

UNIVERSIDADE FEDERAL DO PARANÁ

PRISCILA SAVULSKI FERREIRA

UM ALGORITMO DE FILTRO GLOBALMENTE CONVERGENTE  
SEM DERIVADAS DA FUNÇÃO OBJETIVO PARA  
OTIMIZAÇÃO RESTRITA E  
ALGORITMOS DE PIVOTAMENTO EM BLOCOS PRINCIPAIS  
PARA PROBLEMAS DE COMPLEMENTARIDADE LINEAR

CURITIBA

2016



PRISCILA SAVULSKI FERREIRA

UM ALGORITMO DE FILTRO GLOBALMENTE CONVERGENTE  
SEM DERIVADAS DA FUNÇÃO OBJETIVO PARA  
OTIMIZAÇÃO RESTRITA E  
ALGORITMOS DE PIVOTAMENTO EM BLOCOS PRINCIPAIS  
PARA PROBLEMAS DE COMPLEMENTARIDADE LINEAR

Tese apresentada ao Programa de Pós-Graduação em Matemática da Universidade Federal do Paraná, como requisito parcial à obtenção do título de Doutora em Matemática.

Orientadora: Prof.<sup>a</sup> Dr.<sup>a</sup> Elizabeth W. Karas.

Coorientadora: Prof.<sup>a</sup> Dr.<sup>a</sup> Mael Sachine.

Orientador no exterior: Prof. Dr. Joaquim J. Júdice.

CURITIBA

2016





MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DO PARANÁ  
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO  
Setor CIÊNCIAS EXATAS  
Programa de Pós Graduação em MATEMÁTICA  
Código CAPES: 40001016041P1

## ATA DE SESSÃO PÚBLICA DE DEFESA DE TESE PARA A OBTENÇÃO DO GRAU DE DOUTOR EM MATEMÁTICA

No dia vinte e cinco de Fevereiro de dois mil e dezesseis às 10:30 horas, na sala Auditório da Informática, Coordenação PPGMA, Centro Politécnico, UFPR, do Setor de CIÊNCIAS EXATAS da Universidade Federal do Paraná, foram instalados os trabalhos de arguição da doutoranda **PRISCILA SAVULSKI FERREIRA** para a Defesa Pública de sua Tese intitulada: "**Um algoritmo de filtro globalmente convergente sem derivadas da função objetivo para otimização restrita e algoritmos de pivotamento em blocos principais para problemas de complementaridade linear**". A Banca Examinadora, designada pelo Colegiado do Programa de Pós-Graduação em MATEMÁTICA da Universidade Federal do Paraná, foi constituída pelos seguintes Professores Doutores: ELIZABETH WEGNER KARAS (UFPR), CLOVIS CAESAR GONZAGA (UFSC), GISLAINE APARECIDA PERIÇARO (UNESPAR), LUCAS GARCIA PEDROSO (UFPR), SANDRA AUGUSTA SANTOS (UNICAMP). Dando início à sessão, a presidência passou a palavra a discente, para que a mesma expusesse seu trabalho aos presentes. Em seguida, a presidência passou a palavra a cada um dos Examinadores, para suas respectivas arguições. A aluna respondeu a cada um dos arguidores. A presidência retomou a palavra para suas considerações finais e, depois, solicitou que os presentes e a doutoranda deixassem a sala. A Banca Examinadora, então, reuniu-se sigilosamente e, após a discussão de suas avaliações, decidiu-se pela **APROVAÇÃO** da aluna. A doutoranda foi convidada a ingressar novamente na sala, bem como os demais assistentes, após o que a presidência fez a leitura do Parecer da Banca Examinadora, outorgando-lhe o Grau de **Doutor em MATEMÁTICA**. Nada mais havendo a tratar a presidência deu por encerrada a sessão, da qual eu, ELIZABETH WEGNER KARAS, lavrei a presente ata, que vai assinada por mim e pelos membros da Comissão Examinadora.

Curitiba, 25 de Fevereiro de 2016.

Prof ELIZABETH WEGNER KARAS (UFPR)  
(Presidente da Banca Examinadora)

Prof CLOVIS CAESAR GONZAGA (UFSC)

Prof GISLAINE APARECIDA PERIÇARO (UNESPAR)

Prof LUCAS GARCIA PEDROSO (UFPR)

Prof SANDRA AUGUSTA SANTOS (UNICAMP)





MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DO PARANÁ  
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO  
Setor CIÊNCIAS EXATAS  
Programa de Pós Graduação em MATEMÁTICA  
Código CAPES: 40001016041P1

### TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em MATEMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Tese de Doutorado de **PRISCILA SAVULSKI FERREIRA**, intitulada: "**Um algoritmo de filtro globalmente convergente sem derivadas da função objetivo para otimização restrita e algoritmos de pivotamento em blocos principais para problemas de complementaridade linear**", após terem inquirido a aluna e realizado a avaliação do trabalho, são de parecer pela sua

APROVAÇÃO.

Curitiba, 25 de Fevereiro de 2016.

Prof ELIZABETH WEGNER KARAS (UFPR)  
(Presidente da Banca Examinadora)

Prof CLOVIS CAESAR GONZAGA (UFSC)

Prof GISLAINE APARECIDA PERICARO (UNESPAR)

Prof LUCAS GARCIA PEDROSO (UFPR)

Prof SANDRA AUGUSTA SANTOS (UNICAMP)



UNIVERSIDADE FEDERAL DO PARANÁ  
PRISCILA SAVULSKI FERREIRA

UM ALGORITMO DE FILTRO GLOBALMENTE CONVERGENTE SEM  
DERIVADAS DA FUNÇÃO OBJETIVO PARA OTIMIZAÇÃO RESTRITA E  
ALGORITMOS DE PIVOTAMENTO EM BLOCOS PRINCIPAIS PARA  
PROBLEMAS DE COMPLEMENTARIDADE LINEAR



---

Prof.<sup>a</sup> Dr.<sup>a</sup> Elizabeth Wegner Karas  
Orientadora - Universidade Federal do Paraná - UFPR



---

Prof.<sup>a</sup> Dr.<sup>a</sup> Mael Sachine  
Coorientadora - Universidade Federal do Paraná - UFPR

Membros da banca:



---

Prof. Dr. Clovis Caesar Gonzaga  
Universidade Federal de Santa Catarina - UFSC



---

Prof.<sup>a</sup> Dr.<sup>a</sup> Sandra Augusta Santos  
Universidade Estadual de Campinas - Unicamp



---

Prof.<sup>a</sup> Dr.<sup>a</sup> Gislaïne Aparecida Perigo  
Universidade Estadual do Paraná - Campus Campo Mourão



---

Prof. Dr. Lucas Garcia Pedroso  
Universidade Federal do Paraná - UFPR

Curitiba, 25 de fevereiro de 2016.



## Agradecimentos

Quero agradecer a pessoas especiais que, a sua maneira, muito me auxiliaram na conclusão desse trabalho.

Agradeço primeiramente aos meus familiares, pelo apoio, incentivo, compreensão, ensinamentos, ajuda e, em especial, por todo carinho e amor ao longo deste percurso. Aos meus pais, Marli e Teófilo, exemplos de perseverança, compaixão e bondade, retribuo seu amor, dedicação e confiança e agradeço por estarem sempre ao meu lado. Agradeço à minha avó Erondina pelo exemplo de vida e pelo amor concedido. Ao meu irmão Leandro, pelo carinho, incentivo e conselhos. À Lilian, ao Carlos, ao André e à Daiane por todo apoio, carinho e motivação. Espero que minha determinação e foco sejam um exemplo para as gerações futuras da nossa família. Amo vocês.

Agradeço às minhas orientadoras e amigas, Elizabeth e Mael, pela dedicação destinada ao meu trabalho, amizade, apoio, entrega, incentivo, carinho e, sobretudo, por me auxiliarem a me tornar uma profissional de qualidade e uma pessoa melhor.

Aos meus amigos e colegas de curso, pela cumplicidade e ajuda. Um agradecimento especial aos meus amigos e companheiros Adriano, Aura, Elvis, Janaina, Karla e Leonardo que, em momentos difíceis, me deram forças para continuar e me ajudaram matematicamente a conseguir este triunfo. Aos meus amigos externos agradeço pelo ombro amigo, apoio, carinho, incentivo e motivação.

Ao meu orientador no exterior, Joaquim, agradeço por me acolher com carinho em Portugal e pela dedicação e incentivo ao meu trabalho. Agradeço à Universidade de Coimbra e à CAPES, através do Programa de Doutorado Sanduíche no Exterior, pela oportunidade.

Aos meus mestres agradeço pela dedicação, entusiasmo e ensinamentos concedidos. Àqueles que me incentivaram e apostaram em mim, em especial aos professores, Ademir, Ailin, Marcelo, Matioli e Volmir.

Aos professores Clóvis C. Gonzaga, Sandra A. Santos, Gislaine A. Perçaro e Lucas G. Pedroso agradeço seus pareceres, suas participações na banca, suas críticas, e por fazerem sugestões que muito enriqueceram meu trabalho. Em especial à Sandra que participou das várias etapas deste trabalho sempre com um olhar criterioso. À Gislaine,

pelos algoritmos implementados.

Ao Programa de Pós-Graduação em Matemática da UFPR pela oportunidade e formação de qualidade propiciada. Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico, pelo apoio financeiro.

A Deus pela força espiritual, pela proteção e por seguir ao meu lado.

Ao maior incentivador à minha pesquisa e ao meu crescimento, Caio, agradeço por toda atenção, zelo, apoio, carinho, dedicação ao meu desenvolvimento, amor e, sobretudo, companheirismo. Sei que sem o seu apoio o caminho seria ainda mais árduo e longo. À ele e aos meus pais dedico este trabalho.

*“Cada palavra colocada, um sentimento de atenção  
Cada verso feito, um sentimento de empolgação  
Cada frase feita, um sentimento de determinação  
Cada parágrafo construído, um sentimento de vibração  
Cada rima conseguida, um sentimento de concentração  
Cada leitor tocado, um sentimento de satisfação.”*

Nick Santys

*A*

*Marli, Teófilo e Caio.*

## Resumo

Este trabalho engloba dois temas diferentes. Inicialmente, apresentamos um algoritmo para resolver problemas de otimização restrita que não faz uso das derivadas da função objetivo. O algoritmo mescla conceitos de restauração inexata com técnicas de filtro. Cada iteração é decomposta em duas fases: uma fase de viabilidade e uma fase de otimalidade, as quais visam reduzir os valores da medida de inviabilidade e da função objetivo, respectivamente. A fase de otimalidade é computada por iterações internas de região de confiança sem derivadas, sendo que seus modelos podem ser construídos por qualquer técnica, contanto que sejam aproximações razoáveis para a função objetivo em torno do ponto corrente. Assumindo esta, e hipóteses clássicas, provamos que o algoritmo satisfaz uma certa condição de eficiência, a qual implica sua convergência global. Para a análise prática, são apresentados alguns resultados numéricos. O segundo tema refere-se a problemas de complementaridade linear. Nesta parte são discutidos alguns algoritmos de pivotamento em blocos principais, eficientes para solucionar este tipo de problema. Uma análise sobre algumas técnicas para garantia de convergência desses algoritmos é realizada. Apresentamos alguns resultados numéricos para comparar a eficiência e a robustez dos algoritmos discutidos. Além disso, são apresentadas duas aplicações para o método de pivotamento em blocos principais: decomposição em matrizes não negativas e métodos de gradiente projetado preconditionado. Para finalizar, nesta segunda aplicação, sugerimos uma matriz de preconditionamento.

**Palavras-chave:** Convergência global; otimização sem derivadas; métodos de região de confiança; métodos de filtro; problema de complementaridade linear; métodos de gradiente projetado.

## Abstract

This work covers two different subjects. First we present an algorithm for solving constrained optimization problems that does not make explicit use of the objective function derivatives. The algorithm mixes an inexact restoration framework with filter techniques. Each iteration is decomposed in two phases: a feasibility phase that reduces an infeasibility measure; and an optimality phase that reduces the objective function value. The optimality step is computed by derivative-free trust-region internal iterations, where the models can be constructed by any technique, provided that they are reasonable approximations of the objective function around the current point. Assuming that this and classical hypotheses hold, we prove that the algorithm satisfies an efficiency condition, which provides its global convergence. Preliminary numerical results are presented. In the second subject, we discuss the linear complementarity problem. Some block principal pivoting algorithms, efficient for solving this kind of problem, are discussed. An analysis of some techniques to guarantee convergence results of these algorithms is made. We present some numerical results to compare the efficiency and the robustness of the algorithms. Moreover we discuss two applications of the block principal pivoting: nonnegative matrix factorization and preconditioned projected gradient methods. Furthermore, in this second application, we suggest a preconditioning matrix.

**Keywords:** Global convergence; derivative-free optimization; trust-region methods; filter methods; linear complementarity problem; projected gradient methods.

# Lista de Algoritmos

1.1	SD-F	21
2.1	BPP	60
2.2	KR	62
2.3	BPP-PC	64
2.4	BPP-M	83
2.5	MKR	86
3.1	BPP-NMF	112
3.2	PG	114
3.3	PPG	120



# Lista de Figuras

1.1	Conjunto linearizado $\mathcal{L}$ . . . . .	13
1.2	Filtro e as respectivas regiões dos pares dominados pelo filtro. . . . .	14
1.3	Relação entre o domínio de $f$ e as regiões no plano $f \times h$ . . . . .	15
1.4	Regiões proibidas no plano $f \times h$ segundo os critérios de filtro original e inclinado. . . . .	16
1.5	Regiões proibidas temporárias e permanentes. . . . .	17
1.6	Diferença entre a folga do filtro $H_k$ e a altura do filtro $v_k$ . . . . .	35
1.7	Comportamento do Alg. 1.1 com filtro inclinado para o problema (1.65). . . . .	41
1.8	Gráfico de desempenho das instâncias CD-FO, CD-FI, SD-FO e SD-FI. . . . .	43
1.9	Gráfico de desempenho com exceção dos problemas com saída -4. . . . .	44
1.10	Pefil de dados em relação ao número de avaliações de $f$ . . . . .	45
2.1	Cone complementar associado ao LCP descrito em (2.5), com $\omega_1 = \omega_2 = 0$ . . . . .	51
2.2	Cones complementares para o LCP (2.5), e o vetor $c = (-4, -5)$ . . . . .	52
3.1	Gráfico de desempenho dos Algoritmos F1F2, diagH, SPG e SPG+F1F2. . . . .	129
3.2	Comportamento dos Algoritmos F1F2, diagH, SPG e SPG+F1F2 ao longo das iterações. . . . .	130



# Lista de Tabelas

1.1	Casos considerados na demonstração do Teorema 1.11. . . . .	31
1.2	Quantidades de problemas $\times$ saídas para as instâncias com e sem derivadas. . . . .	42
2.1	Solução periódica gerada pelos algoritmos BPP e KR para $j = 0, 1, 2, \dots$ . . . . .	66
2.2	Ciclo gerado pelo Algoritmo 2.3. . . . .	67
2.3	Propriedades de dois iterandos consecutivos $(x, \omega)$ e $(x^+, \omega^+)$ . . . . .	75
2.4	Atualizações dos conjuntos $F^k$ e $T^k$ para os conjuntos $F^{k+1}$ e $T^{k+1}$ . . . . .	89
2.5	Relação das soluções complementares básicas primais viáveis $(x^k, \omega^k)$ e $(x^{k+1}, \omega^{k+1})$ com os conjuntos $F^k, T^k, F^{k+1}$ e $T^{k+1}$ . . . . .	89
2.6	Dados das matrizes consideradas da coleção [101] e desempenho dos Algoritmos BPP e BPP-PC. . . . .	95
2.7	Dados das matrizes não simétricas dadas por [101] e resultados numéricos. . . . .	102
2.8	Variação do conjunto inicial. . . . .	103
2.9	Melhor e pior desempenho de acordo com os conjuntos iniciais adotados. . . . .	103
2.10	Resultados da Tabela 2.9 apresentados em detalhes. . . . .	104
2.11	Desempenho dos algoritmos para matrizes- $P$ simétricas estritamente diagonalmente dominante. . . . .	105
2.12	Desempenho dos Algoritmos BPP, KR, BPP-M e BPP-PC. . . . .	106
2.13	Desempenho dos algoritmos para um LCP com vetor $c$ conforme (2.70). . . . .	107



# Sumário

Introdução	1
<b>PARTE 1</b>	<b>9</b>
<b>1 Um algoritmo de filtro globalmente convergente sem derivadas da função objetivo para otimização restrita</b>	<b>11</b>
1.1 O problema . . . . .	11
1.2 Método de filtro . . . . .	13
1.3 O algoritmo . . . . .	17
1.4 Convergência global . . . . .	22
1.5 Experimentos computacionais . . . . .	38
1.5.1 Algoritmos testados . . . . .	38
1.5.2 Exemplo ilustrativo . . . . .	41
1.5.3 Resultados numéricos . . . . .	41
1.6 Conclusões do capítulo . . . . .	46
<b>PARTE 2</b>	<b>47</b>
<b>2 Algoritmos de pivotamento em blocos principais para problemas de complementaridade linear</b>	<b>49</b>
2.1 O problema . . . . .	49
2.1.1 Classes de matrizes . . . . .	52
2.1.2 Classes do problema de complementaridade linear . . . . .	56
2.2 Algoritmos de pivotamento em blocos principais . . . . .	57
2.2.1 Pivotamento em blocos principais clássico - BPP . . . . .	58
2.2.2 Pivotamento em blocos principais alternativo - KR . . . . .	61
2.2.3 Algoritmo em duas fases - BPP-PC . . . . .	62

2.3	Ciclos nos algoritmos . . . . .	64
2.3.1	Algoritmos BPP e KR . . . . .	65
2.3.2	Algoritmo BPP-PC . . . . .	66
2.4	Funções de mérito . . . . .	67
2.4.1	O Algoritmo BPP e o método de Newton . . . . .	69
2.5	Análise de convergência . . . . .	73
2.5.1	Caso simétrico . . . . .	74
2.5.2	Combinação com método convergente . . . . .	82
2.5.3	Impossibilidade de ciclos . . . . .	84
2.5.4	Técnica de marcação de variáveis . . . . .	84
2.5.5	Comentários . . . . .	92
2.6	Experimentos computacionais . . . . .	93
2.6.1	Algoritmos BPP e BPP-PC . . . . .	93
2.6.2	Versões modificadas dos algoritmos . . . . .	98
2.7	Conclusões do capítulo . . . . .	100
<b>3</b>	<b>Aplicações dos algoritmos de pivotamento em blocos principais</b>	<b>109</b>
3.1	Decomposição não negativa de uma matriz . . . . .	109
3.2	Algoritmo de gradientes projetados . . . . .	113
3.2.1	Método do gradiente projetado espectral . . . . .	118
3.2.2	Método do gradiente projetado preconditionado . . . . .	119
3.2.3	Obtenção de matrizes de preconditionamento . . . . .	123
3.2.4	Experimentos numéricos . . . . .	127
	<b>Conclusões</b>	<b>131</b>
	<b>Referências</b>	<b>133</b>
	<b>APÊNDICES – PARTE 1</b>	<b>145</b>
<b>A</b>	<b>Resultados auxiliares – Capítulo 1</b>	<b>147</b>
<b>B</b>	<b>Desempenho dos métodos de filtro para os problemas da coleção CU-TEST</b>	<b>159</b>

# Introdução

Esta tese aborda diferentes métodos para abranger duas frentes de trabalho: otimização contínua com restrições suaves porém sem derivadas da função objetivo e otimização restrita especificamente para problemas de complementaridade linear. Este segundo tema surgiu do doutoramento sanduíche no Instituto de Telecomunicações de Coimbra sob orientação do Prof. Dr. Joaquim J. Júdice no período de janeiro a março de 2015.

A seguir fazemos uma breve revisão bibliográfica procurando situar nossa tese no atual estado da arte.

## Otimização sem derivadas

Num primeiro momento abordamos métodos para problemas de programação não linear nos quais as derivadas da função objetivo não estão disponíveis ou são desconhecidas e as avaliações da função objetivo são caras computacionalmente. Existem muitas aplicações de otimização sem derivadas em diversas áreas tais como projetos de engenharia, projetos de circuito, precificação dinâmica, geometria molecular e registro de imagens médicas, como mencionadas em [34]. Tais situações motivaram pesquisadores a desenvolver técnicas para otimização livre de derivadas.

Em particular, M.J.D. Powell tem uma grande contribuição na área de otimização sem derivadas. Em 1964, ele começou seu exaustivo estudo nesta área com o trabalho [108]. Para problemas de otimização irrestrita, Powell propôs os métodos de região de confiança UOBYQA [109] e NEWUOA [110], com bons resultados práticos. Baseados em [110], são apresentados em [7] e [68] algoritmos para resolver problemas com restrições de caixa e com restrições lineares, respectivamente. Em [112], Powell propôs o algoritmo BOBYQA para restrições de caixa. Apesar de bons resultados práticos, os algoritmos apresentados em [109, 110, 112] não fornecem ferramentas necessárias para prova de con-

vergência. A fim de garantir resultados teóricos, ele sugere em [114] um método para problemas irrestritos globalmente convergente.

Durante meu mestrado estudamos métodos de região de confiança [14, 29, 100] para problemas irrestritos sem uso das derivadas da função objetivo, o que culminou com a publicação do artigo [51], onde apresentamos um algoritmo passo a passo. O algoritmo envolve dois tipos de iterações: de região de confiança e alternativas; e os modelos da função objetivo são construídos por interpolação polinomial [30, 34]. Nas iterações de região de confiança resolvemos um subproblema na esperança de que pelo menos uma fração da redução obtida pelo modelo seja herdada pela função objetivo. Já as iterações alternativas têm o objetivo de cuidar da boa geometria dos pontos interpoladores e são divididas em duas variações: alfa e beta. Enquanto as tentativas alfa visam maximizar o volume da envoltória convexa formada pelos pontos interpoladores, as iterações beta evitam que os pontos fiquem muito distantes do centro da região de confiança. Sob hipóteses razoáveis, provamos que todos os pontos de acumulação da sequência gerada pelo algoritmo são estacionários.

Vários métodos sem derivadas vêm sendo desenvolvidos para problemas irrestritos [31, 35, 47, 51, 109, 111, 114], problemas com restrições de caixa [7, 34, 66, 86, 112, 118], problemas com restrições lineares [32, 68, 82, 87, 114, 113], problemas com restrições de igualdade [25], problemas com restrições convexas [28], problemas com restrições *lower-level* [41] e problemas com restrições gerais [8, 10, 11, 25]. Resultados de convergência global para problemas irrestritos são apresentados em [31, 34, 51, 114], para problemas restritos a um conjunto viável convexo e fechado em [28], para problemas com restrições de caixa em [8] e para problemas com restrições de igualdade em [25, 45].

Nesta tese abordamos problemas de otimização não linear restritos cujas derivadas da função objetivo não estão disponíveis, enquanto as derivadas das restrições estão [25, 91]. Em relação à mesma classe de problemas, Bueno et.al. [25] propuseram um trabalho de restauração inexata [53, 93, 94] para problemas com restrições de igualdade no qual o progresso do algoritmo é medido por uma função de mérito. Porém, usar estas funções traz a necessidade de tomar decisões difíceis como, por exemplo, realizar as atualizações dos parâmetros de penalidade. Valores muito grandes ou muito pequenos destes parâmetros tendem a provocar instabilidade numérica [74]. Além disso, uma escolha muito grande amortece o efeito da função objetivo, resultando por exemplo, em convergência lenta quando o método de minimização segue o contorno da região viável. Outro ponto delicado

é que, dependendo do valor do parâmetro de penalidade, o algoritmo pode recusar uma solução ótima ou, por outro lado, ficar muito lento. Como uma alternativa para funções de mérito, Fletcher e Leyffer [55] sugeriram, no contexto de otimização com derivadas, as técnicas de filtro.

Ao longo dos últimos anos, os métodos de filtro vêm sendo estudados por vários autores. Inicialmente proposto para resolver problemas de programação não linear, os métodos de filtro têm sido combinados com abordagens de região de confiança [106, 117], técnicas de programação quadrática sequencial [54, 128], algoritmos de restauração inexata [62, 78], estratégias de pontos interiores [125] e algoritmos de busca linear [67, 103, 127]. Também foram estendidos para outras áreas de otimização, tais como equações e inequações não lineares [44, 56, 63, 65], otimização não suave [80, 104], problemas de complementaridade [89, 90] e otimização sem derivadas [9, 45]. Enquanto em [9] é sugerido um algoritmo de filtro com busca padrão, em [45], publicado recentemente, é proposto um algoritmo de filtro com restauração inexata sem derivadas.

Os métodos de restauração inexata foram propostos inicialmente para problemas de programação não linear com derivadas, por Martínez e Pilotta, em [94, 95]. Nestes métodos, cada iteração é decomposta em duas fases. Na primeira, chamada de *fase de restauração*, a viabilidade é melhorada sem avaliações da função objetivo. Na segunda, chamada de *fase de otimalidade*, busca-se uma melhora da função objetivo em uma aproximação linear das restrições. A decisão se o ponto obtido será aceito ou se outro ponto tentativo será calculado pode ser feita, por exemplo, funções de mérito [25, 93, 94] ou, como optamos neste trabalho, usando o critério de filtro [62, 78, 117]. Vale mencionar que abordagens de restauração inexata se aplicam adequadamente para o caso em que avaliações da função objetivo são caras, situação mais comum em otimização sem derivadas. Nestes casos é preferível tratar da inviabilidade de uma maneira independente da função objetivo. Além disso, se alguma restrição não possuir derivada disponível ou possuir avaliações caras, esta restrição pode ser incluída na função objetivo em um contexto de Lagrangiano aumentado [1].

Nesta parte da tese, propomos um algoritmo de restauração inexata sem derivadas cujo controle do progresso é feito por intermédio de um filtro que pode ser tanto o original [57] como o inclinado [26]. Cada iteração é composta de duas fases. Primeiramente um passo de viabilidade é computado, a partir do ponto corrente, com objetivo de obter um ponto restaurado que reduza uma medida de inviabilidade. Nesta fase, basicamente

qualquer método que reduza a medida de inviabilidade pode ser utilizado [44, 93, 94]. Em seguida, a partir do ponto restaurado, uma fase de otimalidade com técnicas de região de confiança [29, 100, 121, 129] calcula um ponto que não seja proibido pelo filtro e que forneça uma redução no valor da função objetivo. Os modelos de região de confiança podem ser lineares ou quadráticos, sendo obtidos apenas com informações de ordem zero da função objetivo. Para fins teóricos, as Hessianas dos modelos devem ser simétricas e limitadas e os gradientes devem ser aproximações razoáveis para o gradiente da função objetivo em torno do ponto restaurado. Modelos que satisfazem estas propriedades podem ser construídos por várias técnicas sem derivadas, tais como interpolação polinomial [27, 33, 34, 40] e regressão por vetores suporte [79]. Assumindo hipóteses clássicas, provamos que o algoritmo satisfaz uma certa condição de eficiência, como proposta em [117] e considerada em [78, 105, 106], que garante, essencialmente, que perto de um ponto viável não estacionário o decréscimo da função objetivo é grande. A prova desta propriedade é uma das contribuições desta tese. A convergência global do algoritmo é consequência desta propriedade, uma vez que esta é herdada dos resultados provados em [78, 106]. Experimentos numéricos preliminares são apresentados.

Nosso trabalho difere de [45] pois consideramos restrições de desigualdade além de igualdades, os modelos podem ser construídos por qualquer técnica desde que sejam boas aproximações da função objetivo, e por considerar a possibilidade do uso do filtro inclinado, que garante resultados mais fortes de convergência, ou seja, que todo ponto de acumulação da sequência gerada pelo algoritmo é estacionário.

## Problema de complementaridade linear

O problema de complementaridade linear é muito importante na otimização e consiste em, dados  $Q \in \mathbb{R}^{n \times n}$  e  $c \in \mathbb{R}^n$ , encontrar um par  $(x, \omega) \in \mathbb{R}^{2n}$  tal que

$$\begin{aligned} Qx + c &= \omega, \\ x \circ \omega &= 0, \\ x &\geq 0, \omega \geq 0, \end{aligned}$$

sendo  $x \circ \omega := (x_1\omega_1, \dots, x_n\omega_n) \in \mathbb{R}^n$ . Este problema tem uma vasta gama de aplicações em ciência e tecnologia aplicada, por exemplo, em análise sócio-econômica, teoria dos jogos, modelos de equilíbrio espacial, problemas de análise elastoplástica em mecânica e modelagem de energia [69, 76, 98].

O método de pivotamento é um dos primeiros algoritmos desenvolvidos para este tipo de problema e foi inicialmente introduzido por Lemke em [85]. O método escolhe as variáveis por uma regra de complementaridade e pode ser visto como o pioneiro nestes tipos de algoritmos [98]. Existem muitas referências na literatura que fazem uso desses algoritmos, podemos citar, por exemplo, os livros [37, 73, 98]. Estes algoritmos geralmente possuem um excelente desempenho na prática, o que é comprovado em [75] quando  $Q$  é uma matriz simétrica definida positiva ( $PD$ ) de grande porte. Uma comparação entre estes métodos e os métodos de pontos interiores é realizada em [107].

Murty em [97] propõe um algoritmo de pivotamento principal com terminação finita assegurada quando  $Q$  é uma matriz- $P$  [98]. Aquele trabalho inspirou muitos pesquisadores a desenvolver variações deste algoritmo no intuito de herdar a convergência finita do método. Em [123] é sugerida uma modelagem em dígrafos, a qual é utilizada em [43] para examinar o desempenho do método de Murty quando aplicado a classes de matrizes mais gerais. Em [76] os autores propõem o *algoritmo de pivotamento em blocos principais clássico* e o combina com o algoritmo de Murty, e conseqüentemente, herda a convergência finita do método de Murty para toda matriz- $P$ . Como as iterações do tipo Murty quase nunca são utilizadas, este último trabalho conseguiu propor um algoritmo muito eficiente. Porém este método é lento quando as iterações tipo Murty são utilizadas. Além disso, o algoritmo de pivotamento em blocos principais clássico que não usa estas últimas iterações não tem convergência teórica garantida para toda matriz- $P$ , existindo vários trabalhos que apresentam exemplos em que este algoritmo cicla [39, 124]. Em [124], é verificado que para qualquer matriz- $P$  que seja triangular inferior ou superior, o algoritmo clássico encontra a solução em no máximo  $n$  passos. Ademais, os autores também verificam que o algoritmo clássico tem terminação finita quando aplicado a problemas de complementaridade linear de ordem 3 com matrizes- $P$  estritamente diagonalmente dominantes por linhas ou colunas.

Em [83], os autores propõem um algoritmo de conjuntos ativos e garantem que o método proposto tem terminação finita para uma subclasse de matrizes  $PD$ . Um algoritmo de conjuntos ativos que converge para toda matriz  $PD$  é proposto em [72], porém este método é recursivo e não existe um controle sobre quantas recursividades serão necessárias. Verificamos neste trabalho que ambos os algoritmos podem ser considerados como um algoritmo de pivotamento em blocos principais. Deste modo ainda não existe uma prova de convergência para o algoritmo de pivotamento em blocos principais para

uma classe de matrizes satisfatória ou um algoritmo modificado eficiente.

Neste trabalho, discutimos o algoritmo clássico de pivotamento em blocos principais e verificamos que os algoritmos de conjuntos ativos apresentados em [72, 83] podem ser vistos como algoritmos de pivotamento em blocos principais. Também discutimos um algoritmo em duas fases: uma fase de viabilidade, que visa obter um ponto viável, e outra de complementaridade. No intuito de fornecer um texto auxiliar para um futuro resultado de convergência mais geral, é feito um levantamento sobre estas provas de convergência, assim como listamos várias funções de mérito que podem ser utilizadas nestes métodos. Para validar os comentários decorrentes ao longo do texto apresentamos testes numéricos que comprovam a grande eficiência desses algoritmos na prática e a lentidão do algoritmo proposto em [76], quando iterações do tipo Murty são utilizadas. Também é feita uma comparação entre o desempenho dos algoritmos apresentados e a sua dependência com os conjuntos iniciais considerados.

Para finalizar esta parte do trabalho, discutimos duas aplicações do algoritmo de pivotamento em blocos principais. A primeira delas consiste na possibilidade de utilizar este tipo de método para os problemas de decomposição em matrizes não negativas [16, 84, 102], problema decorrente do desenvolvimento tecnológico, principalmente devido ao processamento e transmissão de grandes quantidades de dados. Apresentamos as ideias de Kim e Park [81] para solucionar este tipo de problema utilizando algoritmos de pivotamento em blocos principais. Como segunda aplicação, mostramos que os algoritmos apresentados anteriormente podem ser aplicados no Algoritmo de Gradiente Projetado [60, 100, 116, 119] Precondicionado [15, 92]. Cabe salientar que estes métodos vêm sendo intensamente aplicados em problemas práticos [12, 18, 19, 20, 24, 38, 58, 115, 126]. Neste contexto, nossa principal contribuição é propor uma matriz de condicionamento para os métodos de gradiente projetado precondicionado no intuito de obter uma solução com boa precisão e em poucas iterações. Resultados numéricos preliminares são apresentados.

## **Organização do trabalho**

O trabalho está organizado da seguinte forma. No Capítulo 1, apresentamos um método de filtro com restauração inexata sem derivadas da função objetivo para problemas de otimização contínua com restrições suaves. Além disso, provamos que o algoritmo proposto é globalmente convergente e apresentamos resultados numéricos na resolução

de alguns problemas da coleção CUTEST. Discutimos, no Capítulo 2, o problema de complementaridade linear e os algoritmos de pivotamento em blocos principais como uma possibilidade para solucionar este tipo de problema. Sintetizamos alguns algoritmos existentes numa linguagem unificada e fazemos uma análise sobre alguns resultados de convergência existentes. Enfatizamos nossa conjectura de que o algoritmo de pivotamento em blocos principais clássico tem terminação finita para problemas de complementaridade linear com matrizes- $P$  diagonalmente dominantes por colunas através de resultados numéricos. No Capítulo 3, discutimos duas aplicações para o algoritmo de pivotamento em blocos principais: solução de problemas de decomposição em matrizes não negativas e aplicação no algoritmo de gradiente projetado preconditionado. Uma proposta de matrizes de preconditionamento é apresentada. No Apêndice A, alguns resultados auxiliares para a convergência estabelecida no Capítulo 1 são apresentados. As tabelas referentes aos resultados dos testes numéricos obtidos para o algoritmo de filtro proposto no Capítulo 1 são apresentadas no Apêndice B.



## PARTE 1

# OTIMIZAÇÃO SEM DERIVADAS



# Capítulo 1

## Um algoritmo de filtro globalmente convergente sem derivadas da função objetivo para otimização restrita

Neste capítulo propomos um algoritmo globalmente convergente, que não faz uso das derivadas da função objetivo, para resolver um problema de programação não linear com restrições de igualdade e desigualdade. O algoritmo combina conceitos de restauração inexata com filtro e é decomposto em duas fases: uma fase de viabilidade, que visa obter um ponto mais próximo do conjunto viável, e uma fase de otimalidade, que visa reduzir o valor da função objetivo. A fase de otimalidade consiste de um algoritmo de região de confiança, cujos modelos podem ser obtidos por qualquer técnica sem derivadas que forneça uma aproximação suficientemente boa para a função objetivo próximo ao ponto corrente. Verificamos neste capítulo que este algoritmo satisfaz uma condição de eficiência que implica a sua convergência global. Para finalizar este tema da tese, são apresentados alguns testes computacionais.

### 1.1 O problema

Este capítulo aborda problemas de otimização não linear da seguinte forma

$$(P) \quad \begin{cases} \text{minimizar} & f(x) \\ \text{sujeito a} & c_{\mathcal{E}}(x) = 0 \\ & c_{\mathcal{I}}(x) \leq 0, \end{cases}$$

sendo que a função objetivo  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  e as funções que definem as restrições  $c_i : \mathbb{R}^n \rightarrow \mathbb{R}$ , para  $i \in \mathcal{E} \cup \mathcal{I}$  com  $|\mathcal{E} \cup \mathcal{I}| = m$ , são continuamente diferenciáveis. Assumimos que as derivadas da função objetivo não estão disponíveis ou são caras computacionalmente enquanto que as das restrições estão disponíveis.

O conjunto

$$\Omega = \{x \in \mathbb{R}^n \mid c_{\mathcal{E}}(x) = 0 \text{ e } c_{\mathcal{I}}(x) \leq 0\}$$

é denominado *conjunto viável* do problema  $(P)$ . Se um ponto  $x$  pertence a  $\Omega$ , então este é chamado de *ponto viável* e, caso contrário, de *ponto inviável*. Este problema também pode ser estendido a problemas em que as derivadas de algumas restrições estejam inacessíveis ou sejam caras computacionalmente. Nestes casos, por exemplo, bastaria incluir estas restrições na função objetivo no contexto de Lagrangiano aumentado [1].

Para medir quanto um iterando está longe do conjunto  $\Omega$ , considere a função *medida de inviabilidade*  $h : \mathbb{R}^n \rightarrow \mathbb{R}_+$  definida por

$$h(x) = \|c^+(x)\|, \quad (1.1)$$

sendo que  $\|\cdot\|$  é uma norma arbitrária e  $c^+ : \mathbb{R}^n \rightarrow \mathbb{R}^m$  é dada por

$$c_i^+(x) = \begin{cases} c_i(x) & \text{para } i \in \mathcal{E} \\ \max\{0, c_i(x)\} & \text{para } i \in \mathcal{I}. \end{cases}$$

Note que  $h(x) = 0$  se, e somente se,  $x \in \mathbb{R}^n$  é um ponto viável do problema  $(P)$ .

A sequência de pontos gerada pelo algoritmo proposto será denotada por  $(x^k)_{k \in \mathbb{N}}$ . Para simplificar a notação, referenciaremos  $f(x^k)$  por  $f^k$  e  $h(x^k)$  por  $h^k$ .

Com o propósito de estabelecer a convergência global do algoritmo, as seguintes hipóteses são consideradas.

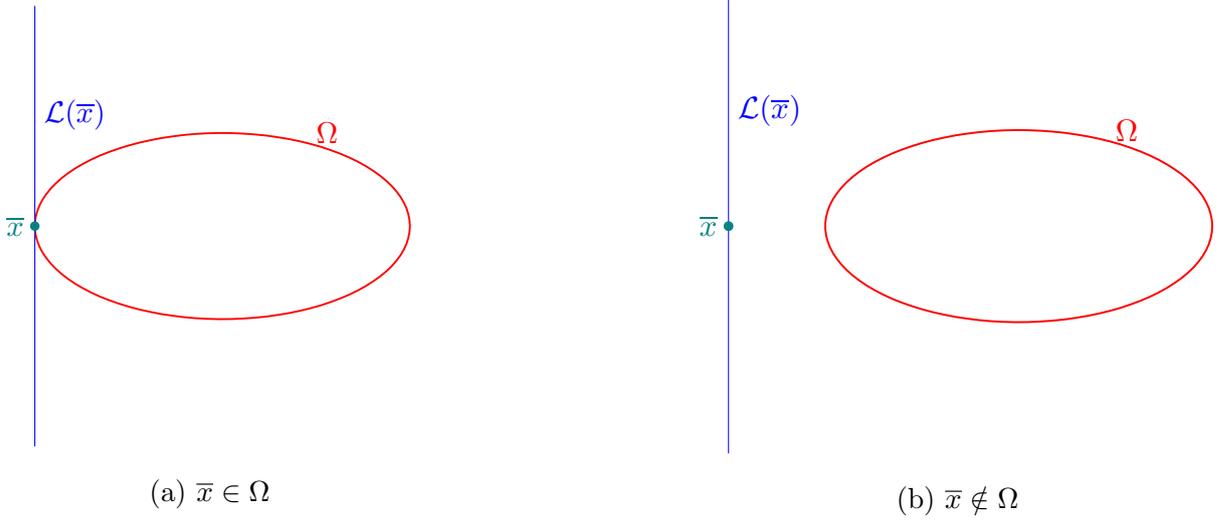
**H1** *Todas as funções  $f$  e  $c_i$ , para  $i \in \mathcal{E} \cup \mathcal{I}$ , são continuamente diferenciáveis.*

**H2** *A sequência  $(x^k)_{k \in \mathbb{N}} \subset \mathbb{R}^n$  gerada pelo algoritmo está contida em um conjunto compacto e convexo  $X$ .*

**H3** *O gradiente  $\nabla f$  é Lipschitz contínuo em um conjunto aberto contendo  $X$ , ou seja, existe uma constante  $L > 0$  tal que*

$$\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|$$

*para todo  $x$  e  $y$  em um conjunto aberto contendo  $X$ .*

Figura 1.1: Conjunto linearizado  $\mathcal{L}$ .

Cabe ressaltar que H2 é uma condição exigida sobre a sequência gerada pelo algoritmo e esta pode ser garantida ao adicionar restrições de caixa ao problema.

A seguir apresentamos a definição de estacionariedade para  $(P)$ .

**Definição 1.1** Dizemos que  $\bar{x} \in \Omega$  é um ponto estacionário para o problema  $(P)$  quando

$$\liminf_{x \rightarrow \bar{x}} \|\mathcal{P}_{\mathcal{L}(x)}(x - \nabla f(x)) - x\| = 0, \quad (1.2)$$

sendo que  $\mathcal{P}_{\mathcal{L}(x)}$  é a projeção ortogonal sobre o conjunto  $\mathcal{L}(x)$  definido por

$$\mathcal{L}(x) = \{x + d \in \mathbb{R}^n \mid Jc_{\mathcal{E}}(x)d = 0 \text{ e } c_{\mathcal{I}}(x) + Jc_{\mathcal{I}}(x)d \leq c_{\mathcal{I}}^+(x)\}, \quad (1.3)$$

onde  $Jc_{\mathcal{E}}(\cdot)$  e  $Jc_{\mathcal{I}}(\cdot)$  são as matrizes Jacobianas das restrições  $c_{\mathcal{E}}$  e  $c_{\mathcal{I}}$ , respectivamente.

