	0.07	0.10
med.	0.16	0.16	0.16	0.15	0.24	0.29	0.24	0.01	0.02

Tabela 10.10: Redução de tempo de CPU - SH10Risk.

N	d_λ -MPC	d_λ -MFE	d_λ -MNE	MFI- d_1	MNI- d_1	MFI- d_λ	MNI- d_λ	MFI-cos	MNI-cos
100	-43.63	-43.45	-35.39	-37.14	-44.99	-63.82	-51.56	-54.64	-68.83
200	-50.15	-51.28	-61.69	-37.44	-75.89	-64.75	-66.29	-72.54	-88.66
300	-48.81	-43.51	-42.69	-34.01	-38.52	-61.92	-58.10	-66.75	-79.05
500	-44.60	-41.49	-61.81	-44.90	-64.49	-55.73	-62.51	-80.86	-88.87
800	-41.67	-40.34	-39.70	-27.94	-36.88	-43.09	-55.10	-73.70	-83.55
1000	-56.29	-50.17	-57.49	-29.73	-54.81	-53.87	-57.80	-81.77	-84.25
1200	-62.60	-60.94	-60.44	-55.98	-77.59	-48.69	-57.03	-79.83	-86.64
1500	-77.17	-74.28	-76.99	-61.01	-77.67	-66.76	-68.70	-84.02	-90.22
1800	-66.74	-64.48	-70.56	-49.23	-62.95	-46.01	-53.21	-81.25	-84.90
2000	-71.71	-68.74	-72.63	-48.81	-66.61	-52.00	-55.05	-83.63	-86.89
2500	-70.28	-63.97	-70.37	-52.95	-65.40	-42.00	-43.60	-83.24	-85.33
min.	-77.17	-74.28	-76.99	-61.01	-77.67	-66.76	-68.70	-84.02	-90.22
max.	-41.67	-40.34	-35.39	-27.94	-36.88	-42.00	-43.60	-54.64	-68.83
med.	-57.60	-54.79	-59.07	-43.56	-60.53	-54.42	-57.18	-76.57	-84.29

10.1.4 Avaliação do Desempenho das Técnicas

Para avaliar a performance das técnicas sobre um conjunto de testes é empregado o método gráfico introduzido em [26], denominado *perfil da performance*²-PP. O PP é interessante para apresentar e comparar um conjunto de métodos de otimização diferentes, aplicados a um conjunto de problemas distintos. Uma explicação sucinta acerca do PP é apresentada no Apêndice B.

Para comparar a acurácia e a redução de tempo de CPU das 9 técnicas empregadas, foi utilizada a rotina de MATLAB dada em [69, Seção 22.4]. Como os dados de entrada devem ser positivos, foi adicionado o valor 1 aos erros percentuais $e\%$, e 100 à redução de CPU. Desta maneira, valores mais altos no PP sempre indicam um desempenho melhor da técnica.

Em adição aos problemas SH10, SH10Risk, e InvestmentRisk, foram considerados os seguintes problemas:

- *SH31* - baseado em *SH10*, multiplicando por 10 o desvio padrão, e com cada elemento diferente de zero na matriz de tecnologia T normalmente distribuído com média $T(i, j)$ e variância $(T(i, j)/10)^2$, resultando em 31 variáveis aleatórias;
- *SH25* - baseado em *SH10*. Esta versão considera o vetor de custos q de segundo estágio independente e normalmente distribuído com média q e variância $q/2$. Há, portanto, 25 variáveis aleatórias;
- *AirCraft* - o problema de alocação de aeronaves dado em [70, pp. 544-546], com as seguintes modificações: cada uma das cinco componentes de h (demanda de passageiros) é independente e normalmente distribuída com média e variância calculadas usando os dados apresentados em [70].
- *ProductMix* - o problema de produção dado em [70, pp. 554-555]. Este tem 10 variáveis aleatórias, com T uniformemente e $h \in \mathcal{R}^2$ normalmente distribuídos;
- *ProjectSchedule* - um problema de agendamento de projetos dado em [70, pp. 549-552]. O tempo para completar cada um dos 25 projetos foi modificado de [70], considerando uma distribuição uniforme inteira no intervalo [2, 25];
- *ExpTerm3* - um problema de expansão da capacidade térmica de um parque térmico de geração de energia elétrica, baseado em [11, Seção 1.3], com 10 termoelétricas. A demanda pela eletricidade $h \in \mathcal{R}^3$ é normalmente distribuída;
- *ExpTerm23* - baseado em *ExpTerm3*, mas com os custos de geração aleatórios, resultando em 23 variáveis aleatórias.

Para cada um dos 10 problemas acima foram resolvidos 11 casos, correspondentes aos diferentes números de cenários, usando os parâmetros dados na Tabela 10.11.

A tolerância ϵ_{cos} é dependente do problema, e varia no intervalo [0.0001, 0.2].

São apresentadas nas Tabelas 10.12 e 10.13 as médias (levando em conta os 11

²Performance profile, em inglês.

Tabela 10.11: Parâmetros

Método	n_{LP}^1	n_{LP}^k , com $k > 1$
MPC	-	-
d_λ -MPC	$\min\{0.5N, 200\}$	-
d_λ -MFE	$\min\{0.5N, 200\}$	-
d_λ -MNE	$\min\{0.5N, 200\}$	-
MFI- d_1	$\min\{0.4N, 150\}$	$\min\{1.1n_{LP}^{k-1}, 0.6N\}$
MNI- d_1	$\min\{0.4N, 150\}$	$\min\{1.1n_{LP}^{k-1}, 0.6N\}$
MFI- d_λ	$\min\{0.4N, 150\}$	$\min\{0.4N, 150\}$
MNI- d_λ	$\min\{0.4N, 150\}$	$\min\{0.4N, 150\}$

casos) dos erros e% e da redução de tempo CPU%, obtidos por cada uma das 9 técnicas empregadas. Nestas tabelas o símbolo \dagger é usado para indicar que os custos q de segundo estágio são determinísticos. Pode-se perceber que MFI-cos e MNI-cos são as técnicas mais acuradas para tais problemas.

Tabela 10.12: Média dos erros percentuais (11 casos).

Problema	d_λ -MPC	d_λ -MFE	d_λ -MNE	MFI-cos	MNI-cos	MFI- d_1	MNI- d_1	MFI- d_λ	MNI- d_λ
AirCraft \dagger	0.02	0.02	0.02	0.00	0.00	0.03	0.09	0.24	0.23
ExpTerm3 \dagger	0.00	0.00	0.00	0.02	0.07	0.36	0.56	0.75	1.17
ExpTerm23	0.01	0.01	0.01	0.72	0.73	0.84	1.45	1.43	1.85
ProductMix \dagger	0.03	0.03	0.03	0.00	0.00	0.07	0.03	0.30	0.14
ProjectSchedule \dagger	0.00	0.00	0.00	0.00	0.00	0.04	0.03	0.08	0.17
SH10 \dagger	0.02	0.02	0.02	0.00	0.00	0.02	0.05	0.02	0.07
SH31 \dagger	0.08	0.09	0.08	0.00	0.01	0.48	0.40	0.54	0.48
SH25	0.02	0.03	0.02	0.05	0.21	0.03	0.05	0.03	0.07
SHRisk \dagger	0.01	0.01	0.01	0.01	0.02	0.33	0.62	1.02	0.32
InvestRisk	0.29	0.29	0.30	1.75	1.93	0.18	0.23	0.28	0.36
Média	0.05	0.05	0.05	0.25	0.30	0.24	0.35	0.47	0.49

Tabela 10.13: Média das reduções de tempo de CPU (11 casos).

Problema	d_λ -MPC	d_λ -MFE	d_λ -MNE	MFI-cos	MNI-cos	MFI- d_1	MNI- d_1	MFI- d_λ	MNI- d_λ
AirCraft \dagger	-62.15	-67.86	-65.33	-67.91	-67.34	-68.96	-67.66	-68.15	-67.28
ExpTerm3 \dagger	-44.96	-37.29	-9.48	-75.54	-80.02	-20.70	-59.52	-52.53	-32.06
ExpTerm23	-14.64	-11.34	-18.70	-49.47	-15.21	-41.57	-67.00	-51.32	-61.40
ProductMix \dagger	-62.26	-62.12	-56.56	-59.18	-54.41	-72.99	-64.93	-62.13	-55.53
ProjectSchedule \dagger	-20.35	-30.46	-9.28	-42.08	-34.74	-51.76	-64.34	-50.87	-56.04
SH10 \dagger	-39.22	-48.93	-51.40	-61.85	-62.04	-73.39	-68.43	-70.63	-66.74
SH31 \dagger	-52.08	-57.54	-56.52	-53.07	-54.82	-78.56	-78.59	-71.44	-67.79
SH25	-31.79	-46.16	-47.79	-51.45	-63.57	-71.35	-62.97	-64.29	-57.03
SHRisk \dagger	-67.50	-65.08	-69.86	-76.57	-83.17	-59.11	-71.51	-68.19	-71.93
InvestRisk	-71.05	-74.83	-78.39	-74.06	-79.56	-72.44	-77.76	-77.89	-79.87
Média	-46.60	-50.16	-46.33	-61.12	-59.49	-61.08	-68.27	-63.74	-61.57

A partir das Tabelas 10.12 e 10.13 se tenderia a concluir que as técnicas estáticas são preferíveis às técnicas dinâmicas, devido à boa acurácia e à uma velocidade competitiva. No entanto, esta seria uma conclusão equivocada, como analisado a seguir utilizando o PP.

São apresentadas na Figura 10.2 as performances das técnicas, relativas a dois atributos: acurácia e redução de tempo de CPU. Cada linha do gráfico pode ser entendida como uma distribuição de probabilidades acumulada do atributo de interesse.

Estão resumidos na Figura 10.2 990 exemplos, referentes aos 11 casos, 10 problemas e 9 técnicas.

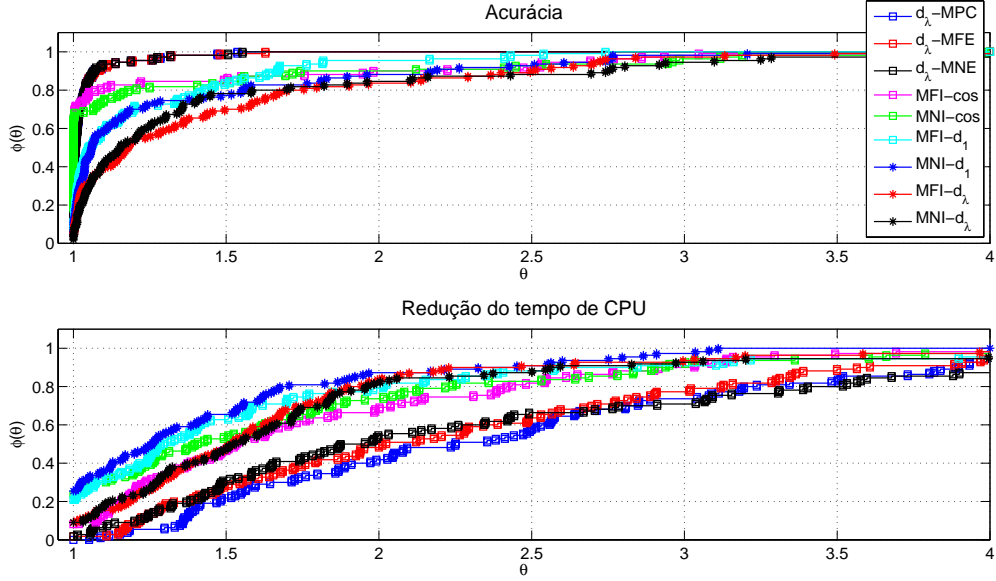


Figura 10.2: Desempenho das técnicas para todos os problemas.

Em termos de acurácia, pode-se perceber pela Figura 10.2 (parte superior) que as técnicas d_λ -MPC, d_λ -MFE, e d_λ -MNE têm, como esperado, desempenhos idênticos. Como $1 - \phi(\theta)$ representa a fração de casos que uma técnica não resolveu dentro de um fator θ da melhor técnica (de acordo com o atributo escolhido), pode-se verificar pela Figura 10.2 que as técnicas MFI-cos e MNI-cos não puderam resolver, para um fator de $\theta = 1.5$, menos de 20% dos casos com a mesma acurácia das técnicas d_λ -MPC, ou d_λ -MFE ou, ainda, d_λ -MNE (este percentual é devido aos problemas com custos de segundo estágio aleatórios). No entanto, quase todas as técnicas conseguiram, em 100% dos casos, uma acurácia de no máximo um fator de 3.5 da técnica mais acurada.