O conjunto  $\mathcal{L}(x)$  é uma linearização do conjunto viável em torno de  $x$ . Na Figura 1.1a ilustramos o conjunto  $\mathcal{L}$  no ponto  $(1, 0) \in \Omega$  e na Figura 1.1b no ponto  $(0, 0) \notin \Omega$  para um problema com conjunto viável dado por

$$\Omega = \{(x, y) \in \mathbb{R}^2 \mid (x - 3)^2 + 4y^2 = 4\}.$$

## 1.2 Método de filtro

Os métodos de filtro, que se inspiram no critério de dominância de Pareto de otimização multiobjetivo, foram propostos no contexto de programação não linear por Fletcher e Leyffer em [55]. As principais referências desta seção são [26, 55].

**Definição 1.2** Dizemos que o par  $(f^i, h^i)$  domina outro par  $(f^j, h^j)$  ou que o par  $(f^j, h^j)$  é dominado pelo par  $(f^i, h^i)$  quando

$$f^i \leq f^j \quad e \quad h^i \leq h^j.$$

Este critério é denominado *Dominância de Pareto*.

No contexto de otimização não linear, quando um par  $(f^i, h^i)$  domina outro par  $(f^j, h^j)$ , isto quer dizer que o ponto  $x^i$  é pelo menos tão bom quanto o ponto  $x^j$  tanto com relação à otimalidade como com relação à viabilidade. Com base neste critério, podemos definir o filtro como sendo uma lista de pares  $(f^i, h^i)$  escolhidos convenientemente levando em consideração os dados das iterações anteriores. Denotamos por  $F_k$  o conjunto dos pares  $(f^i, h^i)$  que constituem o filtro na iteração  $k$ , com  $i < k$ .

A ideia do método de filtro consiste em evitar pontos dominados por pelo menos um dos pares do filtro corrente. Deste modo, o método não permite que seja obtido um ponto que piore tanto a viabilidade quanto a otimalidade comparado aos pares que constituem o filtro. A Figura 1.2 ilustra os pares, no plano  $f \times h$ , que constituem um determinado filtro e, na região preenchida, o conjunto de pontos que são dominados por pelo menos um dos pares deste filtro. Nesta, podemos observar que cada par  $(f^j, h^j)$  do filtro cria uma região em bloco na qual dado um ponto  $x \in \mathbb{R}^n$  tal que o par  $(f(x), h(x))$  pertença a este bloco, temos que

$$h(x) \geq h^j \quad e \quad f(x) \geq f^j,$$

ou seja, qualquer ponto pertencente a este bloco aumenta, ou mantém, tanto o valor da inviabilidade quanto o valor da função objetivo comparado a  $x^j$ . A Figura 1.3 ilustra

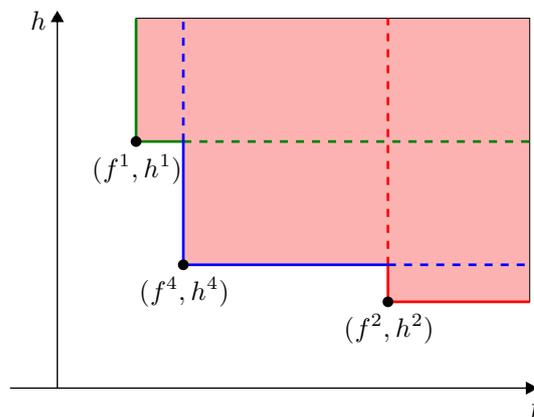


Figura 1.2: Filtro e as respectivas regiões dos pares dominados pelo filtro.

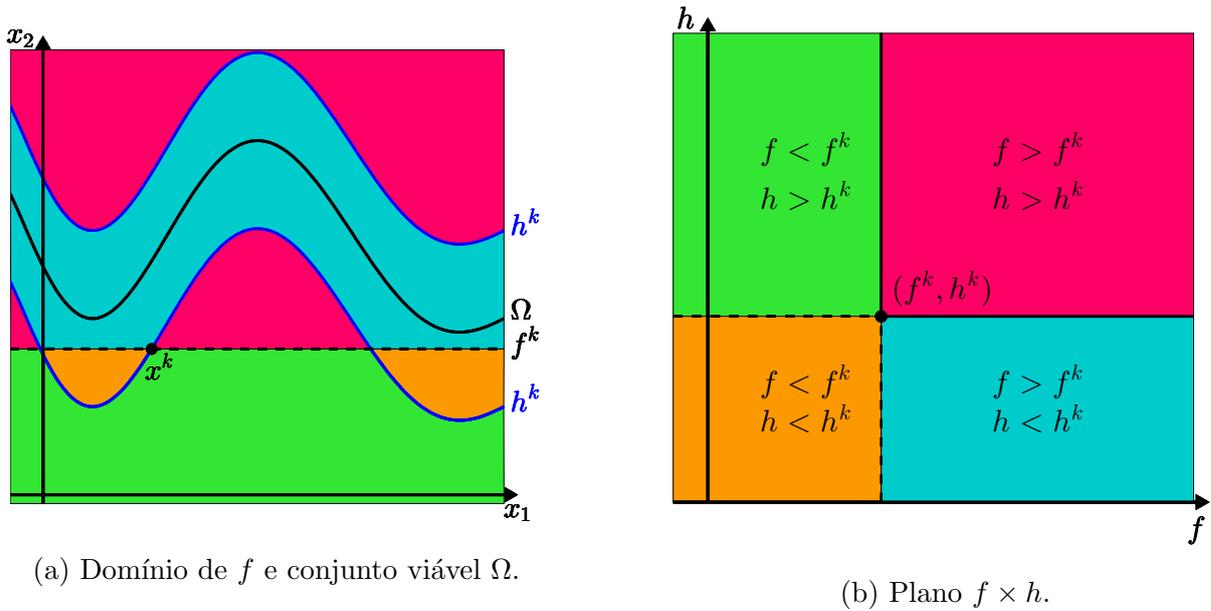
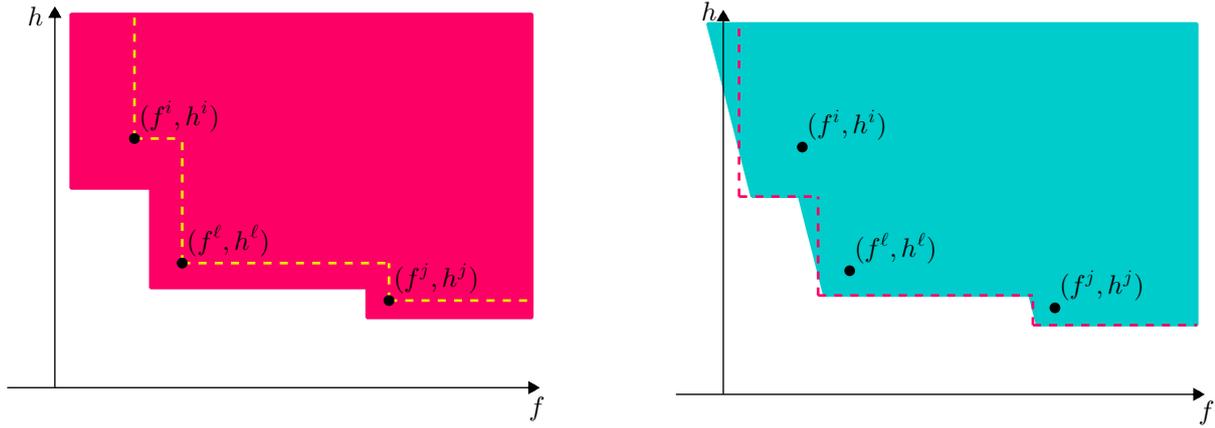


Figura 1.3: Relação entre o domínio de  $f$  e as regiões no plano  $f \times h$ .

a relação entre as regiões no domínio de  $f$  e no plano  $f \times h$  que melhoram a viabilidade, e/ou a otimalidade, ou nenhuma delas. Nesta figura, consideramos que  $(P)$  é um problema bidimensional, com função objetivo  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$  dada por  $f(x_1, x_2) = x_2$  e uma restrição de igualdade  $c : \mathbb{R}^2 \rightarrow \mathbb{R}$ . As curvas contínuas em azul (—) representam as curvas de nível da medida de inviabilidade de valor  $h^k$ , a curva contínua em preto (—) representa o conjunto viável  $\Omega$  e a reta pontilhada em preto (- -) ilustra a curva de nível de  $f$  de valor  $f^k$ . As regiões preenchidas em verde nas Figuras 1.3a e 1.3b representam os pontos e os pares, respectivamente, em que ocorre uma melhora na otimalidade, porém uma piora na viabilidade. Analogamente, em azul claro temos os pontos e os pares em que a medida de inviabilidade é reduzida, mas o valor da função objetivo aumenta. Em laranja, temos a melhor situação, na qual acontece uma redução tanto na inviabilidade quanto na otimalidade. Por fim, em rosa temos os pares dominados pelo par  $(f^k, h^k)$ , ou seja, os pontos e os pares que pioram tanto a viabilidade quanto a otimalidade comparado ao ponto  $x^k$  e ao par  $(f^k, h^k)$ .

Deste modo, podemos dizer que o método de filtro é um método que em cada iteração define uma **região proibida** em  $\mathbb{R}^n$  para o próximo ponto corrente. Esta região consiste de todos os pontos  $x \in \mathbb{R}^n$  cujos pares  $(f(x), h(x))$  são dominados por pelo menos um dos pares do filtro. Dada uma iteração  $k$ , denotaremos por  $\mathcal{F}_k \subset \mathbb{R}^n$  o conjunto dos pontos  $x \in \mathbb{R}^n$  proibidos pelo método nesta iteração.



(a) Região proibida segundo a regra (1.4) e em pontilhado a fronteira da região proibida sem a margem.

(b) Região proibida segundo a regra (1.5) e em pontilhado a fronteira da região proibida com filtro original.

Figura 1.4: Regiões proibidas no plano  $f \times h$  segundo os critérios de filtro original e inclinado.

Para garantir a convergência do método de filtro, em [57] cria-se uma margem em torno da região proibida. Assim, dado um filtro  $F_k$  a região proibida nesta iteração é dada por

$$\mathcal{F}_k = \{\cup \mathcal{R}_j \mid (f^j, h^j) \in F_k\}$$

sendo

$$\mathcal{R}_j = \{x \in \mathbb{R}^n \mid f(x) \geq f^j - \alpha h^j \text{ e } h(x) \geq (1 - \alpha)h^j\} \quad (1.4)$$

e  $\alpha \in (0, 1)$  é uma constante dada. Uma maneira um pouco diferente de definir esta região, proposta inicialmente por [26], considera uma inclinação na margem referente à otimalidade e é dada por

$$\mathcal{R}_j = \{x \in \mathbb{R}^n \mid f(x) + \alpha h(x) \geq f^j \text{ e } h(x) \geq (1 - \alpha)h^j\}. \quad (1.5)$$

Os métodos baseados na regra (1.4) serão denominados de *filtro original* e os baseados em (1.5) de *filtro inclinado*. Ilustramos na Figura 1.4 as respectivas regiões proibidas no plano  $f \times h$  segundo as regras (1.4) na figura da esquerda e (1.5) na da direita. Na Figura 1.4a, a fronteira da região sem margem está representada em pontilhado. Já na Figura 1.4b as linhas pontilhadas demarcam a fronteira da região segundo a regra (1.4).

Definimos  $\widehat{F}_k$  e  $\widehat{\mathcal{F}}_k$  os conjuntos dados por

$$\widehat{F}_k = F_k \cup \{(f^k, h^k)\} \quad \text{e} \quad \widehat{\mathcal{F}}_k = \mathcal{F}_k \cup \mathcal{R}_k,$$

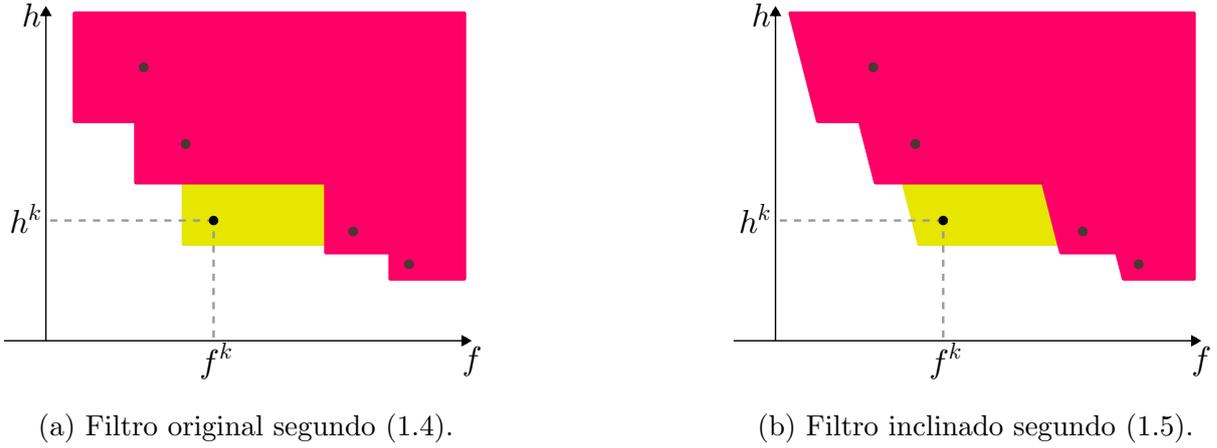


Figura 1.5: Regiões proibidas temporárias e permanentes.

estes são denominados de filtro temporário e região proibida temporária, respectivamente, sendo  $F_k$  e  $\mathcal{F}_k$  o filtro permanente e a região proibida permanente na iteração  $k$ . As regiões proibidas temporárias e permanentes, segundo as regras de filtro (1.4) e (1.5), para um determinado filtro podem ser vistos na Figura 1.5. A região proibida permanente está representada pela região preenchida em tom de rosa, já a região temporária consiste da união da região permanente, em rosa, com a região preenchida em amarelo, de acordo com cada critério de filtro adotado.

### 1.3 O algoritmo

O algoritmo proposto para resolver  $(P)$  mescla conceitos de restauração inexata com conceitos de filtro. Cada iteração  $k$  do algoritmo é composta por uma fase de viabilidade, que reduz a medida de inviabilidade sem avaliações da função objetivo, e uma fase de otimalidade, que visa reduzir o valor da função objetivo. Estas fases são independentes e a ligação entre elas ocorre por intermédio do filtro.

Dado um ponto corrente  $x^k \notin \mathcal{F}_k \subset \mathbb{R}^n$ , a fase de viabilidade encontra um ponto restaurado  $z^k$  que esteja relativamente próximo de  $x^k$ , fora do filtro temporário  $\widehat{\mathcal{F}}_k$  e que seja mais viável que o ponto corrente, ou seja, tal que  $z^k \notin \widehat{\mathcal{F}}_k$  e satisfaça

$$h(z^k) < (1 - \alpha)h(x^k) \quad \text{e} \quad \|z^k - x^k\| \leq \beta h(x^k), \quad (1.6)$$

para  $\alpha \in (0, 1)$  e  $\beta > 0$ . Então é considerada uma aproximação linear das restrições neste ponto mais viável através do conjunto linearizado  $\mathcal{L}(\cdot)$  definido em (1.3). Computado o ponto  $z^k$ , a fase de otimalidade deve encontrar um ponto  $x^{k+1} = z^k + d_k \notin \widehat{\mathcal{F}}_k$  com

$z^k + d_k \in \mathcal{L}(z^k)$ , tal que este forneça uma redução do valor da função objetivo a partir do ponto restaurado, ou seja, tal que  $f(x^{k+1}) < f(z^k)$ . Note que neste algoritmo, a única relação entre as duas etapas é dada pelo filtro. Além disso, a fase de viabilidade obtém um ponto mais próximo do conjunto viável sem que sejam necessárias avaliações da função objetivo.

A fase de viabilidade deve obter um ponto restaurado  $z^k \notin \widehat{\mathcal{F}}_k$  que satisfaça (1.6). Como as restrições são providas de derivadas, o processo usado nesta fase pode ser qualquer algoritmo iterativo de descida para  $h$ , e o término finito deve ser atingido uma vez que  $h^j > 0$  para todo  $(f^j, h^j) \in \widehat{\mathcal{F}}_k$ , (veja Lema A.2). Devido à acessibilidade das derivadas das restrições e como, por exemplo, o passo de viabilidade estudado por Martínez em [93] pode ser aplicado diretamente neste trabalho, não discutimos em detalhes o processo de viabilidade. Note que a fase de viabilidade pode falhar quando a medida de inviabilidade  $h(\cdot)$  tem um ponto estacionário inviável. Neste caso, o método para sem sucesso.

Na fase de otimalidade serão realizadas iterações de região de confiança  $k_j$  para  $j \in \mathbb{N}$ . A ideia do método de região de confiança [29, 116] é definir um modelo para a função objetivo e uma região em torno do ponto corrente na qual o modelo é uma boa aproximação para a função, ou seja, uma região na qual confiamos no modelo. A partir daí, encontramos um minimizador aproximado do modelo e caso este forneça uma redução razoável em  $f$ , aceitamos o ponto. Caso contrário, pode ser que o modelo não esteja representando adequadamente a função nesta região. Neste caso, o ponto é recusado, o tamanho da região é reduzido e construímos um novo modelo para encontrar um novo minimizador.

Os modelos quadráticos  $m_{k_j} : \mathbb{R}^n \rightarrow \mathbb{R}$  de  $f$  em torno do ponto restaurado  $z^k$  são definidos por

$$m_{k_j}(x) = f(z^k) + (x - z^k)^t g_{k_j} + \frac{1}{2}(x - z^k)^t B_{k_j}(x - z^k), \quad (1.7)$$

onde  $g_{k_j} \in \mathbb{R}^n$  e  $B_{k_j} \in \mathbb{R}^{n \times n}$  é uma matriz simétrica. O raio da região de confiança, região em que confiamos que o modelo é uma boa aproximação para  $f$ , é denotado por  $\Delta_{k_j} > 0$ . O modelo pode ou não ser atualizado no começo de cada iteração  $k_j$  e a sua qualidade é controlada por um segundo raio  $\delta_{k_j} \in (0, \Delta_{k_j}]$ . Para garantir um boa aproximação para  $f$  perto do ponto  $z^k$ , é exigido que os modelos quadráticos  $m_{k_j}$  satisfaçam à seguinte condição:

**H4** Existem constantes  $\gamma > 0$  e  $\sigma > 0$  tais que

$$\|B_{k_j}\| \leq \gamma \quad e \quad \|g_{k_j} - \nabla f(z^k)\| \leq \sigma \delta_{k_j}$$

para todo  $k \in \mathbb{N}$ ,  $j \in \mathbb{N}$  e  $\delta_{k_j} > 0$ .

Cada iteração do método de região de confiança começa com um raio  $\Delta_{k_0} \geq \Delta_{\min}$  e  $\delta_{k_0} = \Delta_{k_0}$ , para  $\Delta_{\min} > 0$  fixado. O uso deste raio mínimo simplifica a prova de convergência. Caso  $z^k + d_{k_j} \in \widehat{\mathcal{F}}_k$  ou a redução  $f(z^k) - f(z^k + d_{k_j})$  não seja suficientemente boa, dizemos que a iteração  $k_j$  foi de insucesso. Neste caso, o raio da região de confiança  $\Delta_{k_j}$  e o raio  $\delta_{k_j}$  são reduzidos, o contador  $k_j$  é incrementado, caso necessário um novo modelo satisfazendo H4 é construído e repetimos o processo com os novos raios. Caso contrário, a iteração  $k_j$  é de sucesso e o processo de região de confiança está concluído nesta iteração  $k$ .

Apesar de  $z^k$  estar fixado, a condição H4 e o fato do raio  $\delta_{k_j}$  ser reduzido em uma iteração de insucesso são responsáveis pela atualização dos modelos em cada iteração interna  $k_j$ , uma vez que,  $g_{k_j} = g(z^k, \delta_{k_j})$  e, conseqüentemente,  $m_{k_j}(\cdot) = m(\cdot, z^k, \delta_{k_j})$ . Esta dependência dos modelos com relação a cada raio  $\delta_{k_j}$  ocorre exclusivamente em  $g_{k_j}$ . As Hessianas podem ser sempre as mesmas, em particular nulas. Qualquer técnica que satisfaça a Hipótese H4 pode ser utilizada para obter os modelos quadráticos, os processos mais conhecidos na literatura são os baseados em interpolação polinomial [34, 47, 122]. Existem vários algoritmos capazes de encontrar modelos que satisfaçam esta condição sem calcular  $\nabla f(z^k)$ , como em [34, Capítulo 6].

Em cada iteração  $k_j$  consideramos o subproblema de região de confiança dado por

$$(SP) \quad \begin{cases} \text{minimizar} & m_{k_j}(z^k + d) \\ \text{sujeito a} & z^k + d \in \mathcal{L}(z^k) \\ & \|d\| \leq \Delta_{k_j}, \end{cases} \quad (1.8)$$

sendo que  $\|\cdot\|$  é uma norma arbitrária em  $\mathbb{R}^n$ . A solução obtida é denominada de passo de região de confiança e é denotada por  $d_{k_j} \in \mathbb{R}^n$ . Definimos a medida de estacionariedade para (SP) em um ponto  $x$  por

$$\pi_{k_j}(x) = \|\mathcal{P}_{\mathcal{L}(x)}(x - \nabla m_{k_j}(x)) - x\|. \quad (1.9)$$

A próxima etapa consiste em avaliar o passo. Esperamos que o ponto  $z^k + d_{k_j}$  proporcione uma redução na função objetivo que seja no mínimo uma fração da redução

obtida no modelo. Para tal, definimos  $ared_{k_j}$  como sendo a redução real obtida pela função objetivo, dada por

$$ared_{k_j} = f(z^k) - f(z^k + d_{k_j}) \quad (1.10)$$

e  $pred_{k_j}$  a redução predita e obtida pelo modelo, dada por

$$pred_{k_j} = m_{k_j}(z^k) - m_{k_j}(z^k + d_{k_j}). \quad (1.11)$$

Um passo  $d_{k_j}$  é aceito quando a redução real for menor que uma fração  $\eta \in (0, 1)$  da redução predita, isto é, quando

$$ared_{k_j} > \eta \text{ pred}_{k_j},$$

pois do contrário, o modelo não está representando bem a função objetivo na região delimitada por  $\Delta_{k_j}$  e neste caso, diminuimos o raio e obtemos um novo modelo.

O subproblema (*SP*) é resolvido de forma aproximada no sentido que o ponto obtido forneça uma redução suficiente no valor do modelo. Deste modo, levando em consideração a redução mínima obtida por um passo de Cauchy [34, Teorema 10.1], consideramos que  $d_{k_j}$  é uma **solução aproximada** do subproblema (*SP*) quando

$$\text{pred}_{k_j} \geq \xi \pi_{k_j}(z^k) \min \left\{ \frac{\pi_{k_j}(z^k)}{1 + \|B_{k_j}\|}, \Delta_{k_j} \right\}, \quad (1.12)$$

sendo  $\pi_{k_j}(\cdot)$  a medida de estacionariedade dada em (1.9),  $B_{k_j} \in \mathbb{R}^{n \times n}$  a Hessiana do modelo  $m_{k_j}$ ,  $\Delta_{k_j} > 0$  o raio da região de confiança e  $\xi > 0$  uma constante.

Em algumas iterações  $k_j$  a iteração é incrementada e considerada como insucesso, antes mesmo que um passo de região de confiança  $d_{k_j}$  seja computado. Isto ocorre nas situações em que o raio  $\delta_{k_j}$  está muito grande comparado à medida de estacionariedade  $\pi_{k_j}$ , ou seja, quando  $\delta_{k_j} > \mu \pi_{k_j}(z^k)$  para  $\mu > 1$ . Esta exigência está relacionada com o fato de que os modelos devem ser boas aproximações para a função objetivo em torno de uma vizinhança do ponto restaurado. Note que quando  $\pi_{k_j}(z^k)$  é muito pequeno, o ponto corrente está provavelmente próximo da solução do subproblema (1.8). Por outro lado, se o raio  $\delta_{k_j}$  é consideravelmente grande, não podemos garantir que a função objetivo está bem representada pelo modelo.

O algoritmo sem derivadas da função objetivo com filtro (SD-F) é apresentado a seguir.

---

**Algoritmo 1.1** SD-F

---

Dados:  $x^0 \in \mathbb{R}^n$ ,  $\alpha \in (0, 1)$ ,  $\eta \in (0, 1)$ ,  $\varepsilon > 0$ ,  $\beta > 0$ ,  $\Delta_{\min} > 0$ ,  $\mu > 1$

Considere  $F_0 = \emptyset$ ,  $\mathcal{F}_0 = \emptyset$  e  $k = 0$

REPITA

Defina  $\widehat{F}_k = F_k \cup \{(f^k, h^k)\}$  e

$\widehat{\mathcal{F}}_k = \mathcal{F}_k \cup \mathcal{R}_k$ , sendo  $\mathcal{R}_k$  a região definida em (1.4) ou em (1.5)

Obtenção de  $x^{k+1} \notin \widehat{\mathcal{F}}_k$

FASE DE VIABILIDADE

Se  $h(x^k) = 0$ , então  $z^k = x^k$

Senão, calcule  $z^k \notin \widehat{\mathcal{F}}_k$  satisfazendo (1.6)

Se impossível, então pare sem sucesso

FASE DE OTIMALIDADE

Faça  $j = 0$ , escolha  $\Delta_{k_0} \geq \Delta_{\min}$  e considere  $\delta_{k_0} = \Delta_{k_0}$

REPITA

Construa o modelo  $m_{k_j}$  satisfazendo H4

Se  $\delta_{k_j} > \mu \pi_{k_j}(z^k)$ , então

Se  $\delta_{k_j} \leq \varepsilon$  e  $h(z^k) \leq \varepsilon$ , pare o algoritmo com sucesso

Senão, faça  $d^{k_j} = 0$ ,  $\delta_{k_{j+1}} = \delta_{k_j}/2$  e escolha  $\Delta_{k_{j+1}} \in [\delta_{k_{j+1}}, \Delta_{k_j}]$

Senão

Calcule  $d_{k_j}$  como uma solução aproximada de (1.8)

Se  $z^k + d_{k_j} \notin \widehat{\mathcal{F}}_k$  e  $\text{ared}_{k_j} > \eta \text{pred}_{k_j}$

Faça  $x^{k+1} = z^k + d_{k_j}$ ,  $d_k = d_{k_j}$ ,  $\Delta_k = \Delta_{k_j}$ ,  $\delta_k = \delta_{k_j}$

e pare a fase de otimalidade com sucesso

Senão, faça  $\delta_{k_{j+1}} = \delta_{k_j}/2$  e  $\Delta_{k_{j+1}} = \Delta_{k_j}/2$

$j = j + 1$

*Atualização do filtro*

Se  $f(x^{k+1}) < f(x^k)$ , então

$F_{k+1} = F_k$ ,  $\mathcal{F}_{k+1} = \mathcal{F}_k$  (iteração- $f$ )

Senão,

$F_{k+1} = \widehat{F}_k$ ,  $\mathcal{F}_{k+1} = \widehat{\mathcal{F}}_k$  (iteração- $h$ )

$k = k + 1$

---

No começo de cada iteração, o par  $(f^k, h^k)$  é temporariamente introduzido no filtro através do filtro temporário, e a região proibida temporária é gerada. Depois de completada a iteração, o par  $(f^k, h^k)$  pertencerá permanentemente ao filtro somente quando a iteração não reduziu  $f$ , em outras palavras, quando  $k$  é uma iteração do tipo  $h$ . Quando  $k$  é denominada de iteração- $f$ , a nova entrada  $(f^k, h^k)$  é descartada, o filtro não é atualizado e, conseqüentemente, nem a região proibida. Note que se  $x^k$  é viável, então qualquer ponto  $x$  que não for proibido pelo filtro deve satisfazer  $f(x) < f(x^k)$ . Note também que o método pode adotar tanto o filtro original como o filtro inclinado de acordo com a região  $\mathcal{R}_k$  escolhida. Deste modo, para todo  $x \notin \widehat{\mathcal{F}}_k$ , temos que para cada  $(f^j, h^j) \in \widehat{F}_k$  vale

$$f(x) < f^j - \alpha h^j \quad \text{ou} \quad h(x) < (1 - \alpha)h^j, \quad (1.13)$$

caso seja usado o filtro original e

$$f(x) < f^j - \alpha h(x) \quad \text{ou} \quad h(x) < (1 - \alpha)h^j, \quad (1.14)$$

caso o filtro inclinado esteja em consideração. Os pares de um determinado filtro e suas respectivas regiões proibidas temporárias e permanentes, segundo as regras de filtro (1.4) e (1.5), podem ser vistos na Figura 1.5.

## 1.4 Convergência global

A tarefa desta seção é assegurar a convergência global do Algoritmo 1.1, ou seja, mostrar que a sequência gerada pelo algoritmo tem um ponto de acumulação estacionário. A fim de alcançar este objetivo, primeiramente vamos verificar que o passo de otimalidade é eficiente, no sentido de que perto de um ponto viável não estacionário o decréscimo na função objetivo é relativamente grande. Em seguida, verificamos que esta condição de eficiência é suficiente para estabelecer a convergência do passo todo.

De agora em diante vamos assumir que o algoritmo gerou as sequências infinitas  $(x^k)$  e  $(z^k)$  e que as Hipóteses H1-H4 estão satisfeitas. Dado  $k \in \mathbb{N}$ , considere  $\mathcal{J}_k$  o conjunto de índices das iterações internas do passo de otimalidade que apenas atualizam os modelos sem computar passos de região de confiança, o qual pode ser definido por

$$\mathcal{J}_k = \{j \in \mathbb{N} \mid \delta_{k_j} > \mu \pi_{k_j}(z^k)\}, \quad (1.15)$$

sendo a constante  $\mu > 1$  dada no algoritmo e  $\pi_{k_j}(\cdot)$  definido em (1.9). O lema a seguir verifica que existe uma iteração  $\overline{k}_j$  tal que um passo de região de confiança é calculado em toda iteração interna  $k_j > \overline{k}_j$ .

**Lema 1.3** *Considere  $z^k$  um ponto não estacionário e  $\mu > 1$ . Então o conjunto  $\mathcal{J}_k$  é finito.*

*Demonstração.* Suponha por contradição que o conjunto  $\mathcal{J}_k$  é infinito. Pelo mecanismo do algoritmo temos que  $\delta_{k_j} = \left(\frac{1}{2}\right)^j \delta_{k_0} > 0$  e, conseqüentemente,

$$\delta_{k_j} \longrightarrow 0, \quad (1.16)$$

uma vez que  $\mathcal{J}_k$  é infinito. Assim, pela definição de  $\mathcal{J}_k$ ,

$$\pi_{k_j}(z^k) \xrightarrow{j \in \mathcal{J}_k} 0. \quad (1.17)$$

Por outro lado, como  $z^k$  é não estacionário, por (1.2) temos que existe uma constante  $\bar{c} > 0$  tal que

$$\|\mathcal{P}_{\mathcal{L}(z^k)}(z^k - \nabla f(z^k)) - z^k\| > \bar{c}. \quad (1.18)$$

Por (1.16), existe  $j_0 \in \mathbb{N}$  tal que, para todo  $j > j_0$ , temos  $\delta_{k_j} < \frac{\bar{c}}{2\sigma}$ . Usando a definição de  $\pi_{k_j}(z^k)$  e a desigualdade triangular, temos que para todo  $j \in \mathbb{N}$

$$\begin{aligned} \pi_{k_j}(z^k) &= \|\mathcal{P}_{\mathcal{L}(z^k)}(z^k - g_{k_j}) - z^k\| \\ &= \|\mathcal{P}_{\mathcal{L}(z^k)}(z^k - g_{k_j}) - \mathcal{P}_{\mathcal{L}(z^k)}(z^k - \nabla f(z^k)) + \mathcal{P}_{\mathcal{L}(z^k)}(z^k - \nabla f(z^k)) - z^k\| \\ &\geq -\|\mathcal{P}_{\mathcal{L}(z^k)}(z^k - g_{k_j}) - \mathcal{P}_{\mathcal{L}(z^k)}(z^k - \nabla f(z^k))\| + \|\mathcal{P}_{\mathcal{L}(z^k)}(z^k - \nabla f(z^k)) - z^k\| \end{aligned}$$

Assim, usando a propriedade de contração da projeção [116, Capítulo 3], Hipótese H4 e (1.18) temos que para todo  $j > j_0$

$$\begin{aligned} \pi_{k_j}(z^k) &\geq -\|g_{k_j} - \nabla f(z^k)\| + \|\mathcal{P}_{\mathcal{L}(z^k)}(z^k - \nabla f(z^k)) - z^k\| \\ &> -\sigma\delta_{k_j} + \bar{c} \\ &> \frac{\bar{c}}{2} > 0, \end{aligned}$$

o que contradiz (1.17) e conclui a prova.  $\square$

Cabe ressaltar que a exigência de que os modelos sejam atualizados quando  $\delta_{k_j} > \mu\pi_{k_j}(z^k)$  está relacionada com o fato de que os modelos devem ser boas aproximações para a função objetivo numa vizinhança do ponto restaurado.

Para o próximo resultado é necessária uma condição de qualificação. Embora existam várias condições de qualificação [100, 116], algumas inclusive recentes [3, 5], adotamos aqui a clássica condição de Mangasarian-Fromovitz, definida a seguir.

**Definição 1.4** Dizemos que a condição de qualificação de Mangasarian-Fromovitz é satisfeita em um ponto  $\bar{x}$  quando os gradientes das restrições de igualdade são linearmente independentes e existe um vetor  $d \in \mathbb{R}^n$  tal que

$$\nabla c_i(\bar{x})^t d = 0 \quad e \quad \nabla c_j(\bar{x})^t d < 0,$$

para todo  $i \in \mathcal{E}$  e  $j \in \{\mathcal{I} \mid c_j(\bar{x}) = 0\}$ .

Além das Hipóteses H1 - H4, também assumiremos que a condição a seguir é válida.

**H5** Todo ponto do conjunto viável  $\Omega$  satisfaz a condição de qualificação de Mangasarian-Fromovitz.

O resultado a seguir garante a continuidade de algumas funções auxiliares.

**Lema 1.5** Seja  $\bar{x}$  um ponto satisfazendo a condição de Mangasarian-Fromovitz e  $p : \mathbb{R}^n \rightarrow \mathbb{R}^n$  uma função contínua em  $\bar{x}$ . Então a transformação ponto conjunto  $\mathcal{L}(\cdot)$  e a função  $z \mapsto \mathcal{P}_{\mathcal{L}(z)}(p(z))$  são contínuas em  $\bar{x}$ .

*Demonstração.* Veja [62, Lemas A.1 e A.2]. □

No lema seguinte verificamos que a diferença da medida de inviabilidade ao longo do conjunto linearizado está relacionada com o quadrado da norma do passo.

**Lema 1.6** [62, Lema 3.3] Para todo  $z \in X$  e  $z + d \in \mathcal{L}(z)$ , existe uma constante  $C_1 > 0$  tal que

$$|h(z + d) - h(z)| \leq C_1 \|d\|^2.$$

*Demonstração.* Pela Hipótese H1, temos que para todo  $z \in X$ ,  $z + d \in \mathcal{L}(z)$  e  $i \in \mathcal{E} \cup \mathcal{I}$ , existe uma constante  $C_1 > 0$  tal que

$$c_i(z + d) - c_i(z) \leq \nabla c_i(z)^t d + C_1 \|d\|^2. \quad (1.19)$$

Como  $z + d \in \mathcal{L}(z)$ , decorre diretamente da definição do conjunto linearizado  $\mathcal{L}$  que

$$c_i(z) + \nabla c_i(z)^t d \leq c_i^+(z).$$

Portanto, com isso e (1.19), temos que

$$c_i(z + d) \leq c_i^+(z) + C_1 \|d\|^2.$$

Pela definição da medida de inviabilidade  $h$ , basta provar que

$$c_i^+(z+d) \leq c_i^+(z) + C_1 \|d\|^2. \quad (1.20)$$

Se  $c_i(z+d) < 0$ , para  $i \in \mathcal{I}$ , temos  $c_i^+(z+d) = 0$  e (1.20) vale. Caso contrário,  $c_i^+(z+d) = c_i(z+d)$  e (1.20) é verdadeira. Assim, usando a definição de normas e a definição de  $h$  em (1.20) temos o resultado desejado.  $\square$

O resultado a seguir garante que perto de um ponto viável não estacionário a medida de estacionariedade do subproblema é suficientemente grande.

**Lema 1.7** *Seja  $\bar{x} \in X$  um ponto viável não estacionário,  $\bar{c} = \frac{1}{4} \|\mathcal{P}_{\mathcal{L}(\bar{x})}(\bar{x} - \nabla f(\bar{x})) - \bar{x}\| > 0$  e  $\sigma > 0$  dado por H4. Então existe uma vizinhança  $V_1$  de  $\bar{x}$  tal que para todo  $z^k \in V_1$  e  $j \in \mathbb{N}$*

$$\pi_{k_j}(z^k) > -\sigma \delta_{k_j} + \frac{\bar{c}}{2}.$$

*Demonstração.* Considere  $\tilde{V}_1 = B\left(\bar{x}, \frac{\bar{c}}{2}\chi\right)$ , sendo  $\chi = \min\{1, 1/\gamma\}$  e  $\gamma > 0$  dada por H4. Então, para todo  $x \in \tilde{V}_1$ ,  $z^k \in \tilde{V}_1$  e  $k, j \in \mathbb{N}$ ,

$$\|\nabla m_{k_j}(\bar{x}) - \nabla m_{k_j}(x)\| = \|B_{k_j}(\bar{x} - x)\| < \frac{\bar{c}}{2}. \quad (1.21)$$

Dado  $v \in \mathbb{R}^n$ , a  $\mathcal{P}_{\mathcal{L}(\cdot)}(v)$  é contínua em  $\bar{x}$  pelo Lema 1.5. Consequentemente, existe uma vizinhança  $\tilde{V}_2$  de  $\bar{x}$  tal que para todo  $z^k \in \tilde{V}_2$ ,

$$\|\mathcal{P}_{\mathcal{L}(z^k)}(v) - \mathcal{P}_{\mathcal{L}(\bar{x})}(v)\| < \frac{\bar{c}}{2}. \quad (1.22)$$

Como  $\|\mathcal{P}_{\mathcal{L}(\bar{x})}(\bar{x} - \nabla f(\cdot)) - \bar{x}\|$  é contínua em  $\bar{x}$ , existe uma vizinhança  $\tilde{V}_3$  de  $\bar{x}$  tal que para todo  $z^k \in \tilde{V}_3$

$$\|\mathcal{P}_{\mathcal{L}(\bar{x})}(\bar{x} - \nabla f(z^k)) - \bar{x}\| > \frac{3}{4} \|\mathcal{P}_{\mathcal{L}(\bar{x})}(\bar{x} - \nabla f(\bar{x})) - \bar{x}\| = 3\bar{c}. \quad (1.23)$$

Considere  $V_1 = \tilde{V}_1 \cap \tilde{V}_2 \cap \tilde{V}_3$ . Usando a definição de  $\pi_{k_j}$  dada em (1.9), a desigualdade triangular duas vezes e somando e subtraindo  $(\mathcal{P}_{\mathcal{L}(z^k)}(\bar{x} - \nabla m_{k_j}(\bar{x})) - \bar{x})$ , segue que, para todo  $z^k \in V_1$ ,

$$\begin{aligned} & |\pi_{k_j}(z^k) - \pi_{k_j}(\bar{x})| = \\ & = \left| \|\mathcal{P}_{\mathcal{L}(z^k)}(z^k - \nabla m_{k_j}(z^k)) - z^k\| - \|\mathcal{P}_{\mathcal{L}(\bar{x})}(\bar{x} - \nabla m_{k_j}(\bar{x})) - \bar{x}\| \right| \\ & \leq \left\| \mathcal{P}_{\mathcal{L}(z^k)}(z^k - \nabla m_{k_j}(z^k)) - z^k - \mathcal{P}_{\mathcal{L}(\bar{x})}(\bar{x} - \nabla m_{k_j}(\bar{x})) + \bar{x} \right\| \\ & = \left\| \mathcal{P}_{\mathcal{L}(z^k)}(z^k - \nabla m_{k_j}(z^k)) - z^k - \mathcal{P}_{\mathcal{L}(z^k)}(\bar{x} - \nabla m_{k_j}(\bar{x})) + \bar{x} + \right. \\ & \quad \left. + \mathcal{P}_{\mathcal{L}(z^k)}(\bar{x} - \nabla m_{k_j}(\bar{x})) - \bar{x} - \mathcal{P}_{\mathcal{L}(\bar{x})}(\bar{x} - \nabla m_{k_j}(\bar{x})) + \bar{x} \right\| \\ & \leq \left\| \mathcal{P}_{\mathcal{L}(z^k)}(z^k - \nabla m_{k_j}(z^k)) - z^k - \mathcal{P}_{\mathcal{L}(z^k)}(\bar{x} - \nabla m_{k_j}(\bar{x})) + \bar{x} \right\| + \\ & \quad + \left\| \mathcal{P}_{\mathcal{L}(z^k)}(\bar{x} - \nabla m_{k_j}(\bar{x})) - \mathcal{P}_{\mathcal{L}(\bar{x})}(\bar{x} - \nabla m_{k_j}(\bar{x})) \right\| \end{aligned}$$

Usando a contração da projeção no resultado prévio, temos que para todo  $z^k \in V_1$

$$\begin{aligned} & \left| \pi_{k_j}(z^k) - \pi_{k_j}(\bar{x}) \right| = \\ & \leq \left\| \nabla m_{k_j}(\bar{x}) - \nabla m_{k_j}(z^k) \right\| + \left\| \mathcal{P}_{\mathcal{L}(z^k)}(\bar{x} - \nabla m_{k_j}(\bar{x})) - \mathcal{P}_{\mathcal{L}(\bar{x})}(\bar{x} - \nabla m_{k_j}(\bar{x})) \right\| \\ & \leq \left\| \nabla m_{k_j}(\bar{x}) - \nabla m_{k_j}(z^k) \right\| + \left\| \mathcal{P}_{\mathcal{L}(z^k)}(\bar{x} - \nabla m_{k_j}(\bar{x})) - \mathcal{P}_{\mathcal{L}(\bar{x})}(\bar{x} - \nabla m_{k_j}(\bar{x})) \right\| + \\ & \quad + 2 \left\| \bar{x} - z^k \right\|. \end{aligned}$$

Com isto, (1.21) e (1.22)

$$\pi_{k_j}(z^k) > \pi_{k_j}(\bar{x}) - 2\bar{c}. \quad (1.24)$$

Por outro lado, usando novamente a definição de  $\pi_{k_j}$ , a definição de  $m_{k_j}$  dada em (1.7) e a desigualdade triangular temos que

$$\begin{aligned} \pi_{k_j}(\bar{x}) &= \left\| \mathcal{P}_{\mathcal{L}(\bar{x})}(\bar{x} - g_{k_j} - B_{k_j}(\bar{x} - z^k)) - \bar{x} \right\| \\ &= \left\| \mathcal{P}_{\mathcal{L}(\bar{x})}(\bar{x} - g_{k_j} - B_{k_j}(\bar{x} - z^k)) - \bar{x} - (\mathcal{P}_{\mathcal{L}(\bar{x})}(\bar{x} - \nabla f(z^k)) - \bar{x}) + \right. \\ & \quad \left. + (\mathcal{P}_{\mathcal{L}(\bar{x})}(\bar{x} - \nabla f(z^k)) - \bar{x}) \right\| \\ &\geq - \left\| \mathcal{P}_{\mathcal{L}(\bar{x})}(\bar{x} - g_{k_j} - B_{k_j}(\bar{x} - z^k)) - \mathcal{P}_{\mathcal{L}(\bar{x})}(\bar{x} - \nabla f(z^k)) \right\| + \left\| \mathcal{P}_{\mathcal{L}(\bar{x})}(\bar{x} - \nabla f(z^k)) - \bar{x} \right\|. \end{aligned}$$

Pela propriedade de contração da projeção e desigualdade triangular segue que

$$\begin{aligned} \pi_{k_j}(\bar{x}) &\geq - \left\| g_{k_j} + B_{k_j}(\bar{x} - z^k) - \nabla f(z^k) \right\| + \left\| \mathcal{P}_{\mathcal{L}(\bar{x})}(\bar{x} - \nabla f(z^k)) - \bar{x} \right\| \\ &\geq - \left\| g_{k_j} - \nabla f(z^k) \right\| - \left\| B_{k_j}(\bar{x} - z^k) \right\| + \left\| \mathcal{P}_{\mathcal{L}(\bar{x})}(\bar{x} - \nabla f(z^k)) - \bar{x} \right\|. \end{aligned}$$

Assim, usando a Hipótese H4, propriedade de norma, (1.23) e o fato que  $z^k \in V_1 \subset \tilde{V}_1$ , temos que

$$\pi_{k_j}(\bar{x}) > -\sigma\delta_{k_j} + 3\bar{c} - \gamma \left\| \bar{x} - z^k \right\| \geq -\sigma\delta_{k_j} + \frac{5}{2}\bar{c}.$$

Usando isto em (1.24) obtemos o resultado desejado.  $\square$

Segue imediatamente deste lema que a medida de estacionariedade do subproblema em torno de um ponto viável não estacionário é limitada inferiormente por uma constante positiva.