Uma interpretação errônea do gráfico superior da Figura 10.2 seria afirmar que alguma das técnicas estáticas resolve 100% dos casos com uma precisão de no máximo 1.5 vezes a precisão da técnica mais acurada. Esta interpretação estaria correta se ao invés dos valores $e\%+1$, fossem utilizados apenas os erros $e\%$. Como mencionado anteriormente, somar o valor 1 é importante para que não haja valores nulos (zeros) para o PP. Apesar de fazer com que a interpretação intuitiva do PP seja perdida, este procedimento não prejudica a análise do desempenho de cada uma das técnicas.

Como apresentado na parte inferior da Figura 10.2, em termos de tempo de CPU, d_λ -MPC é sistematicamente mais lenta do que as técnicas d_λ -MFE e d_λ -MNE, um fato conhecido em otimização não diferenciável, devido à instabilidade dos métodos de planos cortantes. Pode ser verificado que, para os parâmetros dados, MFI- d_1 e

MNI- d_1 são as técnicas mais rápidas. Outra informação importante fornecida pela Figura 10.2 é o fato de que o método de feixes proximal e o método de nível proximal apresentam desempenhos comparáveis, independentemente do oráculo empregado.

Um compromisso entre acurácia e esforço computacional pode ser obtido combinando a acurácia e redução de tempo de CPU. Para isto, é importante que os dados estejam na mesma ordem de grandeza. Com o intuito de padronizá-los, tanto a acurácia (e%) quanto a redução de CPU (CPU%) foram divididos pelos seus respectivos valores máximos. É apresentada na Figure 10.3 uma medida combinada, com peso de 50% para ambas acurácia e redução de CPU.

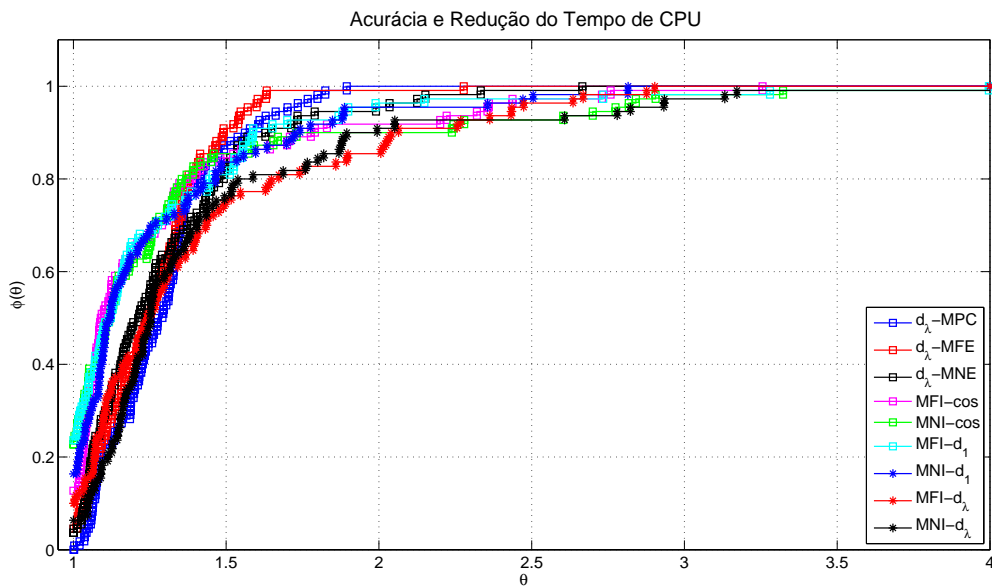


Figura 10.3: Desempenho das técnicas para todos os problemas (combinação entre acurácia e redução de CPU).

O valor $\phi(1)$ nas Figuras 10.2 e 10.3 representa a probabilidade do método de interesse ter o melhor desempenho. Estes valores são apresentados na Tabela 10.14. Pode-se verificar pela Tabela 10.14 que MFI-cos é, em 46.4% dos casos, a técnica

Tabela 10.14: Probabilidade da melhor performance.

Atributo	d_λ -MPC	d_λ -MFE	d_λ -MNE	MFI-cos	MNI-cos	MFI- d_1	MNI- d_1	MFI- d_λ	MNI- d_λ
Acur.	0.073	0.127	0.091	0.464	0.182	0.082	0.036	0.036	0.018
CPU	0.000	0.018	0.018	0.082	0.236	0.209	0.255	0.091	0.091
Acur. e CPU	0.000	0.045	0.036	0.127	0.227	0.236	0.164	0.100	0.064

mais acurada. A técnica MNI-cos apresenta a segunda melhor performance para a acurácia. No entanto, MNI- d_1 é a técnica que exigiu menor esforço computacional. Com relação a medida combinada entre acurácia e redução de CPU, pode-se perceber (pela Tabela 10.14 e Figura 10.3) que a técnica MFI- d_1 é, em 23.6% dos casos, a que tem a melhor performance, seguida da técnica MNI-cos. Ainda com relação a

Figura 10.3, todas as técnicas alcançaram o valor 1 para a abscissa igual a 4. Neste sentido, se é procurada uma técnica que resolva 100% dos problemas até um fator de 4 vezes a melhor, então todas as 9 técnicas consideradas satisfariam. Por outro lado, se o critério desejado para escolha de uma técnica for o melhor desempenho até um fator de 1.5 da técnica que tem a melhor performance, então, pela Figura 10.3, tem-se que d_λ -MFE seria escolhida, pois esta técnica proporciona o maior valor de $\phi(1.5)$.

Com o objetivo de determinar o impacto sobre o oráculo baseado no critério de colinearidade, quando se considera apenas os problemas com recursos fixos e custos q determinísticos, são apresentadas na Figuras 10.4 e 10.5 as performances das técnicas, excluindo os casos dos 3 problemas com custos q aleatórios: InvestmentRisk, SH25, e ExpTerm23. A técnica MFI-cos mantém-se mais acurada do que as técnicas estáticas, e com uma boa vantagem em tempo de CPU. Esta afirmação pode ser comprovada observando a Tabela 10.15, e é consistente com o fato que, para economizar esforço computacional, a técnica MFI-cos não realiza cálculos exatos para os problemas com custos de segundo estágio aleatórios.

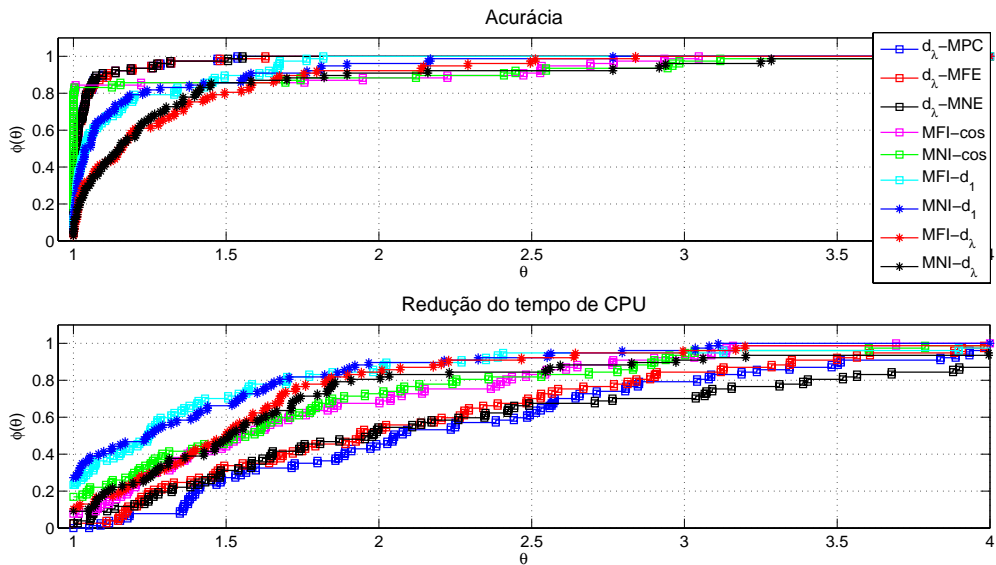


Figura 10.4: Desempenho das técnicas para os problemas com custo q fixo.

Tabela 10.15: Probabilidade da melhor performance - q fixo.

Atributo	d_λ -MPC	d_λ -MFE	d_λ -MNE	MFI-cos	MNI-cos	MFI- d_1	MNI- d_1	MFI- d_λ	MNI- d_λ
Acur.	0.104	0.052	0.039	0.623	0.247	0.026	0.000	0.000	0.013
CPU	0.000	0.026	0.000	0.078	0.247	0.234	0.247	0.104	0.065
Acur. e CPU	0.000	0.039	0.000	0.143	0.325	0.195	0.195	0.065	0.039

Com base nos resultados numéricos, o oráculo fundamentado no critério de colinearidade se apresenta como a melhor estratégia em termos de acurácia e velocidade.

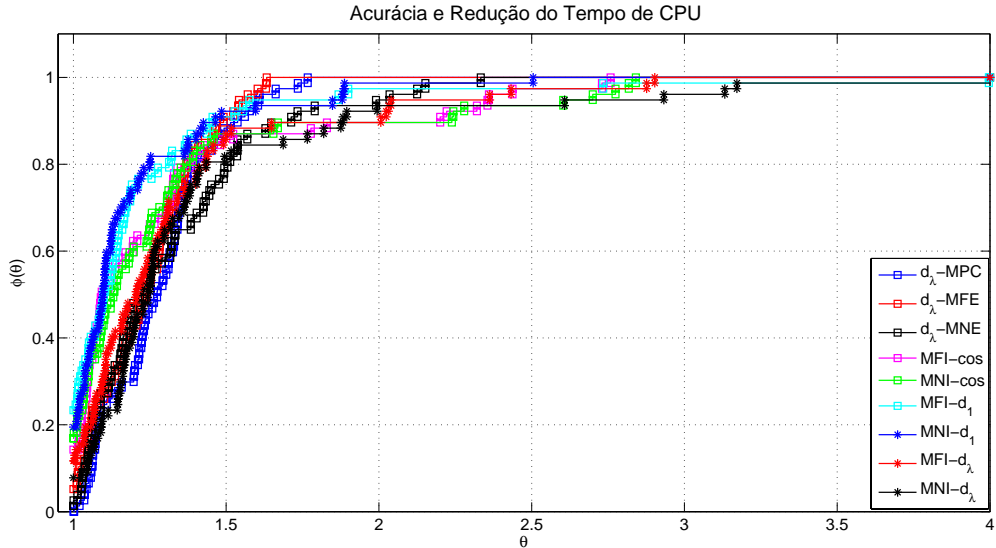


Figura 10.5: Desempenho das técnicas para os problemas com custo q fixo (combinação entre acurácia e redução de CPU).

Se o tempo de CPU não for uma grande preocupação, as variantes estáticas usando o método de feixes proximal ou o método de nível proximal podem ser boas opções (lembrando que o critério de colinearidade é sistematicamente mais rápido). Se uma variante estática for empregada, então os métodos de feixes seriam preferíveis ao método de planos cortantes.

Estas conclusões dependem dos parâmetros, e devem ser tomadas como uma indicação, ao invés de uma afirmação de superioridade do oráculo baseado no critério de colinearidade, sobre os demais oráculos. Claramente, tomando $n_{LP} = \min\{0.5N, 200\}$ a acurácia das técnicas MFI- d_1 , MNI- d_1 , MFI- d_λ , e MNI- d_λ melhoraria. No entanto, estas técnicas poderiam ser as mais demoradas, porque a estratégia de seleção de cenários realizada durante algumas iterações dos métodos de feixes exigiria um esforço computacional maior.

A seguir são analisados os métodos de feixes parcialmente inexatos.

10.1.5 Métodos Parcialmente Inexatos

Nesta seção são empregados os Oráculos Parcialmente Inexatos 1 e 5 conjuntamente com o método de feixes proximal parcialmente inexato, e com o método de nível parcialmente inexato, para resolver os problemas com custos de segundo estágios q fixos, apresentados anteriormente.

Métodos de Otimização

Adicionalmente ao MFPI e MNPI utilizando os Oráculos 1 e 5, são consideradas, a título de comparação, a *decomposição inexata de nível* dada em [18], e a *decomposição inexata de Benders* introduzida em [16]. Além disso, são considerados os métodos de planos cortantes, feixes proximal, nível, e nível proximal, todos utilizando um oráculo exato. A seguir são dados os mnemônicos das técnicas empregadas:

- MPC- método de planos cortantes exato (método L-shaped, dado em [3]);
- DIB- decomposição inexata de Benders [16];
- DIN- decomposição inexata de nível, [18];
- MFE- método de feixes proximal exato, [2];
- MNE- método de nível proximal exato, [9];
- MNE-LNN- método de nível³ exato, [19];
- MFPI-cos- método de feixes proximal parcialmente inexato com o Oráculo Parcialmente Inexato 1;
- MNPI-cos- método de nível parcialmente inexato com o Oráculo Parcialmente Inexato 1;
- MFPI-J- método de feixes proximal parcialmente inexato com o Oráculo Parcialmente Inexato 5, baseado na desigualdade de Jensen;
- MNPI-J- método de nível parcialmente inexato com o Oráculo Parcialmente Inexato 5.

As técnicas MFPI-J e MNPI-J empregam a classificação estática, i.e., o conjunto de grupos é realizada uma única vez.

Teste de Parada e Parâmetros

A cada iteração, as técnicas DIB e DIN exigem que cada programa linear de segundo estágio seja resolvido aproximadamente obtendo uma ε^k -solução. A tolerância ε^k foi definida pela regra

$$\varepsilon^{k+1} := \max\{\varepsilon^k/5, \delta_{PL}\}$$

com $\varepsilon^1 = 1/2$, e $\delta_{PL} = 10^{-8}$ é o valor padrão da rotina de programação linear do MOSEK. O parâmetro de nível para os métodos de nível é dado por $\kappa = 10^{-1}$, sendo o parâmetro ϵ_Δ do algoritmo do MNPI, para o cálculo exato de uma linearização, definido por $\epsilon_\Delta = 10^{-2}$. A tolerância para o teste de parada é $\delta_{\text{To1}} = 10^{-4}$ para os métodos de nível, e planos cortantes. Para os métodos de feixes proximais é utilizada a mesma regra apresentada na Subseção 10.1.1.

³LNN corresponde às iniciais dos autores: Lemaréchal, Nemirovskii, e Nesterov.

Alteração

Para que as técnicas MFPI-J e MNPI-J se tornassem mais competitivas, foi fixado $\zeta_{inex} = 0$ para o Oráculo Parcialmente Inexato 5 após 3 passos sérios do MFPI-J, e após o teste (6.12) ser verificado pelo MNPI-J. Este procedimento evita que o Oráculo 5 seja (por muitas iterações) invocado duas vezes para o mesmo iterado z^k (uma vez com $\zeta_{inex} = 1$, e outra com $\zeta_{inex} = 0$). Vale ressaltar que este procedimento não foi empregado nas técnicas MFPI-cos e MNPI-cos.

O número de cenários para as técnicas MFPI-J e MNPI-J foi fixado por $\min\{0.15N, 100\}$.

Desempenho das Técnicas

Salvo as tolerâncias dos testes de parada dos algoritmos, todos os métodos acima encontraram os mesmos valores ótimos para os problemas considerados. Portanto, são analisados a seguir somente o desempenho do tempo de CPU de cada algoritmo.

São apresentadas na Tabela 10.16 as reduções do esforço computacional para resolver o problema SH10, para os 11 casos. Para confrontar com a DIB, a base de comparação são os tempos de CPU do método de planos cortantes. Logo, esta tabela não é comparável com a Tabela 10.4, cuja base de comparação é o método de feixes proximal exato.

Tabela 10.16: Redução do tempo de CPU - métodos parcialmente inexatos.

N	MFE	MNE	MNE-LNN	MFPI-cos	MNPI-cos	MFPI-J	MNPI-J	DIB	DIN
100	-25.70	-6.47	7.01	-29.07	-48.34	11.97	-8.33	-0.42	12.67
200	-20.12	-20.63	-6.31	-62.58	-53.93	-9.12	-15.17	-1.31	14.26
300	-3.77	-14.84	-12.19	-63.82	-66.79	-5.81	-17.60	-2.77	-21.35
500	-11.57	-8.88	-13.84	1.61	-67.44	81.87	-17.79	0.25	-28.15
800	-18.00	-14.53	4.23	-31.13	-71.21	31.19	4.50	0.55	-13.99
1000	-23.50	-4.21	-2.70	-47.95	-57.64	2.79	9.44	-2.96	-9.62
1200	-37.28	-21.47	-21.13	-63.24	-75.87	-14.14	-20.46	-2.85	-19.59
1500	-30.97	-40.78	-25.84	-46.15	-70.31	26.35	-20.76	-2.83	-14.22
1800	-15.28	-17.36	-11.67	-63.30	-74.33	5.23	-18.00	-0.13	-10.55
2000	-31.37	-27.51	-22.25	-2.95	-74.54	72.07	-14.40	-5.65	-26.73
2500	-21.25	-24.46	-19.10	-58.29	-62.79	2.88	-16.84	-7.21	5.51
min.	-37.28	-40.78	-25.84	-63.82	-75.87	-14.14	-20.76	-7.21	-28.15
max.	-3.77	-4.21	7.01	1.61	-48.34	81.87	9.44	0.55	14.26
med.	-21.71	-18.28	-11.25	-42.44	-65.74	18.66	-12.31	-2.30	-10.16

Como pode ser verificado na Tabela 10.4, o MNPI utilizando o oráculo baseado no critério de colinearidade se apresenta como a técnica mais rápida, proporcionando para o problema SH10, uma redução de tempo de CPU de 65.75%, em média. Sendo deste modo, aproximadamente 23% mais eficiente que a segunda técnica mais rápida, MFPI-cos. É importante frisar que a técnica MFPI-J é mais demorada do que o método de planos cortantes. No entanto, a variante MNPI-J é ligeiramente mais rápida do que MN-LNN, a sua versão exata. Isto comprova na prática o que foi mencionado no Capítulo 6: o MNPI é menos susceptível à qualidade do oráculo do que o MFPI.

São apresentadas na Tabela 10.17 as médias das reduções de tempo de CPU, para os problemas considerados.

Tabela 10.17: Média das reduções de tempo de CPU - métodos parcialmente inexatos.

Problema	MFE	MNE	MNE-LNN	MFPI-cos	MNPI-cos	MFPI-J	MNPI-J	DIB	DIN
SH10	-21.71	-18.28	-11.25	-42.44	-65.74	18.66	-12.31	-2.30	-10.16
SH31	-4.66	-13.87	-11.69	-57.22	-59.26	6.78	-5.82	-1.13	-11.13
AirCraft	-17.20	-20.12	-17.18	-67.58	-70.68	-22.44	-21.38	1.37	-20.24
ExpTerm	-7.43	2.50	5.08	-22.62	-25.50	18.52	-1.28	3.31	15.07
ProductMix	-5.50	1.25	2.33	-15.49	-54.68	26.58	0.16	-0.54	2.95
ProjectSchedule	-32.64	-21.90	-20.09	-14.31	-5.91	-9.15	-10.30	0.43	-9.60
Média	-14.86	-11.74	-8.80	-36.61	-46.96	6.49	-8.49	0.19	-5.52

Apesar dos métodos exatos não serem o foco da análise, é interessante ressaltar a seguinte informação fornecida pela Tabela 10.17: em média, o MFE é mais rápido do que o MNE, que por sua vez, é mais rápido do que o MNE-LNN. Diferentemente de MFE e MNE, o MNE-LNN não é um método proximal. Esta característica pode explicar o desempenho menos rápido do MNE-LNN, em relação aos dois primeiros.

O perfil da performance das 10 técnicas analisadas nesta subseção é apresentado na Figura 10.6. Neste gráfico são utilizados os tempos de CPU para construir as distribuições acumuladas, ao invés dos valores $\text{CPU}\%+100$ como na parte inferior da Figura 10.2. Deste modo, a intuição com relação ao desempenho dos métodos é preservada. Por exemplo, verifica-se que o MFE foi incapaz de resolver aproximadamente 55% dos casos ($1 - \phi(2)$) em um tempo computacional menor que duas vezes ($\theta = 2$) o tempo da variante mais rápida. Em outras palavras, o MFE é, com probabilidade de 45%, no máximo duas vezes menos veloz do que a variante mais rápida.

Analisando os valores $\phi(1)$ para todas as técnicas, pode ser notado que MNPI-cos é a variante mais rápida em aproximadamente 63% dos casos, seguida de MFPI-cos com 21% das “vitórias”. Além disso, o MFE foi o método mais rápido em aproximadamente 15% dos casos. Neste sentido, se é procurado um método que, com probabilidade de 20%, seja o “vencedor”, então qualquer uma das variantes MFPI-cos ou MNI-cos pode ser escolhida. No entanto, a superioridade em termos de tempo de CPU da variante MNPI-cos fica evidenciada, em contraste com a variante MNI-J e, principalmente, com MFPI-J. Como resultado, pode se concluir que o Oráculo Parcialmente Inexato 5 baseado na desigualdade de Jensen não é uma alternativa interessante para reduzir o esforço computacional na resolução de programas estocásticos em dois estágios.

Como pode ser verificado pela Figura 10.6, não houve uma redução significativa do tempo computacional quando comparadas as técnicas MPC com DIB, e MNE-LNN com DIN. Uma explicação possível é o fato que os PLs de segundo estágio não sejam difíceis de serem resolvidos até a tolerância $\delta_{PL} > 0$. É apresentada em [16] uma redução média de quase 27% do tempo de CPU, quando a DIB

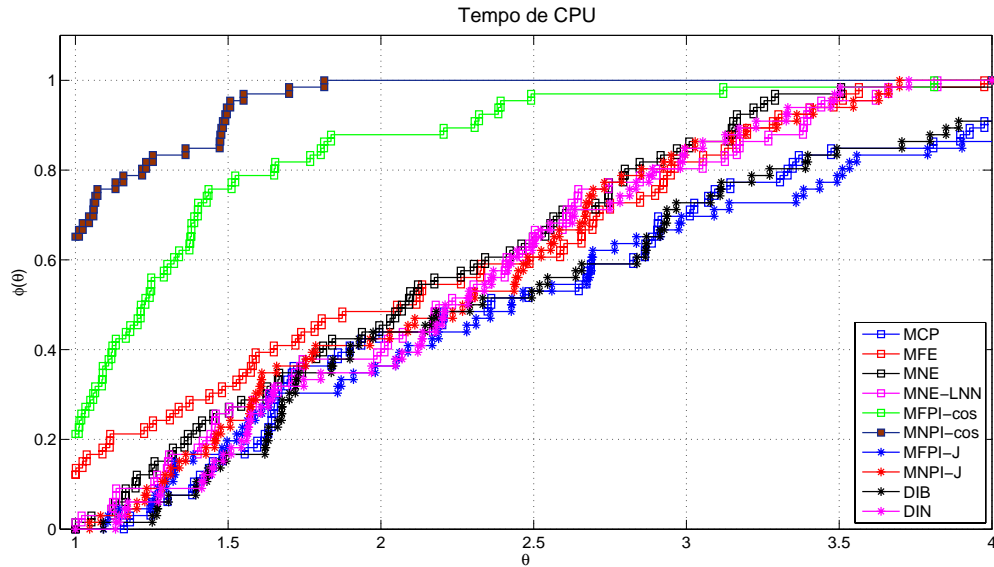


Figura 10.6: Desempenho das técnicas para os problemas com custo q fixo - métodos parcialmente inexatos.

é comparada como método de planos cortantes, para problemas de grande porte. Um bom desempenho da variante DIN é apresentado em [18]. Um estudo acerca da versão exata da DIN, que corresponde neste trabalho à variante MNE-LNN, é apresentado em [64].

A seguir são apresentados alguns resultados numéricos para os programas estocásticos em multiestágios.

10.2 Programas Lineares em Multiestágios

Nesta seção são consideradas duas variantes do problema de planejamento financeiro *PlanFin* apresentado em [52, Exemplo 7], com $T = 3$ e função de utilidade dada por [52, eq. 3.22]:

- *PlanFin3* - possui três opções de investimento;
- *PlanFin5* - possui cinco opções de investimento.

Com exceção de um ativo com retorno fixo, em cada variante deste problema, o retorno de cada investimento é normalmente distribuído com média e desvio padrão gerados aleatoriamente no intervalo $(0, 1)$.