**Corolário 1.8** *Considere um ponto viável não estacionário  $\bar{x} \in X$  e a vizinhança  $V_1$  dada pelo Lema 1.7. Então existem constantes  $\hat{\delta} \in (0, \Delta_{\min})$  e  $C_2 > 0$ , tais que para todo  $z^k \in V_1$  e  $j \notin \mathcal{J}_k$  ou  $j \in \mathbb{N}$  com  $\delta_{k_j} \leq \hat{\delta}$ , vale*

$$\pi_{k_j}(z^k) > C_2.$$

*Demonstração.* Seja  $z^k \in V_1$ . Considere inicialmente que  $j \notin \mathcal{J}_k$ . Pelo Lema 1.7 e a definição de  $\mathcal{J}_k$  dada em (1.15), segue que

$$\pi_{k_j}(z^k) > -\sigma\delta_{k_j} + \frac{\bar{c}}{2} \geq -\sigma\mu\pi_{k_j}(z^k) + \frac{\bar{c}}{2}.$$

Então

$$\pi_{k_j}(z^k) > \frac{\bar{c}}{2(1 + \sigma\mu)}.$$

Considere agora  $\hat{\delta} = \min \left\{ \frac{\bar{c}}{4\sigma}, \frac{\Delta_{\min}}{2} \right\} > 0$  e  $j \in \mathbb{N}$  tal que  $\delta_{k_j} \leq \hat{\delta}$ . Pelo Lema 1.7

$$\pi_{k_j}(z^k) > -\sigma\hat{\delta} + \frac{\bar{c}}{2} \geq \frac{\bar{c}}{4}.$$

Tomando  $C_2 = \min \left\{ \frac{1}{2(1 + \sigma\mu)}, \frac{1}{4} \right\} \bar{c} > 0$ , completamos a prova.  $\square$

O próximo lema estabelece que, próximo de um ponto viável não estacionário, todo ponto obtido com um raio suficientemente pequeno será aceito pelo critério de região de confiança.

**Lema 1.9** *Considere um ponto viável não estacionário  $\bar{x} \in X$ , a vizinhança  $V_1$  dada pelo Lema 1.7. Então existem constantes  $\bar{\Delta} \in (0, \Delta_{\min})$  e  $C_3 > 0$  tais que para todo  $z^k \in V_1$  e  $j \notin \mathcal{J}_k$ ,*

$$\text{pred}_{k_j} > C_3 \min \{ \Delta_{k_j}, \bar{\Delta} \} \quad (1.25)$$

e

$$\text{ared}_{k_j} > \eta \text{pred}_{k_j} > \eta C_3 \Delta_{k_j}, \quad \text{para } \Delta_{k_j} \in (0, \bar{\Delta}]. \quad (1.26)$$

*Demonstração.* Considere  $z^k \in V_1$  e  $j \notin \mathcal{J}_k$ . Então, pela condição (1.12), Hipótese H4, Corolário 1.8 e o fato que  $j \notin \mathcal{J}_k$  segue que

$$\text{pred}_{k_j} \geq \xi \pi_{k_j}(z^k) \min \left\{ \frac{\pi_{k_j}(z^k)}{1 + \|B_{k_j}\|}, \Delta_{k_j} \right\} > \xi C_2 \min \left\{ \frac{C_2}{1 + \gamma}, \Delta_{k_j} \right\}.$$

Assim, considerando

$$\bar{\Delta} = \min \left\{ \frac{\Delta_{\min}}{2}, \frac{C_2}{1 + \gamma} \right\} \in (0, \Delta_{\min})$$

e  $C_3 = C_2\xi > 0$ , obtemos (1.25).

Agora vamos verificar a validade de (1.26). Considere  $z^k \in V_1$ ,  $j \notin \mathcal{J}_k$  com  $\Delta_{k_j} \in (0, \bar{\Delta}]$  e  $d_{k_j}$  o passo de região de confiança obtido na iteração interna  $k_j$ . Pela Hipótese H1 e pelo Teorema do Valor Médio, existe uma constante  $\theta \in (0, 1)$  tal que

$$f(z^k) - f(z^k + d_{k_j}) = -\nabla f(z^k + \theta d_{k_j})^t d_{k_j}.$$

Usando as definições das reduções  $\text{ared}_{k_j}$  e  $\text{pred}_{k_j}$ , o resultado anterior e as definições dos modelos  $m_{k_j}$ , obtemos que

$$\begin{aligned} \left| \text{ared}_{k_j} - \text{pred}_{k_j} \right| &= \left| f(z^k) - f(z^k + d_{k_j}) - (m_{k_j}(z^k) - m_{k_j}(z^k + d_{k_j})) \right| \\ &= \left| -\nabla f(z^k + \theta d_{k_j})^t d_{k_j} + g_{k_j}^t d_{k_j} + \frac{1}{2} d_{k_j}^t B_{k_j} d_{k_j} \right|. \end{aligned}$$

Com isso, a desigualdade de Cauchy-Schwarz, propriedades de norma e a condição H3, segue que

$$\begin{aligned} & \left| \text{ared}_{k_j} - \text{pred}_{k_j} \right| \leq \\ & \leq \left( \left\| \nabla f(z^k + \theta d_{k_j}) - \nabla f(z^k) \right\| + \left\| \nabla f(z^k) - g_{k_j} \right\| + \frac{1}{2} \|B_{k_j}\| \|d_{k_j}\| \right) \|d_{k_j}\| \\ & \leq \left( L\theta \|d_{k_j}\| + \left\| \nabla f(z^k) - g_{k_j} \right\| + \frac{1}{2} \|B_{k_j}\| \|d_{k_j}\| \right) \|d_{k_j}\|. \end{aligned}$$

Portanto, pela Hipótese H4 e pelos fatos que  $\|d_{k_j}\| \leq \Delta_{k_j}$  e  $\Delta_{k_j} \geq \delta_{k_j}$ , temos que

$$\left| \text{ared}_{k_j} - \text{pred}_{k_j} \right| \leq \left( L\theta + \sigma + \frac{1}{2}\gamma \right) \Delta_{k_j}^2. \quad (1.27)$$

Redefinindo  $\bar{\Delta} \in (0, \Delta_{\min})$  como

$$\bar{\Delta} = \min \left\{ \frac{\Delta_{\min}}{2}, \frac{C_2}{1 + \gamma}, \frac{C_2(1 - \eta)}{(L\theta + \sigma + \frac{1}{2}\gamma)} \right\} > 0,$$

usando (1.25), (1.27) e o fato que  $\Delta_{k_j} \in (0, \bar{\Delta}]$ , temos que

$$\left| \frac{\text{ared}_{k_j}}{\text{pred}_{k_j}} - 1 \right| = \left| \frac{\text{ared}_{k_j} - \text{pred}_{k_j}}{\text{pred}_{k_j}} \right| < \frac{(L\theta + \sigma + \frac{1}{2}\gamma)}{C_2} \Delta_{k_j} \leq 1 - \eta.$$

Desta maneira, para todo  $z^k \in V_1$  e  $\Delta_{k_j} \in (0, \bar{\Delta}]$ , obtemos que

$$\text{ared}_{k_j} > \eta \text{pred}_{k_j}.$$

Usando isto e (1.25) obtemos a segunda desigualdade de (1.26), o que completa a prova.  $\square$

O lema a seguir garante que, perto de um ponto viável não estacionário, a recusa de um passo de otimalidade é devida a um grande aumento da inviabilidade. Para tanto, defina a folga do filtro por

$$H_k = \min \left\{ 1, \min \left\{ (1 - \alpha)h^j \mid (f^j, h^j) \in F_k \text{ e } f^j \leq f^k \right\} \right\}. \quad (1.28)$$

A Figura 1.6 ilustra a relação entre  $H_k$  e a região proibida temporária  $\widehat{\mathcal{F}}_k = \mathcal{F}_k \cup \mathcal{R}_k$  quando o critério de filtro adotado é o filtro original. Esta relação é análoga para o critério de filtro inclinado. Além disso, note que pelas Hipóteses H1, H2 e pelo Teorema do Valor Médio de Lagrange, existe uma constante  $\tilde{L} > 0$  tal que

$$\left| f(z^k) - f(x^k) \right| \leq \tilde{L} \|z^k - x^k\|.$$

Usando isto e (1.6), temos que existe uma constante  $C_4 > 0$  tal que

$$\left| f(z^k) - f(x^k) \right| \leq C_4 h(x^k). \quad (1.29)$$

**Lema 1.10** *Seja  $\bar{x} \in X$  um ponto viável não estacionário. Considere as constantes dadas pelos Lemas 1.6 e 1.9 e pelo Corolário 1.8 e a vizinhança  $V_1$  dada pelo Lema 1.7. Defina as constantes  $C_5 = \frac{1}{2} \min \left\{ \frac{1}{\sqrt{C_1}}, \mu C_2 \right\} > 0$  e  $\Delta' \in (0, \Delta_{\min})$  por*

$$\Delta' = \min \left\{ \widehat{\delta}, \overline{\Delta}, \frac{\eta C_3}{8\alpha C_1}, \eta C_3 \left( \frac{1-\alpha}{C_5^2} + C_1\alpha + \frac{C_4}{\alpha C_5^2} \right)^{-1} \right\}. \quad (1.30)$$

*Então existe uma vizinhança  $V_2 \subset V_1$  de  $\bar{x}$  tal que para todo  $x^k \in V_2$  e  $\Delta_{k_j} \in (\Delta'/2, \Delta']$  com  $j \notin \mathcal{J}_k$ , temos que  $z^k \in V_1$  e*

$$f(z^k + d_{k_j}) + \alpha h(x^k) < f(x^k) \quad \text{para o filtro original} \quad (1.31)$$

e

$$f(z^k + d_{k_j}) + \alpha h(z^k + d_{k_j}) < f(x^k) \quad \text{para o filtro inclinado.} \quad (1.32)$$

*Além disso, se o passo  $d_{k_j}$  é recusado pelo algoritmo, então*

$$h(z^k + d_{k_j}) \geq H_k.$$

*Demonstração.* Como  $\bar{x}$  é um ponto viável,  $h$  é uma função contínua e  $\|z^k - x^k\| \leq \beta h(x^k)$ , então existe uma vizinhança  $V_2 \subset V_1$  de  $\bar{x}$  tal que para todo  $x^k \in V_2$  temos que  $z^k \in V_1$  e

$$h(x^k) \leq \frac{\eta C_3 \Delta'}{2} \min \left\{ \frac{1}{\alpha + C_4}, \frac{1}{2C_4}, \frac{1}{4\alpha(1-\alpha)} \right\}. \quad (1.33)$$

Considere  $x^k \in V_2$ ,  $z^k \in V_1$  e  $j \notin \mathcal{J}_k$  com  $\Delta_{k_j} \in \left( \frac{\Delta'}{2}, \Delta' \right] \subset (0, \overline{\Delta}]$ . Como  $\frac{\Delta'}{2} < \Delta_{k_j} \leq \overline{\Delta}$ , pelo Lema 1.9, segue que

$$f(z^k) - f(z^k + d_{k_j}) > \eta C_3 \Delta_{k_j} > \frac{\eta C_3 \Delta'}{2}.$$

Assim, usando isto e (1.29) obtemos que

$$f(x^k) - f(z^k + d_{k_j}) = f(x^k) - f(z^k) + f(z^k) - f(z^k + d_{k_j}) > -C_4 h(x^k) + \frac{\eta C_3 \Delta'}{2}. \quad (1.34)$$

Usando (1.33) segue que

$$f(x^k) - f(z^k + d_{k_j}) > -C_4 h(x^k) + (\alpha + C_4) h(x^k) = \alpha h(x^k),$$

o que implica o resultado (1.31) para o critério de filtro original.

Agora vamos verificar (1.32) para o filtro inclinado. Usando (1.33) em (1.34) temos que

$$f(x^k) - f(z^k + d_{k_j}) > -C_4 \frac{\eta C_3 \Delta'}{4C_4} + \frac{\eta C_3 \Delta'}{2} = \frac{\eta C_3 \Delta'}{4}. \quad (1.35)$$

Por outro lado, pelo Lema 1.6 temos que  $|h(z^k + d_{k_j}) - h(z^k)| \leq C_1 \Delta_{k_j}^2$  e pelo mecanismo do Algoritmo 1.1, que  $h(z^k) < (1 - \alpha)h(x^k)$ . Portanto, usando estes resultados e o fato que  $\Delta_{k_j} \leq \Delta'$ , segue que

$$h(z^k + d_{k_j}) \leq h(z^k) + C_1 \Delta_{k_j}^2 < (1 - \alpha)h(x^k) + C_1 \Delta'^2.$$

Multiplicando por  $\alpha > 0$  e usando (1.30) e (1.33), obtemos que

$$\alpha h(z^k + d_{k_j}) < \alpha(1 - \alpha) \frac{\eta C_3 \Delta'}{8\alpha(1 - \alpha)} + C_1 \alpha \Delta'^2 = \frac{\eta C_3 \Delta'}{8} + C_1 \alpha \Delta'^2 < \frac{\eta C_3}{4} \Delta'.$$

Assim, combinando isto com (1.35), temos que

$$f(x^k) - f(z^k + d_{k_j}) > \alpha h(z^k + d_{k_j}),$$

o que prova (1.32) para o filtro inclinado.

Para completar a prova, suponha que o ponto  $z^k + d_{k_j}$  foi recusado pelo algoritmo. Como  $\Delta_{k_j} \leq \bar{\Delta}$  e  $j \notin \mathcal{J}_k$ , o Lema 1.9 garante que este ponto foi aceito pelo critério de região de confiança e, conseqüentemente, foi recusado pelo critério de filtro. Assim,

$$z^k + d_{k_j} \in \widehat{\mathcal{F}}_k.$$

Portanto, como (1.31) vale para o critério de filtro original e (1.32) vale para o critério de filtro inclinado, temos, pela definição do filtro e de  $H_k$ , dada em (1.28), que

$$h(z^k + d_{k_j}) \geq H_k,$$

e a prova está concluída.  $\square$

O próximo resultado estabelece que, perto de um ponto viável não estacionário, a fase de otimalidade fornece um decréscimo suficiente no valor da função objetivo, comparado com o valor obtido no ponto restaurado.

**Teorema 1.11** *Dado um ponto  $\bar{x} \in X$  viável não estacionário, existem constantes  $C_6, C_7 > 0$  e uma vizinhança  $V_3$  de  $\bar{x}$  tais que, para qualquer  $x^k \in V_3$ , o ponto  $x^{k+1}$  obtido pelo Algoritmo 1.1 satisfaz*

$$f(z^k) - f(x^{k+1}) \geq C_6 \sqrt{H_k} \tag{1.36}$$

e

$$f(z^k) - f(x^{k+1}) \geq C_7 \|z^k - x^{k+1}\|, \tag{1.37}$$

sendo  $H_k > 0$  a folga do filtro definida em (1.28).

*Demonstração.* Primeiramente, cabem ressaltar alguns resultados imediatos. Pelo mecanismo do algoritmo segue que

$$f(z^k) - f(x^{k+1}) = \text{ared}_k > \eta \text{pred}_k, \quad (1.38)$$

sendo  $\text{ared}_k := \text{ared}_{k_j}$  e  $\text{pred}_k := \text{pred}_{k_j}$  para  $j \in \mathbb{N}$  tal que  $d_{k_j} = d_k$ . Além disso, como  $z^k + d_{k_j} \in \mathcal{L}(z^k)$ , pelo Lema 1.6, temos que para todo  $\Delta_{k_j} > 0$  existe uma constante  $C_1 > 0$  tal que

$$|h(z^k + d_{k_j}) - h(z^k)| \leq C_1 \Delta_{k_j}^2. \quad (1.39)$$

Considere a constante  $\bar{\Delta} > 0$  dada pelo Lema 1.9, a constante  $\Delta' \leq \bar{\Delta}$  e a vizinhança  $V_2$  de  $\bar{x}$  dadas pelo Lema 1.10. Seja  $V_3 \subset V_2$  uma vizinhança de  $\bar{x}$  tal que  $h(x^k) < 1$  e  $z^k \in V_2$ , para todo  $x^k \in V_3$ . Esta vizinhança está bem definida uma vez que  $h(\bar{x}) = 0$ ,  $h(\cdot) \geq 0$  é uma função contínua e (1.6) vale.

Considere  $x^k \in V_3$  e, conseqüentemente,  $z^k \in V_2$ . Pela fase de viabilidade, a definição de  $H_k$  e o fato que  $h(x^k) < 1$ , temos que

$$h(z^k) < (1 - \alpha)h(x^k) < (1 - \alpha)H_k. \quad (1.40)$$

Denote  $j^*$  o índice da iteração interna de sucesso, ou seja, tal que  $\Delta_k = \Delta_{k_j^*}$ . Note que pelo mecanismo do algoritmo,  $j^* \notin \mathcal{J}_k$ . Vamos considerar alguns casos ao longo da demonstração conforme representados na Tabela 1.1.

<i>Primeiro caso:</i> $\Delta_k > \frac{\Delta'}{2}$	<i>Segundo caso:</i> $\Delta_k \leq \frac{\Delta'}{2}$
$h(z^k + d_{k_j}) \geq H_k,$ $\forall \Delta_{k_j} \leq (\Delta'/2)$ com $j \notin \mathcal{J}_k$	$\exists j \notin \mathcal{J}_k$ com $\Delta_{k_j} \leq (\Delta'/2)$ tal que $h(z^k + d_{k_j}) < H_k$

Tabela 1.1: Casos considerados na demonstração do Teorema 1.11.

*Primeiro caso:* suponha que  $\Delta_k > \frac{\Delta'}{2}$ . Pelo fato que  $j^* \notin \mathcal{J}_k$ , pelo Lema 1.9 e pelo fato que  $\frac{\Delta'}{2} < \min\{\Delta_k, \bar{\Delta}\}$  temos que

$$\text{pred}_k > C_3 \frac{\Delta'}{2}.$$

Aplicando isto em (1.38), segue que

$$f(z^k) - f(x^{k+1}) > \frac{1}{2} \eta C_3 \Delta'. \quad (1.41)$$

As sequências  $(\sqrt{H_k})_{k \in \mathbb{N}}$  e  $(\|z^k - x^{k+1}\|)_{k \in \mathbb{N}}$  são limitadas, pois para todo  $k \in \mathbb{N}$  temos que  $0 < H_k \leq 1$  e  $(\Delta_k)_{k \in \mathbb{N}}$  é uma sequência positiva não crescente. Assim, existem constantes  $C_6 > 0$  e  $C_7 > 0$  tais que

$$\frac{1}{2}\eta C_3 \Delta' \geq C_6 \sqrt{H_k} \quad \text{e} \quad \frac{1}{2}\eta C_3 \Delta' \geq C_7 \|z^k - x^{k+1}\|.$$

Aplicando estas relações em (1.41), temos (1.36) e (1.37).

*Segundo caso:* suponha que

$$\Delta_k \leq \frac{\Delta'}{2}. \quad (1.42)$$

Pelo Lema 1.3 e pelo mecanismo do algoritmo, existe pelo menos um índice  $j \in \mathbb{N}$  tal que  $j \notin \mathcal{J}_k$ . Além disso, os passos de região de confiança são calculados somente nas iterações internas  $j \notin \mathcal{J}_k$  e  $x^{k+1} = z^k + d_{k_j^*}$  com  $j^* \notin \mathcal{J}_k$ . Assim, é suficiente verificar o resultado para  $j \notin \mathcal{J}_k$ . Vamos novamente considerar duas situações.

*Primeiro subcaso:* suponha que a condição

$$h(z^k + d_{k_j}) \geq H_k \quad (1.43)$$

vale para todo  $\Delta_{k_j} \leq \frac{\Delta'}{2}$  com  $j \notin \mathcal{J}_k$ . Assim, por (1.42) temos que (1.43) vale em particular para  $j^*$ . Portanto, por (1.40) e (1.43) obtemos que

$$h(z^k + d_k) - h(z^k) > \alpha H_k.$$

Usando isto e (1.39) segue que

$$\alpha H_k < h(z^k + d_k) - h(z^k) \leq C_1 \Delta_k^2. \quad (1.44)$$

De (1.38), Lema 1.9 e do fato que  $\Delta_k \leq \frac{\Delta'}{2} < \bar{\Delta}$ , obtemos que

$$f(z^k) - f(x^{k+1}) > \eta C_3 \Delta_k. \quad (1.45)$$

Usando isto, (1.44) e considerando  $C_6 = \eta C_3 \sqrt{\frac{\alpha}{C_1}} > 0$ , obtemos (1.36). Por outro lado, usando o fato que  $\|z^k - x^{k+1}\| \leq \Delta_k$  em (1.45) e considerando  $C_7 = \eta C_3 > 0$ , obtemos (1.37), como desejado.

*Segundo subcaso:* suponha agora que existe  $j \notin \mathcal{J}_k$  com  $\Delta_{k_j} \leq \frac{\Delta'}{2}$  tal que (1.43) não vale. Considere  $\hat{j} \notin \mathcal{J}_k$  o primeiro índice com  $\Delta_{k_{\hat{j}}} \leq \frac{\Delta'}{2}$  tal que (1.43) falha. Denote  $\hat{\Delta} = \Delta_{k_{\hat{j}}}$  e  $\hat{x} = z^k + d_{k_{\hat{j}}}$ .

Em primeiro lugar vamos limitar o raio  $\widehat{\Delta}$  com respeito à folga do filtro  $H_k$ . Note que, como

$$2\widehat{\Delta} \leq \Delta' \leq \overline{\Delta} < \Delta_{\min} \quad \text{e} \quad \Delta_{k_0} > \Delta_{\min},$$

temos que o raio  $2\widehat{\Delta}$  da iteração interna  $\widehat{j} - 1$  foi recusado pelo algoritmo. Suponha que  $\widehat{j} - 1 \notin \mathcal{J}_k$ . Neste caso,  $\Delta_{k_{\widehat{j}-1}} = 2\widehat{\Delta}$  e (1.43) vale em  $2\widehat{\Delta}$ . De fato, se  $2\widehat{\Delta} \leq \frac{\Delta'}{2}$ , isto segue imediatamente da definição de  $\widehat{\Delta}$  e do fato que  $\widehat{j} - 1 \notin \mathcal{J}_k$ . Por outro lado, se  $2\widehat{\Delta} \in \left(\frac{\Delta'}{2}, \Delta'\right]$ , o resultado é decorrente do Lema 1.10. Assim, por (1.39), (1.40) e (1.43) temos que

$$4C_1\widehat{\Delta}^2 \geq h\left(z^k + d_{k_{\widehat{j}-1}}\right) - h(z^k) > H_k - (1 - \alpha)H_k = \alpha H_k.$$

Portanto,

$$\widehat{\Delta} > \frac{1}{2\sqrt{C_1}}\sqrt{\alpha H_k}. \quad (1.46)$$

Suponha agora que  $\widehat{j} - 1 \in \mathcal{J}_k$ . Assim, pelas definições de  $\widehat{\Delta}$  e de  $\mathcal{J}_k$ , pelo mecanismo do algoritmo e pelos fatos que  $\sqrt{H_k} \leq 1$  e  $\alpha \in (0, 1)$ , temos que

$$2\widehat{\Delta} = \Delta_{k_{\widehat{j}-1}} \geq \delta_{k_{\widehat{j}-1}} > \mu\pi_{k_{\widehat{j}-1}}(z^k) > \mu\sqrt{\alpha H_k}\pi_{k_{\widehat{j}-1}}(z^k).$$

Usando isto e o fato que  $\delta_{k_{\widehat{j}-1}} \leq 2\widehat{\Delta} \leq \Delta' \leq \widehat{\delta}$  temos, pelo Corolário 1.8, que

$$\widehat{\Delta} > \frac{\mu C_2}{2}\sqrt{\alpha H_k}. \quad (1.47)$$

Considere  $C_5 > 0$  a constante dada pelo Lema 1.10. Combinando a definição de  $C_5$  com (1.46) e (1.47), temos que, em ambos os casos  $\widehat{j} - 1 \notin \mathcal{J}_k$  e  $\widehat{j} - 1 \in \mathcal{J}_k$ ,

$$\widehat{\Delta} > C_5\sqrt{\alpha H_k}. \quad (1.48)$$

Por outro lado, usando o fato que  $\widehat{j} \notin \mathcal{J}_k$ , o Lema 1.9 e o fato que  $\widehat{\Delta} < \overline{\Delta}$ , resulta que

$$f(z^k) - f(\widehat{x}) > \eta C_3 \widehat{\Delta}. \quad (1.49)$$

Usando (1.48) em (1.49) e considerando  $C_6 = \eta C_3 C_5 \sqrt{\alpha} > 0$ , segue que

$$f(z^k) - f(\widehat{x}) > C_6 \sqrt{H_k}. \quad (1.50)$$

Combinando  $\|z^k - \widehat{x}\| \leq \widehat{\Delta}$  com (1.49) e tomando  $C_7 = \eta C_3 > 0$  temos que

$$f(z^k) - f(\widehat{x}) > C_7 \|z^k - \widehat{x}\|.$$

Portanto, (1.36) e (1.37) valem em  $\hat{x}$ . Para finalizar a prova é suficiente verificar que  $x^{k+1} = \hat{x}$ . Como  $\hat{j} \notin \mathcal{J}_k$  e  $\hat{\Delta} < \bar{\Delta}$ , o Lema 1.9 garante que

$$\text{ared}_{k_{\hat{j}}} > \eta \text{pred}_{k_{\hat{j}}}.$$

Assim, o ponto  $\hat{x}$  satisfaz o critério de região de confiança. Além disso, como (1.43) não vale em  $\hat{x}$ , para que  $x^{k+1}$  seja igual a  $\hat{x}$  é suficiente conferir que

$$f(\hat{x}) < f(x^k) - \alpha h(x^k) \quad (1.51)$$

para o filtro original e

$$f(\hat{x}) < f(x^k) - \alpha h(\hat{x})$$

para o filtro inclinado, pois desta maneira teremos que  $\hat{x} \notin \hat{\mathcal{F}}_k$  e, conseqüentemente,  $x^{k+1} = \hat{x}$ . Vamos verificar primeiramente o resultado para o filtro original. Por (1.29) e (1.50), temos que

$$f(\hat{x}) < f(z^k) - C_6 \sqrt{H_k} \leq f(x^k) + C_4 h(x^k) - C_6 \sqrt{H_k}. \quad (1.52)$$

Se  $h(x^k) = 0$ ,

$$f(\hat{x}) < f(x^k) - C_6 \sqrt{H_k} < f(x^k) = f(x^k) - \alpha h(x^k)$$

e neste caso  $x^{k+1} = \hat{x}$ . Suponha agora que  $h(x^k) > 0$ . Como  $h(x^k) < H_k$ , pois  $h(x^k) < 1$ , temos por (1.52) que

$$f(\hat{x}) < f(x^k) + C_4 h(x^k) - C_6 \sqrt{h(x^k)} = f(x^k) + \left( C_4 - \frac{C_6}{\sqrt{h(x^k)}} \right) h(x^k). \quad (1.53)$$

Uma vez que  $h$  é uma função contínua e  $\bar{x}$  é um ponto viável, podemos restringir  $V_3$ , se necessário, tal que para todo  $x^k \in V_3$ , temos que

$$\sqrt{h(x^k)} < \frac{C_6}{C_4 + \alpha}.$$

Assim, combinando isto com (1.53) temos (1.51) e  $x^{k+1} = \hat{x}$ , como desejado.

Vamos verificar que  $x^{k+1} = \hat{x}$  quando o critério de filtro inclinado é considerado.

Por (1.39) e (1.40),

$$h(\hat{x}) \leq h(z^k) + C_1 \hat{\Delta}^2 < (1 - \alpha) H_k + C_1 \hat{\Delta}^2.$$

Usando (1.48) obtemos que

$$h(\hat{x}) < \left( \frac{(1 - \alpha)}{\alpha C_5^2} + C_1 \right) \hat{\Delta}^2. \quad (1.54)$$

Por outro lado, combinando o fato que  $H_k \geq h(x^k)$  com (1.29) e (1.48) segue que

$$|f(z^k) - f(x^k)| \leq C_4 H_k < \frac{C_4}{\alpha C_5^2} \widehat{\Delta}^2. \quad (1.55)$$

Assim, (1.49), (1.55) e o fato que  $\widehat{\Delta} < \Delta'$  garantem que

$$f(x^k) - f(\widehat{x}) = f(x^k) - f(z^k) + f(z^k) - f(\widehat{x}) > -\frac{C_4}{\alpha C_5^2} \widehat{\Delta}^2 + \eta C_3 \widehat{\Delta} > \left( -\frac{C_4}{\alpha C_5^2} + \frac{\eta C_3}{\Delta'} \right) \widehat{\Delta}^2.$$

Finalmente, usando a definição de  $\Delta'$  dada pelo Lema 1.10 e (1.54), obtemos que

$$f(x^k) - f(\widehat{x}) > \left( \frac{(1-\alpha)}{C_5^2} + C_1 \alpha \right) \widehat{\Delta}^2 > \alpha h(\widehat{x})$$

e a prova está completa.  $\square$

A condição de eficiência do passo de otimalidade provada no teorema anterior é estendida para o passo total no resultado a seguir. Para tanto, defina a altura do filtro por

$$v_k = \min \{1, \min \{(1-\alpha)h^j \mid (f^j, h^j) \in F_k\}\}.$$

Esta altura e a folga do filtro  $H_k$ , dada em (1.28), foram inicialmente definidas em [117] e estão ilustradas na Figura 1.6.

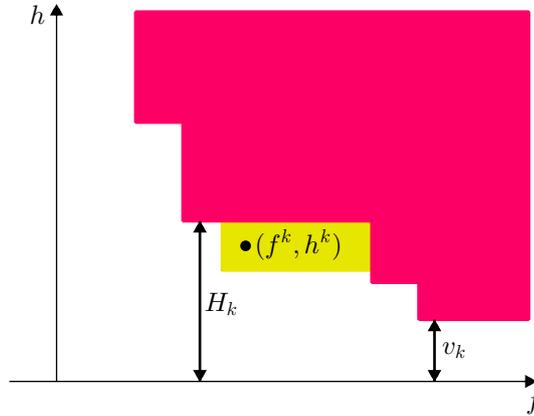


Figura 1.6: Diferença entre a folga do filtro  $H_k$  e a altura do filtro  $v_k$ .

**Lema 1.12** *Considere um ponto viável não estacionário  $\bar{x} \in X$ . Então, existem constantes  $C_8 > 0$  e  $C_9 > 0$  e uma vizinhança  $V_4$  de  $\bar{x}$  tais que para todo  $x^k \in V_4$ ,*

$$f(x^k) - f(x^{k+1}) \geq C_8 v_k \quad (1.56)$$

e

$$f(x^k) - f(x^{k+1}) \geq C_9 \|x^k - x^{k+1}\|. \quad (1.57)$$

*Demonstração.* Considere a vizinhança  $V_3$  e a constante  $C_6$  dadas pelo Teorema 1.11, e  $C_4$  dada em (1.29). Como  $\bar{x}$  é viável e, por H1, a função  $h$  é contínua, existe uma vizinhança  $V_4 \subset V_3$  de  $\bar{x}$  tal que, para todo  $x \in V_4$ ,

$$h(x) \leq \min \left\{ 1, \left( \frac{C_6}{2C_4} \right)^2 \right\}. \quad (1.58)$$

Considere  $x^k \in V_4$ . Usando a relação (1.29), o Teorema 1.11, o fato que  $\sqrt{h(x^k)} \leq \sqrt{H_k}$  e (1.58), temos que

$$\begin{aligned} f(x^k) - f(x^{k+1}) &= f(x^k) - f(z^k) + f(z^k) - f(x^{k+1}) \\ &\geq -C_4 h(x^k) + C_6 \sqrt{H_k} \\ &\geq (-C_4 \sqrt{h(x^k)} + C_6) \sqrt{H_k} \\ &\geq \frac{C_6}{2} \sqrt{H_k}. \end{aligned}$$

Como  $H_k \leq 1$ , temos que  $\sqrt{H_k} \geq H_k \geq v_k$ . Usando isto e tomando  $C_8 = C_6/2 > 0$ , obtemos (1.56), como desejado.

Agora vamos verificar (1.57). Por (1.6) e pelo fato que  $h(x^k) \leq \sqrt{H_k} \leq 1$ , temos que

$$\|z^k - x^k\| \leq \beta h(x^k) \leq \beta \sqrt{H_k}.$$

Usando isto em (1.36), temos que

$$f(z^k) - f(x^{k+1}) \geq C_6 \sqrt{H_k} \geq \frac{C_6}{\beta} \|z^k - x^k\|. \quad (1.59)$$

Pela continuidade Lipschitz de  $f$ , temos que existe uma constante  $\tilde{L} > 0$  tal que

$$|f(z^k) - f(x^{k+1})| \leq \tilde{L} \|z^k - x^{k+1}\|.$$

Com isso e (1.59), segue que

$$\|z^k - x^k\| \leq \frac{\beta \tilde{L}}{C_6} \|z^k - x^{k+1}\|. \quad (1.60)$$

Deste modo, usando a desigualdade triangular e (1.60),

$$\|x^{k+1} - x^k\| \leq \|x^{k+1} - z^k\| + \|z^k - x^k\| \leq \left( 1 + \frac{\beta \tilde{L}}{C_6} \right) \|z^k - x^{k+1}\|. \quad (1.61)$$

Usando (1.29), o fato que  $\sqrt{h(x^k)} \leq \sqrt{H_k}$ , (1.58) e (1.36), temos que

$$|f(x^k) - f(z^k)| \leq C_4 h(x^k) \leq C_3 \sqrt{h(x^k)} \sqrt{H_k} \leq \frac{C_6}{2} \sqrt{H_k} \leq \frac{1}{2} (f(z^k) - f(x^{k+1})).$$

Deste modo,

$$f(x^k) - f(x^{k+1}) = f(x^k) - f(z^k) + f(z^k) - f(x^{k+1}) \geq \frac{1}{2}(f(z^k) - f(x^{k+1})).$$

Com isso, (1.37) e (1.61), temos que

$$f(x^k) - f(x^{k+1}) \geq \frac{1}{2}(f(z^k) - f(x^{k+1})) \geq \frac{C_7}{2} \|z^k - x^{k+1}\| \geq \frac{C_6 C_7}{2(C_6 + \beta \tilde{L})} \|x^k - x^{k+1}\|.$$

Tomando  $C_9 = \frac{C_6 C_7}{2(C_6 + \beta \tilde{L})} > 0$ , obtemos (1.57) e a prova está concluída.  $\square$

Embora o Algoritmo 1.1 decomponha cada iteração em duas fases, ele essencialmente consiste em calcular um ponto  $x^{k+1}$  não proibido pelo filtro, a partir do ponto corrente  $x^k$ , e atualizar o filtro. Este passo satisfaz a condição de eficiência, como mostrado no último lema, proposta inicialmente em [117] e considerada por Peričaro, Ribeiro e Karas em [106, H3]. Além disso, a construção e a atualização do filtro são feitas da mesma maneira e nossas hipóteses implicam nas condições consideradas pelos autores. Consequentemente, nosso algoritmo se encaixa no algoritmo geral de filtro [106, Alg. 1] e os resultados de convergência global seguem imediatamente daquele trabalho, como discutido a seguir. Para melhor esclarecimento, estes resultados estão apresentados em detalhes no Apêndice A.

**Teorema 1.13** [106, Teorema 3.5] *A sequência  $(x^k)$  tem um ponto de acumulação estacionário.*

*Demonstração.* A demonstração pode ser vista no Teorema A.8.  $\square$

O último resultado garante que a sequência  $(x^k)$  gerada pelo Algoritmo 1.1 tem um ponto de acumulação estacionário, independentemente da regra de filtro adotada. No entanto, quando o critério de filtro inclinado é considerado, todo ponto de acumulação da sequência gerada pelo algoritmo é um ponto estacionário. Tal resultado é herdado de [78], onde os autores consideram um algoritmo geral no formato de [106] com o ponto  $x^{k+1} \notin \hat{\mathcal{F}}_k$  computado por restauração inexata e filtro inclinado, e apresentado a seguir.

**Teorema 1.14** [78, Teorema 3.7] *Considere  $\mathcal{R}_k$  como definido em (1.5). Então todo ponto de acumulação de sequência  $(x^k)$  gerada pelo Algoritmo 1.1 é estacionário.*

*Demonstração.* A prova deste fato está apresentada no Teorema A.13.  $\square$

## 1.5 Experimentos computacionais

Nesta seção apresentamos experimentos computacionais realizados com o objetivo de testar o desempenho do algoritmo proposto.

### 1.5.1 Algoritmos testados

A estrutura do nosso algoritmo é semelhante à do [105, Alg. 1] com o ponto  $x^{k+1} \notin \widehat{\mathcal{F}}_k$  calculado por restauração inexata pelo [105, Alg. 3]. A principal diferença reside no uso, ou não, das derivadas da função objetivo.

Assim, foram considerados os dois algoritmos com as regras de filtro original e inclinado, que correspondem às regiões proibidas definidas em (1.4) e (1.5), respectivamente. Em suma, temos quatro instâncias a serem testadas:

- **SD-FO**: Algoritmo 1.1 Sem uso de **D**erivadas da função objetivo e com critério de **F**iltro **O**riginal, isto é, com as regiões proibidas dadas em (1.4);
- **SD-FI**: Algoritmo 1.1 Sem uso de **D**erivadas da função objetivo e com critério de **F**iltro **I**nclinado, isto é, com as regiões proibidas dadas em (1.5);
- **CD-FO**: Algoritmo [105, Alg. 1.1 com Alg. 1.3] de restauração inexata **C**om uso de **D**erivadas da função objetivo e com critério de **F**iltro **O**riginal;
- **CD-FI**: Algoritmo [105, Alg. 1.1 com Alg. 1.3] de restauração inexata **C**om uso de **D**erivadas da função objetivo e com critério de **F**iltro **I**nclinado.