Visto que o desempenho da técnica *aproximação por multiproblemas em dois estágios* depende somente da performance dos métodos de feixes aplicados aos programas em dois estágios (que foi analisada na seção anterior), é considerada nesta seção somente a *decomposição dual* apresentada na Seção 9.3.

Para resolver estes problemas foram empregados conjuntamente com a decomposição dual o método de feixes proximal exato, inexato, e parcialmente inexato. O Oráculo (Parcialmente) Inexato 6 empregado define o conjunto dos subproblemas que são resolvidos de maneira exata por $I^E = \{i : d_i > \mathfrak{q}\}$, com \mathfrak{q} o terceiro quantil da amostra de valores $\{d_1, \dots, d_N\}$, com d_i definido no Oráculo 6.

São apresentados na Tabela 10.18 os erros percentuais $e\%$ e a redução de tempo CPU%, bem como a estrutura da árvore de cenários e a dimensão $\dim(u)$ da variável dual do problema *PlanFin*.

Tabela 10.18: Qualidade da solução e redução do tempo de CPU - PlanFin3.

N_1	N_2	N	$\dim(u)$	MFI		MFPI
				$e\%$	CPU%	CPU%
10	3	30	147	2.13	-41.18	-19.97
10	5	50	267	0.40	-14.07	-10.99
30	3	90	447	1.46	-24.42	-18.37
30	5	150	807	0.75	-27.92	-28.83
50	3	150	747	1.26	-30.29	-23.01
50	5	250	1347	0.34	-37.94	-37.17
70	3	210	1047	2.52	14.62	17.34
70	5	350	1887	1.16	-11.22	-11.32
90	3	270	1347	1.29	-3.54	-12.92
90	5	450	2427	0.92	-25.65	-26.54
100	3	300	1497	0.00	-14.50	-12.72
100	5	500	2697	0.00	-7.31	-14.38
200	3	600	2997	1.51	-14.28	-5.93
200	5	1000	5397	0.13	-6.89	2.11
300	3	900	4497	0.00	-14.82	-12.81
300	5	1500	8097	1.34	92.91	97.71
min.	-	-	-	0.00	-41.18	-37.17
max.	-	-	-	2.52	92.91	97.71
med.	-	-	-	0.95	-10.41	-7.36

O desempenho dos métodos é apresentado na Figura 10.7. Como pode ser veri-

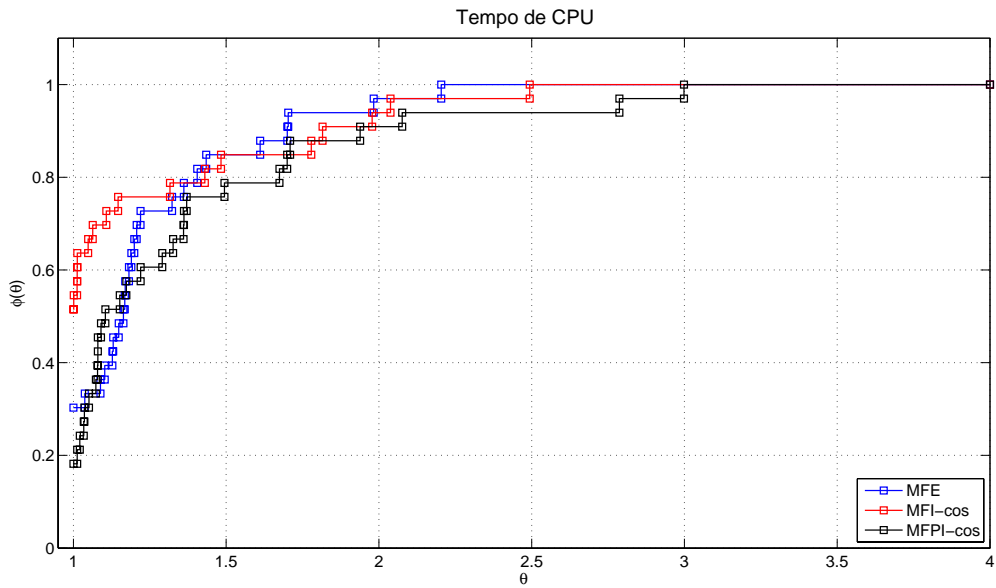


Figura 10.7: Desempenho dos métodos de feixes para problemas em multiestágios.

ficado, o MFI é método mais rápido em aproximadamente 50% dos casos.

Tabela 10.19: Média dos erros e redução de CPU (%).

Problema	MFI		MFPI
	e%	CPU%	CPU%
PlanFin3	0.95	-10.41	-7.36
PlanFin5	1.27	-13.99	-6.68
média	1.11	-12.20	-7.02

Nota-se que o erro médio proporcionado pelo MFI é inferior a 1%, e a a redução média do tempo de CPU de aproximadamente 10%. Para o MFPI a redução média do tempo está por volta de 7%, uma quantia pequena quando comparada às reduções proporcionadas pelos métodos parcialmente inexatos aplicados aos programas lineares estocásticos em dois estágios, como apresentadas na Tabela 10.17.

Para resolver os problemas PlanFin3 e PlanFin5 foram aplicados também o método de planos cortantes, e os métodos de nível exato, inexato e parcialmente inexato. Como esperado, foi verificado que o desempenho destes métodos é fortemente dependente da constante M utilizada na Proposição 9.3, para limitar o conjunto viável das variáveis duais. Em geral, as performances destes métodos foram inferiores às performances dos métodos de feixes proximais, e por isso não são relatadas em detalhe.

A seguir são apresentadas as considerações finais deste trabalho.

Capítulo 11

Considerações Finais e Direções Futuras

Neste trabalho foi apresentado um desenvolvimento teórico introdutório sobre a programação estocástica. Prezando por manter o formalismo matemático necessário, os principais conceitos e resultados desta área foram considerados no Capítulo 2.

Foi evidenciada que a maior dificuldade para lidar com os programas estocásticos consiste na chamada maldição da dimensionalidade. Problemas de otimização estocástica geralmente consideram uma grande quantidade de cenários para representar de forma acurada o processo estocástico subjacente, fazendo com que sua resolução numérica exata seja uma tarefa inviável em termos de tempo computacional. Deste modo, evitar a resolução de alguns subproblemas que definem o programa estocástico com uma degradação aceitável da solução exata é uma alternativa interessante.

Por este motivo, foram considerados neste trabalho alguns métodos de otimização não diferenciável que utilizam oráculos inexatos, que retornam um valor aproximado da função, e um subgradiente aproximado, como um esforço computacional bastante reduzido. O método de feixes proximal inexato desenvolvido por Kiwiel [8] foi detalhadamente apresentado no Capítulo 4, e aplicado aos programas estocásticos em dois e multiestágios, como apresentado nos Capítulos 7, 8, e 9.

Além do método de feixes proximal inexato, foi desenvolvido neste trabalho o método de nível proximal inexato, que é essencialmente o método exato proposto por Kiwiel [9], utilizando um oráculo inexato. Como apresentado no Capítulo 10, ambos os métodos apresentam um desempenho muito satisfatório, no sentido que proporcionam soluções de boa qualidade com uma significativa redução do tempo de CPU.

Para a aplicação destes métodos à programação estocástica, que é a principal contribuição deste trabalho, foram desenvolvidos oráculos inexatos baseados no critério de colinearidade (Oráculos 1, 2, e 6), e oráculos fundamentados na técnica de redução de cenários (Oráculos 3 e 4). Como comprovado pelos resultados numéricos,

ambos oráculos têm desempenhos satisfatoriamente bons, principalmente o Oráculo 1, desenvolvido para os programas lineares estocásticos com recurso e custo fixos, e baseado no critério de colinearidade.

Os oráculos fundamentados no critério de colinearidade são aplicáveis somente aos programas lineares estocásticos. Como apresentado, os erros de aproximação da função podem ser controlados facilmente por uma tolerância $\varepsilon_{cos} \geq 0$ dada. Os oráculos baseados em seleções de cenários podem ser aplicados também aos problemas de otimização estocástica mais gerais, desde que sejam convexos. Assume-se apenas a hipótese de que as funções objetivo dos subproblemas sejam Lipschitz contínuas na variável ξ . Este oráculos fazem uso do funcional de Monge-Kantorovich para medir a qualidade do conjunto de cenários que vai sendo atualizado sucessivamente. Assim como no critério de colinearidade, o agente de decisão é capaz de acurar a solução fornecida pelo método a partir de uma tolerância $\varepsilon_{roc} \geq 0$, fixada a priori.

Tanto para os oráculos fundamentados no critério de colinearidade como para os oráculos baseados em seleções de cenários, o grau de aproximação do problema é controlado pelo decisor de acordo com sua disponibilidade de tempo, ou exigência de informações mais precisas para a tomada de decisões. Todos os desenvolvimentos relacionados a estas propostas foram apresentados nos Capítulos 7 e 8.

Além dos métodos de feixes inexatos, foi considerado o método de feixes proximal parcialmente inexato desenvolvido por Kiwiel, [23]. Foi evidenciado que para os programas lineares estocásticos em dois estágios este método tem um desempenho muito bom; entretanto, quando o oráculo parcialmente inexato utilizado não é de boa qualidade, a performance do método pode ser inexpressiva, ou até mesmo ruim. Com o propósito de obter um método parcialmente inexato menos sensível a qualidade do oráculo parcialmente inexato, foi desenvolvido no Capítulo 6 o método de nível parcialmente inexato. Este método, ao contrário das demais três técnicas consideradas, não é método proximal. Como indicado pelos resultados numéricos apresentados, este método apresenta um desempenho notável em termos de tempo computacional. É importante destacar que os métodos parcialmente inexatos encontram assintoticamente uma solução exata do programa convexo considerado.

Adicionalmente ao Oráculo 1, que satisfaz as condições de um oráculo parcialmente inexato, foi proposto para os programas não lineares estocásticos em dois estágios (convexos em relação ao parâmetro aleatório ξ) o Oráculo 5, baseado na desigualdade de Jensen. Como mencionado no Capítulo 10, este oráculo não se mostrou muito eficiente.

Como mencionado no Capítulo 9, uma das principais dificuldades de aplicar os métodos de feixes (exato, inexato, ou parcialmente inexato) em conjunto com a decomposição aninhada de Benders é a definição dos passos sérios (para o método de feixes proximal), e a definição do conjunto de nível (para os métodos de nível). Por

este motivo, foi proposto combinar os métodos de feixes com a decomposição dual, que relaxa as restrições de mensurabilidade. A técnica resultante foi aplicada em dois problemas numéricos, que indicam um desempenho satisfatório. É importante lembrar que um programa estocástico em multiestágios pode ser modelado como um programa estocástico em dois estágios, quando desprezadas, a partir do segundo estágios, as restrições de mensurabilidade. Neste sentido, as abordagens propostas para os programas estocásticos em dois estágios são imediatamente aplicáveis. Esta é a ideia fundamental da aproximação por multiproblemas em dois estágios, apresentada na Seção 9.2.

Em termos computacionais, este trabalho considera a análise numérica de dez programas lineares estocásticos em dois estágios, e dois programas lineares estocásticos em multiestágio (com $T = 3$), aos quais foram aplicados os métodos propostos. Os resultados numéricos obtidos foram comparados com os principais métodos de otimização empregados em programação estocástica. Ao todo foram consideradas vinte abordagens distintas. Todos os métodos e exemplos considerados foram programados em Matlab, utilizando as rotinas de programação linear e quadrática do MOSEK.

Os resultados numéricos mostram uma grande vantagem computacional em relação ao emprego dos métodos de feixes (parcialmente) inexatos na resolução dos programas estocásticos. Como apresentados no Capítulo 10, os erros de precisão são inversamente proporcionais ao tamanho da amostra de cenários utilizada para representar o processo estocástico subjacente.

É importante salientar que para a programação estocástica em dois estágios, quando o número de cenários é muito grande, o esforço computacional exigido pelos Oráculos 3 e 4 é alto. Deste modo, se o problema de otimização estocástica é linear e possui muitos cenários, torna-se vantajoso utilizar o critério de colinearidade para aproximar o programa estocástico em dois estágios.

A seguir são realizados alguns comentários a cerca de possíveis trabalhos a serem realizados no futuro.

11.1 Direções Futuras

Este trabalho se restringiu aos programas estocásticos com recurso, dando maior ênfase à classe mais importante desta família, composta pelos programas em dois estágios. Entretanto, os métodos de feixes inexatos e parcialmente inexatos podem ser aplicados aos problemas de otimização estocástica com restrições de probabilidades, [1, Capítulo 4], e aos programas estocásticos que envolvem variáveis inteiras. Para problemas deste tipo, há, naturalmente, a necessidade de desenvolver oráculos (parcialmente) inexatos de boa qualidade.

Finalmente, como a dualização da decomposição de Benders resulta na decomposição de Dantizing-Wolfe (e vice-versa), é possível reproduzir as linhas gerais deste trabalho numa perspectiva da decomposição de Dantizing-Wolfe, de modo similar ao trabalho [16].

Referências Bibliográficas

- [1] SHAPIRO, A., DENTCHEVA, D., RUSZCZYŃSKI, A. *Lectures on Stochastic Programming: Modeling and Theory*. MPS-SIAM Series on Optimization. Philadelphia, SIAM - Society for Industrial and Applied Mathematics, 2009.
- [2] BONNANS, J. F., GILBERT, J. C., LEMARÉCHAL, et al. *Numerical Optimization: Theoretical and Practical Aspect*. 2 ed. Berlin, Springer-Verlag, 2006.
- [3] VAN SLYKE, R. M., WETS, R. “L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Linear Programs”, *SIAM Journal of Applied Mathematics*, v. 17, pp. 638–663, 1969.
- [4] KELLEY, J. E. “The Cutting Plane Method for Solving Convex Programs”, *J. Soc. Indust. Appl. Math.*, v. 8, pp. 703–712, 1960.
- [5] CHENEY, E., GOLDSTEIN, A. “Newton’s Method for Convex Programming and Tchebycheff approximations”, *Numerische Mathematik*, v. 1, pp. 253–268, 1959. ISSN: 0029-599X. Disponível em: <<http://dx.doi.org/10.1007/BF01386389>>. 10.1007/BF01386389.
- [6] HIRIART-URRUTY, J.-B., LEMARÉCHAL, C. *Convex Analysis and Minimization Algorithms*. N. 305-306, Grund. der math. Wiss. Springer-Verlag, 1993. (dois volumes).
- [7] RUSZCZYŃSKI, A. “A Regularized Decomposition Method for Minimizing a Sum of Polyhedral Functions”, *Math. Program.*, v. 35, pp. 309–333, July 1986. ISSN: 0025-5610. doi: 10.1007/BF01580883. Disponível em: <<http://portal.acm.org/citation.cfm?id=9504.9509>>.
- [8] KIWIEL, K. C. “A Proximal Bundle Method with Approximate Subgradient Linearizations”, *SIAM Journal on Optimization*, v. 16, n. 4, pp. 1007–1023, 2006.

- [9] KIWIEL, K. C. “Proximal Level Bundle Methods for Convex Nondifferentiable Optimization, Saddle-point Problems and Variational Inequalities”, *Math. Program.*, v. 69, n. 1, pp. 89–109, 1995. ISSN: 0025-5610. doi: <http://dx.doi.org/10.1007/BF01585554>.
- [10] OLIVEIRA, W. L., SAGASTIZÁBAL, C., MAKLER, S. “Inexact Bundle Methods for Two-stage Stochastic Programming”. http://www.optimization-online.org/DB_HTML/2010/09/2729.html, September 2010. Optimization Online.
- [11] BIRGE, J. R., LOUVEAUX, F. *Introduction to Stochastic Programming*. New York, Springer Science, 1997.
- [12] PEREIRA, M. V., PINTO, L. M. V. G. “Multistage Stochastic Optimization Applied to Energy Planning”, *Mathematical Programming*, v. 52, n. 1-3, pp. 359–375, May 1991.
- [13] DONOHUE, C. J., BIRGE, J. R. “The Abridged Nested Decomposition Method for Multistage Stochastic Linear Programs with Relatively Complete Recourse”, *Algorithmic Operations Research*, v. 1, n. 1, pp. 20–30, 2006.
- [14] HINDSBERG, M., PHILPOTT, A. “A Method for Solving Multistage Stochastic Linear Programming”. In: *SPIX Stochastic Programming Symposium*, Berlin, August 2001.
- [15] CHEN, Z. L., POWELL, W. B. “Convergent Cutting-Plane and Partial-Sampling Algorithm for Multistage Stochastic Linear Programs with Recourse”, *Journal of Optimization Theory and Applications*, v. 102, n. 3, pp. 497–524, September 1999.
- [16] ZAKERI, G., PHILPOTT, A. B., RYAN, D. M. “Inexact Cuts in Benders Decomposition”, *SIAM Journal on Optimization*, v. 10, n. 3, pp. 643–657, 1997.
- [17] FÁBIÁN, C. “Bundle-type Methods for Inexact Data”, *Central European Journal of Operations Research*, v. 8 (special issue, T. Csendes and T. Rapcsák, eds.), pp. 35–55, 2000.
- [18] FÁBIÁN, C., SZÓKE, Z. “Solving Two-Stage Stochastic Programming Problems with Level Decomposition”, *Computational Management Science*, v. 4, pp. 313–353, 2007.

- [19] LEMARÉCHAL, C., NEMIROVSKII, A., NESTEROV, Y. “New variants of Bundle Methods”, *Math. Program.*, v. 69, n. 1, pp. 111–147, 1995. ISSN: 0025-5610. doi: <http://dx.doi.org/10.1007/BF01585555>.
- [20] HIGLE, J. L., SEN, S. *Stochastic Decomposition: A Statistical Method for Large Scale Stochastic Linear Programming*. Nonconvex Optimization and Its Applications. 1 ed. New York, Springer, February 1996.
- [21] AU, K. T., HIGLE, J. L., SEN, S. “Inexact Subgradient Methods with Applications in Stochastic Programming”, *Math. Programming*, v. 63, n. 1, Ser. A, pp. 65–82, 1994. ISSN: 0025-5610. doi: 10.1007/BF01582059. Disponível em: <http://dx.doi.org/10.1007/BF01582059>.
- [22] HIGLE, J. L., SEN, S. “Statistical Approximations for Stochastic Linear Programming Problems”, *Annals of Operations Research*, v. 85, pp. 173–192, 1999.
- [23] KIWIEL, K. C. “Bundle Methods for Convex Minimization with Partially Inexact Oracles”. 2010. Disponível em: http://www.optimization-online.org/DB_HTML/2009/03/2257.html. Optimization Oline.
- [24] KALL, P., WALLACE, S. W. *Stochastic Programming*. 1 ed. Chichester, John Wiley & Sons, February 1994.
- [25] BIRGE, J. R. “Decomposition and Partitioning Methods for Multistage Stochastic Linear Programs”, *Operations Research*, v. 33, n. 5, pp. 989–1007, 1985. Disponível em: <http://www.jstor.org/stable/170850>.
- [26] DOLAN, E. D., MORÉ, J. J. “Benchmarking Optimization Software with Performance Profiles”, *Mathematical Programming*, v. 91, pp. 201–213, 2002. ISSN: 0025-5610. Disponível em: <http://dx.doi.org/10.1007/s101070100263>.
- [27] SHAPIRO, A. “Analysis of stochastic dual dynamic programming method”, *European Journal of Operational Research*, v. In Press, pp. 1–8, 2010. ISSN: 0377-2217. doi: DOI:10.1016/j.ejor.2010.08.007. Disponível em: <http://www.sciencedirect.com/science/article/B6VCT-50SPVJK-2/2/c0950df61b541a9b5eb50875692563a2>.
- [28] BORTOLOSSI, H. J., PAGNONCELLI, B. K. “Uma Introdução à Otimização sob Incerteza”. UFG, Novembro 2006. Disponível em: <http://www.mat.ufg.br/bienal/2006/minicursos.php>. III Bienal da Sociedade Brasileira de Matemática, Universidade Federal de Goiás.

- [29] ROCKAFELLAR, R. T. *Convex Analysis*. Princeton Mathematical Series. 10 ed. New Jersey, Princeton University Press, June 1970.
- [30] MIRKOV, R., PFLUG, G. C. “Tree Approximations of Dynamic Stochastic Programs”, *SIAM Journal on Optimization*, v. 18, n. 3, pp. 1082–1105, October 2007.
- [31] KAUT, M., WALLACE, S. W. “Evaluation of Scenario-Generation Methods for Stochastic Programming”. In: *World Wide Web, Stochastic Programming E-Print Series*, pp. 14–2003, 2003. Disponível em: <<http://edoc.hu-berlin.de/series/speps/2003-14/PDF/>>.
- [32] OLIVEIRA, W. L. *Redução Ótima de Cenários em Programação Estocástica. Aplicação às Vazões Afluentes aos Aproveitamentos Hidroelétricos*. Tese de Mestrado, Instituto Nacional de Matemática Pura e Aplicada, Rio de Janeiro, Agosto 2007. Disponível em: <http://www.impa.br/opencms/pt/ensino/mestrado/mestrado_dissertacoes_defendidas/mestrado_dissertacoes_defendidas_2007.html>.
- [33] KUHN, D. *Generalized Bounds for Convex Multistage Stochastic Programs*. Lecture Notes in Economics and Mathematical Systems. 1 ed. Berlin, Springer, 2005.
- [34] IZMAILOV, A., SOLODOV, M. *Otimização - Volume 1. Condições de Otimalidade, Elementos de Análise Convexa e de Dualidade*. 1 ed. Rio de Janeiro, IMPA, 2005.
- [35] DUPAČOVÁ, J., GRÖWE-KUSKA, N., RÖMISCH, W. “Scenario Reduction in Stochastic Programming: An Approach using Probability Metrics”, *Mathematical Programming*, v. 95, pp. 493–511, 2003.
- [36] CASEY, M. S., SEN, S. “The Scenario Generation Algorithm for Multistage Stochastic Linear Programming”, *Math. Oper. Res.*, v. 30, n. 3, pp. 615–631, 2005. ISSN: 0364-765X. doi: <http://dx.doi.org/10.1287/moor.1050.0146>.
- [37] OLIVEIRA, W. L., SAGASTIZABAL, C., PENNA, D. D. J., et al. “Optimal scenario tree reduction for stochastic streamflows in power generation planning problems”, *Optimization Methods Software*, v. 25, pp. 917–936, December 2010. ISSN: 1055-6788. doi: <http://dx.doi.org/10.1080/10556780903420135>. Disponível em: <<http://dx.doi.org/10.1080/10556780903420135>>.

- [38] HEITSCH, H., RÖMISCH, W. “Scenario Reduction Algorithms in Stochastic Programming”, *Computational Optimization and Applications*, v. 24, n. 2-3, pp. 187–206, February-March 2003.
- [39] HEITSCH, H., RÖMISCH, W., STRUGAREK, C. “Stability of Multistage Stochastic Programs”, *SIAM Journal on Optimization*, v. 17, n. 2, pp. 511–525, August 2006.
- [40] HOCHREITER, R., PFLUG, G. C. *Scenario Generation for Multi-Stage Decision Models: An Approach Based on Multidimensional Facility Location*. Relatório Técnico TR 2003-1, Department of Statistics and Decision Support Systems, University of Vienna, 2003. Disponível em: <<http://www.hochreiter.net/ronald/research/>>.
- [41] SHAPIRO, A., NEMIROVSKI, A. “On Complexity of Stochastic Programming Problems”, *Applied Optimization*, v. 99, pp. 111–146, March 2006.
- [42] KIWIEL, K. C. “Approximations in Proximal Bundle Methods and Decomposition of Convex Programs”, *Journal of Optimization Theory and Applications*, v. 84, pp. 529–548, 1995. ISSN: 0022-3239. Disponível em: <<http://dx.doi.org/10.1007/BF02191984>>. 10.1007/BF02191984.
- [43] MILLER, S. A. *An Inexact Bundle Method for Solving Large Structured Linear Matrix Inequalities*. Tese de Doutorado, University of California, Santa Barbara, 2001. Disponível em: <<http://proquest.umi.com/pqdlink?Ver=1\&Exp=11-27-2014\&Fmt=7\&Did=726140631\&Rqt=309\&attempt=1\&cfc=1>>. Chair-Smith, Roy.
- [44] SOLODOV, M. V. “On Approximations with Finite Precision in Bundle Methods for Nonsmooth Optimization”, *Journal of Optimization Theory and Applications*, 119(1):Springer Netherlands, October 2003. Disponível em: <<http://www.springerlink.com/content/m0100406p1rn1611/>>.
- [45] KIWIEL, K. C. “An Inexact Bundle Approach to Cutting-Stock Problems”, *INFORMS J. on Computing*, v. 22, pp. 131–143, January 2010. ISSN: 1526-5528. doi: <http://dx.doi.org/10.1287/ijoc.1090.0326>. Disponível em: <<http://dx.doi.org/10.1287/ijoc.1090.0326>>.
- [46] KIWIEL, K. C., LEMARÉCHAL, C. “An Inexact Bundle Variant Suited to Column Generation”, *Mathematical Programming*, v. 118, n. 1, pp. 177–206, April 2009. ISSN: 0025-5610. Disponível em: <<http://dx.doi.org/10.1007/s10107-007-0187-4>>.