As implementações em MATLAB das instâncias com derivada, CD-FO e CD-FI, foram fornecidas pela autora de [105]. Os detalhes de implementação destas versões estão descritos no Capítulo 2 daquele trabalho.

Partindo desta versão fornecida, foi realizada a implementação das instâncias sem uso de derivada (SD-FO e SD-FI) que diferem de CD-FO e CD-FI essencialmente na fase de otimalidade e no critério de parada do algoritmo, como explicado a seguir.

#### Fase de otimalidade sem derivada

Dada uma iteração  $k$ , para cada iteração interna  $j$ , a fase de otimalidade requer um modelo quadrático  $m_{k_j}$  que aproxime adequadamente a função objetivo em torno do ponto restaurado  $z^k$  e que satisfaça a hipótese H4. Estes modelos foram obtidos

por interpolação polinomial [70, 100]. Cada iteração interna  $j$  possui um conjunto de  $(n+1)(n+2)/2$  pontos interpoladores, incluindo entre eles o ponto restaurado corrente. Na primeira iteração este conjunto é obtido por [34, Alg. 6.3] com um conjunto inicial contendo apenas o ponto  $z^k$ . Cabe lembrar que este algoritmo garante que a hipótese H4 é satisfeita. Na implementação, os raios  $\delta$  e  $\Delta$  coincidem e, no início de cada iteração  $j$ , seus valores são herdados da iteração anterior. Caso esta iteração não calcule um passo de região de confiança, o raio do modelo  $\delta_{k_j}$  é reduzido pela metade e o novo conjunto interpolador é recalculado segundo este mesmo Algoritmo 6.3 de [34] com conjunto inicial  $\{z^k\}$ . Por outro lado, se um passo de região de confiança for calculado e não for proibido pelo filtro, a iteração  $j$  é de sucesso, o novo ponto  $x^{k+1} = z^k + d^k$  é incluído no conjunto interpolador no lugar do ponto mais distante com relação a  $z^k$ , o raio  $\delta$  não é alterado e a fase de otimalidade é finalizada com sucesso. A fim de melhorar o desempenho do algoritmo, a partir de  $10n$  iterações  $k$ , o raio  $\delta$  é reduzido por uma fração de 5% na fase de otimalidade de sucesso. Por fim, se um passo de região de confiança for calculado e o ponto obtido for proibido pelo filtro, os raios  $\delta$  e  $\Delta$  são reduzidos pela metade, o ponto  $z^k + d^{k_j}$  é incluído no conjunto interpolador no lugar do ponto mais distante do ponto restaurado corrente e uma nova iteração interna é realizada com estes novos raios e novo conjunto interpolador.

### Critérios de parada

Dados  $\varepsilon_h > 0$  e  $\varepsilon > 0$ , o critério de parada do Algoritmo 1.1 é

$$h(z^k) < \varepsilon_h \quad \text{e} \quad \mu\pi_{k_j}(z^k) < \delta_{k_j} < \varepsilon. \quad (1.62)$$

No entanto, para fins práticos, adotamos o critério alternativo

$$h(z^k) < \varepsilon_h \quad \text{e} \quad \pi_{k_j}(z^k) < \varepsilon. \quad (1.63)$$

Note que (1.62) implica (1.63).

Os critérios que consideram que uma solução aproximada foi obtida para os algoritmos com derivadas foram os mesmos adotados em [106, (44)], ou seja, quando

$$\|\mathcal{P}_{\mathcal{L}(x^k)}(x^k - \nabla f(x^k)) - x^k\|_\infty \leq \varepsilon \quad \text{e} \quad h(x^k) \leq \varepsilon_h. \quad (1.64)$$

Como usual em testes numéricos, os algoritmos às vezes precisam ser interrompidos antes de encontrar uma possível solução. Assim, a seguir são descritos os critérios

de parada adotados com os respectivos indicadores de saída dos algoritmos com e sem derivada:

- 1: o algoritmo encontrou uma solução aproximada, satisfazendo (1.64) para os algoritmos com derivada e (1.62) ou (1.63) para os algoritmos sem derivadas, com precisão  $\varepsilon_h > 0$  para viabilidade e  $\varepsilon > 0$  para otimalidade;
- -1: o número máximo de iterações externas ( $k_{\max}$ ) ou de iterações internas ( $k_{\max_i}$ ) foi atingido;
- -2: falha na fase de otimalidade, um valor não numérico foi encontrado;
- -3: o procedimento de restauração retornou um ponto inviável, estacionário de  $h$ ;
- -4: falha na obtenção do modelo interpolador, a Hessiana ou o gradiente do modelo obtido continha componentes, em módulo, maiores que  $10^{18}$ , ou um valor não numérico foi obtido.

## Parâmetros

Tanto para as versões com quanto sem derivadas foram adotados os seguintes parâmetros:

- Parâmetro de definição da margem dos filtros:  $\alpha = 0.1$ ;
- Parâmetro de aceitação do passo de região de confiança:  $\eta = 0.01$ ;
- Raio de região de confiança inicial:  $\Delta_{0_0} = 1$ ;
- Tolerância para o critério de otimalidade:  $\varepsilon = 10^{-4}$ ;
- Tolerância para que o ponto seja considerado viável:  $\varepsilon_h = 10^{-6}$ ;
- Número máximo de iterações externas:  $k_{\max} = 5000$ ;
- Número máximo de iterações internas:  $k_{\max_i} = 1000$ ;

Para o raio mínimo foi adotado o valor  $\Delta_{\min} = 1$  nas versões com derivada e  $\Delta_{\min} = 10^{-30}$  nas versões sem derivada. Tomamos valores diferentes pois este parâmetro desempenha papel diferente nas duas versões. Adotou-se  $\mu = 1.0001$  para o parâmetro usado na fase de otimalidade do Algoritmo 1.1. Cabe salientar que, como consideramos parâmetros diferentes dos adotados em [105], as soluções obtidas podem ser diferentes.

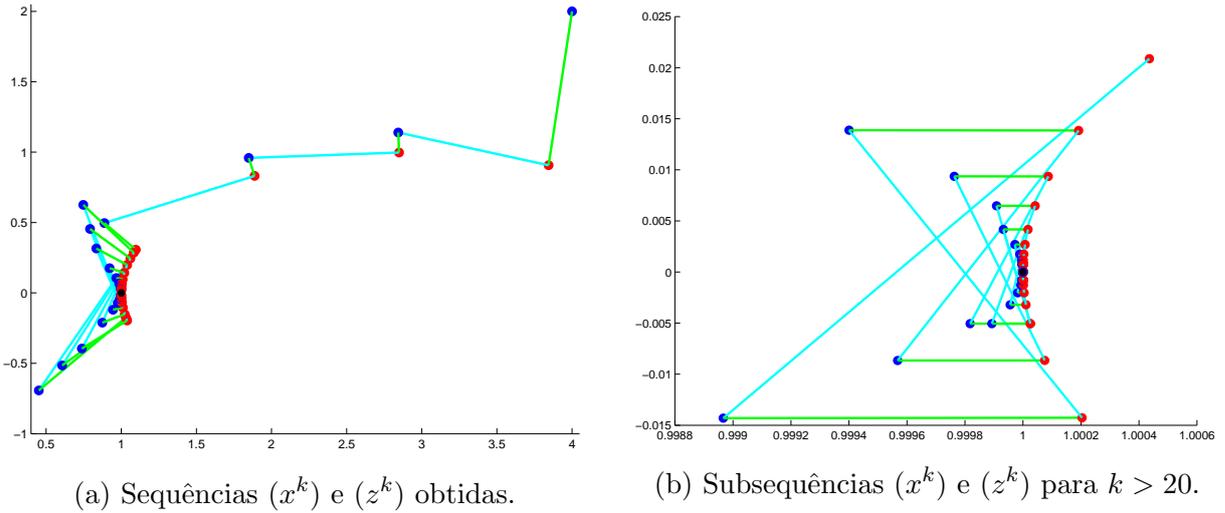


Figura 1.7: Comportamento do Alg. 1.1 com filtro inclinado para o problema (1.65).

## 1.5.2 Exemplo ilustrativo

Primeiramente ilustramos o comportamento do Algoritmo 1.1, especificamente a instância SD-FI, aplicado ao problema em  $\mathbb{R}^2$  dado por

$$\begin{aligned} &\text{minimizar} && 2x_1^2 + \frac{1}{2}x_2^2 \\ &\text{sujeito a} && (x_1 - 3)^2 + 4(x_2)^2 - 4 = 0, \end{aligned} \quad (1.65)$$

cuja solução é  $x^* = (1, 0)$ . Note que o conjunto viável é uma elipse centrada em  $(3, 0)$  e vértices  $(1, 0)$ ,  $(3, 1)$ ,  $(5, 0)$  e  $(3, -1)$ .

Partindo do ponto inicial  $x^0 = (4, 2)$ , o algoritmo obteve uma solução  $\bar{x}$  com  $\|x^* - \bar{x}\|$  da ordem de  $10^{-6}$  após 136 iterações. Nosso objetivo com esta ilustração é mostrar o desempenho do algoritmo de restauração inexata com relação às fases de viabilidade e otimalidade. Os pontos da sequência  $(x^k)$  estão representados em azul ( $\bullet$ ), os da sequência  $(z^k)$  em vermelho ( $\bullet$ ) e a solução encontrada  $\bar{x}$  em preto ( $\bullet$ ). As trajetórias dos pontos de otimalidade para os pontos restaurados estão plotadas em verde, enquanto que, as dos pontos restaurados para os pontos de otimalidade são representadas em azul. A Figura 1.7b ilustra o comportamento do algoritmo próximo à solução obtida.

## 1.5.3 Resultados numéricos

Os testes foram realizados em um computador DELL Intel Core i7-4510U, CPU 2.00GHz  $\times$  4, com 16GB RAM, modelo Inspiron 5000 - i15 5547-A30, 64-bit, usando MATLAB 2014a, v. 8.3.

Foram rodadas as quatro instâncias do algoritmo para a resolução de cada um dos 196 problemas da coleção CUTE<sub>ST</sub> [64] com dimensão menor ou igual a 20 e que foi resolvido com sucesso pelo algoritmo de restauração inexata com derivadas proposto em [105]. Os pontos iniciais considerados foram os preestabelecidos pela coleção de problemas. Os dados dos problemas selecionados e os resultados numéricos das quatro instâncias consideradas estão representados em forma de tabelas no Apêndice B. Apesar de escolhermos os problemas resolvidos com sucesso em [105] nos nossos testes as instâncias CD-FO e CD-FI podem não resolver alguns destes problemas, uma vez que consideramos o parâmetro que determina a margem ( $\alpha$ ) diferente do adotado naquele trabalho.

Saídas	Quantidade de problemas			
	SD-FO	SD-FI	CD-FO	CD-FI
1	109	111	191	190
-1	2	3	4	5
-2	18	18	0	0
-3	0	0	1	1
-4	64	64	0	0

Tabela 1.2: Quantidades de problemas  $\times$  saídas para as instâncias com e sem derivadas.

A Tabela 1.2 sumariza a quantidade de problemas relacionada com o indicador de saída obtido em cada uma das quatro instâncias. A primeira coluna representa o tipo de saída. As demais colunas representam a quantidade de problemas com as respectivas saídas obtidas pelos Algoritmos SD-FO, SD-FI, CD-FO e CD-FI. Nesta tabela constatamos que os algoritmos sem derivadas pararam com saída  $-4$  em 32.7% dos problemas, o que pode indicar dificuldades na construção dos modelos. No entanto, observamos que, em muitos casos, este tipo de saída ocorreu já próximo a uma solução do problema. Veja, por exemplo, no Apêndice B as soluções obtidas pelas instâncias SD-FO e SD-FI para os problemas: BT7, BT8, CANTILVR, CB2, CHACONN1, HS10, HS108, HS27, HS31, HS43, HS63, HS66, MIFFLIN2, OET2, POLAK3 e TRUSPYR2. Deste modo, uma discussão é necessária na forma de atualizar o conjunto interpolador dando atenção ao bom posicionamento do mesmo. É preciso fazer um estudo cuidadoso sobre a escolha do ponto que deve deixar o conjunto interpolador ou ainda, qual ponto deve entrar no conjunto numa iteração de região de confiança de insucesso. Algumas possíveis referências para

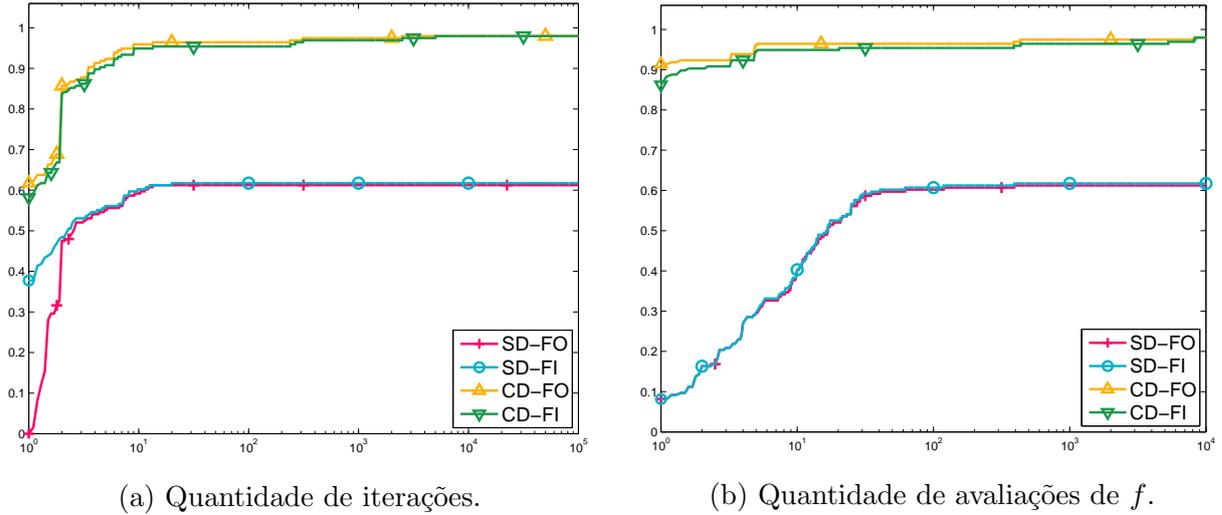


Figura 1.8: Gráfico de desempenho das instâncias CD-FO, CD-FI, SD-FO e SD-FI.

tanto são [66, 99, 122].

Sejam  $h$  e  $f$  os valores da medida de inviabilidade e da função objetivo, respectivamente, obtidos por uma das instâncias quando esta é aplicada a um dado problema teste. Como em [59], consideramos que esta instância resolveu este problema, quando

$$h \leq 10^{-6} \quad \text{e} \quad \frac{f - f_{\min}}{\max\{1, |f_{\min}|\}} \leq 10^{-4},$$

sendo  $f_{\min}$  o menor valor de função objetivo encontrado entre todas as instâncias consideradas para o mesmo problema. Deste modo, consideramos que o algoritmo com melhor desempenho é aquele com uma medida de inviabilidade suficientemente boa e menor valor de função objetivo, independentemente do indicador de saída. Caso alguma outra instância não obtenha um valor de função objetivo próximo a  $f_{\min}$  ou uma medida de inviabilidade suficientemente pequena, consideramos que este algoritmo não resolveu o problema.

Para facilitar a apresentação dos dados obtidos, utilizamos o conceito de perfil de desempenho (*performance profile*) [42]. Adotamos o número de iterações e o número de avaliações da função objetivo como medidas de desempenho nas comparações dos algoritmos. As Figuras 1.8a e 1.8b apresentam os gráficos com os perfis de desempenho das quatro instâncias SD-FO, SD-FI, CD-FO e CD-FI com relação ao número de iterações e a quantidade de avaliações da função objetivo, respectivamente. Como é esperado, os algoritmos com derivadas são mais robustos e mais eficientes do que os que não fazem uso da derivada. Note que a robustez dos algoritmos sem derivadas é de aproximadamente 60%, valor semelhante à porcentagem dos problemas com saída 1 na Tabela 1.2.

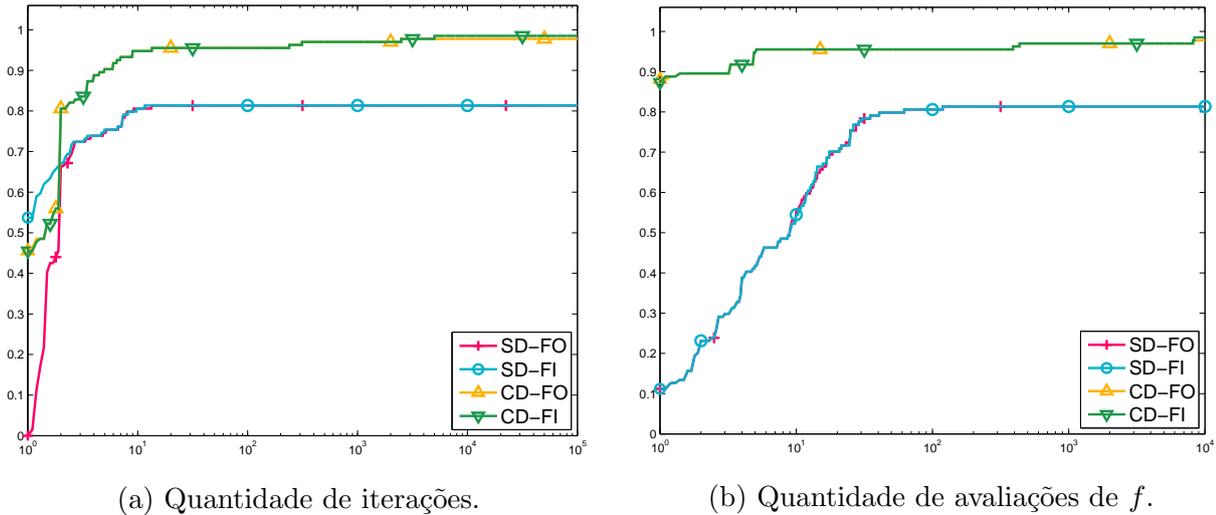


Figura 1.9: Gráfico de desempenho com exceção dos problemas com saída -4.

Com base na dificuldade que o algoritmo sem derivada apresentou na construção dos modelos, vamos desconsiderar os problemas em que pelo menos uma das instâncias sem derivadas obteve saída -4, o que resulta em 134 problemas. A Figura 1.9 ilustra o desempenho das quatro instâncias com relação à quantidade de iterações e o número de avaliações da função objetivo. Com relação ao número de iterações externas, o algoritmo SD-FI foi o mais eficiente, resolvendo 53.7% dos problemas com o menor número de iterações, em seguida temos os algoritmos com derivadas, ambos resolvendo 45.5% dos problemas com o menor número de iterações, e o SD-FO não resolveu nenhum problema com o menor número de iterações. Já com relação à quantidade de avaliações da função objetivo, os algoritmos sem derivadas são os menos eficientes, resolvendo apenas 11.2% dos problemas com o menor número de avaliações, enquanto que as instâncias CD-FO e CD-FI resolveram 88.0% e 87.2%, respectivamente. Isto é plenamente justificado visto que as instâncias SD-FO e SD-FI não usam derivadas e portanto exigem mais avaliações de função na construção dos modelos. A robustez dos algoritmos para estes 134 problemas não teve grande diferença entre os critérios de filtro original e inclinado, sendo 97.7% e 98.5% para os algoritmos com derivadas, respectivamente, e ambos os sem derivadas com robustez de 81.3%. Notamos que uma melhora na forma de atualização dos conjuntos interpoladores poderá contribuir significativamente para um melhor desempenho do algoritmo.

A Figura 1.10 apresenta o gráfico de perfil de dados (*data profile*) [96] em relação ao número de avaliações da função objetivo  $f$ . A figura da direita é uma ampliação do gráfico apresentado à esquerda. Permitindo usar até 500 avaliações de função, os métodos

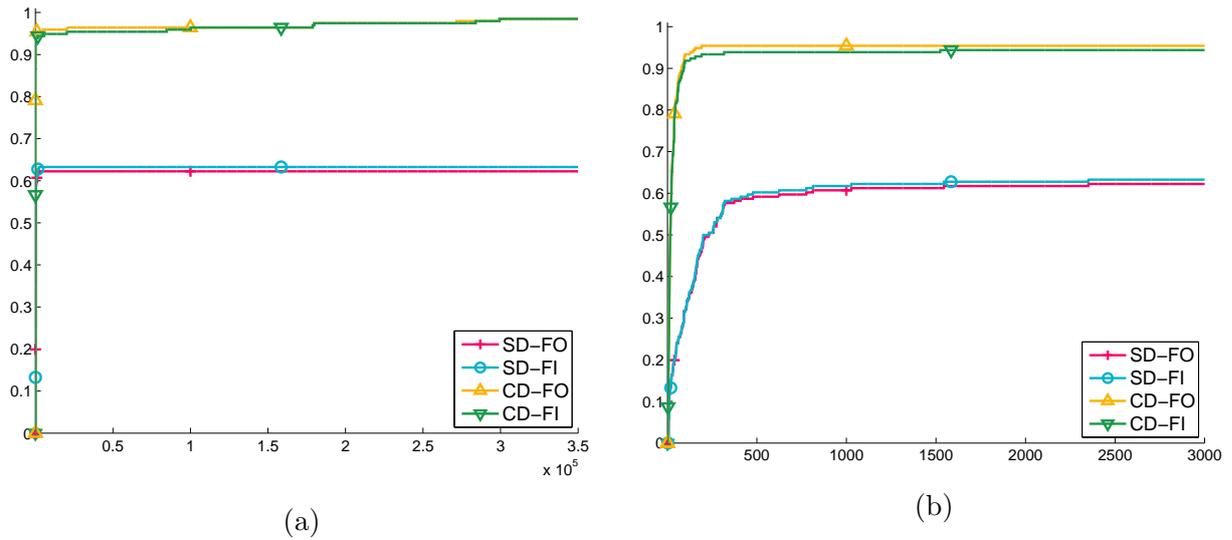


Figura 1.10: Perfil de dados em relação ao número de avaliações de  $f$ .

sem derivadas resolvem cerca de 60% dos problemas, enquanto os métodos que fazem uso das derivadas resolvem cerca de 94% dos problemas. Como era de se esperar o algoritmo com derivadas é o mais eficiente com relação ao número de avaliações da função objetivo para resolver os problemas.

## 1.6 Conclusões do capítulo

Nesta parte do trabalho propomos um algoritmo de restauração inexata com filtro para problemas de programação não linear, o qual não faz uso explícito das derivadas da função objetivo. Cada iteração consiste de duas fases: uma fase de restauração para reduzir a medida de inviabilidade e uma de otimalidade para melhorar o valor da função objetivo no conjunto linearizado. Os modelos lineares ou quadráticos podem ser construídos por qualquer técnica desde que os gradientes dos modelos aproximem suficientemente bem os gradientes da função objetivo perto do ponto restaurado. Nós mostramos que os passos obtidos satisfazem uma condição de eficiência, usando tanto a regra de filtro original quanto a de filtro inclinado. Esta condição implica que a sequência gerada pelo algoritmo tem um ponto de acumulação estacionário independentemente do critério de filtro utilizado. No entanto, se o filtro inclinado é adotado, garante-se que todo ponto de acumulação é estacionário. Alguns resultados numéricos preliminares são apresentados, com os modelos construídos por interpolação polinomial. A forma como os modelos são atualizados ainda precisa de ajustes, uma vez que a maioria dos problemas não resolvidos pelo algoritmo proposto são decorrentes de falhas na obtenção dos modelos. Um dos principais motivos consiste na escolha do ponto que deve sair e entrar no conjunto interpolador quando a iteração interna é de insucesso, ou ainda na escolha de qual ponto deve deixar o conjunto interpolador quando a iteração interna é de sucesso. Excluindo os problemas em que houve tal tipo de falha, o algoritmo se comportou muito bem, encontrando um ponto muito próximo do obtido com o algoritmo com derivadas proposto em [106]. Uma possibilidade para trabalhos futuros é estender o algoritmo proposto para o caso em que tanto as derivadas da função objetivo quanto as das restrições não estão disponíveis. E também propor uma implementação para a interpolação polinomial com um estudo mais profundo sobre a obtenção e atualização dos modelos.

## **PARTE 2**

### **PROBLEMA DE COMPLEMENTARIDADE LINEAR**



## Capítulo 2

# Algoritmos de pivotamento em blocos principais para problemas de complementaridade linear

Neste capítulo, o problema de complementaridade linear é descrito. O problema tem como dados um vetor  $c$  e uma matriz quadrada  $Q$  e, para cada vetor  $c$ , tem solução única quando  $Q$  é uma matriz- $P$ , isto é, quando todos os subdeterminantes principais de  $Q$  são positivos. Nesta tese discutimos alguns algoritmos eficientes para a resolução do problema, nesse caso, e fazemos uma análise sobre algumas estratégias existentes para garantia de convergência dos algoritmos. Finalmente, investigamos o desempenho computacional desses algoritmos.

### 2.1 O problema

Considere  $Q \in \mathbb{R}^{n \times n}$  e  $c \in \mathbb{R}^n$ . O problema de complementaridade linear (*Linear Complementarity Problem* - LCP) consiste em encontrar  $x \in \mathbb{R}^n$  e  $\omega \in \mathbb{R}^n$  tais que

$$\begin{aligned} Qx + c &= \omega, \\ x \circ \omega &= 0, \\ x &\geq 0, \omega \geq 0, \end{aligned}$$

sendo  $x \circ \omega := (x_1\omega_1, \dots, x_n\omega_n) \in \mathbb{R}^n$ . Para facilitar a notação, denotaremos  $(x, \omega) := \begin{pmatrix} x \\ \omega \end{pmatrix} \in \mathbb{R}^{2n}$ . Note que o LCP é equivalente ao seguinte sistema

$$\begin{cases} Qx + c = \omega, & (2.1) \\ x^t\omega = 0, & (2.2) \\ x \geq 0, & (2.3) \\ \omega \geq 0. & (2.4) \end{cases}$$

Note também que, devido às condições (2.3) e (2.4), os vetores  $x$  e  $\omega$  devem ser não negativos e por isso a igualdade (2.2) só é verificada quando, para todo  $i \in \mathcal{N} := \{1, 2, \dots, n\}$ , pelo menos um elemento do par  $(x_i, \omega_i)$  é nulo. Ademais, quando  $c \geq 0$ , o par  $(x, \omega) = (0, c)$  é trivialmente uma solução para o problema.

No decorrer do trabalho denominaremos a variável  $x$  de *variável primal* do LCP e  $\omega$  de *variável dual*. Um par de vetores  $(x, \omega)$  é uma *solução viável* para este problema quando

$$x \geq 0 \quad \text{e} \quad \omega = Qx + c \geq 0.$$

Além disso, dizemos que um par  $(x, \omega)$  é uma *solução complementar* para o problema quando

$$Qx + c = \omega \quad \text{e} \quad x \circ \omega = 0.$$

Portanto, um par  $(x, \omega)$  é uma *solução* do LCP quando este é uma solução complementar e viável para o problema. Por fim, dizemos que um LCP é *viável* quando existe uma solução viável para o problema. Vamos assumir que estamos considerando LCPs viáveis.

Considere a matriz  $Q$  e o vetor  $c$  dados por

$$Q = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \quad \text{e} \quad c = \begin{pmatrix} -4 \\ -5 \end{pmatrix},$$

e o LCP associado dado por

$$\begin{aligned} \omega_1 &= 2x_1 + x_2 - 4 \\ \omega_2 &= x_1 + 2x_2 - 5 \end{aligned} \tag{2.5}$$

$$x_1, x_2, \omega_1, \omega_2 \geq 0 \quad \text{e} \quad x_1\omega_1 = x_2\omega_2 = 0.$$

Note que este pode ser reescrito na seguinte forma matricial

$$\omega_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \omega_2 \begin{pmatrix} 0 \\ 1 \end{pmatrix} + x_1 \begin{pmatrix} -2 \\ -1 \end{pmatrix} + x_2 \begin{pmatrix} -1 \\ -2 \end{pmatrix} = \begin{pmatrix} -4 \\ -5 \end{pmatrix} \tag{2.6}$$

$$x_1, x_2, \omega_1, \omega_2 \geq 0 \quad \text{e} \quad x_1\omega_1 = x_2\omega_2 = 0. \tag{2.7}$$

Como em qualquer solução deste sistema pelo menos um elemento do par  $(x_i, \omega_i)$  tem que ser nulo para  $i \in \{1, 2\}$ . Uma maneira de solucioná-lo é considerar uma variável de cada par  $(x_1, \omega_1)$  e  $(x_2, \omega_2)$  como sendo nula em (2.6). Se a solução de (2.6), considerando essas variáveis nulas, fornecer uma solução não negativa, então esta é uma solução para o sistema.

Por exemplo, fixe as variáveis  $\omega_1$  e  $\omega_2$  como sendo nulas em (2.6). O sistema resultante é dado por

$$x_1 \begin{pmatrix} -2 \\ -1 \end{pmatrix} + x_2 \begin{pmatrix} -1 \\ -2 \end{pmatrix} = \begin{pmatrix} -4 \\ -5 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = c \quad (2.8)$$

$$x_1, x_2 \geq 0.$$

Note que este sistema possui solução se, e somente se, o vetor  $c$  pode ser escrito como uma combinação não negativa dos vetores  $(-2, -1)$  e  $(-1, -2)$ . O conjunto de todas combinações lineares não negativas entre esses vetores é um cone no espaço  $c_1 \times c_2$  como apresentado na Figura 2.1. Este cone é chamado de um *cone complementar* do LCP (2.5). Como a solução de (2.8) é  $(x_1, x_2) = (1, 2) > 0$ , o vetor  $c$  pode ser escrito como uma combinação não negativa dos vetores associados aos coeficientes  $x_1$  e  $x_2$  (veja Figura 2.1) e portanto,  $c$  pertence a este cone complementar e  $(x_1, x_2, \omega_1, \omega_2) = (1, 2, 0, 0)$  é uma solução para (2.5).

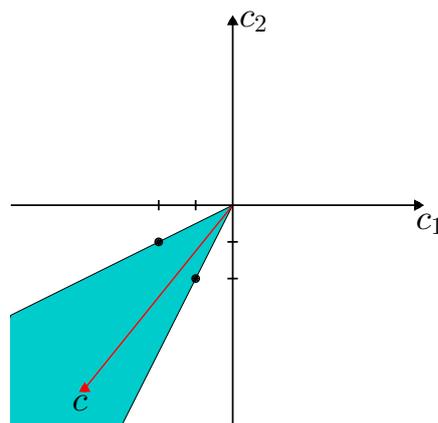


Figura 2.1: Cone complementar associado ao LCP descrito em (2.5), com  $\omega_1 = \omega_2 = 0$ .

Cabe salientar que a complementaridade também pode ser garantida quando fixamos, separadamente, cada um dos pares  $(x_1, x_2)$ ,  $(x_1, \omega_2)$  e  $(x_2, \omega_1)$  como sendo nulos. A cada vez que um desses pares é anulado em (2.6), o sistema resultante consiste em escrever, quando possível, o vetor  $c$  como uma combinação linear não negativa dos vetores associados aos coeficientes não nulos. Cada conjunto contendo todas as combinações não

negativas desses vetores gera um cone complementar. Na Figura 2.2 apresentamos todos os possíveis cones complementares para (2.5). Note que o vetor  $c$  só pertence a um dos cones complementares quando  $\omega_1 = \omega_2 = 0$ . Assim, a solução  $(x_1, x_2, \omega_1, \omega_2) = (1, 2, 0, 0)$  é a única solução para (2.5). Um estudo mais detalhado sobre cones complementares pode ser encontrado em [98, Capítulo 1].

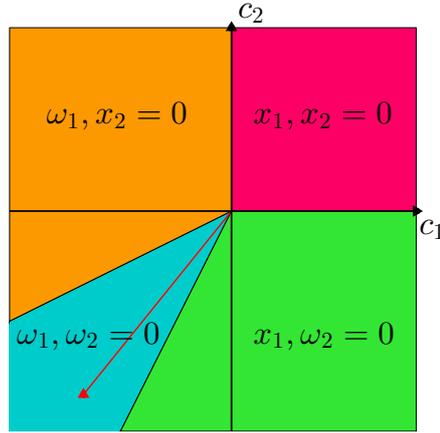


Figura 2.2: Cones complementares para o LCP (2.5), e o vetor  $c = (-4, -5)$ .

### 2.1.1 Classes de matrizes

Nesta seção ressaltamos as definições de algumas classes de matrizes que serão úteis no decorrer do trabalho.

Seja  $A = (a_{ij})$  uma matriz quadrada de ordem  $n$  e  $\mathcal{N} = \{1, \dots, n\}$ . Considere  $F = \{i_1, \dots, i_r\} \subset \mathcal{N}$  um conjunto com elementos arranjados em ordem crescente. A *submatriz principal* de  $A$  determinada por  $F$  é denotada por  $A_F$  e obtida suprimindo de  $A$  todos os elementos das linhas e colunas  $i$  tais que  $i \notin \{i_1, \dots, i_r\}$ , [98, pag. 12], isto é,

$$A_F = \begin{pmatrix} a_{i_1, i_1} & \dots & a_{i_1, i_r} \\ \vdots & \ddots & \vdots \\ a_{i_r, i_1} & \dots & a_{i_r, i_r} \end{pmatrix}.$$

Como existem  $2^n - 1$  subconjuntos com elementos distintos do conjunto  $\mathcal{N}$ , a matriz  $A$  possui  $2^n - 1$  submatrizes principais. O determinante de uma submatriz principal de  $A$  é denominado de *subdeterminante principal* de  $A$ . Considere  $T \subset \mathcal{N}$  com  $T \neq F$ , denotamos por  $A_{FT}$  a submatriz de  $A$  cujos elementos pertencem às linhas e colunas indexadas por elementos de  $F$  e  $T$ , respectivamente.

Para que o leitor possa se familiarizar com esta notação, vamos apresentar dois exemplos. Considere

$$A = \begin{pmatrix} 11 & 12 & 13 & 14 \\ 21 & 22 & 23 & 24 \\ 31 & 32 & 33 & 34 \\ 41 & 42 & 43 & 44 \end{pmatrix}.$$

Se  $F = \{1, 3, 4\}$  e  $T = \{2\}$ , então

$$A_F = \begin{pmatrix} 11 & 13 & 14 \\ 31 & 33 & 34 \\ 41 & 43 & 44 \end{pmatrix}, \quad A_{TF} = (21 \ 23 \ 24), \quad A_{FT} = \begin{pmatrix} 12 \\ 32 \\ 42 \end{pmatrix} \quad \text{e} \quad A_{FN} = \begin{pmatrix} 12 \\ 22 \\ 32 \\ 42 \end{pmatrix}.$$

Considere agora

$$x = (101 \ 102 \ 103 \ 104 \ 105 \ 106) \quad \text{e} \quad \omega = (201 \ 202 \ 203 \ 204 \ 205 \ 206).$$

Se  $F = \{1, 4, 6\}$  e  $T = \{2, 3, 5\}$ , então

$$(x_F, \omega_T) = (101 \ 202 \ 203 \ 104 \ 205 \ 106).$$

Dizemos que  $A$  é uma matriz- $P$  ( $A \in P$ ) quando todos os seus subdeterminantes principais são estritamente positivos, [98, pag. 18]. Deste modo,

$$P = \{A \in \mathbb{R}^{n \times n} \mid \det(A_F) > 0, \forall F \subset \mathcal{N}\}.$$

Uma matriz  $A$  é dita ser *definida positiva* (*semidefinida positiva*) quando  $x^t A x > 0$  ( $x^t A x \geq 0$ ) para todo vetor  $x \in \mathbb{R}^n \setminus \{0\}$  ( $x \in \mathbb{R}^n$ ). A classe de matrizes definidas positivas (semidefinidas positivas) é denotada por  $PD$  ( $PSD$ ).

**Lema 2.1** *Seja uma matriz  $A \in \mathbb{R}^{n \times n}$ . Se  $A$  é definida positiva, então  $A$  é uma matriz- $P$ .*

*Demonstração.* [98, Teorema 1.3]. □

A recíproca do resultado anterior não é válida. Seja  $A \in \mathbb{R}^{2 \times 2}$  dada por

$$A = \begin{pmatrix} 1 & 0 \\ 3 & 2 \end{pmatrix}. \tag{2.9}$$

Apesar de  $A$  ser uma matriz- $P$ , temos que  $A \notin PD$ , uma vez que, para todo  $x = \left(-\frac{3}{2}y, y\right)$  com  $y \neq 0$ ,

$$x^t Ax = -\frac{1}{4}y^2 < 0.$$

As classes de matrizes- $P$  *simétricas* e *simétricas definidas positivas* são denotadas por  $SP$  e  $SPD$ . Deste modo,

$$SP = \{A \in P \mid A^t = A\}$$

e

$$SPD = \{A \in PD \mid A^t = A\}.$$

Analogamente, as classes de matrizes *simétricas semidefinidas positivas* são denotadas por  $SPSD$ .

**Lema 2.2** *A classe de matrizes- $P$  simétricas coincide com a classe de matrizes  $PD$  simétricas.*

*Demonstração.* [98, Teorema 1.9]. □

**Lema 2.3** [98, Resultado 1.9] *Seja  $A \in \mathbb{R}^{n \times n}$ . Então  $A \in PD$  ( $PSD$ ) se, e somente se,  $(A + A^t) \in SPD$  ( $SPSD$ ).*

*Demonstração.* Seja  $A \in \mathbb{R}^{n \times n}$  e  $x \in \mathbb{R}^n \setminus \{0\}$  ( $\mathbb{R}^n$ ). Assim,

$$x^t(A + A^t)x = x^t Ax + (Ax)^t x = 2x^t Ax.$$

Pela igualdade acima temos que  $A \in PD$  ( $PSD$ ) se, e somente se,  $(A + A^t) \in PD$  ( $PSD$ ).

Como a simetria de  $(A + A^t)$  é imediata, a demonstração está concluída. □

**Lema 2.4** [98, Resultado 1.11] *Considere  $B \in \mathbb{R}^{m \times n}$  de posto coluna completo. Então  $B^t B \in SPD$ .*

*Demonstração.* A simetria de  $B^t B$  é imediata. Assim, considere  $x \in \mathbb{R}^n \setminus \{0\}$ . Deste modo,

$$x^t(B^t B)x = \|Bx\|^2 \geq 0.$$

Mas como  $B$  tem posto  $n$ ,  $Bx = 0$  se, e somente se,  $x = 0$ . Logo  $x^t(B^t B)x > 0$  para todo  $x \neq 0$ , o que completa a prova. □

A classe de matrizes *diagonalmente dominantes por linhas* é denotada por  $RDD$  e definida por

$$RDD = \left\{ A \in \mathbb{R}^{n \times n} \mid |a_{ii}| \geq \sum_{j \neq i} |a_{ij}|, \forall i \in \mathcal{N} \right\}.$$

A subclasse de  $RDD$  contendo as matrizes *estritamente diagonalmente dominantes por linhas* é denotada por  $SRDD$  e usa a desigualdade estrita na definição anterior. A subclasse que contém as matrizes *diagonalmente dominantes por linhas com elementos diagonais positivos* é denotada por  $RDD_+$ . A intersecção dessas duas classes é denotada por  $SRDD_+$  e por isso satisfaz

$$A \in SRDD_+ \Leftrightarrow a_{ii} > \sum_{j \neq i} |a_{ij}|, \forall i \in \mathcal{N}.$$

A classe de matrizes *diagonalmente dominantes por colunas* é denotada por  $CDD$  e definida por

$$CDD = \left\{ A \in \mathbb{R}^{n \times n} \mid |a_{jj}| \geq \sum_{i \neq j} |a_{ij}|, \forall j \in \mathcal{N} \right\}.$$

Analogamente denotamos por  $SCDD$  e  $CDD_+$  às subclasses de  $CDD$  contendo as matrizes *estritamente diagonalmente dominantes por colunas* e as matrizes *diagonalmente dominantes por colunas com elementos diagonais positivos*, respectivamente. Por fim,  $SCDD_+ = CDD_+ \cap SCDD$ , isto é,

$$A \in SCDD_+ \Leftrightarrow a_{jj} > \sum_{i \neq j} |a_{ij}|, \forall j \in \mathcal{N}.$$

Note que a matriz- $P$  dada por

$$A = \begin{pmatrix} 1 & 0 \\ -5 & 6 \end{pmatrix} \tag{2.10}$$

pertence a  $SRDD_+$ , mas não pertence a  $SCDD_+$ .

**Lema 2.5** *Seja  $A \in \mathbb{R}^{n \times n}$ . Então  $A \in CDD$  se, e somente se,  $A^t \in RDD$ .*

*Demonstração.* Segue imediatamente da definição das classes de matrizes envolvidas.  $\square$

**Lema 2.6** *Se  $A \in SCDD_+ \cup SRDD_+$ , então  $A \in P$ . Além disso, se  $A \in SCDD_+ \cap SRDD_+$ , então  $A \in PD$ .*

*Demonstração.* Veja Proposição 2.2.20 em [37].  $\square$

Note que as recíprocas deste lema não valem. De fato, a matriz  $A$  dada em (2.9) é uma matriz- $P$ , mas não pertence a  $SCDD_+ \cup SRDD_+$ . Já a matriz

$$\begin{pmatrix} 1 & -2 \\ 2 & 1 \end{pmatrix}$$

é  $PD$  mas não pertence a  $SCDD_+ \cap SRDD_+$ . Além disso,  $A \in SCDD_+ \cup SRDD_+$  não é suficiente para que  $A \in PD$ . Por exemplo, a matriz  $A$  dada em (2.10) é  $P$ , pertence a  $SCDD_+ \cup SRDD_+$ , mas não é definida positiva, pois para todo  $x = \begin{pmatrix} 5 \\ 2 \end{pmatrix} y, y$  com  $y \neq 0$ ,

$$x^t Ax = -\frac{1}{4}y^2 < 0.$$

### 2.1.2 Classes do problema de complementaridade linear

Com base nas classes de matrizes apresentadas na seção anterior, discutimos a seguir algumas classes do LCP.