- [47] EMIEL, G., SAGASTIZÁBAL, C. “Incremental-like Bundle Methods with Application to Energy Planning”, *Computational Optimization and Applications*, v. 46, pp. 305–332, 2010. ISSN: 0926-6003. Disponível em: <<http://dx.doi.org/10.1007/s10589-009-9288-8>>. 10.1007/s10589-009-9288-8.
- [48] HINTERMÜLLER, M. “A Proximal Bundle Method Based on Approximate Subgradients”, *Computational Optimization and Applications*, v. 20, pp. 245–266, 2001.
- [49] MACULAN, N., FAMPA, M. H. *Otimização Linear*. 1 ed. Brasília, Editora UnB, 2006.
- [50] KIWIEL, K. C. “Proximity Control in Bundle Methods for Convex Nondifferentiable Minimization”, *Mathematical Programming*, v. 46, pp. 105–122, 1990.
- [51] BURACHIK, R., SCHEIMBERG, S., SAGASTIZÁBAL, C. “An Inexact Method of Partial Inverses and a Paralell Bundle Method”, *Optimization Methods and Software*, v. 20, pp. 1–16, 2005. Disponível em: <<http://www.informaworld.com/10.1080/10556780500094887>>.
- [52] RUSZCZYŃSKI, A., SHAPIRO, A. *Stochastic Programming. Handbook in Operations Research and Management Science*, v. 10. Elsevier, 2003.
- [53] KIWIEL, K. C. “Exact Penalty Functions in Proximal Bundle Methods for Constrained Convex Nondifferentiable Minimization”, *Math. Programming*, v. 52, pp. 285–302, 1991.
- [54] MANGASARIAN, O. L., SHIAU, T. H. “Lipschitz continuity of solutions of linear inequalities, programs and complementarity problems”, *SIAM J. Control Optim.*, v. 25, n. 3, pp. 583–595, 1987. ISSN: 0363-0129. doi: 10.1137/0325033. Disponível em: <<http://dx.doi.org/10.1137/0325033>>.
- [55] RACHEV, S. T., RÜSCHENDORF, L. *Mass Transportation Problems. Volume I: Theory (Probability and its Applications)*. 1 ed. New York, Springer - Verlag, March 1998.
- [56] GRÖWE-KUSKA, N., HEITSCH, H., RÖMISCH, W. “Scenario Reduction and Scenario Tree Construction for Power Management Problems”. In: *IEEE Bologna Power Tech Proceedings*, v. 3, p. 7. (A. Borghetti, C.A. Nucci, M. Paolone eds.) IEEE, June 23-26 2003.

- [57] KÜCHLER, C. “On Stability of Multistage Stochastic Programs”, *SIAM J. on Optimization*, v. 19, n. 2, pp. 952–968, 2008. ISSN: 1052-6234. Disponível em: <<http://dx.doi.org/10.1137/070690365>>.
- [58] HEITSCH, H., RÖMISCH, W. “Scenario tree Reduction for Multistage Stochastic Programs”, *Computational Management Science*, v. 6, pp. 117–133, 2009.
- [59] MORALES, J., PINEDA, S., CONEJO, A., et al. “Scenario Reduction for Futures Market Trading in Electricity Markets”, *IEEE Transactions on Power Systems*, v. 24, n. 2, pp. 878 – 888, May 2009.
- [60] HARTIGAN, J. A., WONG, M. A. “A k-means Clustering Algorithm”, *JSTOR: Applied Statistics*, v. 28, n. 1, pp. 100–108, 1979. Disponível em: <<http://www.bibsonomy.org/bibtex/20399e12b4e411a03eda28ebaf11553ec/enitsirhc>>.
- [61] XAVIER, A. E. “The Hyperbolic Smoothing Clustering Method”, *Pattern Recognition*, v. 43, n. 3, pp. 731–737, 2010. ISSN: 0031-3203. doi: DOI:10.1016/j.patcog.2009.06.018. Disponível em: <<http://www.sciencedirect.com/science/article/B6V14-4WRD3G0-2/2/cbb627f11e8d7b71ec2e0c317c9bbf77>>.
- [62] ROCKAFELLAR, R. T., WETS, R. J. B. “Scenario and Policy Aggregation in Optimization under Uncertainty”, *Mathematical Programming*, v. 58, pp. 201–228, 1993.
- [63] ZHAO, G. “A Lagrangian Dual Method with Self-Concordant Barriers for Multi-Stage Stochastic Convex Programming”, *Mathematical Programming*, v. 102, pp. 1–24, 2005.
- [64] ZVEROVICH, V., FÁBIÁN, C., ELLISON, F., et al. “A Computational Study of a Solver System for Processing Two-stage Stochastic Linear Programming Problems”. 2009. Disponível em: <<http://edoc.hu-berlin.de/docviews/abstract.php?id=29838>>. [Online: Stand 2010-10-30T17:57:37Z].
- [65] DUPAČOVÁ, J., BERTOCCHI, M., MORIGGIA, V. “Testing the Estructure of Multistage Stochastic Programs”, *Computational Management Science*, v. 6, pp. 161–185, 2009. doi: 10.1007/s10287-008-0092-1.
- [66] SHAPIRO, A., HOMEM-DE-MELLO, T. “A Simulation Based Approach to Two-Stage Stochastic Programming with Recourse”, *Mathematical Programming*, v. 81, pp. 301–325, 1998.

- [67] DEÁK, I. “Two-Stage Stochastic Problems with Correlated Normal Variables: Computational Experiences”, *Annals of Operations Research*, v. 142, pp. 79–97, February 2006. ISSN: 0254-5330. Disponível em: <<http://dx.doi.org/10.1007/s10479-006-6162-2>>. 10.1007/s10479-006-6162-2.
- [68] MILLER, N., RUSZCZYNSKI, A. *Risk-Averse Two-Stage Stochastic Linear Programming: Modeling and Decomposition*. Relatório técnico, 2009. Disponível em: <http://www.optimization-online.org/DB_HTML/2009/08/2385.html>.
- [69] HIGHAM, D. J., HIGHAM, N. J. *MATLAB Guide*. Philadelphia, PA, USA, Society for Industrial and Applied Mathematics, 2000. ISBN: 0-89871-516-4.
- [70] ERMOLIEV, Y., WETS, R. J.-B. *Numerical Techniques for Stochastic Optimization Problems*. Springer Series in Computational Mathematics. Springer-Verlag, 1988.
- [71] JAMES, B. J. *Probabilidade: Um Curso em Nível Intermediário*. 1 ed. Rio de Janeiro, IMPA, 1981.

Apêndice A

Análise Convexa e Teoria de Probabilidade

A.1 Análise Convexa

Esta seção se restringe à apresentação dos conceitos de análise convexa utilizados neste trabalho. Recomenda-se àqueles interessados em maiores informações, uma consulta às publicações especializadas no assunto, como por exemplo [29] e [34].

Inicia-se esta seção com as principais definições de convexidade.

Definição A.1 (*Convexidade de um conjunto.*) Um conjunto $\mathcal{X} \subset \mathcal{R}^n$ é convexo quando para quaisquer x e y pertencentes a \mathcal{X} e $\lambda \in [0, 1]$, tem-se que $\lambda x + (1 - \lambda)y \in \mathcal{X}$.

Definição A.2 (*Convexidade de uma função.*) Se $\mathcal{X} \subset \mathcal{R}^n$ é um conjunto convexo, uma função $f : \mathcal{X} \rightarrow \mathcal{R}$ é dita ser convexa em \mathcal{X} quando

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) \quad \text{para todo } x, y \in \mathcal{X}.$$

Segue das definições acima a definição de um programa convexo.

Definição A.3 (*Problema de otimização convexa.*) Se $\mathcal{X} \subset \mathcal{R}^n$ é um conjunto convexo e a função $f : \mathcal{R}^n \rightarrow \mathcal{R}$ é convexa, o problema

$$\min_{x \in \mathcal{X}} f(x)$$

é dito ser um problema de otimização convexa, ou equivalentemente, um programa convexo.

Uma importante definição em otimização não linear é a seguinte.

Definição A.4 (Semicontinuidade.) Uma função $f : \mathcal{X} \rightarrow \mathcal{R}$ é *semicontínua inferiormente* (sci) no ponto $x \in \mathcal{X}$, se para toda sequência $\{x^k\} \subset \mathcal{X}$ tal que $x^k \rightarrow x$, tem-se que

$$\liminf_{k \rightarrow \infty} f(x^k) \geq f(x).$$

Quando a desigualdade é contrária, diz-se que f é *semicontínua superiormente*.

A fim de definir a condição que assegura a otimalidade do ponto $x^* \in \arg \min_{x \in \mathcal{X}} f(x)$ é necessário estabelecer o cone das direções normais, e o subdiferencial da função f .

Definição A.5 (Cone normal.) Sejam $\mathcal{X} \subset \mathcal{R}^n$ um conjunto convexo e $z \in \mathcal{X}$ um ponto qualquer. O cone normal no ponto z em relação ao conjunto \mathcal{X} é dado por

$$\mathcal{N}_{\mathcal{X}}(z) = \{d \in \mathcal{R}^n : d^\top(x - z) \leq 0, \text{ para todo } x \in \mathcal{X}\}.$$

Definição A.6 (Subgradiente.) Um ponto $g \in \mathcal{R}^n$ é dito ser um subgradiente da função convexa $f : \mathcal{R}^n \rightarrow \mathcal{R}$ em z , se

$$f(x) \geq f(z) + g^\top(x - z) \text{ para todo } x \in \mathcal{R}^n.$$

Adicionalmente, para $\varepsilon \geq 0$ o vetor g é um ε -subgradiente de f em z se

$$f(x) \geq f(z) - \varepsilon + g^\top(x - z) \text{ para todo } x \in \mathcal{R}^n.$$

O conjunto de todos os subgradientes de f no ponto $z \in \mathcal{R}^n$ é chamado de *subdiferencial* de f em z , e é denotado por $\partial f(z)$. Em particular, se f é diferenciável em z , o subdiferencial $\partial f(z)$ coincide com o gradiente $\nabla f(z)$.

Teorema A.1 Seja $\mathcal{X} \subset \mathcal{R}^n$ um conjunto convexo. Então, a função indicadora do conjunto \mathcal{X} dada por

$$i_{\mathcal{X}}(x) := \begin{cases} 0, & \text{se } x \in \mathcal{X}; \\ \infty, & \text{se } x \notin \mathcal{X}, \end{cases}$$

é convexa. Adicionalmente, para cada $x \in \mathcal{X}$, o subdiferencial de $i_{\mathcal{X}}(x)$ coincide com o cone das direções normais em x , i.e.,

$$\partial i_{\mathcal{X}}(x) = \begin{cases} \mathcal{N}_{\mathcal{X}}(x), & \text{se } x \in \mathcal{X}; \\ \emptyset, & \text{se } x \notin \mathcal{X}. \end{cases}$$

Encerra-se esta seção com um dos principais teoremas da análise convexa.

Teorema A.2 *Sejam $f : \mathcal{R}^n \rightarrow \mathcal{R}$ uma função convexa e $\mathcal{X} \subset \mathcal{R}^n$ um conjunto convexo. Então $x^* \in \arg \min_{x \in \mathcal{X}} f(x)$ se, e somente se,*

$$0 \in \partial f(x^*) + \mathcal{N}_{\mathcal{X}}(x^*).$$

A demonstração do resultado acima pode ser obtida em [34, p. 168]. A seguir são dadas algumas definições da teoria de probabilidades.

A.2 Probabilidades

Em otimização estocástica se fazem necessários alguns conceitos de probabilidades. Esta seção trata sucintamente deste assunto. Maiores informações podem ser encontradas em [71].