Dizemos que um LCP é *estritamente monótono* quando  $Q$  é uma matriz definida positiva, ou seja,  $Q \in PD$ , e *monótono* quando  $Q \in PSD$  [37].

Quando a matriz  $Q \in P$  é simétrica (e portanto  $PD$ ) uma propriedade interessante é que o LCP estritamente monótono e o problema quadrático estritamente convexo dado por

$$(PQ) \quad \begin{cases} \text{minimizar} & \frac{1}{2}x^t Qx + c^t x \\ \text{sujeito a} & x \geq 0, \end{cases}$$

são equivalentes. De fato, como o conjunto viável do problema  $(PQ)$  possui interior não vazio, a condição de qualificação de Slater [100, 116] é satisfeita. Deste modo, usando isto e o Teorema Karush-Kuhn-Tucker (KKT) para  $(PQ)$ , temos que  $x^*$  é o minimizador (global) deste problema se, e somente se, existe  $\omega^*$  tal que  $(x^*, \omega^*)$  satisfaz o sistema KKT, o qual é exatamente o sistema definido pelo LCP.

No caso em que  $Q \in P$ , o LCP possui solução única para cada  $c \in \mathbb{R}^n$  [98, Capítulo 3]. Assim, iremos abordar LCPs que possuem matrizes pertencentes à ampla classe de matrizes- $P$ . Cabe salientar que, como vimos nos Lemas 2.1 e 2.2,  $PD \subset P$  e  $P = PD$  para matrizes simétricas.

## 2.2 Algoritmos de pivotamento em blocos principais

Esta seção apresenta alguns métodos de pivotamento que visam resolver LCPs com matrizes- $P$ . Estes métodos trabalham com maneiras de pivotar a igualdade (2.1) e fornecem, iterativamente, soluções complementares inviáveis até que seja encontrada uma solução complementar viável e, portanto, a solução do problema.

Considere  $F \subset \mathcal{N}$  e  $T \subset \mathcal{N}$  dois conjuntos de índices, tais que

$$F \cap T = \emptyset \quad \text{e} \quad F \cup T = \mathcal{N}. \quad (2.11)$$

A ideia geral dos métodos de pivotamento em blocos principais consiste em, a partir desses conjuntos, obter um par corrente  $(x, \omega)$ , tal que este seja uma solução complementar para o LCP. Caso o par obtido também seja viável, então a solução do problema foi encontrada e o método termina com sucesso. Caso contrário, os conjuntos  $F$  e  $T$  são modificados e o processo é repetido.

No decorrer do trabalho, denotamos o complementar de um conjunto  $T$  em  $\mathcal{N}$  por  $\bar{T}$ , isto é,  $\bar{T} = \mathcal{N} \setminus T$ . Além disso, dado um vetor  $x \in \mathbb{R}^n$ , o vetor em  $\mathbb{R}^{|T|}$  contendo as componentes  $x_i$  tais que  $i \in T$  é denotado por  $x_T$ .

Para formalizar o processo realizado pelos métodos de pivotamento em blocos principais, considere dois conjuntos  $F$  e  $T$  satisfazendo (2.11). A cada iteração desses métodos, o par corrente  $(x, \omega)$  é definido como a solução do subsistema do LCP dado por

$$\begin{cases} \omega = Qx + c \\ x_T = 0 \\ \omega_F = 0. \end{cases} \quad (2.12)$$

Note que este subsistema satisfaz (2.1) e (2.2) e, portanto, garante que o par  $(x, \omega)$  é uma solução complementar do LCP. A solução de (2.12) é denominada de *solução básica complementar* do LCP relacionada aos conjuntos  $F$  e  $T$  quando a submatriz principal  $Q_F$  é não-singular. As variáveis nulas  $x_i$  e  $\omega_j$  para  $i \in T$  e  $j \in F$  são denominadas de *variáveis não básicas* e as variáveis  $x_i$  e  $\omega_j$  para  $i \in F$  e  $j \in T$  de *variáveis básicas*, que podem ser nulas ou não.

Vamos agora descrever como obter os valores das variáveis básicas de uma solução complementar, isto é, o vetor  $c' = (x_F, \omega_T) \in \mathbb{R}^n$ . Note primeiramente que, ao determinar  $c'$  determinamos também o par corrente  $(x, \omega)$ , uma vez que, as componentes  $x_T$  e  $\omega_F$  já

foram fixadas como nulas. Para tanto, vamos particionar (2.1) da seguinte forma:

$$\begin{pmatrix} \omega_F \\ \omega_T \end{pmatrix} = \begin{pmatrix} c_F \\ c_T \end{pmatrix} + \begin{pmatrix} Q_F & Q_{FT} \\ Q_{TF} & Q_T \end{pmatrix} \begin{pmatrix} x_F \\ x_T \end{pmatrix}.$$

Uma vez que  $\omega_F$  e  $x_T$  são nulos e que  $x_F = c'_F$  e  $\omega_T = c'_T$ , temos que

$$Q_F c'_F = -c_F \quad \text{e} \quad c'_T = c_T + Q_{TF} x_F.$$

Deste modo, para determinar  $c'$  de modo que a inversa de  $Q_F$  não seja calculada, basta

$$\begin{cases} \text{RESOLVER} & Q_F c'_F = -c_F \end{cases} \quad (2.13)$$

$$\begin{cases} \text{CALCULAR} & c'_T = c_T + Q_{TF} c'_F. \end{cases} \quad (2.14)$$

Portanto, o par corrente  $(x, \omega)$ , solução básica complementar de (2.12), é dado por

$$x = \begin{pmatrix} x_F \\ x_T \end{pmatrix} = \begin{pmatrix} c'_F \\ 0 \end{pmatrix} \quad \text{e} \quad \omega = \begin{pmatrix} \omega_F \\ \omega_T \end{pmatrix} = \begin{pmatrix} 0 \\ c'_T \end{pmatrix}, \quad (2.15)$$

sendo  $c'_F$  e  $c'_T$  dados por (2.13) e (2.14), respectivamente. Se  $c' = (x_F, \omega_T) \geq 0$ , então  $(x, \omega)$  é a solução do LCP e o método termina. Caso contrário, dizemos que esta solução básica é *inviável*, os conjuntos  $F$  e  $T$  são atualizados e uma nova iteração é realizada.

Como veremos nas próximas seções, a grande diferença entre os métodos de pivotamento em blocos principais consiste nas formas de atualizações dos conjuntos  $F$  e  $T$  realizadas de uma iteração para outra.

### 2.2.1 Pivotamento em blocos principais clássico - BPP

Nesta seção apresentaremos o algoritmo de pivotamento em blocos principais (*Block Principal Pivoting Algorithm - (BPP)*) proposto em [76] para resolver LCPs com matrizes- $P$ .

Dados  $T \subseteq \mathcal{N}$  e  $F = \mathcal{N} \setminus T$  dois conjuntos de índices, a solução básica associada  $(x, \omega)$  é dada por (2.15). Deste modo,

$$x_i = \begin{cases} c'_i & \text{se } i \in F \\ 0 & \text{caso contrário} \end{cases} \quad \text{e} \quad \omega_i = \begin{cases} c'_i & \text{se } i \in T \\ 0 & \text{caso contrário,} \end{cases}$$

sendo  $c'$  o vetor obtido por (2.13) e (2.14). Se  $c' \geq 0$ , então  $(x, \omega)$  é a solução do LCP. Caso contrário, considere o conjunto  $H$  dado por

$$H = \{i \in \mathcal{N} \mid c'_i < 0\}, \quad (2.16)$$

ou, equivalentemente

$$H = \{i \in \mathcal{N} \mid x_i < 0 \text{ ou } \omega_i < 0\}.$$

Note que, como  $c' \not\equiv 0$ , temos que  $H \neq \emptyset$ . Qualquer índice  $i$  satisfazendo  $c'_i < 0$  é chamado de índice de *inviabilidade* e o conjunto  $H$  é denominado de *conjunto das inviabilidades*.

Um algoritmo de pivotamento em blocos principais é um processo que, a partir de conjuntos  $F$  e  $T$ , satisfazendo (2.11), constrói novos conjuntos até obter uma solução básica complementar associada que seja viável, ou seja, até que  $H = \emptyset$ . Para tanto, os conjuntos  $F$  e  $T$  são modificados da seguinte maneira

$$F = F - (F \cap H') \cup (T \cap H') \tag{2.17}$$

e

$$T = T - (T \cap H') \cup (F \cap H'), \tag{2.18}$$

sendo  $H'$  um conjunto contido em, igual a, ou contendo  $H$ . Salientamos que esta notação representa as seguintes atualizações

$$F \leftarrow F - (F \cap H') \cup (T \cap H') \quad \text{e} \quad T \leftarrow T - (T \cap H') \cup (F \cap H'),$$

as quais foram escritas nas formas (2.17) e (2.18) para simplificar a notação. Além disso, nessas atualizações primeiramente são calculadas as operações entre parênteses e depois as operações na ordem em que estão, primeiro a subtração e depois a união. Suponha, por exemplo, que  $F = \{1, 2, 3\}$ ,  $T = \{4, 5, 6\}$  e  $H' = \{2, 6\}$ , então de acordo com (2.17) e (2.18), as atualizações em  $F$  e  $T$  são dadas por

$$F = \{1, 2, 3\} - \{2\} \cup \{6\} = \{1, 3, 6\}$$

e

$$T = \{4, 5, 6\} - \{6\} \cup \{2\} = \{2, 4, 5\}.$$

De acordo com (2.17) e (2.18), se  $H' = H$ , então estas atualizações podem ser reescritas da seguinte forma

$$F = F - \{i \in F \mid x_i < 0\} \cup \{i \in T \mid \omega_i < 0\}$$

e

$$T = T - \{i \in T \mid \omega_i < 0\} \cup \{i \in F \mid x_i < 0\}.$$

Observe também que estas atualizações são razoáveis, uma vez que, se um índice  $i \in F$  ( $i \in T$ ) corresponde a uma variável básica viável  $x_i \geq 0$  ( $\omega_i \geq 0$ ), então este índice

permanece no conjunto para a próxima iteração. Caso contrário, este índice trocará de conjunto. Deste modo, na próxima iteração a solução básica inviável de índice  $i$  da iteração anterior se tornará viável para a iteração seguinte, uma vez que esta será anulada ao trocar de conjunto.

O Algoritmo BPP também pode ser visto como um método de conjuntos ativos em blocos (*Block Active-Set*) [6, 100]. De fato, seja uma iteração qualquer com conjuntos  $F$  e  $T$  que fornecem a solução básica complementar  $(x, \omega)$ . Na próxima iteração o novo conjunto  $T$ , de restrições ativas ( $x_i = 0$ ), é definido por

$$T = \{i \in F \mid x_i < 0\} \cup \{i \in T \mid \omega_i \geq 0\}, \quad (2.19)$$

e, como anteriormente, o novo conjunto  $F$  é atualizado, com base no novo conjunto  $T$ , por

$$F = \mathcal{N} \setminus T.$$

Em outras palavras, em cada iteração o conjunto das restrições ativas ( $x_i = 0$ ) contém os índices viáveis ( $i$ ) das variáveis básicas duais ( $\omega_i \geq 0$ ) e os índices inviáveis ( $i$ ) das variáveis básicas primais ( $x_i < 0$ ) da iteração anterior.

A seguir dispomos o algoritmo de pivotamento em blocos principais clássico que satisfaz as propriedades descritas nesta seção.

---

### Algoritmo 2.1 BPP

---

Dados:  $k = 0$ ,  $T \subseteq \mathcal{N}$  e  $F = \mathcal{N} \setminus T$

REPITA

    Calcule  $c'$  por (2.13) e (2.14) e  $(x, \omega)$  por (2.15)

    Calcule  $H$  por (2.16)

    SE  $H = \emptyset$

        Pare com  $(x, \omega)$  solução do LCP

    SENÃO

        Atualize  $F$  e  $T$  por (2.17) e (2.18), respectivamente, considerando  $H' = H$

$k = k + 1$

---

Observe que este algoritmo é muito simples de entender e de ser implementado. Porém, resultados de convergência ainda estão sendo estudados e serão abordados mais

adiante. Note também que o número de iterações  $k$  do algoritmo coincide com o número de sistemas (2.13) resolvidos.

O lema a seguir garante que os conjuntos  $F$  e  $T$  mudam em cada iteração, a menos que a solução seja obtida. Deste modo, como a solução do sistema (2.12) é única para cada conjunto  $F$ , o algoritmo não gera duas soluções básicas consecutivas iguais.

**Lema 2.7** *Considere  $F, T$  e  $F^+, T^+$  conjuntos de duas iterações consecutivas e  $(x, \omega)$  a solução básica complementar associada aos conjuntos  $F$  e  $T$ . Então,  $(x, \omega)$  satisfaz (2.1)–(2.4) ou,  $T \neq T^+$  e  $F \neq F^+$ .*

*Demonstração.* Suponha que  $T = T^+$  e, conseqüentemente,  $F = F^+$ . Considere  $i \in T^+$ . Então,  $\omega_i \geq 0$  ou  $x_i < 0$ . Mas,  $T^+ = T$  e para  $i \in T$  temos que  $x_i = 0$ . Logo, não existe  $i \in \mathcal{N}$  tal que  $x_i < 0$ . Assim,  $x \geq 0$  e  $\omega_{T^+} \geq 0$ . Como  $\omega_{F^+} = 0$ , temos que  $\omega \geq 0$  e, conseqüentemente,  $H = \emptyset$ . Portanto, a solução  $(x, \omega)$  de (2.12) é solução do LCP e o algoritmo para antes de gerar o conjunto  $T^+$ .  $\square$

## 2.2.2 Pivotamento em blocos principais alternativo - KR

Apesar do Algoritmo Kunisch-Rendl - (KR), proposto em [83], ser introduzido como um algoritmo de conjuntos ativos, apresentamos nesta seção que este pode ser visto como um algoritmo de pivotamento em blocos principais que usa em cada iteração os conjuntos  $T$  e  $F$  dados por

$$T = \{i \in \mathcal{N} \mid x_i < 0 \text{ ou } \omega_i > 0\} \quad \text{e} \quad F = \mathcal{N} \setminus T, \quad (2.20)$$

onde  $(x, \omega)$  é uma solução básica da iteração anterior. Note que a definição de  $T$  difere de (2.19) apenas pelas desigualdades estritas nas variáveis duais. Como em implementações computacionais a igualdade é raramente atingida, os dois métodos, BPP e KR, coincidem na prática.

Seja  $(x, \omega)$  uma solução básica complementar de (2.12), dada por (2.15). Cada iteração do algoritmo KR utiliza as regras de atualização (2.17) e (2.18) com o conjunto  $H'$  dado por

$$H' = \{i \in F \mid c'_i < 0\} \cup \{i \in T \mid c'_i \leq 0\}, \quad (2.21)$$

ou, equivalentemente,

$$H' = \{i \in F \mid x_i < 0\} \cup \{i \in T \mid \omega_i \leq 0\}.$$

Note que  $H \subseteq H'$ , sendo  $H$  o conjunto de inviabilidades definido em (2.16) e considerado no método BPP. Cabe salientar que, como o conjunto  $H'$  possui as variáveis básicas duais ativas e não apenas as inviabilidades, a cardinalidade nula deste conjunto não pode ser um critério de parada, como foi considerado no Algoritmo BPP. Deste modo, o Algoritmo KR é o Algoritmo BPP com  $H'$  dado por (2.21), como apresentado a seguir.

---

**Algoritmo 2.2** KR

---

Dados:  $k = 0$ ,  $T \subseteq \mathcal{N}$  e  $F = \mathcal{N} \setminus T$

REPITA

    Calcule  $c'$  por (2.13) e (2.14) e  $(x, \omega)$  por (2.15)

    Calcule  $H'$  por (2.21)

    SE  $c' \geq 0$

        Pare com  $(x, \omega)$  solução do LCP

    SENÃO

        Atualize  $F$  e  $T$  por (2.17) e (2.18)

$k = k + 1$

---

### 2.2.3 Algoritmo em duas fases - BPP-PC

Discutimos nesta seção um algoritmo de pivotamento principal alternativo chamado de algoritmo de pivotamento em blocos principais - ciclos primais (*Block Principal Pivoting - Primal Cycle (BPP-PC)*). Este algoritmo é baseado no proposto em [72]. Cada iteração deste algoritmo é decomposta em duas fases: uma *fase de viabilidade primal*, que fornece uma solução básica primal viável e uma *fase dual*, que a partir da fase primal obtém uma solução básica dual.

Sejam  $F$  e  $T$  dois conjuntos de índices satisfazendo (2.11) e  $x = (c'_F, 0)$  a solução básica associada, com  $c'_F$  dado por (2.13). Se o ponto  $x$  for viável ( $x \geq 0$ ), então a fase de viabilidade é finalizada com a solução básica primal viável  $x$ . Caso contrário, esta fase realiza atualizações consecutivas do conjunto  $F$  segundo (2.17), com  $H' = H_x$  sendo  $H_x$  definido por

$$H_x = \{i \in F \mid c'_i < 0\}, \quad (2.22)$$

até obter um conjunto  $F$  tal que a solução básica primal associada seja viável

$(x = (c'_F, 0) \geq 0)$ , ou seja, até que  $H_x = \emptyset$ . Então esta fase é finalizada com sucesso. Como  $H_x \subseteq F$ , podemos reescrever a atualização de  $F$ , definida em (2.17), por

$$F = F \setminus H_x.$$

Note que o número máximo de atualizações necessárias para obter uma solução primal viável é  $n$ , uma vez que, a cada solução básica primal inviável, o conjunto  $F$  se torna menor, até que, no pior dos casos, este se torne vazio. Além disso, como o objetivo desta fase é obter uma solução primal viável, as variáveis duais e as atualizações no conjunto  $T$  não precisam ser calculadas.

A partir do ponto  $x = (c'_F, 0)$  viável, obtido na fase de viabilidade primal, a fase dual calcula o conjunto  $T$  por  $T = \mathcal{N} \setminus F$  e a variável básica dual associada por  $\omega = (c'_T, 0)$ , com  $c'_T$  dado por (2.14). Note que o conjunto  $T$  não pode ser atualizado segundo (2.18), pois durante a fase de viabilidade primal este conjunto não foi atualizado.

Suponha que se tenha uma solução básica complementar  $(x, \omega)$  com  $x$  viável e considere o conjunto das inviabilidades duais, associado a esta solução, dado por

$$H_\omega = \{i \in T \mid c'_i < 0\}. \quad (2.23)$$

Se  $H_\omega = \emptyset$ , então a solução básica complementar  $(x, \omega)$  é também viável, a solução do LCP é encontrada com sucesso e o algoritmo para. Caso contrário, o conjunto  $F$  é modificado segundo (2.17) com  $H' = H_\omega$  e uma nova fase de viabilidade primal é realizada. Cabe salientar que, ao realizar esta atualização no conjunto  $F$ , a viabilidade da variável primal pode ser perdida, e por isso é necessário realizar a fase de viabilidade primal novamente. Note também que o conjunto  $H_\omega$  calculado na fase dual é o mesmo que o conjunto  $H$  considerado no algoritmo BPP, dado por (2.16), devido ao fato de  $x$  ser um ponto viável. Além disso, a atualização de  $F$  na fase dual pode ser reescrita por

$$F = F \cup H_\omega = F \cup H.$$

Para estabelecer o algoritmo, denote  $k_v$  o número de fases realizadas para recuperar a viabilidade primal e  $k$  o número de sistemas (2.13) resolvidos. Adotamos que o número de iterações coincide com o número  $k$  de sistemas resolvidos.

Cabe salientar que, usualmente, o conjunto inicial  $F$  é considerado como  $F = \{i \mid c_i < 0\}$  em todos os algoritmos discutidos até o momento, a não ser que as características do LCP a resolver sugiram uma escolha do conjunto inicial  $F$  mais conveniente.

---

**Algoritmo 2.3** BPP-PC
 

---

Dados:  $k_v = 0$ ,  $k = 0$ ,  $F \subseteq \{1, 2, \dots, n\}$ ,  $T = \mathcal{N} \setminus F$

REPITA

Calcule  $c'_F$ ,  $x$  e  $H_x$  por (2.13), (2.15) e (2.22), respectivamente

$k = k + 1$

SE  $H_x \neq \emptyset$

$k_v = k_v + 1$

ENQUANTO  $H_x \neq \emptyset$

Atualize  $F$  por (2.17) com  $H' = H_x$

Calcule  $c'_F$ ,  $x$  e  $H_x$  por (2.13), (2.15) e (2.22), respectivamente

$k = k + 1$

Faça  $T = \mathcal{N} \setminus F$

Calcule  $c'_T$ ,  $\omega$  e  $H_\omega$  por (2.14), (2.15) e (2.23), respectivamente

SE  $H_\omega = \emptyset$

Pare com  $(x, \omega)$  solução do LCP

SENÃO

Atualize  $F$  por (2.17) com  $H' = H_\omega$

---

## 2.3 Ciclos nos algoritmos

Apesar dos algoritmos descritos anteriormente estarem bem definidos quando  $Q$  é uma matriz- $P$ , não há garantia teórica de convergência desses processos num número finito de iterações. Nesta seção são apresentados alguns exemplos em que os algoritmos ciclam e, conseqüentemente, não convergem. Notemos que esses casos ocorrem muito raramente na prática e que, como veremos mais adiante, na prática os algoritmos requerem em geral poucas iterações para obter a solução única do LCP.

Inicialmente, considere o LCP, sugerido em [71], cujo vetor  $c$  é dado por

$$c = ( -1852, -9850, 8930, -1624, -27127, -11927, 3889, 5579, 203, -652, -1781, -353 ) \quad (2.24)$$

e a matriz  $Q$  por (2.25). Note que a matriz  $Q$  é simétrica e definida positiva e portanto  $Q \in P$ , mas não é diagonalmente dominante. Além disso, como a matriz  $Q$  é simétrica, resolver um LCP com esta matriz é equivalente a resolver o problema  $(PQ)$ , independentemente do vetor  $c$  adotado.

$$Q = \begin{pmatrix} 102 & 1 & -105 & -1 & 182 & -2 & -1 & -23 & 1 & 0 & 0 & 0 \\ 1 & 92 & 1 & 16 & 56 & -27 & -16 & 3 & -13 & 1 & 10 & -2 \\ -105 & 1 & 114 & 0 & -196 & -4 & 1 & 25 & 0 & 1 & 1 & 0 \\ -1 & 16 & 0 & 36 & 1 & -2 & 5 & -5 & -26 & 1 & 1 & -3 \\ 182 & 56 & -196 & 1 & 541 & 206 & -53 & -121 & 14 & -5 & 9 & -1 \\ -2 & -27 & -4 & -2 & 206 & 427 & -43 & -123 & 4 & 0 & 5 & -4 \\ -1 & -16 & 1 & 5 & -53 & -43 & 130 & 12 & -2 & 1 & 14 & 3 \\ -23 & 3 & 25 & -5 & -121 & -123 & 12 & 218 & -13 & -9 & 4 & 8 \\ 1 & -13 & 0 & -26 & 14 & 4 & -2 & -13 & 339 & 11 & 15 & -6 \\ 0 & 1 & 1 & 1 & -5 & 0 & 1 & -9 & 11 & 590 & 82 & -3 \\ 0 & 10 & 1 & 1 & 9 & 5 & 14 & 4 & 15 & 82 & 685 & -13 \\ 0 & -2 & 0 & -3 & -1 & -4 & 3 & 8 & -6 & -3 & -13 & 457 \end{pmatrix}. \quad (2.25)$$

### 2.3.1 Algoritmos BPP e KR

A diferença entre os Algoritmos BPP e KR está na atualização dos conjuntos  $F$  e  $T$ , a qual utiliza os conjuntos  $H$ , para o método BPP, e  $H'$  para o método KR. Esta diferença decorre exclusivamente da adição dos índices das restrições duais ativas de  $T$  em  $H'$ , como descrito em (2.21), que não ocorre no conjunto  $H$ , como definido em (2.16). Porém, na prática esta diferença acaba não existindo. Isso pode ser visto no desempenho idêntico desses algoritmos para resolver o problema anterior e nos demais testes realizados e apresentados mais adiante na Seção 2.6.2.

Considere agora o exemplo descrito anteriormente e os Algoritmos 2.1 e 2.2. Para este exemplo, estes algoritmos geram sequências periódicas para vários conjuntos iniciais  $F$  e  $T$ . Por exemplo, considere os conjuntos iniciais  $T^0 = \{1, 2, 3, 6, 9\}$  e  $F^0 = \mathcal{N} \setminus T^0$ . Assim, para todo  $j \in \mathbb{N}$  e  $\ell = 0, 1, 2, 3$ , temos que  $T^{4j+\ell} = T^\ell$ . Conseqüentemente, como para cada conjunto  $T$  e  $F$  existe uma única solução básica do problema (2.12), temos que  $(x^{4j+\ell}, \omega^{4j+\ell}) = (x^\ell, \omega^\ell)$ , para  $j \in \mathbb{N}$  e  $\ell = 0, 1, 2, 3$ . Este ciclo está resumido na Tabela 2.1. Além deste conjunto inicial, os algoritmos também ciclam, por exemplo, para os conjuntos  $T^0 = \mathcal{N}$ ,  $T^0 = \{3, 4, 7, 8, 10, 11\}$  e  $T^0 = \{1, 7, 9, 12\}$ .

Em [39], os autores exibem um exemplo de menor dimensão em que os Algoritmos BPP e KR ciclam para vários conjuntos iniciais. Este exemplo é um LCP cuja matriz  $Q$

$k$	$f(x^k)$ ( $\times 10^6$ )	$T^k$
$4j$	-0.726676	{1, 2, 3, 6, 9}
$4j + 1$	-1.181320	{1, 7 }
$4j + 2$	-0.792242	{1, 3, 4, 5, 7, 8}
$4j + 3$	-1.613570	{3, 4, 7, 8}

Tabela 2.1: Solução periódica gerada pelos algoritmos BPP e KR para  $j = 0, 1, 2, \dots$

e vetor  $c$  são dados por

$$Q = \begin{pmatrix} 4 & 5 & -5 \\ 5 & 9 & -5 \\ -5 & -5 & 7 \end{pmatrix} \quad \text{e} \quad c = \begin{pmatrix} -2 \\ -1 \\ 3 \end{pmatrix}. \quad (2.26)$$

Note que a matriz  $Q$  é simétrica e definida positiva, mas não é diagonalmente dominante. Os Algoritmos BPP e KR ciclam para os conjuntos iniciais  $F^0 = \mathcal{N} \setminus T^0$ , com  $T^0 \in \{\emptyset, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{1, 2, 3\}\}$ . Cabe salientar que o Algoritmo BPP-PC encontra a solução do problema em no máximo 3 iterações, para todos os conjuntos iniciais possíveis em  $\mathcal{N}$ .

Os Algoritmos BPP e KR também ciclam para matrizes- $P$  não simétricas. Em [98] os autores sugerem o exemplo em que a matriz  $Q$  e o vetor  $c$  do LCP são dados por

$$Q = \begin{pmatrix} 1 + \varepsilon & 0 & -2 \\ -2 & 1 & 4 \\ -4 & 2 & 9 + \varepsilon \end{pmatrix} \quad \text{e} \quad c = \begin{pmatrix} 1 \\ -1 \\ -3 \end{pmatrix}, \quad (2.27)$$

para  $0 < \varepsilon < \frac{1}{3}$ . Considerando  $\varepsilon = \frac{1}{4}$ , os algoritmos entram em ciclo para todos os conjuntos iniciais, com exceção dos conjuntos  $T^0 = \{3\}$  e  $T^0 = \{1, 2\}$ .

### 2.3.2 Algoritmo BPP-PC

Considere o problema  $(PQ)$  cuja matriz  $Q$  e o vetor  $c$  são dados por (2.25) e (2.24), respectivamente. Quando o Algoritmo 2.3 é utilizado para resolver este problema, sequências periódicas são geradas para diferentes conjuntos iniciais  $F$  e  $T$ . Por exemplo, caso o conjunto inicial seja  $F^0 = \{2, 3, 4, 5, 6, 8, 10, 11, 12\}$  e  $T^0 = \mathcal{N} \setminus F^0$ , o algoritmo gera  $F^{4j+\ell} = F^\ell$ , para todo  $j = 0, 1, \dots$  e  $\ell = 0, 1, 2, 3$ . Conseqüentemente,

$(x^{4j+\ell}, \omega^{4j+\ell}) = (x^\ell, \omega^\ell)$ . Este ciclo é resumido na Tabela 2.2. O algoritmo também cicla para, por exemplo, os seguintes conjuntos iniciais:  $F^0 = \emptyset$ ,  $F^0 = \{1, 2, 5, 6, 9, 12\}$  e  $F^0 = \{2, 3, 4, 5, 6, 8, 10, 11\}$ .

$k$	$f(x^k)$ ( $\times 10^6$ )	$F^k$
$4j$	-1.17740	$\{2, 3, 4, 5, 6, 8, 10, 11, 12\}$
$4j + 1$	-0.79019	$\{2, 6, 10, 11, 12\}$
$4j + 2$	-1.61357	$\{1, 2, 5, 6, 9, 10, 11, 12\}$
$4j + 3$	-0.68208	$\{5, 10, 11, 12\}$

Tabela 2.2: Ciclo gerado pelo Algoritmo 2.3.

O Algoritmo 2.3 também cicla para o LCP com a matriz não simétrica  $Q$  dada em (2.27). Para  $\varepsilon = \frac{1}{4}$  o algoritmo cicla para todos os conjuntos iniciais, com exceção do conjunto  $T^0 = \{3\}$ . O interessante é que o método BPP-PC cicla para  $T^0 = \{1, 2\}$ , enquanto que os Algoritmos BPP e KR não ciclam.

Resumindo, nesta seção vimos que os algoritmos apresentados podem ciclar para LCPs com matrizes  $Q$  definidas positivas simétricas e não simétricas. Cabe salientar que toda matriz definida positiva é uma matriz- $P$ . Note ainda que até então não foi possível gerar um LCP com matriz  $SRDD_+$  ou  $SCDD_+$  para o qual os algoritmos ciclam.

## 2.4 Funções de mérito

As funções de mérito são funções que fornecem um valor escalar que pode indicar o progresso do algoritmo de uma iteração para outra. Deste modo, podemos por exemplo verificar se um novo ponto é melhor ou pior que o ponto corrente. Nesta seção expomos algumas funções de mérito para o problema de complementaridade linear, com matriz  $Q \in P$ , associadas aos algoritmos de pivotamento em blocos principais. Em [49] é realizado um extensivo estudo sobre funções de mérito para LCPs.

Neste contexto, uma função é considerada de mérito para o LCP quando um minimizador global desta é solução do LCP. Deste modo, considere  $\mathcal{K}$  um conjunto que engloba os iterandos do algoritmo. Dizemos que  $\phi : \mathcal{K} \rightarrow \mathbb{R}$  é uma função de mérito para

o LCP quando um minimizador global do problema

$$\begin{aligned} & \text{minimizar} && \phi(u) \\ & \text{sujeito a} && u \in \mathcal{K} \end{aligned}$$

é solução do LCP.

Quando a matriz  $Q$  do problema é  $P$  e simétrica, isto é,  $PD$ , vimos que resolver o LCP é equivalente a resolver o problema  $(PQ)$ . Deste modo, a função

$$f(x) = \frac{1}{2}x^tQx + c^tx \quad (2.28)$$

pode ser considerada como uma função de mérito para o problema.

Os algoritmos de pivotamento em blocos principais geram em cada iteração uma solução básica complementar, isto é, cada iterando satisfaz (2.1) e (2.2). Deste modo, para que a função de mérito forneça alguma informação sobre o desempenho do algoritmo, esta deve estar relacionada com a não-negatividade do par corrente, ou seja, levar em consideração quão perto as equações (2.3) e (2.4) estão de serem satisfeitas. A seguir, listamos algumas quantificações deste tipo que podem ser usadas como funções de mérito para o LCP. Uma função que analisa estas condições é a soma das variáveis inviáveis, ou seja,

$$\Phi_1(x, \omega) = - \sum_{\{i|x_i < 0\}} x_i - \sum_{\{i|\omega_i < 0\}} \omega_i.$$

Podemos também, utilizar uma função de mérito que realiza a soma dos quadrados das variáveis inviáveis

$$\Phi_2(x, \omega) = \sum_{\{i|x_i < 0\}} (x_i)^2 + \sum_{\{i|\omega_i < 0\}} (\omega_i)^2.$$

Uma outra possibilidade seria contar o número de variáveis inviáveis,

$$\Phi_3(x, \omega) = \sum_{\{i|x_i < 0\}} 1 + \sum_{\{i|\omega_i < 0\}} 1.$$

Para finalizar, podemos considerar a norma das inviabilidades dada por

$$\Phi_\rho(x, \omega) = \left\| \min \left\{ x, \frac{\omega}{\rho} \right\} \right\|^2,$$

para algum  $\rho > 0$ . O próximo resultado verifica que estas funções são de fato funções de mérito para o LCP.

**Lema 2.8** [36, Teorema 3] *O par  $(x, \omega)$  é solução do LCP se, e somente se, o par é uma solução complementar e um minimizador global das funções  $\Phi_1$ ,  $\Phi_2$ ,  $\Phi_3$  e  $\Phi_\rho$ .*

*Demonstração.* Note que as quatro funções de mérito são não negativas. Considere  $(x, \omega)$  uma solução complementar tal que

$$\Phi_1(x, \omega) = \Phi_2(x, \omega) = \Phi_3(x, \omega) = \Phi_\rho(x, \omega) = 0.$$

Como isso ocorre se, e somente se, o par  $(x, \omega)$  é uma solução complementar e viável do LCP, ou seja, é a solução do LCP, e a demonstração está concluída.  $\square$

Considere a seguinte função  $\widehat{\xi} : \mathbb{R}^{2n} \rightarrow \mathbb{R}^n$  definida por

$$\widehat{\xi}(x, \omega) = \min(x, \omega),$$

sendo o mínimo considerado componente a componente.

**Teorema 2.9** [36, Seção 4.1] *O par  $(x, \omega)$  é solução do LCP se, e somente se,  $\widehat{\xi}(x, \omega) = 0$  e  $\omega = Qx + c$ .*

*Demonstração.* De fato, considere  $(x, \omega)$  tal que  $\omega = Qx + c$  e  $\widehat{\xi}(x, \omega) = 0$ . Deste modo, como  $\widehat{\xi}(x, \omega) = 0$ , temos que o par é viável e que  $x \circ \omega = 0$ . Assim, o par é uma solução do LCP, uma vez que este é viável e complementar. A recíproca é imediata.  $\square$

Como os métodos de pivotamento em blocos principais sempre consideram soluções complementares  $(x, \omega)$  do LCP, a variável  $\omega$  é dada implicitamente pela variável  $x$ . Assim, podemos considerar o caso particular  $\xi : \mathbb{R}^n \rightarrow \mathbb{R}^n$  de  $\widehat{\xi}$  dada por

$$\xi(x) = \min(x, \omega), \tag{2.29}$$

sendo  $\omega = Qx + c$ .

### 2.4.1 O Algoritmo BPP e o método de Newton

Esta seção apresenta uma relação entre o algoritmo de pivotamento em blocos principais clássico e o método de Newton.

Considere dois conjuntos  $F$  e  $T$  satisfazendo (2.11). Seja  $x = (x_F, 0)$  e  $\omega = (0, \omega_T)$  uma solução básica complementar associada a esses conjuntos. Com base no par  $(x, \omega)$  vamos particionar os conjuntos  $F$  e  $T$  da seguinte forma:

$$F_1 = \{i \in F \mid x_i < 0\}, \quad F_2 = \{i \in F \mid x_i \geq 0\} \tag{2.30}$$

$$T_1 = \{i \in T \mid \omega_i \geq 0\} \quad \text{e} \quad T_2 = \{i \in T \mid \omega_i < 0\}. \tag{2.31}$$

Note que,  $F = F_1 \cup F_2$  e  $T = T_1 \cup T_2$ . Além disso, pelas atualizações dos conjuntos  $F$  e  $T$  consideradas no método BPP, dadas em (2.17) e (2.18), segue que os novos conjuntos  $F^+$  e  $T^+$  são dados por

$$F^+ = T_2 \cup F_2 \quad \text{e} \quad T^+ = T_1 \cup F_1. \quad (2.32)$$

Para obter a relação entre os métodos, iremos considerar a função de mérito  $\xi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ , definida em (2.29), e empregar o método de Newton para sistemas [100] para resolver

$$\xi(x) = 0. \quad (2.33)$$

Note que se  $\xi(x) = 0$ , então  $(x, Qx + c)$  é a solução do LCP considerado. Para este problema, sabemos que o passo  $d$  do método de Newton é a solução do seguinte sistema

$$J\xi(x)d = -\xi(x), \quad (2.34)$$

sendo  $J\xi(x)$  a matriz Jacobiana da função  $\xi$  avaliada em  $x$ . Obtido  $d$ , o novo iterando é dado por

$$x^+ = x + d. \quad (2.35)$$

Consequentemente,

$$\omega^+ = Qx^+ + c. \quad (2.36)$$

O próximo resultado garante que os métodos de Newton e BPP são equivalentes quando os iterandos são soluções básicas complementares não degeneradas. Recordamos que um par complementar  $(x, \omega) \in \mathbb{R}^{2n}$  é não degenerado, quando  $x_i + \omega_i \neq 0$  para todo  $i \in \mathcal{N}$ .

**Teorema 2.10** [36, Seção 4.1] *Considere  $(x, \omega)$  uma solução básica complementar não degenerada do LCP. Então, o próximo iterando, a partir de  $(x, \omega)$ , do método de Newton para resolver o problema (2.33) coincide com o próximo iterando do método BPP para o LCP.*

*Demonstração.* Considere os conjuntos  $F$  e  $T$  satisfazendo (2.11) e  $(x, \omega)$  a solução básica complementar associada. Suponha que este par seja não degenerado. Sejam  $F_1, F_2, T_1$  e  $T_2$  os conjuntos definidos em (2.30) e (2.31). Pela definição de  $\xi$ , para  $i \in \mathcal{N}$ , temos que

$$[\xi(x)]_i = \begin{cases} x_i, & \text{se } i \in F_1 \cup T_1 \\ \omega_i, & \text{se } i \in F_2 \cup T_2. \end{cases}$$

Por isto e pelo fato que  $(x, \omega)$  é não degenerado, obtemos que

$$\xi(x) = (x_{F_1}, \omega_{F_2}, x_{T_1}, \omega_{T_2}). \quad (2.37)$$

Como  $\omega = Qx + c$ , segue que

$$\xi(x) = (x_{F_1}, Q_{F_2\mathcal{N}}x + c_{F_2}, x_{T_1}, Q_{T_2\mathcal{N}}x + c_{T_2}).$$

Deste modo,

$$J\xi(x) = \begin{pmatrix} I_{F_1} & 0 & 0 & 0 \\ Q_{F_2F_1} & Q_{F_2} & Q_{F_2T_1} & Q_{F_2T_2} \\ 0 & 0 & I_{T_1} & 0 \\ Q_{T_2F_1} & Q_{T_2F_2} & Q_{T_2T_1} & Q_{T_2} \end{pmatrix},$$

sendo  $I$  a matriz identidade. Assim, pela definição do passo de Newton, dada em (2.34), e por (2.37), temos que

$$\begin{pmatrix} I_{F_1} & 0 & 0 & 0 \\ Q_{F_2F_1} & Q_{F_2} & Q_{F_2T_1} & Q_{F_2T_2} \\ 0 & 0 & I_{T_1} & 0 \\ Q_{T_2F_1} & Q_{T_2F_2} & Q_{T_2T_1} & Q_{T_2} \end{pmatrix} \begin{pmatrix} d_{F_1} \\ d_{F_2} \\ d_{T_1} \\ d_{T_2} \end{pmatrix} = - \begin{pmatrix} x_{F_1} \\ \omega_{F_2} \\ x_{T_1} \\ \omega_{T_2} \end{pmatrix} \quad (2.38)$$

Consequentemente,

$$d_{F_1} = -x_{F_1} \quad \text{e} \quad d_{T_1} = -x_{T_1}. \quad (2.39)$$

Para verificar que o novo iterando  $x^+$  do passo de Newton, como definido em (2.35), ( $\omega^+$  dado em (2.36)) coincide com o obtido pelo método BPP, vamos primeiramente constatar que  $x^+$  e  $\omega^+$  são complementares. De fato, como a validação da igualdade (2.1) decorre imediatamente da definição de  $\omega^+$ , basta garantir que  $x^+$  e  $\omega^+$  satisfazem (2.2). Usando a definição de  $x^+$  e (2.39), temos que

$$x_{F_1}^+ = x_{F_1} - x_{F_1} = 0 \quad \text{e} \quad x_{T_1}^+ = x_{T_1} - x_{T_1} = 0.$$

Além disso, pelas definições de  $x^+$  e  $\omega^+$  e por (2.1), temos que

$$\begin{aligned} \omega_{T_2}^+ &= c_{T_2} + (Qx^+)_{T_2} \\ &= c_{T_2} + (Q(x+d))_{T_2} \\ &= c_{T_2} + (Qx)_{T_2} + (Qd)_{T_2} \\ &= c_{T_2} + (\omega - c)_{T_2} + (Qd)_{T_2} \\ &= \omega_{T_2} + Q_{T_2\mathcal{N}}d. \end{aligned}$$

Logo, por isso e (2.38), obtemos que

$$\omega_{T_2}^+ = \omega_{T_2} - \omega_{T_2} = 0.$$

Analogamente, temos que

$$\omega_{F_2}^+ = c_{F_2} + (\omega - c)_{F_2} + (Qd)_{F_2} = \omega_{F_2} - \omega_{F_2} = 0.$$

Deste modo, como  $\mathcal{N} = F_1 \cup F_2 \cup T_1 \cup T_2$  e  $x_{T_1}^+ = x_{F_1}^+ = \omega_{T_2}^+ = \omega_{F_2}^+ = 0$ , segue que (2.2) é satisfeita. Portanto, com isso e o fato que  $\omega^+ = Qx^+ + c$ , temos que o par  $(x^+, \omega^+)$  é uma solução complementar do LCP.

Para finalizar, vamos garantir que  $x^+$  e  $\omega^+$ , decorrentes do método de Newton, satisfazem (2.13) e (2.14), ou seja, vamos verificar que  $(x^+, \omega^+)$  é uma solução básica complementar do LCP associada aos conjuntos  $F^+$  e  $T^+$ , como definidos em (2.32). Por (2.38), temos que

$$Q_{F_2F_1}d_{F_1} + Q_{F_2}d_{F_2} + Q_{F_2T_1}d_{T_1} + Q_{F_2T_2}d_{T_2} = -\omega_{F_2}.$$

Com isso, (2.39) e (2.1), segue que

$$\begin{aligned} Q_{F_2}d_{F_2} + Q_{F_2T_2}d_{T_2} &= -\omega_{F_2} + Q_{F_2F_1}x_{F_1} + Q_{F_2T_1}x_{T_1} \\ &= -(Qx + c)_{F_2} + Q_{F_2F_1}x_{F_1} + Q_{F_2T_1}x_{T_1} \\ &= -(Qx)_{F_2} - c_{F_2} + Q_{F_2F_1}x_{F_1} + Q_{F_2T_1}x_{T_1}. \end{aligned}$$

Com isso e o fato que

$$-(Qx)_{F_2} = -Q_{F_2\mathcal{N}}x = -Q_{F_2T_1}x_{T_1} - Q_{F_2T_2}x_{T_2} - Q_{F_2F_1}x_{F_1} - Q_{F_2}x_{F_2},$$

segue que

$$Q_{F_2}d_{F_2} + Q_{F_2T_2}d_{T_2} = -Q_{F_2}x_{F_2} - Q_{F_2T_2}x_{T_2} - c_{F_2}.$$

Deste modo,

$$(Q_{F_2} \quad Q_{F_2T_2}) \begin{pmatrix} d_{F_2} \\ d_{T_2} \end{pmatrix} = - (Q_{F_2} \quad Q_{F_2T_2}) \begin{pmatrix} x_{F_2} \\ x_{T_2} \end{pmatrix} - c_{F_2}. \quad (2.40)$$

Novamente, por (2.38), temos que

$$Q_{T_2F_1}d_{F_1} + Q_{T_2F_2}d_{F_2} + Q_{T_2T_1}d_{T_1} + Q_{T_2}d_{T_2} = -\omega_{T_2}.$$

Analogamente ao que fizemos anteriormente, usando a relação anterior, (2.39) e (2.1), segue que

$$\begin{aligned} Q_{T_2F_2}d_{F_2} + Q_{T_2}d_{T_2} &= -(Qx + c)_{T_2} + Q_{T_2F_1}x_{F_1} + Q_{T_2T_1}x_{T_1} \\ &= -Q_{T_2F_2}x_{F_2} - Q_{T_2}x_{T_2} - c_{T_2}. \end{aligned}$$

Logo,

$$(Q_{T_2 F_2} \quad Q_{T_2}) \begin{pmatrix} d_{F_2} \\ d_{T_2} \end{pmatrix} = - (Q_{T_2 F_2} \quad Q_{T_2}) \begin{pmatrix} x_{F_2} \\ x_{T_2} \end{pmatrix} - c_{T_2}. \quad (2.41)$$

Portanto, juntando (2.40) e (2.41), obtemos que

$$\begin{pmatrix} Q_{F_2} & Q_{F_2 T_2} \\ Q_{T_2 F_2} & Q_{T_2} \end{pmatrix} \begin{pmatrix} d_{F_2} \\ d_{T_2} \end{pmatrix} = - \begin{pmatrix} Q_{F_2} & Q_{F_2 T_2} \\ Q_{T_2 F_2} & Q_{T_2} \end{pmatrix} \begin{pmatrix} x_{F_2} \\ x_{T_2} \end{pmatrix} - \begin{pmatrix} c_{F_2} \\ c_{T_2} \end{pmatrix}.$$

Com isso e (2.32), temos que

$$d_{F^+} = -x_{F^+} - (Q_{F^+})^{-1} c_{F^+}.$$

Mas  $d_{F^+} = x_{F^+}^+ - x_{F^+}$ . Logo

$$x_{F^+}^+ = -(Q_{F^+})^{-1} c_{F^+},$$

o que mostra que (2.13) vale. Para finalizar, pela definição de  $\omega^+$  e pelo fato que  $\omega_{F^+} = 0$ , segue que

$$\omega_{T^+}^+ = c_{T^+} + Q_{T^+ F^+} x_{F^+}^+$$

e, conseqüentemente,  $\omega_{T^+}^+$  satisfaz (2.14). Logo, os novos iterandos  $x^+$  e  $\omega^+$  do método de Newton satisfazem  $x_{T^+} = \omega_{F^+} = 0$ , e  $x_{F^+}$  e  $\omega_{T^+}$  verificam (2.13) e (2.14), respectivamente. Portanto, os novos iterandos dos métodos de Newton e BPP coincidem e a prova está concluída.  $\square$

Uma explicação para o fato dos métodos de pivotamento em blocos principais serem tão eficientes na prática, para solucionar LCPs com matrizes  $P$ , simétricas ou não, pode estar relacionada com a sua equivalência com o método de Newton, provada no resultado anterior. Cabe salientar que o método de Newton tem convergência quadrática quando o algoritmo converge para um ponto no qual a Jacobiana é não-singular e isto ocorre para a função  $\xi$  quando a solução básica complementar do LCP é não degenerada.

## 2.5 Análise de convergência

Vimos na Seção 2.3 que os Algoritmos BPP, KR e BPP-PC podem ciclar para matrizes- $P$  simétricas e não simétricas, apesar desse fenômeno ser raro na prática. Deste modo, não é possível garantir que esses algoritmos convergem. Portanto, a convergência pode ser garantida somente para algumas subclasses de matrizes- $P$  ou para versões modificadas dos algoritmos.

Nesta seção é realizada uma análise sobre alguns resultados de convergência existentes para os algoritmos de pivotamento em blocos principais. Primeiramente, exibimos a convergência dos Algoritmos BPP e KR para uma subclasse de matrizes simétricas definidas positivas. Em seguida, expomos um algoritmo que combina o Algoritmo BPP com um algoritmo convergente, porém lento, a fim de garantir a convergência do algoritmo para qualquer matriz- $P$  simétrica ou não, e em particular para um LCP estritamente monótono. Ademais, discutimos a impossibilidade de ciclos, quando o Algoritmo BPP é aplicado a problemas em que as matrizes pertencem a uma subclasse de matrizes  $PD$ . Uma versão modificada do Algoritmo BPP-PC é apresentada com convergência para toda matriz  $PD$  simétrica. Por fim, fazemos alguns apontamentos sobre trabalhos futuros.

### 2.5.1 Caso simétrico

Nesta seção é apresentada uma análise de convergência para os Algoritmos BPP e KR. O resultado é garantido exclusivamente para LCPs com matrizes simétricas pertencentes a uma subclasse de matrizes definidas positivas. Esta subclasse é decorrente de uma condição suficiente exigida para as matrizes dos LCPs. O resultado discutido aqui pode ser visto em [83]. Apesar do resultado de convergência apresentado ser estabelecido para o Algoritmo KR, ele pode ser facilmente estendido para Algoritmo BPP.

Para obtenção dos resultados teóricos, suponha que a matriz  $Q \in \mathbb{R}^{n \times n}$  do LCP é simétrica e definida positiva. Cabe salientar que neste caso o LCP é equivalente a  $(PQ)$ .

Considere os seguintes conjuntos

$$X = \{i \in F \mid x_i < 0\}, \quad (2.42)$$

$$W = \{i \in T \mid \omega_i \leq 0\}, \quad (2.43)$$

$$K_1 = \{i \in W \mid x_i^+ < 0\},$$

$$K_2 = \{i \in F - X \mid x_i^+ < 0\},$$

e

$$K = K_1 \cup K_2. \quad (2.44)$$

Note que o conjunto  $K \subset F^+$  contém os índices inviáveis da variável primal  $x^+$ .

Para garantir a convergência do método, vamos analisar as mudanças ocorridas em duas iterações consecutivas. Suponha que alguma iteração é realizada com os conjuntos  $F$  e  $T$  e que  $(x, \omega)$  seja a solução básica complementar associada. Então, vamos denotar

por  $F^+$  e  $T^+$  os conjuntos da iteração consecutiva, atualizados por (2.20), e por  $(x^+, \omega^+)$  a solução básica complementar associada. A Tabela 2.3 apresenta as propriedades das soluções básicas complementares  $(x, \omega)$  e  $(x^+, \omega^+)$  com relação a esses conjuntos.

$T$		$F$			
$(x_i = 0)$		$(\omega_i = 0)$			
$W$	$T \setminus W$	$X$	$F \setminus X$		
$(\omega_i \leq 0)$	$(\omega_i > 0)$	$(x_i < 0)$	$(x_i \geq 0)$		
$\subset F^+$		$T^+$		$\subset F^+$	
$(\omega_i^+ = 0)$		$(x_i^+ = 0)$		$(\omega_i^+ = 0)$	
$K_1$			$K_2$		
$(x_i^+ < 0)$	$(x_i^+ \geq 0)$		$(x_i^+ < 0)$	$(x_i^+ \geq 0)$	

Tabela 2.3: Propriedades de dois iterandos consecutivos  $(x, \omega)$  e  $(x^+, \omega^+)$ .

Para estabelecer resultados de convergência, requeremos a função de mérito Lagrangiana aumentada do problema  $(PQ)$  dada por

$$\mathcal{L}_\rho(x, \omega) = f(x) + \frac{\rho}{2} \|\max\{-x, 0\}\|^2, \quad (2.45)$$

para  $\rho > 0$  e  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  a função de mérito quadrática definida por (2.28). Esta é uma variante da função de mérito Lagrangiana aumentada, estabelecida em [17], dada por

$$L_\rho(x, \omega) = f(x) + \omega^t \max\left\{-x, -\frac{\omega}{\rho}\right\} + \frac{\rho}{2} \left\| \max\left\{-x, -\frac{\omega}{\rho}\right\} \right\|^2.$$

Note que  $\mathcal{L}_\rho(x, \omega) = L_\rho(x, \max\{0, \omega\})$ , uma vez que as condições de complementaridade são satisfeitas, isto é,  $x_i \omega_i = 0$ , para todo  $i \in \mathcal{N}$ .

O próximo resultado estabelece uma relação sobre a variação da função objetivo quadrática (2.28) entre dois iterandos. Cabe salientar que este resultado é obtido sem considerar que os iterandos sejam consecutivos.

**Lema 2.11** *Considere  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  a função de mérito quadrática definida por (2.28). Sejam  $(x, \omega)$  e  $(x^+, \omega^+)$  dois iterandos primal-duais e  $X$  o conjunto definido em (2.42). Considere  $z = x^+ - x$ . Então*

$$f(x^+) - f(x) = z^t \omega^+ - z^t Q z.$$

*Demonstração.* Pelas definições da função  $f$  e de  $z$ , temos que

$$\begin{aligned}
f(x^+) - f(x) &= \frac{1}{2}x^{+t}Qx^+ + c^t x^+ - \frac{1}{2}x^t Qx - c^t x \\
&= \frac{1}{2} \left( x^{+t}Qx^+ - x^t Qx \right) + c^t(x^+ - x) \\
&= x^{+t}Qx^+ - \frac{1}{2}x^{+t}Qx^+ - \frac{1}{2}x^t Qx - x^t Qx^+ + x^t Qx^+ + c^t z \\
&= (x^+ - x)^t Qx^+ - \frac{1}{2}(x^+ - x)^t Q(x^+ - x) + c^t z \\
&= z^t Qx^+ - z^t Qz + c^t z.
\end{aligned}$$

Usando o fato que  $Qx^+ + c = \omega^+$ , temos que

$$f(x^+) - f(x) = z^t \omega^+ - z^t Qz$$

e a demonstração está concluída.  $\square$

Este resultado merece ênfase, uma vez que, este pode ser o começo para uma prova de convergência do algoritmo clássico para LCPs com matrizes- $P$  simétricas. Caso seja possível provar que  $z^t \omega^+ - z^t Qz < 0$  para  $z = x^+ - x$ , a convergência é garantida.

A seguir discutimos alguns resultados auxiliares. Cabe salientar que dado um conjunto de índice  $X$ , denotamos o seu complementar em  $\mathcal{N}$  por  $\bar{X}$ , ou seja,  $\bar{X} = \mathcal{N} \setminus X$ .

**Lema 2.12** [83, Lema 3.1] *Considere  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  a função de mérito quadrática definida por (2.28). Sejam  $(x, \omega)$  e  $(x^+, \omega^+)$  dois iterandos primal-duais consecutivos e  $X$  o conjunto definido em (2.42). Considere  $z = x^+ - x$ . Então*

$$f(x^+) - f(x) = \frac{1}{2} z^t \begin{pmatrix} Q_X & 0 \\ 0 & -Q_{\bar{X}} \end{pmatrix} z.$$

*Demonstração.* Pelas propriedades para dois iterandos consecutivos, como pode ser visto na Tabela 2.3, temos que  $\omega_i^+ = 0$  para todo  $i \in W \cup (F \setminus X)$  e que  $z_i = 0$  para todo  $i \in T \setminus W$ . Por isso e pelo fato que  $\mathcal{N} = [W \cup (T \setminus W)] \cup [X \cup (F \setminus X)]$ , segue que

$$z^t \omega^+ = \sum_{i \in X} z_i \omega_i^+.$$

Além disso,  $\omega_i^+ = [\omega^+ - \omega]_i$ , para todo  $i \in X$ , e por (2.1), obtemos que

$$\omega^+ - \omega = Qx^+ + c - Qx - c = Qz.$$

Portanto,

$$z^t \omega^+ = \sum_{i \in X} z_i [Qz]_i = z_X^t Q_{X\mathcal{N}} z.$$

Usando o lema anterior, o resultado pr vio e a simetria de  $Q$ , temos que

$$\begin{aligned}
 f(x^+) - f(x) &= z_X^t Q_{XN} z - z^t Q z \\
 &= z_X^t (Q_X z_X + Q_{X\bar{X}} z_{\bar{X}}) - z^t Q z \\
 &= z^t \begin{pmatrix} Q_X & \frac{1}{2} Q_{X\bar{X}} \\ \frac{1}{2} Q_{\bar{X}X} & 0 \end{pmatrix} z - \frac{1}{2} z^t \begin{pmatrix} Q_X & Q_{X\bar{X}} \\ Q_{\bar{X}X} & Q_{\bar{X}} \end{pmatrix} z \\
 &= \frac{1}{2} z^t \begin{pmatrix} Q_X & 0 \\ 0 & -Q_{\bar{X}} \end{pmatrix} z
 \end{aligned}$$

e a prova est  completa. □

**Lema 2.13** [83, Lema 3.2] *Sejam  $(x, \omega)$  e  $(x^+, \omega^+)$  duas solu es b sicas complementares consecutivas e  $X$  e  $K$  os conjuntos definidos em (2.42) e (2.44), respectivamente. Ent o,*

$$\|\max\{-x^+, 0\}\|^2 - \|\max\{-x, 0\}\|^2 = \sum_{i \in K} |x_i^+|^2 - \sum_{i \in X} |x_i|^2.$$

*Demonstra o.* O resultado segue imediatamente das defini es dos conjuntos  $X$  e  $K$ . De fato, pelas propriedades presentes na Tabela 2.3, temos que

$$x_i \begin{cases} < 0, & \text{se } i \in X \\ \geq 0, & \text{se } i \in \bar{X} \end{cases} \Rightarrow \max\{-x_i, 0\} = \begin{cases} -x_i, & \text{se } i \in X \\ 0, & \text{se } i \in \bar{X} \end{cases}$$

e

$$x_i^+ \begin{cases} < 0, & \text{se } i \in K = K_1 \cup K_2 \\ \geq 0, & \text{se } i \in \bar{K} \end{cases} \Rightarrow \max\{-x_i^+, 0\} = \begin{cases} -x_i^+, & \text{se } i \in K \\ 0, & \text{se } i \in \bar{K} \end{cases}$$

e a prova est  concluída. □

Para os demais resultados desta se o, vamos denotar por  $\lambda_{\min}$  e  $\lambda_{\max}$  o menor e o maior autovalores de  $Q$ , respectivamente.

**Lema 2.14** *Sejam  $\mathcal{L}_\rho : \mathbb{R}^{2n} \rightarrow \mathbb{R}$  a fun o definida em (2.45),  $\rho > 0$ ,  $(x, \omega)$  e  $(x^+, \omega^+)$  duas solu es complementares consecutivas e  $X$  e  $K$  os conjuntos definidos em (2.42) e (2.44), respectivamente. Considere  $z = x^+ - x$ . Ent o,*

$$2(\mathcal{L}_\rho(x^+, \omega^+) - \mathcal{L}_\rho(x, \omega)) \leq (\|Q\| - \rho) \|z_X\|^2 - \lambda_{\min} \|z_{\bar{X}}\|^2 + \rho \|z_K\|^2. \quad (2.46)$$

*Demonstração.* Pela definição do conjunto  $K$  temos que, para todo  $i \in K$ ,  $x_i \geq 0$  e  $x_i^+ < 0$  (veja Tabela 2.3). Então,  $0 < -x_i^+ \leq x_i - x_i^+ = -z_i$ , para  $i \in K$ , e conseqüentemente

$$\sum_{i \in K} (x_i^+)^2 \leq \sum_{i \in K} (z_i)^2 = \|z_K\|^2. \quad (2.47)$$

Além disso, como  $x_X^+ = 0$  temos que

$$\sum_{i \in X} (x_i)^2 = \sum_{i \in X} (x_i^+ - x_i)^2 = \|z_X\|^2. \quad (2.48)$$

Considere  $\lambda_{\max}^X$  o maior autovalor de  $Q_X$ . Assim,

$$\lambda_{\max}^X = \|Q_X\| \leq \|Q\| = \lambda_{\max}. \quad (2.49)$$

Considere  $\lambda_{\min}^{\bar{X}}$  o menor autovalor de  $Q_{\bar{X}}$ , então

$$-z_{\bar{X}}^t(Q_{\bar{X}})z_{\bar{X}} \leq -\lambda_{\min}^{\bar{X}} \|z_{\bar{X}}\|^2 \leq -\lambda_{\min} \|z_{\bar{X}}\|^2. \quad (2.50)$$

Usando a definição de  $\mathcal{L}_\rho(\cdot, \cdot)$  e os Lemas 2.12 e 2.13, segue que

$$\begin{aligned} 2(\mathcal{L}_\rho(x^+, \omega^+) - \mathcal{L}_\rho(x, \omega)) &= 2(f(x^+) - f(x)) + \rho \left( \|\max\{-x^+, 0\}\|^2 - \|\max\{-x, 0\}\|^2 \right) \\ &= z^t \begin{pmatrix} Q_X & 0 \\ 0 & -Q_{\bar{X}} \end{pmatrix} z + \rho \left( \sum_{i \in K} |x_i^+|^2 - \sum_{i \in X} |x_i|^2 \right). \end{aligned}$$

Com isso, os resultados (2.47) e (2.48), a desigualdade de Cauchy-Schwarz, propriedades de normas, (2.50), a definição de  $\lambda_{\min}$  e (2.49) temos que

$$\begin{aligned} 2(\mathcal{L}_\rho(x^+, \omega^+) - \mathcal{L}_\rho(x, \omega)) &\leq (z_X)^t(Q_X)(z_X) - (z_{\bar{X}})^t(Q_{\bar{X}})(z_{\bar{X}}) + \rho (\|z_K\|^2 - \|z_X\|^2) \\ &\leq \|Q_X\| \|z_X\|^2 - \lambda_{\min} \|z_{\bar{X}}\|^2 + \rho (\|z_K\|^2 - \|z_X\|^2) \\ &\leq \|Q\| \|z_X\|^2 - \lambda_{\min} \|z_{\bar{X}}\|^2 + \rho (\|z_K\|^2 - \|z_X\|^2) \end{aligned}$$

e a prova está completa. □

Consideramos agora as seguintes quantidades:

- $\nu = \max\{\|Q_{T\bar{T}}\| \mid T \subset \mathcal{N}, T \neq \emptyset, T \neq \mathcal{N}\}$ , sendo  $\|\cdot\|$  a norma espectral,
- $D = \text{diag}(Q)$ ,
- $q = \min\{Q_{ii} \mid i \in \mathcal{N}\}$  o menor elemento diagonal de  $Q$ ,
- $r = \|Q - D\|$ ,

- $\text{cond}(Q) = \frac{\lambda_{\max}}{\lambda_{\min}}$ .

De agora em diante, iremos considerar a subclasse de matrizes definidas positivas e satisfazem pelo menos uma das seguintes condições consideradas em [83] dadas por

$$(C_1) \quad \text{cond}(Q) < \left(\frac{\lambda_{\min}}{\nu}\right)^2 - 1 \quad (2.51)$$

ou

$$(C_2) \quad \text{cond}(Q) < \left(\frac{q}{r}\right)^2 - 1. \quad (2.52)$$

Como mencionado naquele trabalho, estas condições exigem que a matriz  $Q$  satisfaça uma condição mais forte do que ser diagonalmente dominante estrita com elementos diagonais positivos. Observe que  $\nu$  é independente da diagonal de  $Q$  e pode ser limitado por  $0 < \nu \leq \|Q - D\| = r$ . Assim,

$$\left(\frac{1}{r}\right)^2 \leq \left(\frac{1}{\nu}\right)^2.$$

Portanto a condição

$$\text{cond}(Q) < \left(\frac{\lambda_{\min}}{r}\right)^2 - 1,$$

implica  $(C_1)$  e esta exige que  $r = \|Q - D\| < \lambda_{\min}$ . Além disso, para que a condição  $(C_2)$  seja verificada, temos que  $\frac{r}{q} < 1$ , então

$$1 > \frac{r}{q} = \frac{\|Q - D\|}{\min_{i \in \mathcal{N}} \{Q_{ii}\}} = \|Q - D\| \|D^{-1}\| \geq \|QD^{-1} - I\|.$$

Como

$$[QD^{-1} - I]_{ij} = \begin{cases} 0, & \text{para } i = j \\ \frac{Q_{ij}}{Q_{jj}}, & \text{para } i \neq j, \end{cases}$$

o resultado anterior mostra a exigência sobre a diagonal, requerida para a matriz  $Q$ . Esta dominância diagonal é discutida com mais detalhes em [83].

Por outro lado, note que uma matriz estritamente diagonalmente dominante não satisfaz necessariamente  $(C_1)$  e nem  $(C_2)$ . Apresentamos a seguir um exemplo que comprova este fato. Considere a matriz  $PD$  e estritamente diagonalmente dominante dada por

$$Q = \begin{pmatrix} 2 & 1 \\ 1 & 1.5 \end{pmatrix}, \quad (2.53)$$

com  $\nu = r = 1$ ,  $q = 1.5$  e autovalores  $\lambda_{\min} = 0.7192$  e  $\lambda_{\max} = 2.7808$ . Assim,

$$\left(\frac{\lambda_{\min}}{\nu}\right)^2 - 1 = -0.4827$$

e

$$\left(\frac{q}{r}\right)^2 - 1 = 1.25.$$

Como  $\text{cond}(Q) = 3.8665$ , concluímos que esta matriz não satisfaz  $(C_1)$  e nem  $(C_2)$ . Este exemplo confirma que as condições  $(C_1)$  e  $(C_2)$  são, de fato, muito restritivas.

A seguir expomos a convergência do Algoritmo KR para LCPs satisfazendo uma destas condições.

**Teorema 2.15** [83, Teorema 3.4] *Sejam  $(x, \omega)$  e  $(x^+, \omega^+)$  duas soluções básicas complementares consecutivas e  $\rho = \|Q\| + \lambda_{\min}$ . Se pelo menos uma das condições  $(C_1)$  ou  $(C_2)$  vale, então existe uma constante  $\hat{c} < 0$  tal que*

$$2(\mathcal{L}_\rho(x^+, \omega^+) - \mathcal{L}_\rho(x, \omega)) \leq \hat{c} \|x^+ - x\|^2 < 0. \quad (2.54)$$

*Em ambos os casos, o algoritmo termina em um número finito de iterações.*

*Demonstração.* Considere  $z = x^+ - x$ . Primeiramente, vamos limitar  $\|z_K\|$  em termos de  $\|z\|$ . Pela definição do conjunto  $K_1$  (veja Tabela 2.3) temos que,  $\omega_{K_1} \leq 0$ ,  $\omega_{K_1}^+ = 0$  e  $z_{K_1} < 0$ . Consequentemente,

$$(Qz)_{K_1} = (Qx^+ - Qx)_{K_1} = (\omega^+ - \omega)_{K_1} = -\omega_{K_1} \geq 0. \quad (2.55)$$

Além disso, pela definição de  $K_2$  temos que  $\omega_{K_2} = 0$  e  $\omega_{K_2}^+ = 0$  e assim,

$$(Qz)_{K_2} = (\omega^+ - \omega)_{K_2} = 0. \quad (2.56)$$

Usando a definição de  $K$ , (2.55) e (2.56), obtemos

$$(Qz)_K = (-\omega_{K_1}^t \ 0)^t. \quad (2.57)$$

Por fim, usando (2.57) e os fatos que  $z_{K_1} < 0$  e  $\omega_{K_1} \leq 0$  segue que

$$z_K^t (Q_K z_K + Q_{K\bar{K}} z_{\bar{K}}) = z_K^t (Qz)_K = -z_{K_1}^t \omega_{K_1} \leq 0. \quad (2.58)$$

Agora vamos analisar as duas condições separadamente.

Primeiramente, vamos assumir que a condição  $(C_1)$  é satisfeita. Usando (2.58) e o fato que  $Q_K$  é uma matriz definida positiva, temos que

$$0 \leq z_K^t Q_K z_K \leq -z_K^t Q_{K\bar{K}} z_{\bar{K}}.$$

Usando a desigualdade acima, propriedades de normas, a definição de  $\lambda_{\min}$ , a desigualdade de Cauchy-Schwarz e a definição de  $\nu$ , segue que

$$\lambda_{\min} \|z_K\|^2 \leq |z_K^t Q_K z_K| \leq |z_K^t Q_{K\bar{K}} z_{\bar{K}}| \leq \|z_K\| \|z_{\bar{K}}\| \|Q_{K\bar{K}}\| \leq \nu \|z_K\| \|z_{\bar{K}}\|.$$

Assim, pela relação anterior e pelas propriedades de normas, obtemos que

$$\|z_K\|^2 \leq \frac{\nu^2}{\lambda_{\min}^2} \|z_{\bar{K}}\|^2 \leq \frac{\nu^2}{\lambda_{\min}^2} (\|z_X\|^2 + \|z_{\bar{X}}\|^2). \quad (2.59)$$

Pelo Lema 2.14, (2.59) e a hipótese que  $\rho = \|Q\| + \lambda_{\min}$ , temos que

$$\begin{aligned} 2(\mathcal{L}_\rho(x^+, \omega^+) - \mathcal{L}_\rho(x, \omega)) &\leq \left( \|Q\| - \rho + \rho \frac{\nu^2}{\lambda_{\min}^2} \right) \|z_X\|^2 + \left( \rho \frac{\nu^2}{\lambda_{\min}^2} - \lambda_{\min} \right) \|z_{\bar{X}}\|^2 \\ &= \left( (\|Q\| + \lambda_{\min}) \frac{\nu^2}{\lambda_{\min}^2} - \lambda_{\min} \right) (\|z_X\|^2 + \|z_{\bar{X}}\|^2). \end{aligned}$$

Pela condição  $(C_1)$ , temos que  $\|Q\| + \lambda_{\min} < \lambda_{\min} \frac{\lambda_{\min}^2}{\nu^2}$ . Deste modo, tomando

$$\hat{c} = \left( (\|Q\| + \lambda_{\min}) \frac{\nu^2}{\lambda_{\min}^2} - \lambda_{\min} \right) < 0,$$

obtemos (2.54).

Vamos supor agora que a condição  $(C_2)$  é válida. Por (2.58) obtemos que

$$0 \geq z_K^t (Qz)_K = z_K^t (D_K z_K + (Q_{KN} - D_{KN})z).$$

Assim, usando propriedades de normas, a definição de  $q$ , o fato acima, a desigualdade de Cauchy-Schwarz e a definição de  $r$ , segue que

$$q \|z_K\|^2 \leq z_K^t D_K z_K \leq z_K^t (D_{KN} - Q_{KN})z \leq \|D_{KN} - Q_{KN}\| \|z_K\| \|z\| \leq r \|z_K\| \|z\|.$$

Deste modo,

$$\|z_K\|^2 \leq \frac{r^2}{q^2} \|z\|^2 = \frac{r^2}{q^2} (\|z_X\|^2 + \|z_{\bar{X}}\|^2). \quad (2.60)$$

Similarmente ao primeiro caso, pelo Lema 2.14, por (2.60) e tomando  $\rho = \|Q\| + \lambda_{\min}$ , obtemos

$$\begin{aligned} 2(\mathcal{L}_\rho(x^+, \omega^+) - \mathcal{L}_\rho(x, \omega)) &\leq \left( \|Q\| - \rho + \rho \frac{r^2}{q^2} \right) \|z_X\|^2 + \left( \rho \frac{r^2}{q^2} - \lambda_{\min} \right) \|z_{\bar{X}}\|^2 \\ &= \left( (\|Q\| + \lambda_{\min}) \frac{r^2}{q^2} - \lambda_{\min} \right) (\|z_X\|^2 + \|z_{\bar{X}}\|^2). \end{aligned}$$

Tomando

$$\hat{c} = \left( (\|Q\| + \lambda_{\min}) \frac{r^2}{q^2} - \lambda_{\min} \right),$$

usando a condição  $(C_2)$ , temos que  $\hat{c} < 0$  e o resultado segue.

Resumindo, caso a matriz satisfaça a condição  $(C_1)$  ou  $(C_2)$ , o valor da função de mérito sempre diminui de uma iteração para a outra. Portanto, como cada solução básica é única, é impossível que o algoritmo considere um conjunto  $T$  mais de uma vez. Como existem no máximo  $2^n$  possibilidades de conjuntos  $T$ , o algoritmo obtém a solução ótima em um número finito de iterações.  $\square$

O Algoritmo BPP com  $H' = H$  também converge em um número finito de iterações se a condição  $(C_1)$  ou  $(C_2)$  vale. A prova é similar, basta trocar a definição do conjunto  $W$  dado em (2.43) por

$$W = \{i \in T \mid \omega_i < 0\}.$$

Apesar da convergência garantida, as condições  $(C_1)$  e  $(C_2)$  são muito fortes, como pode ser visto no parágrafo que contém (2.53). A procura por condições suficientes mais fracas será um tópico muito interessante de investigação futura.

## 2.5.2 Combinação com método convergente

Nesta seção apresentamos o algoritmo proposto em [76] que combina o Algoritmo BPP com o Algoritmo Murty proposto em [97]. O algoritmo Murty considera, em cada iteração, o conjunto de atualizações com cardinalidade um. Assim, embora seja convergente, é lento na prática. Este algoritmo híbrido herda a convergência finita do Algoritmo Murty para todo LCP com uma matriz- $P$ , em particular para um LCP estritamente monótono.

A ideia sugerida em [76] é manter as atualizações (2.17) e (2.18) do algoritmo BPP com  $H' = H$ , como definido em (2.16), enquanto o número de inviabilidades está reduzindo. Se em alguma iteração este número aumenta, então o valor da cardinalidade da iteração anterior é guardado na variável  $n_{inf}$  e começa a contagem de  $(p-1)$  iterações, para alguma constante  $p \in \mathbb{N}$  (autores de [76] sugerem  $p = 10$ ). Se durante estas iterações o número de inviabilidade se reduz para um valor menor que  $n_{inf}$ , então o algoritmo finaliza esta contagem de  $(p-1)$  iterações e continua normalmente. Caso contrário, são realizadas iterações do algoritmo Murty, o qual considera

$$H' = \min\{i \in H\}, \tag{2.61}$$

até que seja obtida uma solução básica complementar com um número de inviabilidades menor que  $n_{inf}$ . A partir deste ponto, o algoritmo volta a realizar atualizações com  $H' = H$ . Caso não seja obtida uma solução básica complementar com um número de inviabilidades menor que  $n_{inf}$ , o Algoritmo Murty é utilizado até encontrar a solução. A convergência deste algoritmo híbrido é garantida pela terminação finita do Algoritmo Murty [97]. Denominaremos o algoritmo proposto em [76] de BPP-M.

---

**Algoritmo 2.4** BPP-M
 

---

Dados:  $k = 0$ ,  $T \subseteq \mathcal{N}$ ,  $F = \mathcal{N} \setminus T$ ,  $p \in \mathbb{N}$ ,  $n_{inf} = n$ ,  $PV_{max} = n$

REPITA

    Calcule  $c'$  por (2.13) e (2.14) e  $(x, \omega)$  por (2.15)

    Calcule  $H$  por (2.16)

    SE  $H = \emptyset$

        Pare com  $(x, \omega)$  solução do LCP

    SENÃO

        SE  $|H| < n_{inf}$

$n_{inf} = |H|$ ,

$PV_{max} = k + p$

$H' = H$

        SENÃO

            SE  $k \leq PV_{max}$

$H' = H$

            SENÃO

$H' = \min\{i \in H\}$

    Atualize  $F$  e  $T$  por (2.17) e (2.18), respectivamente

$k = k + 1$

---

Este algoritmo tem convergência finita para a única solução de um LCP com matriz- $P$  não necessariamente simétrica. Em particular, obtém teoricamente a solução única de um LCP estritamente monótono. A convergência é garantida pela existência das atualizações de cardinalidade um do método de Murty. Esse tipo de atualizações provoca lentidão no Algoritmo BPP. No entanto, iterações deste tipo raramente são requeridas na prática [50].

### 2.5.3 Impossibilidade de ciclos

A subclasse de matrizes definidas positivas considerada na Seção 2.5.1 não é a única para a qual o Algoritmo BPP converge. Em [124], é verificado que para qualquer matriz  $Q \in P$  que seja triangular inferior ou superior, o Algoritmo BPP encontra a solução em no máximo  $n$  passos. Ademais, os autores também verificam que o Algoritmo BPP encontra a solução em um número finito de iterações quando aplicado a LCPs de ordem 3 com matrizes  $SRDD_+$  e  $SCDD_+$ .

A verificação do primeiro fato é realizada por indução e pode ser vista nas Proposições 1 e 2 do mesmo artigo. Já a demonstração do segundo resultado é árdua, pois tem de mostrar que não é possível haver ciclos nesse caso. Esse processo é difícil de estender para matrizes de ordem superior a 3, mas indica que possivelmente os Algoritmos BPP e BPP-PC tenham convergência finita para matrizes das classes  $SRDD_+$  e  $SCDD_+$ , uma conjectura apresentada por Júdice em [124], mas que até o momento se encontra em aberto.

### 2.5.4 Técnica de marcação de variáveis

Nesta seção discutimos uma versão modificada do algoritmo BPP-PC que possui convergência finita para todo LCP estritamente monótono com uma matriz simétrica. Este algoritmo é denominado de Algoritmo KR Modificado (*Modified KR Algorithm - (MKR)*) e foi proposto em [72] como um algoritmo de conjuntos ativos. Salientamos que, neste contexto, o LCP é equivalente ao  $(PQ)$ .

De acordo com os autores, o Algoritmo MKR é uma extensão do Algoritmo KR apresentado anteriormente. Como a ideia deste algoritmo consiste em, a cada iteração, obter uma solução básica complementar primal viável e em seguida obter a solução básica complementar dual, não necessariamente viável, podemos identificar este algoritmo como uma extensão do algoritmo BPP-PC. Veremos que, como no Algoritmo KR, o Algoritmo MKR também pode ser considerado como um algoritmo de pivotamento em blocos principais.

Considere dois conjuntos  $F$  e  $T$  satisfazendo (2.11) e a solução básica complementar  $(x, \omega)$  associada a estes conjuntos dada por (2.15). Como no Algoritmo BPP-PC, se esta solução é primal viável, então  $H' = H_\omega$  sendo  $H_\omega$  dado por

$$H_\omega = \{i \in T \mid c'_i < 0\}$$

e os conjuntos  $F$  e  $T$  são atualizados de acordo com (2.17) e (2.18), respectivamente. Os conjuntos resultantes destas atualizações são denotados de  $\widehat{F}$  e  $\widehat{T}$  e, segundo (2.17) e (2.18), são dados por

$$\widehat{T} = T \setminus H_\omega \quad (2.62)$$

e

$$\widehat{F} = F \cup H_\omega.$$

Por outro lado, se a solução não for primal viável, então  $H' = H_x$  com

$$H_x = \{i \in F \mid c'_i \leq 0\}$$

e o conjunto  $F$  é atualizado segundo (2.17). Este processo, considerando  $H' = H_x$  é repetido sucessivamente, até que uma solução primal viável seja obtida e, então o novo conjunto  $T$  é dado por  $T = \mathcal{N} \setminus F$ . Note que estes são exatamente os processos utilizados no Algoritmo BPP-PC, porém como já vimos, este algoritmo é cíclico. Para evitar ciclos, o algoritmo utiliza a função de mérito quadrática  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  definida por (2.28) e realiza algumas regras de atualizações específicas que garantem um decréscimo em  $f$ , como será apresentado nos resultados de convergência. Nessas atualizações, dado um conjunto  $S \subset \mathcal{N}$ , o algoritmo anula as componentes da variável  $x$  referentes aos índices de  $S$  e resolve, com o mesmo algoritmo, o subproblema resultante, ou seja, o subproblema dado por

$$\begin{cases} \text{minimizar} & \frac{1}{2}x^t Qx + c^t x \\ \text{sujeito a} & x_{\overline{S}} \geq 0, \\ & x_S = 0. \end{cases} \quad (2.63)$$

O conjunto  $T_0 \subset \overline{S}$  é dito *conjunto ótimo* deste subproblema quando  $x_{T_0} = 0$  e  $x_{\overline{T_0}} \geq 0$ . É neste momento que a recursividade é requerida no algoritmo. Sintetizamos a seguir o Algoritmo MKR num formato de algoritmos de pivotamento em blocos principais, que embora seja diferente do formato apresentado em [72], é equivalente. Diferentemente dos algoritmos anteriores, adotamos que o número de sistemas (2.13) resolvidos é denotado por  $k_s$  e o número de iterações por  $k$ . A quantidade de fases realizadas para recuperação da viabilidade primal é denotada por  $k_v$ .

Cabe salientar que este algoritmo é muito mais complexo que o Algoritmo BPP-PC. Além disso, o número de sistemas (2.13) resolvidos coincide com  $k_s$ .

---

**Algoritmo 2.5** MKR
 

---

Dados:  $k = 0$ ,  $k_v = 0$ ,  $k_s = 0$ ,  $F \subseteq \{1, 2, \dots, n\}$ ,  $T = \mathcal{N} \setminus F$

Calcule  $c'_F$ ,  $x$  e  $H_x$  por (2.13), (2.15) e (2.22), respectivamente e faça  $k_s = 1$

Se  $H_x \neq \emptyset$ , então  $k_v = k_v + 1$

ENQUANTO  $H_x \neq \emptyset$

    Atualize  $F$  por (2.17) com  $H' = H_x$

    Calcule  $c'_F$ ,  $x$  e  $H_x$  por (2.13), (2.15) e (2.22), respectivamente

$k_s = k_s + 1$

Faça  $T = \mathcal{N} \setminus F$  e calcule  $c'_T$  e  $\omega$  por (2.14) e (2.15), respectivamente

Faça  $F^0 = F$ ,  $T^0 = T$ ,  $x^0 = x$  e  $\omega^0 = \omega$

REPITA

    Calcule  $H_w$  por (2.23)

    Atualize  $F$  e  $T$  por (2.17) e (2.18), respectivamente, com  $H' = H_w$

    Calcule  $c'_F$ ,  $x$  e  $H_x$  por (2.13), (2.15) e (2.22), respectivamente

$k_s = k_s + 1$ ,  $\widehat{T} = T$  e  $\widehat{F} = F$

    Se  $H_x \neq \emptyset$ , então  $k_v = k_v + 1$

    ENQUANTO  $H_x \neq \emptyset$

        Atualize  $F$  por (2.17) com  $H' = H_x$

        Calcule  $c'_F$ ,  $x$  e  $H_x$  por (2.13), (2.15) e (2.22), respectivamente

$k_s = k_s + 1$

    Se  $f(x) \geq f(x^k)$

        Se  $|T^k| = 1$ , resolva o subproblema com a única variável de  $T$  removida

    Senão

        Tome  $T_1 = T \cap T^k \setminus \widehat{T}$  e  $F_1 = T^k \cap F$

        Considere  $\widehat{T}_0$  tal que  $\widehat{T}_0 \neq \emptyset$ ,  $\widehat{T}_0 \subseteq T^k$  e  $T_1 \cup F_1 \setminus \widehat{T}_0 \neq \emptyset$  e  $\widehat{F}_0 = \mathcal{N} \setminus \widehat{T}_0$

        Seja  $T_0$  o conjunto ótimo do subproblema (2.63) com  $S = \widehat{T}_0$

$T = T_0 \cup \widehat{T}_0$  e  $F = \mathcal{N} \setminus T$

        Calcule  $\bar{c}$  por (2.13) e (2.14) e  $(x, \omega)$  por (2.15)

$k_s = k_s + 1$

    Senão

        Faça  $T = \mathcal{N} \setminus F$  e calcule  $c'_T$  e  $\omega$  por (2.14) e (2.15), respectivamente

$(x^{k+1}, \omega^{k+1}) = (x, \omega)$ ,  $T^{k+1} = T$ ,  $F^{k+1} = F$  e  $k = k + 1$

---

### Convergência global do algoritmo

Vamos denotar por  $\mathcal{P} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  a função projeção Euclidiana de um ponto em  $\mathbb{R}^n$  sobre o conjunto viável do problema  $(PQ)$ . Como o conjunto viável deste problema é o  $\mathbb{R}_+^n$ , esta projeção possui a seguinte fórmula fechada

$$\mathcal{P}(x) = x + r(x), \quad (2.64)$$

sendo

$$r(x) = \max\{-x, 0\}. \quad (2.65)$$

Note que

$$[\mathcal{P}(x)]_i = \begin{cases} x_i & \text{se } x_i \geq 0 \\ 0 & \text{se } x_i < 0. \end{cases} \quad (2.66)$$

Além disso, se  $(x, \omega)$  é uma solução básica complementar, então  $(r(x))^t \omega = 0$ .