Definição A.7 (*Sigma álgebra.*) *Seja Ξ um conjunto abstrato. O conjunto \mathcal{F} de todos os subconjuntos de Ξ é uma σ -álgebra se:*

- i. $A, B \in \mathcal{F}$, então $A \cap B \in \mathcal{F}$, $A \cup B \in \mathcal{F}$ e $A \setminus B \in \mathcal{F}$;*
- ii. $\Xi \in \mathcal{F}$;*
- iii. $A_i \in \mathcal{F}$, para $i \in \mathbb{N}$, então $\cup_{i \in \mathbb{N}} A_i \in \mathcal{F}$.*

Se toda σ -álgebra \mathcal{S} contendo os subconjuntos de Ξ contém também \mathcal{F} , é dito que \mathcal{F} é gerada por Ξ . Quando o conjunto Ξ é finito ($|\Xi| < \infty$), então $|\mathcal{F}| = 2^{|\Xi|}$.

O conjunto Ξ associado a sua σ -álgebra \mathcal{F} é chamado de espaço mensurável e é representado por (Ξ, \mathcal{F}) . Deste modo, um subconjunto $A \subset \Xi$ é \mathcal{F} -mensurável se A é um elemento de \mathcal{F} . Um conjunto A \mathcal{F} -mensurável é elementar se os únicos dois subconjuntos de A é o próprio A , e o conjunto vazio.

A fim de definir uma medida de probabilidades, precisa-se da definição seguinte.

Definição A.8 (*Medida.*) *Seja (Ξ, \mathcal{F}) um espaço mensurável. Uma função $P : (\Xi, \mathcal{F}) \rightarrow \mathcal{R} \cup \{+\infty\}$ é uma medida se:*

- i. $P(\emptyset) = 0$, $P(A) \geq 0$ para todo $A \in \mathcal{F}$; e*
- ii. P é contavelmente aditiva, i.e.,*

$$P(\cup_{i \in \mathbb{N}} A_i) = \sum_{i \in \mathbb{N}} P(A_i),$$

onde $A_i \in \mathcal{F}$ para todo $i \in \mathbb{N}$, tal que $A_i \cap A_j = \emptyset$ se $i \neq j$.

Em particular, uma medida P é chamada de probabilidade se $P(\Xi) = 1$. Neste caso, o espaço mensurável (Ξ, \mathcal{F}) é chamado de *espaço amostral*, o subconjunto $A \in \mathcal{F}$ é um evento, e o trio (Ξ, \mathcal{F}, P) é denominado de *espaço de probabilidades*. É dito que um evento $A \in \mathcal{F}$ acontece *P -quase certamente* (q.c.) quando $P(A) = 1$.

Um mapeamento $\vartheta : \Xi \rightarrow \mathcal{R}^n$ é dito ser mensurável se para qualquer conjunto $B \subset \mathcal{R}^n$, o conjunto $\vartheta^{-1}(B) := \{\xi \in \Xi; \vartheta(\xi) \in B\}$ é \mathcal{F} -mensurável. Com este resultado, tem-se a definição de *variável aleatória*.

Definição A.9 (*Variável aleatória.*) *Uma variável aleatória é um mapeamento mensurável*

$$\vartheta : (\Xi, \mathcal{F}, P) \rightarrow \mathcal{R}^n.$$

É importante observar que uma variável aleatória $\vartheta : (\Xi, \mathcal{F}, P) \rightarrow \mathcal{R}^n$ induz uma medida de probabilidades P_ϑ em \mathcal{F} dada por

$$P_\vartheta(A) = P(\{\xi : \vartheta(\xi) \in A\}), \text{ para todo evento } A \in \mathcal{F}.$$

Deste resultado seguem as definições a seguir.

Definição A.10 (*Densidade.*) *Uma função $dP_\vartheta : \Xi \rightarrow \mathcal{R}$ é uma função de densidade da variável aleatória $\vartheta : (\Xi, \mathcal{F}, P) \rightarrow \mathcal{R}^n$ se $\int_A dP_\vartheta(\xi) = P_\vartheta(A)$, para todo evento $A \in \mathcal{F}$.*

Definição A.11 (*Valor esperado.*) *O valor esperado da variável aleatória ϑ é dado por*

$$\mathbb{E}_{P_\vartheta}[\vartheta] := \int_{\Xi} \xi dP_\vartheta(\xi).$$

Adicionalmente, o valor esperado da variável aleatória ϑ em relação à medida de probabilidades P é representado por

$$\mathbb{E}_P[\vartheta] := \int_{\Xi} \vartheta(\xi) dP(\xi),$$

e vale a igualdade $\mathbb{E}_{P_\vartheta}[\vartheta] = \mathbb{E}_P[\vartheta]$, [71].

Com a definição do valor esperado, define-se a seguir as principais noções de estatística associadas a uma variável aleatória.

Definição A.12 (*Estatísticas.*) *Sejam $\vartheta : (\Xi, \mathcal{F}, P) \rightarrow \mathcal{R}^n$ e $\zeta : (\Xi, \mathcal{F}, P) \rightarrow \mathcal{R}^n$ variáveis aleatórias. Então*

- i. $var(\vartheta) := \mathbb{E}_P(\vartheta - \mathbb{E}_P[\vartheta])^2$ é a variância de ϑ , e $var(\zeta) := \mathbb{E}_P(\zeta - \mathbb{E}_P[\zeta])^2$ é a variância de ζ ;*
- ii. $cov(\vartheta, \zeta) := \mathbb{E}_P[(\vartheta - \mathbb{E}_P[\vartheta])(\zeta - \mathbb{E}_P[\zeta])]$ é a covariância entre ϑ e ζ ;*
- iii. $corr(\vartheta, \zeta) := \frac{cov(\vartheta, \zeta)}{\sqrt{var(\vartheta)var(\zeta)}}$ é a correlação entre ϑ e ζ .*

Apêndice B

Perfil da Performance

O *Perfil da Performance* - PP - introduzido em [26] é uma ferramenta para avaliar e comparar o desempenho computacional de um método de otimização. O PP de um método é a função de distribuição acumulada de um dado atributo, que pode ser, por exemplo, a razão entre o tempo de CPU do método, e o tempo computacional do método mais rápido.

A fim de descrever o PP com mais detalhes, seja \mathcal{P} um conjunto com $n_{\mathcal{P}}$ problemas de otimização, e \mathcal{M} um conjunto com $n_{\mathcal{M}}$ métodos de otimização. O desenvolvimento seguinte consiste em comparar o atributo tempo de CPU, entretanto, a mesma análise pode ser realizada com outros atributos, como por exemplo, o número de avaliações da função, o número de iterações e, particularmente para este trabalho, a acurácia dos métodos.

Para cada problema $p \in \mathcal{P}$ e método $m \in \mathcal{M}$, seja

$t_{p,m}$ o tempo computacional requerido para resolver o problema p pelo método m .

Seja também a *razão da performance*

$$r_{p,m} := \frac{t_{p,m}}{t_p^*}, \quad \text{com } t_p^* := \min\{t_{p,m} : m \in \mathcal{M}\},$$

que é a razão entre o desempenho do método m para resolver o problema p , com o tempo computacional do método que mais rapidamente resolve o problema p . É suposto que a razão da performance seja limitada por alguma constante $R > 1$. Deste modo, $r_{p,m} \leq R$ para todo problema p , e método m . Além disso, fixa-se $r_{p,m} = R$ se, e somente se, o método m é incapaz de resolver o problema p . Como apresentado em [26], a escolha de R não afeta a análise do desempenho dos métodos.

Fixando um parâmetro $1 < \theta \leq R$, o tempo computacional para resolver o problema p pelo método m é no máximo θ vezes o tempo computacional do método

mais rápido para o mesmo problema p , se

$$t_{p,m} \leq \theta t_p^*.$$

Quando a desigualdade acima é verificada, é dito que o método m resolve o problema p dentro de (ou até) um fator θ do método mais rápido.

A performance de um método m em um dado problema p pode ser interessante; entretanto, na maioria das aplicações o principal objetivo consiste em analisar o desempenho do método m sobre todos os problemas $p \in \mathcal{P}$. Com este intuito, define-se por

$$\phi_m(\theta) := \frac{1}{n_{\mathcal{P}}} |p \in \mathcal{P} : r_{p,m} \leq \theta|$$

a probabilidade do método $m \in \mathcal{M}$ resolver o problema p com um tempo de CPU inferior a θ vezes o método mais rápido. Naturalmente, $\phi_m(R) = 1$ para todo $m \in \mathcal{M}$. Assim sendo, a função ϕ_m é a distribuição de probabilidades acumulada da razão da performance $r_{p,m}$.

Em particular, se o conjunto \mathcal{P} de problemas é representativo, no sentido de possuir muitos problemas com as características dos problemas que ocorrem em aplicações, então o método que possui maior probabilidade $\phi_m(\theta)$ é preferível (dentro de um fator θ do melhor método) aos demais, [26]. Neste sentido, o valor $1 - \phi_m(\theta)$ corresponde à fração dos problemas que o método m foi incapaz de resolver dentro de um fator θ do melhor método. Entretanto, se o interesse é escolher um método que tenha o maior número de “vitórias”, i.e, o método que tem o menor atributo (por exemplo, o tempo de CPU) para o maior número de problemas, então basta comparar o valor $\phi_m(1)$ de cada método $m \in \mathcal{M}$. O valor $\phi_m(1)$ é a probabilidade do método m ser o “vencedor”. Ao invés do número de vitórias, outro critério para escolher um método pode ser a robustez. Neste sentido, o método que atinge $\phi_m(\theta) = 1$ para o menor θ é o método mais robusto; e baseado neste critério, tal método é preferível.

As principais vantagens do PP sobre as demais técnicas de comparação de métodos são as seguintes:

- o PP minimiza a influência de um conjunto pequeno de problemas, para a análise do desempenho dos métodos;
- não há necessidade de descartar os métodos que falharam em resolver determinados problemas (basta fixar R apropriadamente, por exemplo, $R = \infty$);
- permite a visualização de um grande conjunto de resultados.

Maiores esclarecimentos acerca do PP podem ser encontrados em [26].

Apêndice C

Exemplos de Programas Lineares Estocásticos em Dois Estágios

C.1 Planejamento da Expansão e Operação Termoelétrica - *ExpTerm*

É considerado nesta seção o problema de planejamento da expansão e operação de geração de energia termoelétrica. Em cada estágio do planejamento n usinas termoelétricas podem ser consideradas. Mais precisamente, cada usina i pode ter sua capacidade de geração expandida até um limite superior especificado. O custo de expansão da usina i é estocástico. De forma análoga, o custo de operação pode também ser incerto, pois depende dos preços dos combustíveis¹ fósseis utilizados pelas usinas térmicas convencionais.

Para planejar a geração de energia elétrica de um parque com n usinas térmicas tem-se os dados a seguir:

- $\text{CapacAnt}_i \rightarrow$ capacidade de geração de energia elétrica instalada até o início do período de planejamento. Se $\text{CapacAnt}_i = 0$, a usina i não existe (ou não produz energia elétrica) até o primeiro estágio do planejamento.
- $x_i^t \rightarrow$ capacidade de geração elétrica da usina i decidida no estágio t .
- $s_i^t \rightarrow$ total da capacidade de geração de energia elétrica decidida até o estágio t . Matematicamente, $s_i^1 = x_i^1$ e $s_i^2 = s_i^1 + x_i^2$.

Para um planejamento multiestágios, a capacidade total de geração de energia elétrica da usina i no estágio t é dada por $s_i^t = s_i^{t-1} + x_i^t - x_i^{t-L_i}$, onde L_i é a vida útil da capacidade decidida $x_i^{t-L_i}$. Por exemplo, se a capacidade decidida no estágio 1

¹O custo de operação se torna conhecido quando se contrata combustível.

(x_i^1) tem vida útil de 12 estágios, $L_i = 12$, a capacidade total de geração decidida para usina i no estágio 13 é $s_i^{13} = s_i^{12} + x_i^{13} - x_i^1$.

A capacidade decidida x_i^t demora um tempo para ser construída, diga-se Δ_i . Com esta notação, a capacidade total de geração de energia elétrica da usina i no estágio t (não confundir com a capacidade total decidida s_i^t) é

$$\text{CapacTot}_i^t := \text{CapacAnt}_i + s_i^{t-\Delta_i}.$$

Por simplicidade, neste exemplo é assumido que o tempo de construção da capacidade decidida é de 1 estágio ($\Delta_i = 1$ para $i = 1, \dots, n$), i.e., a capacidade decidida no estágio $t - 1$ esta disponível para a produção de energia elétrica no estágio t .