Para a prova de convergência iremos considerar a função de mérito  $L : \mathbb{R}^{2n} \rightarrow \mathbb{R}$  dada por

$$L(x) = f(x) + \frac{1}{2}(r(x))^t Q(r(x)),$$

sendo  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  a função de mérito quadrática dada em (2.28). No lema a seguir, uma propriedade interessante desta função é apresentada.

**Lema 2.16** [72, Lema 4] *Seja  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  a função de mérito quadrática definida por (2.28). Considere  $(x, \omega)$  uma solução básica complementar de (2.12). Então,*

$$L(x) = f(\mathcal{P}(x)),$$

sendo  $\mathcal{P}(\cdot)$  como definido em (2.64).

*Demonstração.* Como  $(x, \omega)$  é uma solução básica complementar, temos que  $(r(x))^t \omega = 0$  e  $\omega = Qx + c$ . Assim, por estes fatos e pela definição de  $f$ , dada em (2.28), segue que

$$\begin{aligned} \frac{1}{2}(r(x))^t Q(r(x)) &= \frac{1}{2}(r(x))^t Q(r(x)) + (r(x))^t \omega \\ &= \frac{1}{2}(r(x))^t Q(r(x)) + (r(x))^t (Qx + c) \\ &= f(r(x)) + (r(x))^t Qx. \end{aligned}$$

Deste modo, usando isto na definição de  $L(x)$ , juntamente com a definição de  $f$ , da projeção  $\mathcal{P}(\cdot)$  e de  $r(\cdot)$  dada em (2.65), temos que

$$\begin{aligned} L(x) &= f(x) + f(r(x)) + (r(x))^t Qx \\ &= \frac{1}{2}x^t Qx + x^t c + \frac{1}{2}(r(x))^t Q(r(x)) + (r(x))^t c + (r(x))^t Qx \\ &= f(x + r(x)) \\ &= f(\mathcal{P}(x)) \end{aligned}$$

e a prova está completa.  $\square$

Note que quando um par  $(x, \omega)$  é primal viável, então  $\mathcal{P}(x) = x$ . Deste modo, de acordo com o último lema, quando estivermos em uma solução primal viável,

$$L(x) = f(x),$$

e  $L(x)$  coincide com a função de mérito dada em (2.28). Logo, para este algoritmo, podemos considerar ambas as funções de mérito  $f$  e  $L$ .

A seguir, vamos analisar as mudanças ocorridas em duas iterações consecutivas. Para tanto, considere uma iteração  $k$  do algoritmo. Sejam  $F^k$ ,  $T^k$ ,  $\widehat{T} = \widehat{T}(k)$  e  $\widehat{F} = \widehat{F}(k)$  os conjuntos relacionados à iteração  $k$  e  $T^{k+1}$  e  $F^{k+1}$ . Note que, pelo mecanismo do algoritmo, os elementos de  $\widehat{T}(k)$  pertencerão ao conjunto  $T^{k+1}$ . Já os elementos de  $T^k$  podem pertencer tanto ao conjunto  $T^{k+1}$  como ao  $F^{k+1}$ . Assim, denote por  $\widehat{T} \cup T_1$  o conjunto dos elementos de  $T^k$  que pertencem a  $T^{k+1}$  e por  $F_1$  o conjunto dos elementos de  $T^k$  que pertencem a  $F^{k+1}$ . De maneira similar, denote os elementos de  $F^k$  que pertencem a  $T^{k+1}$  por  $T_2$  e que pertencem a  $F^{k+1}$  por  $F_2$ . Note que desta forma,  $T_1 \cup F_1 \subseteq T^k$ ,

$$T^k = \widehat{T} \cup T_1 \cup F_1, \quad T^{k+1} = \widehat{T} \cup T_1 \cup T_2,$$

$$F^k = T_2 \cup F_2 \quad \text{e} \quad F^{k+1} = F_1 \cup F_2.$$

Estes fatos estão resumidos na Tabela 2.4. A relação das variáveis primais e duais com estes conjuntos é apresentada na Tabela 2.5.

**Lema 2.17** [72, Lema 6] *Considere um conjunto  $T^k$  com solução básica complementar associada  $(x^k, \omega^k)$  primal viável. Então,  $\widehat{T} \subseteq T^k$  e  $\widehat{T} = T^k$  se, e somente se,  $(x^k, \omega^k)$  é uma solução do LCP. Além disso, se existe  $i \in T^k$  tal que  $\omega_i^k < 0$ , então  $T^k \neq T^{k+1}$  e  $(x^k - x^{k+1}) \neq 0$ .*

$T^k$			$F^k$	
$\widehat{T}$	$\widehat{F}$			
$\widehat{T}$	$T_1$	$F_1$	$T_2$	$F_2$
$\subseteq T^{k+1}$		$\subseteq F^{k+1}$	$\subseteq T^{k+1}$	$\subseteq F^{k+1}$

Tabela 2.4: Atualizações dos conjuntos  $F^k$  e  $T^k$  para os conjuntos  $F^{k+1}$  e  $T^{k+1}$ .

	$T^k$			$F^k$	
	$\widehat{T}$	$T_1$	$F_1$	$T_2$	$F_2$
$x^k$	= 0	= 0	= 0	> 0	> 0
$x^{k+1}$	= 0	= 0	> 0	= 0	> 0
$\omega^k$	$\geq 0$	< 0	< 0	= 0	= 0
$\omega^{k+1}$	?	?	= 0	?	= 0
	$\subseteq T^{k+1}$		$\subseteq F^{k+1}$	$\subseteq T^{k+1}$	$\subseteq F^{k+1}$

Tabela 2.5: Relação das soluções complementares básicas primais viáveis  $(x^k, \omega^k)$  e  $(x^{k+1}, \omega^{k+1})$  com os conjuntos  $F^k$ ,  $T^k$ ,  $F^{k+1}$  e  $T^{k+1}$ .

*Demonstração.* Note que, pela definição de  $\widehat{T}$ , dada em (2.62), segue imediatamente que  $\widehat{T} \subseteq T^k$ . Além disso, se  $T^k$  fornece uma solução básica complementar viável, ou seja, solução do LCP, então  $H_\omega = \emptyset$  e por (2.62), temos que  $\widehat{T} = T^k$ . Reciprocamente, caso  $\widehat{T} = T^k$ , temos que  $H_\omega = \emptyset$  e, portanto, a solução básica complementar é dual viável ( $\omega^k \geq 0$ ). Como  $x^k \geq 0$  e  $(x^k, \omega^k)$  é dado por (2.15), então todas as condições KKT são satisfeitas em  $(x^k, \omega^k)$ . Portanto,  $(x^k, \omega^k)$  é a solução do LCP.

Para provar a última implicação do lema, considere  $T^k$  tal que  $(x^k, \omega^k)$  é primal viável, mas é dual inviável. Note que  $T^k \neq \emptyset$ , pois caso contrário  $\omega^k = 0$ , o que contradiz o fato de ser inviável. Suponha, por contradição, que  $T^k = T^{k+1}$ . Consequentemente,  $F^k = F^{k+1}$ . Assim, segundo as propriedades descritas na Tabela 2.4, segue que  $F_1 = T_2 = \emptyset$ . Além disso, como  $\omega^k$  não é viável, segue da primeira parte desta demonstração que  $\widehat{T} \neq T^k$ . Portanto, juntando estes fatos, temos que  $F_2 = F^k = F^{k+1}$  e  $T_1 = T^k \setminus \widehat{T} \neq \emptyset$ . Assim, usando o fato que  $F^k = F^{k+1}$  e que  $x^k$  e  $x^{k+1}$  são dados por (2.13), obtemos que

$$x_{F^k}^k = -Q_{F^k}^{-1} c_{F^k} \quad \text{e} \quad x_{F^k}^{k+1} = -Q_{F^k}^{-1} c_{F^k}. \quad (2.67)$$

Além disso, por (2.1), pelo fato que  $\mathcal{N} = \widehat{T} \cup T^1 \cup F^k$  e pelas propriedades descritas na

Tabela 2.5, temos que

$$Q_{T_1\mathcal{N}} \begin{pmatrix} x_{\widehat{T}}^k \\ x_{T_1}^k \\ x_{F^k}^k \end{pmatrix} + c_{T_1} = Q_{T_1\mathcal{N}} \begin{pmatrix} 0 \\ 0 \\ x_{F^k}^k \end{pmatrix} + c_{T_1} = \omega_{T_1}^k < 0$$

e

$$Q_{T_1\mathcal{N}} \begin{pmatrix} x_{\widehat{T}}^{k+1} \\ x_{T_1}^{k+1} \\ x_{F^k}^{k+1} \end{pmatrix} + c_{T_1} = Q_{T_1\mathcal{N}} \begin{pmatrix} 0 \\ 0 \\ x_{F^k}^{k+1} \end{pmatrix} + c_{T_1} = \omega_{T_1}^{k+1} = 0.$$

Usando isto e (2.67)

$$-Q_{T_1F^k}(Q_{F^k}^{-1}c_{F^k}) + c_{T_1} < -Q_{T_1F^k}(Q_{F^k}^{-1}c_{F^k}) + c_{T_1}$$

o que é uma contradição. Portanto,  $T^k \neq T^{k+1}$ .

Para finalizar, note que pelas propriedades descritas na Tabela 2.5,

$$x_{F^k}^k > 0, \quad x_{F^{k+1}}^{k+1} > 0, \quad \omega_{F^k}^k = 0, \quad \omega_{F^{k+1}}^{k+1} = 0.$$

Assim, como  $T^k \neq T^{k+1}$ ,  $F^k \neq F^{k+1}$  e vale a complementaridade, temos que  $(x^k - x^{k+1}) \neq 0$  e a prova está concluída. □

**Lema 2.18** [72, Lema 7] *Considere  $(x^k, \omega^k)$  e  $(x^{k+1}, \omega^{k+1})$  duas soluções complementares básicas primais viáveis consecutivas. Seja o conjunto  $T_2$  como definido anteriormente e a função  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  definida em (2.28). Então,*

$$f(x^{k+1}) - f(x^k) = -\frac{1}{2}(x^{k+1} - x^k)^t Q(x^{k+1} - x^k) + \sum_{i \in T_2} (x_i^{k+1} - x_i^k) \omega_i^{k+1}.$$

Além disso,  $[x^{k+1} - x^k]_{T_2} < 0$ .

*Demonstração.* Pela definição de  $f$  e pelo fato que  $Qx^{k+1} + c = \omega^{k+1}$  temos que

$$\begin{aligned} f(x^{k+1}) - f(x^k) &= -\frac{1}{2}(x^{k+1} - x^k)^t Q(x^{k+1} - x^k) + (x^{k+1} - x^k)^t (Qx^{k+1} + c) \\ &= -\frac{1}{2}(x^{k+1} - x^k)^t Q(x^{k+1} - x^k) + (x^{k+1} - x^k)^t \omega^{k+1}. \end{aligned}$$

Segundo as propriedades descritas na Tabela 2.5, temos que  $[x^{k+1} - x^k]_i = 0$  para  $i \in \widehat{T} \cup T_1$  e  $\omega_i^{k+1} = 0$  para  $i \in F^{k+1} = F_1 \cup F_2$ . Assim,

$$(x^{k+1} - x^k)^t \omega^{k+1} = \sum_{i \in T_2} (x_i^{k+1} - x_i^k) \omega_i^{k+1}$$

e a demonstração está concluída.  $\square$

O resultado a seguir verifica que a forma como os conjuntos  $F_0$  e  $T_0$  são considerados no algoritmo garante um decréscimo no valor de  $f$  entre duas iterações consecutivas.

**Lema 2.19** [72, Lema 8] *Seja  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  a função de mérito definida por (2.28). Considere  $(x^k, \omega^k)$  uma solução básica complementar primal viável não ótima e  $|T^k| > 1$ . Então, existe um conjunto  $\widehat{T}_0 \neq \emptyset$  tal que  $\widehat{T}_0 \subseteq T^k$  e  $(T_1 \cup F_1) \setminus \widehat{T}_0 \neq \emptyset$ . Considere  $T_0$  o conjunto ótimo do subproblema (2.63) com  $S = \widehat{T}_0$ . Se  $T^{k+1} = \widehat{T}_0 \cup T_0$ , então  $f(x^{k+1}) < f(x^k)$ .*

*Demonstração.* Se  $\widehat{T} \neq \emptyset$ , então é suficiente considerar  $\widehat{T}_0 = \widehat{T} \subseteq T^k$ . De fato, como  $(x^k, \omega^k)$  não é solução ótima de  $(PQ)$ , pelo Lema 2.17 temos que  $\widehat{T} \neq T^k$  e consequentemente,  $(T_1 \cup F_1) \setminus \widehat{T}_0 = (T^k \setminus \widehat{T}) \setminus \widehat{T} = T^k \setminus \widehat{T} \neq \emptyset$ . Caso contrário, temos que  $T^k = T_1 \cup F_1 \neq \emptyset$ , pois por hipótese  $|T^k| > 1$ . Neste caso, basta tomar  $\widehat{T}_0 = \{j\}$  para algum  $j \in T_1 \cup F_1$ .

Considere  $T_0$  o conjunto ótimo do subproblema (2.63) com  $S = \widehat{T}_0$ . Este conjunto é obtido quando este subproblema é resolvido recursivamente até obter otimalidade, sendo  $T_0$  o conjunto ótimo. Considere  $T^{k+1} = \widehat{T}_0 \cup T_0$ . Note que, como  $x^k \geq 0$  e  $x_{\widehat{T}_0}^k = 0$ , o ponto  $x^k$  é viável para este subproblema. Porém,  $(x^k, \omega^k)$  não é a solução ótima de  $(PQ)$  e assim, como  $x^{k+1}$  é a solução ótima do subproblema, obtemos que  $f(x^{k+1}) < f(x^k)$  como queríamos demonstrar.  $\square$

Cabe salientar que, como cada par iterando é primal viável, pelo Lema 2.16, segue que  $L(x^k) = f(x^k)$ , para todo  $k$ . Deste modo, o lema anterior afirma que se  $T^{k+1}$  é atualizado para  $T'_0 \cup T_0$ , então as funções de mérito são reduzidas de uma iteração para outra.

**Teorema 2.20** [72, Teorema 9] *O Algoritmo MKR encontra a solução ótima em um número finito de iterações.*

*Demonstração.* Considere  $\mathcal{T}$  o conjunto de todos os possíveis conjuntos que geram soluções complementares primais viáveis, ou seja,

$$\mathcal{T} = \{T \mid (x, \omega) \text{ é dado por (2.15) e satisfaz } x \geq 0\}.$$

O número possível de conjuntos que geram soluções complementares é no máximo  $2^n$ , deste modo,  $|\mathcal{T}| \leq 2^n$ . Sejam  $k$  uma iteração qualquer,  $x^k$  o ponto corrente, os conjuntos

corrente  $T^k$ ,  $\widehat{F} = \widehat{F}(k)$  e  $\widehat{T} = \widehat{T}(k)$  e a função de mérito  $f$  dada em (2.28). Seja  $(x, \omega)$  uma solução básica complementar de (2.12) com  $\widehat{F}$  e  $\widehat{T}$ .

Se  $f(x) \geq f(x^k)$  e  $|T^k| = 1$ , então como  $x^k$  é primal viável, mas não é ótimo de  $(PQ)$ , segue que  $\omega_j < 0$ , onde  $T^k = \{j\}$ . Assim, o valor da função objetivo melhora, ao permitir que  $x_j$  se afaste da fronteira. Portanto, em uma solução ótima temos  $x_j > 0$  e, assim, um problema com apenas  $n-1$  restrições é resolvido, a solução ótima é encontrada e o algoritmo termina com a solução do problema. Caso contrário, temos que  $f(x) \geq f(x^k)$  e  $|T^k| > 1$  ou  $f(x) < f(x^k)$ . Se  $f(x) < f(x^k)$ , temos pelo mecanismo do algoritmo que  $x^{k+1} = x$  e, portanto,  $f(x^{k+1}) < f(x^k)$ . Caso contrário, pelo mecanismo do algoritmo temos que  $T^{k+1} = T'_0 \cup T_0$ . Assim, pelo Lema 2.19,  $f(x^{k+1}) < f(x^k)$ .

Deste modo, a sequência gerada  $\{f(x^k)\}$  é estritamente decrescente. Portanto, por isso, pelos fatos que  $f(\cdot)$  é estritamente convexa, que a sequência  $\{x^k\}$  é viável e que cada solução básica complementar é única, segue que os conjuntos da sequência  $\{T^k\} \subseteq \mathcal{T}$  são distintos, ou seja,  $T^j \neq T^k$ , para todo  $j \neq k$ . Assim, usando isto, o fato que  $|\mathcal{T}| \leq 2^n$  e que o conjunto que fornece a solução ótima está contido em  $\mathcal{T}$ , obtemos que o algoritmo encontra a solução em um número finito de iterações.  $\square$

### 2.5.5 Comentários

Foram apresentados, nas seções anteriores, vários resultados de convergência para os algoritmos descritos na Seção 2.2.

Na Seção 2.5.1 a convergência dos Algoritmos BPP e KR é estabelecida para LCPs cujas matrizes consideradas são simétricas definidas positivas e satisfazem pelo menos uma das condições  $(C_1)$  e  $(C_2)$ , definidas em (2.51) e (2.52), respectivamente. Porém, esta subclasse é muito restritiva. Verificamos que a matriz simétrica

$$Q = \begin{pmatrix} 2 & 1 \\ 1 & 1.5 \end{pmatrix},$$

diagonalmente dominante, não satisfaz nenhuma das condições.

Um algoritmo que combina o método BPP com o método convergente de Murty é apresentado na Seção 2.5.2. A convergência finita do algoritmo é garantida por causa das atualizações de cardinalidade um. Estes tipos de iterações provocam lentidão no algoritmo. A vantagem deste tipo de processo híbrido é que essas atualizações lentas, de cardinalidade um, raramente são utilizadas na prática e, portanto, o Algoritmo BPP

funciona na sua versão original [50].

Foi recordado que o Algoritmo BPP converge para matrizes- $P$  triangulares inferiores e superiores na Seção 2.5.3. Ainda nesta seção, foi discutido que o algoritmo não cicla quando o LCP de ordem 3 considera matrizes estritamente diagonalmente dominantes por colunas ou por linhas. Porém, este resultado só está provado para matrizes de ordem 3.

Por fim, na Seção 2.5.4 foi apresentada uma versão modificada do Algoritmo BPP-PC globalmente convergente, para todo problema LCP com matriz- $P$  simétrica. Entretanto, o algoritmo é bastante complexo, truncado, utiliza iterações recursivas e não se sabe quantas recursões são necessárias para um determinado problema.

Deste modo, um foco de pesquisa é garantir que o Algoritmo BPP seja convergente para uma subclasse de matrizes- $P$  que não seja tão restritiva. Ou ainda, a construção de um algoritmo modificado simples, que garanta a convergência sem perder a eficiência dos algoritmos.

Uma subclasse das matrizes- $P$  interessante é a das matrizes estritamente diagonalmente dominantes por colunas com elementos diagonais positivos. Em [124], Takriti e Murty comentam que Joaquim J. Júdice já havia conjecturado em 1995 que o algoritmo BPP não possui ciclos e termina com sucesso quando o LCP possui matriz diagonalmente dominante por colunas com elementos diagonais positivos. Esta conjectura ainda permanece em aberto para matrizes de ordem maior que 3.

## 2.6 Experimentos computacionais

Nesta seção são apresentados alguns experimentos numéricos realizados com intuito de discutir o desempenho computacional dos algoritmos descritos anteriormente.

Os algoritmos foram implementados em MATLAB R2012b (8.0.0.783) e os testes numéricos foram realizados num computador Dell Vostro 5470 edição Windows 10 Home Single Language com processador Intel(R) Core(TM) i7-4510U CPU @ 2,00GHz, memória RAM de 8,00 GB e sistema operacional de 64 bits.

### 2.6.1 Algoritmos BPP e BPP-PC

Nesta seção discutimos os desempenhos dos Algoritmos BPP e BPP-PC para solucionar LCPs com matrizes- $P$  não necessariamente simétricas. Inicialmente consideramos matrizes de um repositório disponível na internet para verificar a eficiência desses algo-

ritmos. Também, fazemos uma relação entre a eficiência desses algoritmos e os conjuntos iniciais  $F$  e  $T$  considerados. Por fim, para motivar a nossa conjectura, apresentamos o excelente comportamento dos algoritmos quando aplicados a problemas com matrizes- $P$  estritamente diagonalmente dominantes. Para tanto consideramos as seguintes notações:

- $k$  - quantidade de iterações que coincide com a quantidade de sistemas (2.13) resolvidos pelo algoritmo para encontrar a solução
- $k_v$  - quantidade de vezes que o Algoritmo BPP-PC entra na fase de viabilidade primal

### Eficiência dos Algoritmos BPP e BPP-PC

Para verificar que os algoritmos são de fato eficientes na prática, vamos considerar nesta seção LCPs com matrizes obtidas pelo repositório *Matrix Market* [101]. Este repositório consiste de uma coleção de matrizes esparsas oriundas de uma variedade de aplicações, bem como ferramentas para construção de matrizes, para estudos comparativos de algoritmos.

A primeira bateria de testes foi realizada para 12 problemas com vetores  $c = -e$ , sendo  $e := (1, \dots, 1)^t$ , e matrizes  $Q \in \mathbb{R}^{n \times n}$  simétricas definidas positivas de dimensões entre 1806 e 90499, com números de condição acima de  $10^5$ . Os dados dessas matrizes e do desempenho dos algoritmos estão apresentados na Tabela 2.6. As primeiras cinco colunas correspondem aos dados do problema: identificação do problema, nome da matriz no repositório, ordem  $n$ , quantidade de variáveis nulas na solução  $x^*$  do LCP e número de condição da matriz. As últimas três colunas estão relacionadas com o desempenho dos Algoritmos BPP e BPP-PC de acordo com o número de sistemas (2.13) resolvidos e o número de fases de viabilidade primal realizadas pelo Algoritmo BPP-PC. Para estes problemas, consideramos os conjuntos iniciais  $F^0 = \{i \mid c_i < 0\}$  e  $T^0 = \mathcal{N} \setminus F^0$ . A quantidade de sistemas (2.13), resolvidos para obter as soluções destes problemas, confirma que estes algoritmos são muito eficientes para matrizes esparsas, de alta dimensão e mesmo mal condicionadas. Também podemos perceber como a recuperação da viabilidade primal provoca um aumento no número de sistemas resolvidos para esses problemas.

Em seguida, foi realizada uma bateria de testes com 20 problemas com vetores  $c = -e$  e matrizes- $P$  não simétricas de dimensões entre 59 e 9661 e número de condição entre  $10^0$  e  $10^{15}$ . Como essas matrizes não simétricas dadas pelo repositório não são

Dados da matriz $Q$					BPP	BPP-PC	
Id	Matriz	Dimensão	$\#(x_i^* = 0)$	$\text{cond}(Q)$	$k$	$k$	$k_v$
1	bcsstk14	1806	501	1.3100E+10	5	7	2
2	bcsstk15	3948	769	7.9661E+09	8	13	6
3	bcsstk16	4884	187	7.0084E+09	5	7	2
4	bcsstk17	10974	1704	1.9534E+10	13	33	10
5	bcsstk18	11948	2525	6.4860E+11	8	15	5
6	bcsstm26	1922	0	2.6157E+05	1	1	0
7	s1rmq4m1	5489	784	3.5798E+06	15	23	8
8	s1rmt3m1	5489	620	5.3685E+06	18	22	4
9	s2rmq4m1	5489	271	3.1736E+08	12	19	6
10	s2rmt3m1	5489	246	4.8402E+08	14	17	3
11	s3dkt3m2	90449	3918	6.2755E+11	31	64	26
12	s3rmq4m1	5489	67	3.0667E+10	12	15	4

Tabela 2.6: Dados das matrizes consideradas da coleção [101] e desempenho dos Algoritmos BPP e BPP-PC.

necessariamente matrizes- $P$ , para serem aplicadas no nosso contexto, foram transformadas em matrizes estritamente diagonalmente dominantes alterando os termos da diagonal da seguinte forma,

$$Q_{ii} = 10^{-6} + \max \left\{ \sum_{j \neq i} |Q_{ij}|, \sum_{j \neq i} |Q_{ji}| \right\}, \forall i \in \mathcal{N}. \quad (2.68)$$

Deste modo, de acordo com o Lema 2.6, as matrizes se tornam matrizes- $P$ . Para estes testes consideramos os conjuntos iniciais  $F^0 = \{i \mid c_i < 0\}$  e  $T^0 = \mathcal{N} \setminus F^0$ . Ambos os algoritmos obtiveram as soluções dos problemas resolvendo no máximo 5 sistemas (2.13). Os dados das matrizes consideradas e os resultados obtidos estão dispostos na Tabela 2.7. A primeira coluna apresenta a identificação do problema, a segunda o nome da matriz no repositório e, as duas seguintes, a dimensão e o número de condicionamento das matrizes modificadas, conforme (2.68). As demais colunas estão relacionadas com o desempenho dos Algoritmos BPP e BPP-PC, de acordo com as quantidades  $k$  e  $k_v$ . Cabe salientar que esta tabela reforça a nossa conjectura de que os algoritmos convergem de forma finita e rápida para matrizes estritamente diagonalmente dominantes.

## Conjuntos iniciais

Dependendo da configuração dos problemas considerados, a escolha de um bom conjunto inicial é fundamental para a eficiência do algoritmo. Visando isto, apresentamos a seguir testes preliminares sobre a escolha dos conjuntos iniciais. Cabe salientar que a busca por conjuntos iniciais adequados é um assunto interessante para pesquisas futuras.

A primeira bateria de testes consiste em resolver o LCP cuja matriz  $Q \in \mathbb{R}^{50 \times 50}$  é dada por  $Q = LL^t$ , com  $L \in \mathbb{R}^{50 \times 50}$  a matriz triangular inferior dada por

$$L = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 2 & 1 & 0 & \dots & 0 \\ 2 & 2 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 2 & \dots & 2 & 2 & 1 \end{pmatrix} \quad (2.69)$$

e o vetor  $c$  dado por

$$c_I = -Q_I e_I \quad \text{e} \quad c_J = \max\{-e_J, -Q_{JI} e_I\} \geq -Q_{JI} e_I, \quad (2.70)$$

sendo  $I$  e  $J = \mathcal{N} \setminus I$  dois conjuntos de índices. Nestes testes consideramos  $I = \{1, \dots, 37\}$ . Em (2.71) apresentamos o vetor resultante de (2.70) com este conjunto  $I$ . Cabe salientar que  $Q$  é uma matriz densa com número de condição de ordem de  $10^7$ . Escolhemos o LCP com esta matriz específica, pois este é um dos problemas em que iterações do tipo Murty são necessárias.

$$c = \begin{pmatrix} -73 & -217 & -357 & -493 & -625 & -753 & -877 & -997 & -1113 & -1225 \\ -1333 & -1437 & -1537 & -1633 & -1725 & -1813 & -1897 & -1977 & -2053 & -2125 \\ -2193 & -2257 & -2317 & -2373 & -2425 & -2473 & -2517 & -2557 & -2593 & -2625 \\ -2653 & -2677 & -2697 & -2713 & -2725 & -2733 & -2737 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{pmatrix}^t. \quad (2.71)$$

Resolvemos o problema considerando 7 conjuntos iniciais distintos. A Tabela 2.8 apresenta os conjuntos iniciais considerados e os resultados obtidos, para o LCP com  $Q = LL^t \in \mathbb{R}^{50 \times 50}$ , sendo  $L \in \mathbb{R}^{50 \times 50}$  dada por (2.69) e vetores  $c$  considerados de acordo com (2.70). Em todos os testes,  $T^0 = \mathcal{N} \setminus F^0$ . A primeira coluna apresenta os conjuntos iniciais considerados, a segunda e a terceira as quantidades  $k$  de sistemas (2.13) resolvidos pelos Algoritmos BPP e BPP-PC, respectivamente, e a última coluna ilustra o número

$k_v$  de fases de viabilidade primal realizadas pelo Algoritmo BPP-PC. Note que o número de sistemas (2.13) resolvidos pelos Algoritmos BPP e BPP-PC foi de 30 e 56 no pior caso, respectivamente, e de 2 no melhor caso. Podemos ainda observar como a eficiência dos algoritmos, para problemas densos, possui relação com os conjuntos iniciais utilizados. Além disso, enquanto o Algoritmo BPP resolve mais sistemas à medida que  $|F^0|$  aumenta, este número não tem grande variação para o Algoritmo BPP-PC, o que evidencia que a busca por um conjunto inicial eficaz não é uma tarefa fácil.

Em seguida, apresentamos a relação entre o conjunto inicial e a eficiência dos algoritmos para resolver LCPs com matrizes esparsas fornecidas pelo repositório [101]. Inicialmente foram considerados LCPs com matrizes simétricas definidas positivas, como descritas nas cinco primeiras colunas da Tabela 2.6 e vetores  $c = -e$ . Resolvemos cada um desses 12 problemas considerando 8 diferentes conjuntos iniciais com  $T^0 = \mathcal{N} \setminus F_i^0$  e, para  $i = 1, \dots, 8$ ,  $F_i^0$  dados por

- $F_1^0 = \emptyset$ ,
- $F_2^0 = \{i \mid c_i < -\varepsilon\}$ ,
- $F_3^0 = \operatorname{argmin}\{c_i\}$ ,
- $F_4^0 = \{r, s \mid c_r = \min\{c_i\}, c_s = \min_{i \neq r}\{c_i\}\}$ ,
- $F_5^0 = \{n\}$ ,
- $F_6^0 = \{n/2\}$ ,
- $F_7^0 = \{1, \dots, n/2\}$ ,
- $F_8^0 = \{1, \dots, n\}$ .

Os resultados estão apresentados na Tabela 2.9. Na primeira coluna dispomos as identificações dos problemas e nas colunas seguintes, o melhor caso (MC) e o pior caso (PC), em relação ao número de sistemas (2.13) resolvidos por cada algoritmo, para os oito conjuntos iniciais descritos. Note que para estes problemas, com matrizes esparsas, diferentes conjuntos iniciais não provocaram grandes diferenças com relação ao número de sistemas resolvidos, os números dos piores casos são 2,77 vezes e 4,08 vezes maiores que os números nos melhores casos para os Algoritmos BPP e BPP-PC, respectivamente. Os resultados são apresentados com mais detalhes na Tabela 2.10. Na primeira coluna da Tabela 2.10

disponemos as identificações dos problemas, nas colunas seguintes apresentamos o número  $k$  de sistemas (2.13) resolvidos por cada algoritmo, de acordo com o conjunto inicial considerado.

### Matrizes estritamente diagonalmente dominantes

Nesta seção apresentamos o desempenho dos Algoritmos BPP e BPP-PC para matrizes estritamente diagonalmente dominantes.

Primeiramente, consideramos versões modificadas das matrizes do repositório como descritas na Tabela 2.6. Nessas novas versões, alteramos seus elementos diagonais, conforme (2.68), de modo a torná-las estritamente diagonalmente dominantes. Então, resolvemos os LCPs com essas matrizes modificadas, vetores  $c = -e$  e conjuntos iniciais  $F^0 = \{i \mid c_i < 0\}$  e  $T^0 = \mathcal{N} \setminus F^0$ . Os resultados para estes testes estão dispostos na Tabela 2.11. A primeira coluna representa a identificação das matrizes, conforme Tabela 2.6, já as demais colunas apresentam o número de sistemas (2.13) que os algoritmos resolveram para encontrar a solução. Podemos perceber, nesta tabela, que os algoritmos são muito eficientes para matrizes estritamente diagonalmente dominantes, o que fortalece a nossa conjectura.

Esta boa eficiência para matrizes estritamente diagonalmente dominantes também ocorre em matrizes não simétricas, como pode ser visto na Tabela 2.7.

#### 2.6.2 Versões modificadas dos algoritmos

Nesta seção apresentamos o desempenho dos Algoritmos BPP e BPP-PC, juntamente com os Algoritmos KR e BPP-M descritos na Seção 2.5. Para tanto, vamos considerar as seguintes notações:

- $k_m$  - quantidade de iterações do tipo Murty utilizadas pelo Algoritmo BPP-M;
- $k_v$  - quantidade de fases de viabilidade primal utilizadas pelo Algoritmo BPP-PC.

Inicialmente, consideramos LCPs cujas matrizes são dadas pelo repositório [101], conforme as primeiras colunas da Tabela 2.6 e, os vetores  $c$  possuem as primeiras  $n/2$  componentes iguais a  $-1$  e as demais iguais a  $1$ . Nestes testes, consideramos  $F^0 = \{i \mid c_i < 0\}$ ,  $k_{\max} = 5.000$  e uma tolerância de  $10^{-6}$ . Os resultados obtidos estão dispostos na Tabela 2.12. A primeira coluna apresenta o número de identificação da matriz, de

acordo com a Tabela 2.6, as demais colunas apresentam o desempenho dos algoritmos com relação às quantidades  $k$ ,  $k_m$  e  $k_v$ . Podemos perceber, com estes testes, que iterações do tipo Murty no Algoritmo BPP-M quase nunca são utilizadas, porém quando estes tipos de iterações são requeridas, o algoritmo se torna muito lento. Além disso, nesta tabela confirmamos que os Algoritmos BPP e KR não possuem diferença na prática.

Realizamos também uma bateria de problemas semelhantes aos testes anteriores, com as mesmas matrizes do repositório descritas na Tabela 2.6, porém com vetores  $c$  de acordo com (2.70) considerando  $I = \{1, 2, \dots, \lfloor 3n/4 \rfloor\}$  e  $J = \mathcal{N} \setminus I$ . Utilizamos nestes testes os conjuntos iniciais  $F^0 = \{i \mid c_i < 0\}$ ,  $k_{\max} = 1.000$  e uma tolerância de  $10^{-6}$ . Apresentamos os resultados obtidos na Tabela 2.13. A primeira coluna fornece a identificação das matrizes consideradas, enquanto as demais colunas fornecem as quantidades  $k$ ,  $k_m$  e  $k_v$  gastas pelos algoritmos. Novamente podemos perceber, nesta tabela, que iterações do tipo Murty são raramente utilizadas no Algoritmo BPP-M, mas, quando são utilizadas, o número de sistemas (2.13) resolvidos cresce drasticamente.

## 2.7 Conclusões do capítulo

Neste capítulo abordamos o método de pivotamento em blocos principais e suas variantes para solucionar problemas de complementaridade linear com matrizes- $P$ .

Inicialmente discutimos o método clássico BPP sugerido por [77] que pode ser utilizado para qualquer matriz- $P$ , não necessariamente simétrica. Em seguida, verificamos que o Algoritmo KR, proposto por [83], é o Algoritmo BPP a menos de uma igualdade nas atualizações dos conjuntos  $F$  e  $T$ . Então, discutimos uma versão em duas fases em que a cada iteração o método gera um par complementar  $(x, \omega)$  primal viável. Apesar destes métodos serem muito eficientes na prática, não possuem convergência finita para todo LCP com matriz- $P$ . Expomos na Seção 2.3 exemplos com matrizes- $P$  simétricas e não simétricas em que os algoritmos geram sequências periódicas e, portanto, não convergem para a solução. Deste modo, concluímos que os algoritmos só convergem para subclasses de matrizes- $P$  ou para versões modificadas dos algoritmos.

A fim de verificar a convergência, uma das estratégias é considerar alguma função de mérito e verificar que o método produz algum decréscimo nesta função ao longo das iterações. Para o caso simétrico, como o LCP é equivalente ao  $(PQ)$  podemos utilizar a própria função objetivo  $f$ , dada em (2.28), como função de mérito. A dificuldade está em qual função de mérito considerar para o caso não simétrico que não possui uma equivalência como no caso simétrico. Na Seção 2.4 listamos algumas possíveis funções de mérito para os LCPs. Infelizmente não conseguimos provar a convergência dos Algoritmos BPP e BPP-PC para subclasses de matrizes utilizando estas funções. Ainda nesta mesma seção verificamos que quando uma determinada função de mérito é considerada e se todas as soluções geradas pelo Algoritmo BPP são não degeneradas, então os iterandos deste algoritmo coincidem com os do método de Newton, quando este é aplicado ao problema de minimizar a função de mérito, ou seja, ao problema (2.33). Esta equivalência justifica a eficiência e rapidez dos métodos de pivotamento em blocos principais.

Foi feito também um apanhado entre alguns resultados de convergência existentes. Dispomos na Seção 2.5.1 que os Algoritmos BPP e KR convergem para a subclasse de matrizes- $P$  que são simétricas, diagonalmente dominantes, e satisfazem pelo menos uma das condições  $(C_1)$  ou  $(C_2)$ . Porém, estas condições são muito fortes. Na Seção 2.5.3 discutimos que o algoritmo converge para matrizes- $P$  triangulares inferiores e superiores e para LCPs de ordem 3 com matrizes, não necessariamente simétricas, estritamente dia-

gonalmente dominantes por colunas. Porém, estas classes são também muito restritivas. A conjectura que o algoritmo clássico é convergente para toda matriz- $P$  diagonalmente dominante por colunas ainda permanece em aberto.

Discutimos também algumas versões modificadas dos Algoritmos BPP e BPP-PC existentes na literatura que são convergentes. Primeiramente discutimos um algoritmo híbrido que combina o método BPP com o método convergente de Murty. A convergência para toda matriz- $P$  não necessariamente simétrica é herdada do método de Murty, porém quando essas iterações são necessárias este algoritmo híbrido se torna muito lento. Este fato pode ser visto nos experimentos numéricos, sobretudo do problema 11, apresentados nas Tabelas 2.12 e 2.13. Também discutimos uma versão do Algoritmo BPP-PC que marca algumas variáveis e realiza iterações recursivas para essas variáveis, a fim de garantir que a função de mérito reduza de uma iteração para outra. Este algoritmo é convergente para toda matriz- $P$  simétrica. O problema é que não há uma estimativa de quantas marcações e, conseqüentemente, quantas recursividades são necessárias. Podemos verificar na Tabela 2.13 que este número pode se tornar muito grande. Além disso, para dar liberdade ao método, acreditamos que não seja necessária a exigência de que a função de mérito reduza a cada iteração, e sim ao longo de um certo intervalo de iterações.

### Trabalhos futuros

A conjectura de que o Algoritmo BPP em sua forma clássica é convergente para toda matriz- $P$  estritamente diagonalmente dominante por colunas ainda permanece em aberto, uma vez que não foi possível encontrar um contra-exemplo. Além disso, apresentar uma versão modificada que garanta a convergência sem prejudicar a eficiência do algoritmo também é um foco de pesquisa futura interessante. Por fim, uma análise sobre os conjuntos iniciais considerados pode ser um estudo promissor para uma melhora na eficiência dos algoritmos.

Dados da matriz $Q$				BPP	BPP-PC	
Id	Matriz	Dimensão	cond( $Q$ )	$k$	$k$	$k_v$
1	bp-1200	822	9.069635E+02	3	4	2
2	e05r0000	236	4.867966E+01	1	1	0
3	e05r0500	236	2.965420E+02	1	1	0
4	e20r4000	4142	2.700482E+03	1	1	0
5	e30r2000	9661	1.262343E+03	1	1	0
6	fs-183-1	183	1.275978E+15	2	2	1
7	gre-343	343	2.027358E+02	5	5	1
8	hor-131	434	3.109987E+02	2	2	1
9	impcol-b	59	1.933818E+01	2	2	1
10	impcol-c	137	1.453379E+02	2	2	1
11	impcol-d	425	9.580044E+00	3	3	1
12	impcol-e	225	1.055994E+04	3	3	1
13	orani678	2529	1.646207E+03	2	2	1
14	orsirr-1	1030	1.542291E+02	1	1	0
15	shl-400	663	8.578374E+04	2	2	1
16	west0067	67	6.415539E+00	2	2	1
17	west0132	132	7.913953E+05	2	2	1
18	west0156	156	4.307162E+07	2	2	1
19	west0167	167	6.306060E+05	3	4	2
20	west0381	381	3.490720E+03	2	2	1

Tabela 2.7: Dados das matrizes não simétricas dadas por [101] e resultados numéricos.

$F^0$	BPP		BPP-C	
	$k$	$k$	$k_v$	
{1}	2	2	0	
{1, ..., 5}	4	48	14	
{1, ..., 10}	7	45	14	
{1, ..., 20}	12	56	16	
{1, ..., 30}	17	52	15	
{1, ..., 40}	30	49	14	
{1, ..., 50}	30	51	14	

Tabela 2.8: Variação do conjunto inicial.