Uma abordagem mais realística consideraria a capacidade total de geração como sendo

$$\text{CapacTot}_i^t := \delta_i^t (\text{CapacAnt}_i + s_i^{t-\Delta_i}),$$

onde $\delta_i \in [0, 1]$ é um parâmetro que determina a disponibilidade da usina i no estágio t . Se $\delta_i^t = 0.5$ significa que a usina i possui apenas 50% de sua capacidade total disponível para a produção de energia elétrica no estágio t . A adoção do parâmetro δ permite a representação de possíveis manutenções planejadas.

O principal fator a ser considerado no planejamento da operação e expansão de uma configuração energética é a demanda. Em geral, considera-se que a demanda de energia elétrica é disposta em K patamares de carga que geralmente são realizações incertas de um processo estocástico. Neste exemplo, assume-se que a demanda no estágio $t > 1$ e patamar j (d_j^t) tem distribuição normal com média μ_j^t e desvio padrão σ_j^t , i.e.,

$$d_j^t \sim N(\mu_j^t, \sigma_j^t).$$

A demanda de energia elétrica para o primeiro estágio d_j^1 , $j = 1, \dots, K$ é conhecida, não havendo assim incertezas.

Para atender à demanda d_j^t , a soma das produções de energia elétrica das usinas deve ser no mínimo igual a demanda. Denominando y_{ij} a produção de energia elétrica da usina i no patamar j , deseja-se que

$$\sum_{i=1}^n y_{ij}^t = d_j^t, \text{ para } j = 1, \dots, K \text{ e } t = 1, \dots, T,$$

onde T é o número de estágios (horizonte) de planejamento.

Naturalmente, a produção de cada usina não pode ultrapassar a capacidade total disponível:

$$\sum_{j=1}^K y_{ij}^t \leq \text{CapacTot}_i^t, \text{ para } i = 1, \dots, n \text{ e } t = 1, \dots, T.$$

Com esta formulação, pode-se definir o problema de otimização que consiste em minimizar o custo de expansão e o custo de produção de energia elétrica, sujeito ao atendimento à demanda, aos limites de produção, aos limites de expansão, e ao atendimento de uma meta de expansão do parque termoeletrico.

A seguir é realizada uma listagem dos parâmetros e variáveis do problema considerado:

- **Parâmetros**

- n é o número de usinas (cada usina é indexada por $i = 1, \dots, n$);
- K é o número de patamares (cada patamar é indexado por $j = 1, \dots, K$);
- T é o número de estágios;
- **CapacAnt** é a capacidade das usinas existentes antes do período de planejamento;
- L é a vida útil da capacidade decidida;
- Δ é o tempo de construção da capacidade decidida;
- **MetaExp** é a meta mínima de expansão do parque termoeletrico ao fim do segundo estágio;
- LimExp_i é o limite de expansão de cada usina.

- **Variáveis estocásticas**

- CustoExp_i^2 é o custo de expansão de cada usina no segundo estágio;
- CustoOper_i^2 é o custo de operação de cada usina no segundo estágio;
- d_j^2 é a demanda de energia elétrica no patamar j do segundo estágio.

- **Variáveis de decisão**

- x_i^t é a capacidade expandida decidida para a usina i no estágio t ;
- s_i^t é a capacidade total expandida decidida até o estágio t ;
- y_{ij} é a produção de energia elétrica da usina i no patamar j .

Neste trabalho são fixados

$$n = 10, \quad K = 3, \quad T = 2, \quad \text{e} \quad \text{MetaExp} = 17.$$

Os valores adotados para os demais parâmetros são apresentados nas Tabelas C.1, C.2 e C.3.

Os custo de expansão e operação, bem como os limites de expansão, são representados *por unidade pu*, que pode ser por exemplo, $1\text{pu} = 1 \cdot 10^3$.

Tabela C.1: *ExpTerm*. Parâmetros das usinas 1-5.

	Usina ₁	Usina ₂	Usina ₃	Usina ₄	Usina ₅
CapacAnt	4	3.8	5	6	4
Δ	1	1	1	1	1
LimExp ¹	4	6	1	5	2
LimExp ²	0.5	0.5	0.5	0.5	2
CustoExp ¹	14	9.8	18.2	8.4	21
CustoExp ²	$N(14, 6)$	$N(9.8, 4.5)$	$N(18.2, 0.3)$	$N(8.4, 6)$	$N(21, 6)$
CustoOper ¹	4.8	5.4	3.84	6.6	4.8
CustoOper ²	$N(4.8, 1.2)$	$N(5.4, 2.1)$	$N(3.84, 0.3)$	$N(6.6, 1.5)$	$N(4.8, 1.2)$
$\delta^1 = \delta^2$	1	1	1	1	1

Tabela C.2: *ExpTerm*. Parâmetros das usinas 6-10.

	Usina ₆	Usina ₇	Usina ₈	Usina ₉	Usina ₁₀
CapacAnt	1	3	3	1	1
Δ	1	1	1	1	1
LimExp ¹	5.5	6	4	4	4
LimExp ²	2	4	4	4	4
CustoExp ¹	22.4	22.4	22.4	22.4	22.4
CustoExp ²	$N(22.4, 6)$	$N(22.4, 6)$	$N(22.4, 6)$	$N(22.4, 6)$	$N(22.4, 6)$
CustoOper ¹	4.56	4.68	4.8	4.56	4.56
CustoOper ²	$N(4.56, 1.5)$	$N(4.68, 2.7)$	$N(4.8, 1.8)$	$N(4.56, 1.2)$	$N(4.56, 0.6)$
$\delta^1 = \delta^2$	1	1	1	1	1

Tabela C.3: *ExpTerm*. Demanda.

	Patamar ₁	Patamar ₂	Patamar ₃
d ¹	13	10	6.5
d ²	$N(12, 6)$	$N(11, 5)$	$N(7, 3.2)$

Para esta formulação o problema de otimização pode ser escrito como

$$\left\{ \begin{array}{l} \min \quad c^\top x + \sum_{i=1}^N p_i q_i^\top y_i \\ \text{s.a} \quad Ax = b \\ \quad \quad Wy_i = h_i - Tx \\ \quad \quad x, y_i \geq 0, \quad i = 1, \dots, N, \end{array} \right. \quad (\text{C.1})$$

(que possui a mesma estrutura do problema (7.1) ou mais especificamente,

$$\left\{ \begin{array}{l} \min \quad \left\{ \begin{array}{l} \sum_{i=1}^n (\text{CustoExp}_i^1 x_i^1 + \sum_{j=1}^K \text{CustoOper}_j^1 y_{ij}^1) + \\ \mathbb{E} \left[\sum_{i=1}^n (\text{CustoExp}_i^2 x_i^2 + \sum_{j=1}^K \text{CustoOper}_j^2 y_{ij}^2) \right] \end{array} \right\} \Rightarrow c^\top x + \sum_{i=1}^N q_i^\top y_i \\ \text{s.a} \quad \left\{ \begin{array}{l} \left\{ \begin{array}{l} s_i^1 = x_i^1, \quad i = 1, \dots, n \\ \sum_{i=1}^n y_{ij}^1 = d_j^1, \quad j = 1, \dots, K \\ \sum_{j=1}^K y_{ij}^1 \leq \text{CapacAnt}_i, \quad i = 1, \dots, n \\ x^1, s^1, y^1 \geq 0 \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} Ax = b \\ x \geq 0 \end{array} \right. \\ \left\{ \begin{array}{l} s_i^2 = s_i^1 + x_i^2, \quad i = 1, \dots, n \\ \sum_{i=1}^n y_{ij}^2 = d_j^2, \quad j = 1, \dots, K \\ \sum_{j=1}^K y_{ij}^2 \leq \text{CapacTot}_i^2, \quad i = 1, \dots, n \\ x^2, s^2, y^2 \geq 0 \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} Wy_i = h_i - Tx \\ y_i \geq 0. \end{array} \right. \end{array} \right.$$

C.2 Programa Linear Estocástico com Recurso Completo - *SH10*

A seguir é apresentado o problema artificial *SH10* publicado em [66], e considerado no Capítulo 10. O problema *SH10* tem a mesma formulação do problema (C.1), com a importante diferença de que o vetor de custo e a matriz de tecnologias do segundo estágio não envolve incertezas, i.e., $\xi = h$. A variável aleatória ξ é normalmente distribuída como média e desvio padrão dados, respectivamente, por

$$\mu = [-3.88 \quad 1.12 \quad -4.63 \quad 5.04 \quad 2.05 \quad 5.19 \quad -5.53 \quad 3.80 \quad 1.81 \quad -9.29],$$

$$\sigma = [0.15 \quad 0.01 \quad 0.21 \quad 0.25 \quad 0.04 \quad 0.27 \quad 0.31 \quad 0.14 \quad 0.03 \quad 0.86].$$

Os demais parâmetros são determinísticos, e apresentados a seguir.

$$c = [0.73 \quad -2.16 \quad -0.31 \quad 9 \quad -5.33 \quad 4.30 \quad 5.80 \quad 6.17 \quad -0.09 \quad 2.65]^\top;$$

$$q = [0 \quad 0 \quad 4.80 \quad 5.99 \quad 9.46 \quad 7.01 \quad 0 \quad 6.46 \quad 2.88 \quad 0 \quad 0 \quad 4.95 \quad 0 \quad 0 \quad 1.29]^\top;$$

$$b = [-1.24 \quad -2.79 \quad 8.00 \quad -1.94 \quad 2.61]^\top;$$

$$T = \begin{bmatrix} -8.42 & 0 & 0 & 0 & 0 & 0 & 6.91 & 0 & -2.07 & 0 \\ -5.23 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2.14 & 0 & 0 & 0 & 1.16 & 0 & 0 & 0 & 0 & 0 \\ 1.19 & 0 & 0 & 0 & 0 & 0 & 0 & -6.25 & 0 & -4.82 \\ 0 & 0 & 5.48 & 0 & 0 & 0 & 0 & -4.75 & 0 & 0 \\ 2.38 & 0 & 2.90 & 0 & 0 & -0.88 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.04 & 0 & 0 \\ 2.61 & 0 & 0 & -0.91 & 0 & 0 & 0 & 0 & -4.93 & 0 \\ -5.79 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9.60 & 0 \\ -2.64 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & -4.19 & 0 & 0 & 0 & 4.12 & 0 & 0 & -3.53 & 0 \\ -0.34 & -1.88 & 0 & 0 & 0 & 0 & -1.32 & 0 & 0 & -4.54 \\ 0 & 3.04 & 8.34 & 3.41 & -7.90 & 0 & 0 & 6.45 & 0 & 9.80 \\ 0 & 0 & -9.97 & 0 & 0 & 0 & 5.26 & 0 & 0 & -0.89 \\ 0 & -0.92 & 0 & 6.57 & 0 & 0 & 2.05 & 0 & 2.17 & -2.31 \end{bmatrix}$$

$$W^T = \begin{bmatrix} 0 & 0 & 0 & -0.07 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 9.17 & 5.48 & 0 & 0 & 0 & -7.35 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7.93 & -7.41 & 0 & 0 & 0 \\ 0 & 4.36 & 0 & 0 & 0 & 9.69 & 0 & 5.35 & 0 & 0 \\ 0 & 0 & 0 & -1.60 & 0 & 0 & 0 & 0 & 0 & 7.43 \\ -2.30 & 0 & 0 & 5.38 & 2.96 & 0 & 0 & 0 & 0 & 3.74 \\ 0 & 5.13 & 0 & 0 & 2.15 & 1.65 & 0 & 0 & -5.73 & 2.09 \\ 0 & 0 & 0 & -2.74 & 0 & 0 & 0 & 0 & -2.58 & 0 \\ 0.18 & 0 & 0 & -5.49 & -7.52 & -5.92 & 0 & 0 & 0 & 8.96 \\ -6.37 & 0 & 0 & -3.05 & 0 & 0 & 0 & 0 & 0 & -5.88 \\ 8.49 & 8.96 & 3.45 & 0 & 2.41 & -10.36 & 0 & 2.69 & 0 & 0 \\ 0 & 0 & -6.02 & 0.29 & 0 & 0 & 7.41 & -0.69 & 5.27 & 0 \\ 0 & -18.45 & 0 & 0 & 0 & 0 & 0 & 0 & 3.05 & -16.36 \\ 0 & 0 & 0 & 1.80 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -6.60 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Como verificado por Deák [67], há dois erros de tipografia em [66]: o vetor com os desvios padrão são na verdade as variâncias de cada coordenada do vetor h ; a coordenada $W(10, 13)$ da matriz W é -16.33 , em vez de $W(10, 13) = -16.36$.