Id	BPP		BPP-C	
	MC	PC	MC	PC
1	4	7	7	15
2	7	17	14	19
3	5	6	5	6
4	13	15	33	46
5	8	12	13	27
6	1	2	1	2
7	15	23	23	52
8	18	26	21	59
9	12	26	19	51
10	14	38	17	55
11	31	86	64	261
12	12	27	15	60

Tabela 2.9: Melhor e pior desempenho de acordo com os conjuntos iniciais adotados.

Id	BPP								BPP-C							
	$F_1^0$	$F_2^0$	$F_3^0$	$F_4^0$	$F_5^0$	$F_6^0$	$F_7^0$	$F_8^0$	$F_1^0$	$F_2^0$	$F_3^0$	$F_4^0$	$F_5^0$	$F_6^0$	$F_7^0$	$F_8^0$
1	5	4	5	5	5	5	7	4	7	7	8	8	8	8	15	7
2	8	7	8	8	8	8	17	7	14	14	15	15	15	15	19	14
3	6	5	6	6	6	6	5	5	5	6	6	6	6	6	6	5
4	14	13	14	14	14	14	15	13	33	33	34	34	34	34	46	33
5	9	8	9	9	9	9	12	8	13	13	14	14	14	14	27	13
6	2	1	2	2	2	2	2	1	1	1	2	2	2	2	2	1
7	16	15	16	16	16	23	22	15	23	23	24	24	24	47	52	23
8	19	18	19	19	19	25	26	18	21	21	22	22	22	50	59	21
9	13	12	13	13	13	26	24	12	19	19	20	20	20	51	46	19
10	15	14	15	15	15	38	26	14	17	17	18	18	18	55	43	17
11	32	31	32	32	32	32	86	31	64	64	65	65	65	65	261	64
12	13	12	13	13	13	27	24	12	15	15	16	18	16	60	40	15

Tabela 2.10: Resultados da Tabela 2.9 apresentados em detalhes.

	BPP	BPP-PC
Id	$k$	$k$
1	4	5
2	1	1
3	1	1
4	1	1
5	6	7
6	1	1
7	1	1
8	1	1
9	1	1
10	1	1
11	1	1
12	1	1

Tabela 2.11: Desempenho dos algoritmos para matrizes- $P$  simétricas estritamente diagonalmente dominante.

	BPP	KR	BPP-M		BPP-PC	
Id	$k$	$k$	$k$	$k_m$	$k$	$k_v$
1	11	11	11	0	22	11
2	16	16	16	0	41	26
3	8	8	8	0	15	7
4	33	33	33	0	87	55
5	22	23	22	0	48	26
6	1	1	1	0	1	0
7	32	32	32	0	66	35
8	34	34	34	0	69	35
9	25	25	25	0	56	31
10	28	28	28	0	57	29
11	231	231	$k_{\max}$	4925	679	450
12	27	27	27	0	74	47

Tabela 2.12: Desempenho dos Algoritmos BPP, KR, BPP-M e BPP-PC.

Id	BPP	KR	BPP-M		BPP-PC	
	$k$	$k$	$k$	$k_m$	$k$	$k_v$
1	9	9	9	0	21	7
2	12	12	12	0	23	8
3	8	8	8	0	18	7
4	21	21	21	0	61	16
5	13	13	13	0	30	10
6	1	1	1	0	1	0
7	8	8	8	0	24	7
8	11	11	11	0	27	9
9	15	15	15	0	34	10
10	16	16	16	0	50	15
11	39	39	$k_{\max}$	970	126	32
12	12	12	12	0	50	12

Tabela 2.13: Desempenho dos algoritmos para um LCP com vetor  $c$  conforme (2.70).



# Capítulo 3

## Aplicações dos algoritmos de pivotamento em blocos principais

Este capítulo é dedicado a duas aplicações muito interessantes dos algoritmos de pivotamento em blocos principais. A primeira delas é chamada Decomposição Não Negativa de uma matriz, que consiste em decompor esta matriz como um produto de duas matrizes não negativas. Essa decomposição tem aplicações em problemas de transmissão de dados [16, 84]. Na segunda aplicação, os algoritmos de pivotamento em blocos principais são usados para calcular as projeções na norma elíptica requeridas por um algoritmo de gradientes projetados com condicionamento, para a determinação de um ponto estacionário de um problema não linear com restrições de não negatividade. Nossa principal contribuição neste capítulo é a proposta de uma matriz de condicionamento. Testes numéricos preliminares são apresentados.

### 3.1 Decomposição não negativa de uma matriz

Nos últimos anos a quantidade e a dimensão de dados a serem armazenados vêm sofrendo um grande aumento, consequência inevitável do desenvolvimento tecnológico [16]. O processamento dessa grande quantidade de dados criou novas preocupações com relação à representação dos mesmos, redução dessas grandes dimensões e confiabilidade de dados compactados. Cabe ainda salientar que, na grande maioria dos casos, esses dados são não negativos, e manter esta restrição nos modelos reduzidos é uma condição necessária, a fim de evitar contradições físicas.

Este problema gera a modelagem matemática denominada de Decomposição Não

Negativa de uma matriz (*Nonnegative Matrix Factorization - (NMF)*) [84, 102], que pode ser representada pela formulação matemática a seguir. Dada uma matriz  $A \in \mathbb{R}^{m \times n}$  (matriz de dados) e  $p \leq \min\{m, n\}$ , o problema consiste em encontrar duas matrizes não negativas  $Y \in \mathbb{R}^{m \times p}$  e  $Z \in \mathbb{R}^{p \times n}$  tais que

$$A = YZ.$$

Na prática pretende-se que  $p$  seja pequeno e a decomposição pode não existir para uma dada matriz  $A$ . Em outros casos, a decomposição existe, mas pode ser muito difícil de ser obtida. Para contornar estas dificuldades, procuram-se obter matrizes  $Y$  e  $Z$  cujo produto de  $Y$  por  $Z$  seja aproximadamente  $A$  ( $A \approx YZ$ ) através da resolução do seguinte problema

$$\begin{aligned} & \underset{Y, Z}{\text{minimizar}} && \frac{1}{2} \|A - YZ\|_F^2 \\ & \text{sujeito a} && Y_{ij} \geq 0, \quad \forall i \in \mathcal{N}_m \text{ e } j \in \mathcal{N}_p \\ & && Z_{jk} \geq 0, \quad \forall j \in \mathcal{N}_p \text{ e } k \in \mathcal{N}_n, \end{aligned} \tag{3.1}$$

sendo  $\|\cdot\|_F$  a norma de Frobenius e  $\mathcal{N}_j := \{1, \dots, j\}$  para todo  $j \in \mathbb{N}$ .

Vários algoritmos vêm sendo desenvolvidos para solucionar este problema de otimização não convexa [16, 52, 88]. Em particular Kim e Park em [81] apresentam um algoritmo que utiliza pivotamento em blocos principais como uma alternativa para resolver o NMF.

Para apresentar as ideias gerais deste algoritmo, vamos primeiramente considerar o problema de mínimos quadrados (*Nonnegative Least Squares - (NNLS)*) que, dados  $\tilde{A} \in \mathbb{R}^{r \times s}$  e  $B \in \mathbb{R}^{r \times q}$ , consiste em encontrar  $X \in \mathbb{R}^{q \times s}$  como solução do problema

$$\begin{aligned} & \text{minimizar} && \frac{1}{2} \|\tilde{A} - BX\|_F^2 \\ & \text{sujeito a} && X_{ij} \geq 0, \quad \forall i \in \mathcal{N}_q \text{ e } j \in \mathcal{N}_s. \end{aligned} \tag{3.2}$$

Note que cada coluna da matriz variável  $X$  pode ser encontrada de forma independente das demais colunas.

Vamos apresentar a seguir como encontrar cada uma das colunas de  $X$ . Suponha que  $B$  tenha posto completo por colunas. Considere  $j \in \mathcal{N}_s$  e denote por

$$x = X_{\bullet, j} \in \mathbb{R}^q \text{ e } a = \tilde{A}_{\bullet, j} \in \mathbb{R}^r,$$

os vetores representando a  $j$ -ésima coluna de  $X$  e de  $\tilde{A}$ , respectivamente. Assim, resolvemos o subproblema de (3.2) dado por

$$\begin{aligned} & \text{minimizar} && \frac{1}{2} \|a - Bx\|^2 \\ & \text{sujeito a} && x \geq 0. \end{aligned} \tag{3.3}$$

Este é equivalente ao problema  $(PQ')$

$$(PQ') \quad \begin{cases} \text{minimizar} & \frac{1}{2}x^t B^t Bx - a^t Bx + \frac{1}{2}a^t a \\ \text{sujeito a} & x \geq 0. \end{cases}$$

Como  $B$  tem posto completo por colunas, pelo Lema 2.4,  $B^t B$  é uma matriz SPD. Portanto o  $(PQ')$  é equivalente ao LCP

$$\begin{cases} B^t Bx - B^t a = \omega, & (3.4) \end{cases}$$

$$\begin{cases} x \circ \omega = 0, & (3.5) \end{cases}$$

$$\begin{cases} x \geq 0, & (3.6) \end{cases}$$

$$\begin{cases} \omega \geq 0. & (3.7) \end{cases}$$

Como a matriz desse LCP é SPD, os métodos de pivotamento em blocos principais podem ser usados para resolvê-lo com

$$Q = B^t B \quad \text{e} \quad c = -B^t a.$$

O algoritmo proposto em [81] é o Algoritmo BPP-M com as iterações tipo Murty calculadas com o máximo, ao invés de mínimo, no conjunto  $H'$  dado em (2.61), isto é,

$$H' = \max\{i \in H\}.$$

Vamos agora retornar ao problema  $(NMF)$  dado por (3.1) e apresentar o algoritmo proposto em [81] para este problema. O mecanismo do algoritmo consiste em, de forma iterativa, fixar uma das matrizes  $Y$  ( $Z$ ) como constante e determinar a outra matriz  $Z$  ( $Y$ ) a partir da resolução de um LCP com uma matriz SPD usando um dos algoritmos de pivotamento em blocos principais. Assim, dadas duas matrizes  $\hat{Y} \in \mathbb{R}^{m \times p}$  e  $\hat{Z} \in \mathbb{R}^{p \times n}$ , resolvemos o subproblema a seguir para obter uma matriz tentativa  $Z$

$$\begin{aligned} & \underset{Z}{\text{minimizar}} && \frac{1}{2} \left\| A - \hat{Y} Z \right\|_F^2 \\ & \text{sujeito a} && Z \geq 0, \end{aligned} \quad (3.8)$$

e o subproblema a seguir para obter uma matriz tentativa  $Y$

$$\begin{aligned} & \underset{Y}{\text{minimizar}} && \frac{1}{2} \left\| A^t - \hat{Z}^t Y^t \right\|_F^2 \\ & \text{sujeito a} && Y \geq 0. \end{aligned} \quad (3.9)$$

Esses dois problemas reduzem-se a problemas de mínimos quadrados, que por sua vez se reduzem a LCPs, conforme discutido anteriormente. Sejam  $\tilde{Y}$  e  $\tilde{Z}$  as matrizes obtidas a

partir das resoluções dos problemas (3.8) e (3.9). Se  $\tilde{Y}$  e  $\tilde{Z}$  são aproximadamente iguais a  $\hat{Y}$  e  $\hat{Z}$ , respectivamente, então o algoritmo termina com as matrizes  $\hat{Y}$  e  $\hat{Z}$ . Senão, é aplicada uma nova iteração com as matrizes  $\tilde{Y}$  e  $\tilde{Z}$ .

Antes de descrever o algoritmo para resolver o problema (3.1) vamos definir algumas notações. Dada uma matriz  $A \in \mathbb{R}^{m \times n}$  e índices  $i \in \mathcal{N}_m$  e  $j \in \mathcal{N}_n$ , denotamos a  $i$ -ésima linha e a  $j$ -ésima coluna de  $A$  por  $A_{i\bullet}$  e  $A_{\bullet j}$ , respectivamente. Apresentamos a seguir o Algoritmo BPP-NMF proposto em [81].

---

**Algoritmo 3.1** BPP-NMF

---

Dados:  $k = 0$ ,  $\varepsilon > 0$ ,  $Y^0 \in \mathbb{R}^{m \times p}$ ,  $Z^0 \in \mathbb{R}^{p \times n}$

REPITA

PARA  $j \in \{1, \dots, p\}$  faça

Utilize o Algoritmo BPP (ou uma de suas extensões) para encontrar  $Z_j^{k+1}$   
solução do seguinte subproblema

$$\begin{aligned} &\text{minimizar} && \frac{1}{2} \|A_{\bullet j} - Y^k Z_{\bullet j}\|_2^2 \\ &\text{sujeito a} && Z_{\bullet j} \in \mathbb{R}_+^n \end{aligned}$$

Utilize o Algoritmo BPP (ou uma de suas extensões) para encontrar  $Y_j^{k+1}$   
solução do seguinte subproblema

$$\begin{aligned} &\text{minimizar} && \frac{1}{2} \|A_{\bullet j}{}^t - (Z^k)^t Y_{\bullet j}{}^t\|_2^2 \\ &\text{sujeito a} && Y_{\bullet j}{}^t \in \mathbb{R}_+^m \end{aligned}$$

FIM PARA

Se  $(\|Y^k - Y^{k+1}\| < \varepsilon)$  e  $(\|Z^k - Z^{k+1}\| < \varepsilon)$

Pare com sucesso

$k = k + 1$

---

Durante as iterações, as matrizes correntes  $Y^k$  e  $(Z^k)^t$  podem não ser de posto coluna completo. Em [81], para contornar esta situação, os autores sugerem um problema regularizado dado por

$$\begin{aligned} &\text{minimizar}_{Y,Z} && \frac{1}{2} \|A - YZ\|_F^2 + \alpha \|Y\|_F^2 + \beta \|Z\|_F^2 \\ &\text{sujeito a} && Y_{ij} \geq 0, \quad \forall i \in \mathcal{N}_m \text{ e } j \in \mathcal{N}_p \\ &&& Z_{jk} \geq 0, \quad \forall j \in \mathcal{N}_p \text{ e } k \in \mathcal{N}_n, \end{aligned} \tag{3.10}$$

para  $\alpha, \beta > 0$ . Assim, podemos resolver iterativamente os seguintes subproblemas

$$\begin{aligned} \underset{Z}{\text{minimizar}} \quad & \frac{1}{2} \left\| \begin{pmatrix} A \\ 0_{1 \times p} \end{pmatrix} - \begin{pmatrix} Y \\ \sqrt{\beta} e_1^t \end{pmatrix} Z \right\|_F^2 \\ \text{sujeito a} \quad & Z \geq 0 \end{aligned} \quad (3.11)$$

e

$$\begin{aligned} \underset{Y}{\text{minimizar}} \quad & \frac{1}{2} \left\| \begin{pmatrix} A^t \\ 0_{p \times m} \end{pmatrix} - \begin{pmatrix} Z^t \\ \sqrt{\alpha} I_p \end{pmatrix} Y^t \right\|_F^2 \\ \text{sujeito a} \quad & Y \geq 0, \end{aligned} \quad (3.12)$$

sendo  $e_1 \in \mathbb{R}^p$ .

Em [81], os autores utilizam o método BPP-M para resolver os LCPs requeridos em cada iteração do algoritmo. Além disso sugerem uma forma de reordenar as colunas para aproveitar os conjuntos  $F$  iguais e assim evitar cálculos computacionais desnecessários. Além disso, são realizadas comparações computacionais deste algoritmo com a maioria dos algoritmos apresentados na literatura para solucionar problemas da forma NMF. Para tanto, é considerado um conjunto de problemas testes constituído de dados sintéticos e decorrentes da transmissão de texto e imagem. Os experimentos mostram que este algoritmo é geralmente muito eficiente para a decomposição NMF.

## 3.2 Algoritmo de gradientes projetados

Esta seção é dedicada aos algoritmos de gradientes projetados para a determinação de um ponto estacionário de um problema não linear com restrições de não negatividade. Em particular, veremos que os métodos de pivotamento em blocos principais têm grande utilidade na implementação de formas preconditionadas desses métodos, que requerem projeções usando a norma elíptica.

O método do gradiente projetado (*The Projected Gradient method* - (PG)) foi inicialmente proposto por Rosen em [119, 120] e é um dos métodos clássicos para problemas de otimização contínua com restrições da forma

$$\begin{aligned} (P') \quad & \text{minimizar} \quad f(z) \\ & \text{sujeito a} \quad z \in \Omega, \end{aligned}$$

sendo  $\Omega \subset \mathbb{R}^n$  um conjunto convexo e  $f : \Omega \rightarrow \mathbb{R}$  uma função continuamente diferenciável. Este método combina técnicas de otimização sem restrições de funções continuamente

diferenciáveis com projeções no conjunto viável  $\Omega$ . Dado um ponto corrente  $z^k \in \Omega$ , é calculado o ponto  $z^k - \eta_k \nabla f(z^k)$  na direção oposta ao gradiente e usando um passo  $\eta_k$  previamente definido. A direção  $d^k$  é a diferença do ponto corrente e da projeção no conjunto viável do ponto  $z^k - \eta_k \nabla f(z^k)$ , ou seja,

$$d^k = \mathcal{P}_\Omega(z^k - \eta_k \nabla f(z^k)) - z^k, \quad (3.13)$$

sendo  $\mathcal{P}_\Omega(\cdot)$  a projeção ortogonal Euclidiana sobre o conjunto  $\Omega$ . Computada a direção  $d^k$ , o próximo iterando  $z^{k+1}$  é obtido ao caminhar a partir de  $z^k$  ao longo de  $d^k$  por um comprimento  $\alpha_k \in (0, 1]$ , sendo este obtido por algum processo de busca unidimensional. Deste modo,

$$z^{k+1} = z^k + \alpha_k d^k,$$

e o processo é repetido. O algoritmo de gradiente projetado é apresentado a seguir.

---

**Algoritmo 3.2** PG

---

Dados:  $k = 0$ ,  $z^0 \in \Omega$ ,  $\varepsilon > 0$

REPITA

Calcule  $\nabla f(z^k)$

Escolha  $\eta_k > 0$

Calcule  $d^k = \mathcal{P}_\Omega(z^k - \eta_k \nabla f(z^k)) - z^k$

Se  $\|d^k\|_2 < \varepsilon$

pare com sucesso

Calcule  $\alpha_k \in (0, 1]$  por um processo de busca linear

$z^{k+1} = z^k + \alpha_k d^k$

$k = k + 1$

---

Para apresentar a aplicação dos algoritmos de pivotamento em blocos principais no método do gradiente projetado, iremos, no decorrer desta seção, nos restringir ao problema ( $P'$ ) cujo conjunto viável é dado por  $\Omega = \mathbb{R}_+^n$ . Cabe salientar que como  $\Omega = \mathbb{R}_+^n$ , a projeção ortogonal Euclidiana de um ponto em  $\Omega$  tem uma estrutura especial e de fácil obtenção. Dados  $z^k$  e  $\eta_k > 0$ , considere  $x = z^k - \eta_k \nabla f(z^k)$ . Assim,

$$[\mathcal{P}_\Omega(x)]_i = \begin{cases} x_i & \text{se } x_i \geq 0 \\ 0 & \text{caso contrário,} \end{cases} \quad (3.14)$$

em outras palavras,  $\mathcal{P}_\Omega(x) = \max\{0, x\}$ .

### Convergência global do algoritmo

A convergência do algoritmo pode ser vista em [2, 21] com base nos resultados discutidos nesta seção.

Ao longo desta seção iremos considerar a definição de ponto estacionário segundo [23]. Dizemos que um ponto  $z^* \in \mathbb{R}^n$  é um *ponto estacionário* do problema ( $P'$ ) quando  $z^* \in \Omega$  e para todo  $z \in \Omega$

$$\nabla f(z^*)^t(z - z^*) \geq 0.$$

Na discussão da convergência global do algoritmo, omitiremos o índice  $k$  para simplificar a notação. Seja  $z^k = \hat{z} \in \mathbb{R}^n$ , considere a direção  $d$  a partir de  $\hat{z}$  como calculada no método PG, ou seja,

$$d = \mathcal{P}_\Omega(\hat{z} - \eta \nabla f(\hat{z})) - \hat{z}, \quad (3.15)$$

para  $\eta > 0$ .

O lema a seguir estabelece que se  $d \neq 0$ , então o algoritmo consegue reduzir o valor da função objetivo se for usada uma técnica de busca linear ao longo dessa direção.

**Lema 3.1** *Dado  $\hat{z} \in \mathbb{R}^n$ , considere  $d$  como descrito em (3.15). Se  $d \neq 0$ , então  $d$  é uma direção de descida para  $f$  a partir de  $\hat{z}$ .*

*Demonstração.* Considere  $u = \hat{z} - \eta \nabla f(\hat{z})$  e  $v = \mathcal{P}_\Omega(u)$ . Pela propriedade do operador projeção, temos que para todo  $z \in \Omega$ ,

$$(u - v)^t(z - v) \leq 0. \quad (3.16)$$

Além disso,

$$u - v = \hat{z} - \eta \nabla f(\hat{z}) - \mathcal{P}_\Omega(\hat{z} - \eta \nabla f(\hat{z})) = -d - \eta \nabla f(\hat{z}). \quad (3.17)$$

Como  $\hat{z} \in \Omega$ , temos que a desigualdade (3.16) vale em particular para  $z = \hat{z}$ . Assim, por (3.16), (3.17) e pela definição de  $v$ , segue que

$$0 \geq (u - v)^t(\hat{z} - v) = (-d - \eta \nabla f(\hat{z}))^t(\hat{z} - \mathcal{P}_\Omega(\hat{z} - \eta \nabla f(\hat{z}))) = (-d - \eta \nabla f(\hat{z}))^t(-d).$$

Consequentemente, como  $\eta > 0$  e  $d \neq 0$ , obtemos que

$$\nabla f(\hat{z})^t d \leq -\frac{1}{\eta} \|d\|^2 < 0$$

e, como desejado,  $d$  é uma direção de descida a partir de  $\hat{z}$ .  $\square$

A fim de provar a convergência global do algoritmo para um ponto estacionário é importante estabelecer o resultado a seguir.

**Lema 3.2** *Considere  $\hat{z} \in \mathbb{R}^n$  e  $d \in \mathbb{R}^n$  dado por (3.15). Se  $d = 0$ , então  $\hat{z}$  é um ponto estacionário do problema  $(P')$ .*

*Demonstração.* Considere  $u = \hat{z} - \eta \nabla f(\hat{z})$ . Pelas definições de  $d$  e  $u$  e pelo fato que  $d = 0$ , segue que

$$\hat{z} = \mathcal{P}_\Omega(u).$$

Deste modo, para todo  $z \in \Omega$ , obtemos que

$$(u - \hat{z})^t(z - \hat{z}) \leq 0. \quad (3.18)$$

Pela definição de  $u$ , temos que  $u - \hat{z} = -\eta \nabla f(\hat{z})$ . Logo, usando isto em (3.18) e o fato que  $\eta > 0$ , temos que, para todo  $z \in \Omega$ ,

$$\nabla f(\hat{z})^t(z - \hat{z}) \geq 0.$$

Portanto,  $\hat{z}$  é um ponto estacionário de  $(P')$  e a demonstração está concluída.  $\square$

De acordo com estes dois lemas, dado um ponto corrente  $\hat{z}$  e a direção de gradiente projetado  $d$ , então há dois casos. Se  $d$  é nula, então  $\hat{z}$  é um ponto estacionário. Caso contrário, é possível reduzir o valor da função objetivo ao longo da direção  $d$  usando uma técnica de busca linear. Assim, não é de se estranhar que o algoritmo possua convergência global para um ponto estacionário sob hipóteses relativamente fracas [2].

## Busca linear

Apresentamos a seguir as definições mais utilizadas na prática para o parâmetro  $\alpha_k$  referente à busca linear utilizada no Algoritmo PG.

O parâmetro  $\alpha_k \in (0, 1]$  é obtido por algum processo de busca linear. Para algumas funções é possível usar a busca exata para o comprimento do passo  $\alpha_k$ , obtida ao minimizar a função  $g : (0, 1] \rightarrow \mathbb{R}$  dada por  $g(t) = f(z^k + td^k)$ . Quando não há uma fórmula explícita, é usual obter  $\alpha_k$  através do critério de Armijo [116], o qual fornece um decréscimo suficiente para  $f$  ao longo da direção  $d^k$ . Este parâmetro também pode ser obtido por um processo de busca não monótona como sugerido em [22]. Na Rotina 1 são dispostas algumas formas de obter este comprimento.

**Rotina 1 (Busca linear)** - cálculo de  $\alpha_k \in (0, 1]$

Dados:  $z^k, d^k, \nabla f(z^k), \beta \in (0, 1), \alpha = 1$

Se  $f(z^k + \alpha d^k) \leq f(z^k) + \alpha \beta \nabla f(z^k)^t d^k$

Faça  $\alpha^k = \alpha$  e pare a rotina

Senão

**Função quadrática:**  $f(x) = c^t x + \frac{1}{2} x^t Q x$

Se  $(d^k)^t Q d^k > 0$

$$\alpha = \min \left\{ 1, -\frac{\nabla f(z^k)^t d^k}{(d^k)^t Q d^k} \right\}$$

Senão

$$\alpha = 1$$

**Função fracionária [77]:**  $f(x) = \frac{x^t A x}{x^t B x}$

Calcule

$$\mu_0 = 2(((z^k)^t A z^k)(B z^k)^t d^k) - ((z^k)^t B z^k)((A z^k)^t d^k)$$

$$\mu_1 = 2(((d^k)^t B d^k)(z^k)^t A z^k) - ((d^k)^t A d^k)((z^k)^t B z^k)$$

$$\mu_2 = 2(((d^k)^t B d^k)(A z^k)^t d^k) - ((d^k)^t A d^k)((B z^k)^t d^k)$$

$$\theta = \mu_1^2 - 4\mu_0\mu_2$$

Se  $\theta < 0$ , faça  $\alpha = 1$

Senão

Calcule  $\tilde{\alpha}_i, i = 1, 2$  soluções da equação  $\mu_0 + \mu_1 \alpha + \mu_2 \alpha^2 = 0$

Considere  $\varphi(\alpha) = f(z^k + \alpha d^k)$

Se  $\theta = 0$ , faça  $\alpha = \operatorname{argmin} \{\varphi(\tilde{\alpha}_1), \varphi(1)\}$

Senão, faça  $\alpha = \operatorname{argmin} \{\varphi(\tilde{\alpha}_1), \varphi(\tilde{\alpha}_2), \varphi(1)\}$

**Funções gerais:**

- **Armijo [116]**

ENQUANTO  $f(z^k + \alpha d^k) > f(z^k) + \alpha \beta \nabla f(z^k)^t d^k$

$$\alpha = \frac{\alpha}{2}$$

FIM

- **Busca não monótona [22]**

Dados:  $M \in \mathbb{N}, \gamma \in (0, 1), 0 < \sigma_1 < \sigma_2 < 1,$

$$Z = \{z^{k-j} \mid 0 \leq j \leq \min\{k, M-1\}\}$$

Calcule  $f_{\max} = \max\{f(z) \mid z \in Z\}$  e  $\delta = \nabla f(z^k)^t d^k$

ENQUANTO  $f(x^k + \alpha d^k) > f_{\max} + \alpha \gamma \delta$

$$\alpha_t = -\frac{\alpha^2 \delta}{f(x^k + \alpha d^k) - f(x^k) - \alpha \delta}$$

Se  $\sigma_1 \leq \alpha_t \leq \sigma_2 \alpha$

$$\alpha = \alpha_t$$

Senão,  $\alpha = \alpha/2$

FIM

Faça  $\alpha_k = \alpha$

FIM

O parâmetro  $\eta_k > 0$  pode ser fixo ou ser alterado em cada iteração. Caso seja constante, normalmente este é considerado como um número positivo pequeno, por exemplo  $10^{-1}$ . Os diferentes Algoritmos PG dependem da maneira como o parâmetro  $\eta_k$  é escolhido, como veremos a seguir.

### 3.2.1 Método do gradiente projetado espectral

Nesta seção apresentamos o método do gradiente projetado espectral (*The Spectral Projected Gradient method - (SPG)*) proposto inicialmente por Barzilai e Borwein em [13] para problemas irrestritos e estendido em [21, 22, 23] por Birgin, Martínez e Raydan para problemas com restrições convexas. Este método é um caso particular do Algoritmo PG em que o parâmetro  $\eta_k$  é denominado de parâmetro espectral e é modificado a cada iteração de acordo com a regra apresentada a seguir.

Primeiramente, este parâmetro é inicializado por

$$\eta_0 = \min \left\{ \eta_{\max}, \max \left\{ \eta_{\min}, \frac{1}{\|\mathcal{P}_\Omega(z^0 - \nabla f(z^0)) - z^0\|_\infty} \right\} \right\}, \quad (3.19)$$

sendo  $\eta_{\min} > 0$  um número suficientemente pequeno e  $\eta_{\max} = \eta_{\min}^{-1}$ . Na prática, de acordo com [60], é usual considerar  $\eta_{\min} = \varepsilon_M$ , para  $\varepsilon_M > 0$  a precisão da máquina, a qual é aproximadamente  $10^{-30}$ . Em seguida, os demais parâmetros  $\eta_k$  são atualizados pela rotina a seguir.

#### Rotina 2 (Parâmetro espectral - [22])

Dados:  $k > 0$ ,  $\varepsilon_e > 0$ ,  $z^{k-1} \in \mathbb{R}^n$ ,  $z^k \in \mathbb{R}^n$ ,  $0 < \eta_{\min} \leq \eta_{\max}$

Calcule  $s^k = z^k - z^{k-1}$ ,  $y^k = \nabla f(z^k) - \nabla f(z^{k-1})$ ,  $r_k = (s^k)^t y^k$

Se  $r_k \leq \varepsilon_e$

$$\eta_k = \eta_{\max}$$

Senão

$$\text{Calcule } \eta_k = \min \left\{ \eta_{\max}, \max \left\{ \eta_{\min}, \frac{(s^k)^t s^k}{r_k} \right\} \right\}$$

FIM

A convergência do Algoritmo SPG é herdada da convergência do Algoritmo PG.

### 3.2.2 Método do gradiente projetado preconditionado

O algoritmo do gradiente projetado espectral, apresentado na seção anterior, é de fácil implementação e, normalmente, é eficiente para obter pontos estacionários com baixa precisão (tolerância  $\varepsilon$  do método PG igual a  $10^{-3}$  ou  $10^{-4}$ ), em particular quando a Hessiana nesse ponto é mal condicionada. No entanto, existem problemas em que o algoritmo tem dificuldade em resolver com precisões mais exigentes, como por exemplo os abordados em [4]. Nesta seção discutimos o método do gradiente projetado preconditionado (*Preconditioned Projected-Gradient method - (PPG)*) como uma alternativa para obter um ponto estacionário para o problema ( $P'$ ) com maior precisão.

No Algoritmo PPG, o parâmetro  $\eta_k$  usado no Algoritmo PG é substituído pela matriz  $B_k^{-1}$ , com  $B_k \in \mathbb{R}^{n \times n}$  uma matriz simétrica definida positiva escolhida convenientemente e denominada de matriz de preconditionamento.

Para manter as propriedades do Algoritmo PG em relação à direção de busca (Lemas 3.1 e 3.2), o algoritmo determina as projeções usando a norma elíptica definida por  $\|u\|_B^2 := u^t B u$ , sendo  $B \in \mathbb{R}^{n \times n}$  uma matriz simétrica definida positiva. A projeção em  $\Omega$  segundo a norma elíptica em relação a uma matriz  $B$  será denotada por  $\mathbb{P}_{(\Omega, B)}(\cdot)$ . Assim, a direção  $d^k$  é definida por

$$d^k = \mathbb{P}_{(\Omega, B_k)}(z^k - B_k^{-1} \nabla f(z^k)) - z^k. \quad (3.20)$$

com a norma elíptica a ser usada para calcular a projeção com a matriz  $B_k$ . Usando manipulações algébricas simples para evitar o cálculo da matriz inversa, é possível escrever os passos do Algoritmo PPG da seguinte forma:

Note que se  $B_k = \eta_k I$ , o algoritmo se reduz ao Algoritmo PG.

#### Convergência global do algoritmo

Novamente omitiremos o índice  $k$  para simplificar a notação na discussão da convergência global. Seja  $z^k = \hat{z} \in \mathbb{R}^n$ . Considere a direção  $d$  como calculada no

---

**Algoritmo 3.3** PPG
 

---

Dados:  $z^0 \in \Omega$ ,  $\varepsilon > 0$ ,  $k = 0$

REPITA

Calcule  $\nabla f(z^k)$

Escolha  $B_k \in SPD$

Resolva  $B_k v = \nabla f(z^k)$

Calcule  $d^k = \mathbb{P}_{(\Omega, B_k)}(z^k - v) - z^k$

Se  $\|d^k\|_2 < \varepsilon$

pare, declarando que  $z^k$  é um ponto estacionário aproximado

Calcule  $\alpha_k \in (0, 1]$  por um processo de busca linear

$z^{k+1} = z^k + \alpha_k d^k$

$k = k + 1$

---

algoritmo, isto é,

$$d = \mathbb{P}_{(\Omega, B)}(\hat{z} - B^{-1}\nabla f(\hat{z})) - \hat{z}. \quad (3.21)$$

No lema a seguir verificamos que se o algoritmo não parou, então  $d$  é uma direção de descida para  $f$ .

**Lema 3.3** [15, Teorema 3.1] *Dado um ponto  $\hat{z}$ , considere  $d$  dada por (3.21). Se  $d \neq 0$ , então  $d$  é uma direção de descida para  $f$  a partir de  $\hat{z}$ .*

*Demonstração.* Considere  $v = \mathbb{P}_{(\Omega, B)}(u)$  com  $u = \hat{z} - B^{-1}\nabla f(\hat{z})$ . Seja  $z \in \Omega$ . Como  $\Omega$  é um conjunto convexo e  $v, z \in \Omega$ , segue que  $((1-t)v + tz) \in \Omega$ , para todo  $t \in (0, 1)$ . Usando isto e o fato de  $v$  ser a projeção de  $u$  sobre  $\Omega$ , obtemos que, para todo  $t \in (0, 1)$ ,

$$\|u - v\|_B^2 \leq \|u - ((1-t)v + tz)\|_B^2.$$

Assim,

$$\|u - v\|_B^2 \leq \|u - v\|_B^2 + 2t(u - v)^t B(v - z) + t^2 \|v - z\|_B^2.$$

Tomando  $t \rightarrow 0$ , obtemos que

$$(u - v)^t B(z - v) \leq 0.$$

Portanto, usando a definição de  $u$  acima, temos, para todo  $z \in \Omega$ , que

$$(\hat{z} - B^{-1}\nabla f(\hat{z}) - v)^t B(z - v) \leq 0. \quad (3.22)$$

Como  $\hat{z} \in \Omega$ , a desigualdade (3.22) vale em particular para  $z = \hat{z}$ . Assim, como

$$\hat{z} - v = \hat{z} - \mathbb{P}_{(\Omega, B)}(\hat{z} - B^{-1}\nabla f(\hat{z})) = -d,$$

segue que

$$(-d - B^{-1}\nabla f(\hat{z}))^t B(-d) \leq 0.$$

Como  $B$  é uma matriz definida positiva e  $d \neq 0$ , obtemos que

$$\nabla f(\hat{z})^t d \leq -d^t B d < 0,$$

o que conclui a prova. □

A fim de provar a convergência do método PPG é ainda importante estabelecer a propriedade a seguir.

**Lema 3.4** [15, Teorema 3.1] *Dado um ponto  $\hat{z}$ , considere  $d$  dado por (3.21). Se  $d = 0$ , então  $\hat{z}$  é um ponto estacionário do problema ( $P'$ ).*

*Demonstração.* Se  $d = 0$ , então

$$\hat{z} = \mathbb{P}_{(\Omega, B)}(\hat{z} - B^{-1}\nabla f(\hat{z})).$$

Assim,  $\hat{z}$  é a solução única do seguinte problema

$$\begin{aligned} &\text{minimizar} && \frac{1}{2} \|y - (\hat{z} - B^{-1}\nabla f(\hat{z}))\|_B^2 = g(y) \\ &\text{sujeito a} && y \in \Omega. \end{aligned}$$

Portanto, para todo  $y \in \Omega$ , segue que

$$\nabla g(\hat{z})^t (y - \hat{z}) \geq 0. \tag{3.23}$$

Como

$$\nabla g(\hat{z}) = B(\hat{z} - \hat{z} + B^{-1}\nabla f(\hat{z})) = \nabla f(\hat{z}),$$

por (3.23) temos que  $\hat{z}$  é um ponto estacionário do problema ( $P'$ ) e a demonstração está concluída. □

Tal como no Algoritmo PG, os dois lemas anteriores permitem estabelecer a convergência global do Algoritmo PPG sob hipóteses relativamente fracas [15].

## Projeções

As projeções calculadas no Algoritmo PPG merecem uma discussão mais detalhada, uma vez que essas não são mais obtidas sobre a norma Euclidiana e sim sobre normas elípticas.

Dados  $z^k \in \Omega$  e  $v = B_k^{-1} \nabla f(z^k)$ , a projeção  $x^k$  dada por

$$x^k = \mathbb{P}_{(\Omega, B_k)}(z^k - v),$$

pode ser obtida como a solução do problema quadrático dado por

$$\begin{aligned} &\text{minimizar} && \frac{1}{2} \|x - u\|_{B_k}^2 \\ &\text{sujeito a} && x \in \Omega, \end{aligned} \tag{3.24}$$

para  $u = z^k - v$ . Como

$$\|x - u\|_{B_k}^2 = x^t B_k x - 2(B_k u)^t x + u^t B_k u,$$

o vetor  $u$  está fixado,  $B_k u = B_k z^k - \nabla f(z^k)$  e  $\Omega = \mathbb{R}_+^n$ , temos que o minimizador de (3.24) coincide com a solução do problema

$$\begin{aligned} &\text{minimizar} && \frac{1}{2} x^t B_k x + (\nabla f(z^k) - B_k z^k)^t x \\ &\text{sujeito a} && x \geq 0. \end{aligned} \tag{3.25}$$

Assim, como a matriz  $B_k$  é SPD, o problema (3.25) é equivalente ao problema LCP, como definido na Seção 2.1, com a matriz  $Q = B_k$  e o vetor  $c = \nabla f(z^k) - B_k z^k$ . Portanto, os algoritmos de pivotamento em blocos principais, discutidos no Capítulo 2, podem ser considerados para calcular estas projeções. Assim, a direção  $d^k$  pode ser escrita por

$$d^k = x^k - z^k, \tag{3.26}$$

sendo  $x^k$  a solução de (3.25). Note que a direção  $d^k$  é calculada por (3.25) e (3.26) sem necessidade de calcular a inversa da matriz  $B_k$  nem de resolver o sistema com a matriz  $B_k$  apresentado nos passos do Algoritmo PPG.

A eficiência do Algoritmo PPG depende da escolha da matriz  $B_k$  em cada iteração. Em [15], os autores sugerem que no caso particular em que a função objetivo é uma função quadrática convexa,  $B_k$  seja a diagonal da Hessiana ou uma outra aproximação dessa matriz. Apresentamos a seguir uma sugestão para obtenção destas matrizes.

### 3.2.3 Obtenção de matrizes de condicionamento

Nesta seção propomos uma maneira para obtenção de matrizes de condicionamento  $B_k \in SPD$ . Esta proposta é baseada em ideias discutidas em [46, 48].

Considere  $(P')$  com  $\Omega = \mathbb{R}_+^n$ . Seja  $\bar{z} \in \Omega$  um ponto estacionário de  $(P')$ . Deste modo, existe  $\bar{\lambda} \in \mathbb{R}^n$  tal que

$$\begin{cases} \nabla f(\bar{z}) = \bar{\lambda}, \\ \bar{z} \circ \bar{\lambda} = 0, \\ \bar{\lambda} \geq 0, \bar{z} \geq 0, \end{cases} \quad (3.27)$$

sendo  $\bar{z} \circ \bar{\lambda} := (\bar{z}_1 \bar{\lambda}_1, \dots, \bar{z}_n \bar{\lambda}_n) \in \mathbb{R}^n$ . Considere os seguintes conjuntos de índices

$$\bar{L} = \{i \in \mathcal{N} \mid \bar{z}_i = 0\} \quad \text{e} \quad \bar{F} = \{i \in \mathcal{N} \mid \bar{z}_i > 0\}.$$

Note que  $\bar{L}$  coincide com o conjunto de índices das restrições ativas em  $\bar{z}$ . Além disso,  $\bar{F} \cup \bar{L} = \mathcal{N}$  e  $\bar{F} \cap \bar{L} = \emptyset$ . Usando estes conjuntos e (3.27), podemos concluir que, em um ponto estacionário  $\bar{z}$ , as seguintes propriedades são satisfeitas:

$$\begin{aligned} [\nabla f(\bar{z})]_i &\geq 0 && \text{para } i \in \bar{L} \\ [\nabla f(\bar{z})]_i &= 0 && \text{para } i \in \bar{F}. \end{aligned}$$

Para cada  $i \in \mathcal{N}$ , considere uma função contínua  $a_i : \mathbb{R}^n \rightarrow \mathbb{R}$  limitada em  $\Omega$  tal que  $a_i(z) \geq 0$  para todo  $z \in \Omega$  e  $a_i(z) > 0$  se  $z_i = 0$ . Em [46] os autores sugerem as aproximações para os conjuntos  $\bar{L}$  e  $\bar{F}$  dadas por

$$L(z) = \{i \in \mathcal{N} \mid z_i \leq a_i(z)[\nabla f(z)]_i\}$$

e

$$F(z) = \mathcal{N} \setminus L(z).$$

O resultado a seguir verifica que, perto de um ponto estacionário  $\bar{z}$  de  $(P')$ , os conjuntos  $L(z)$  e  $F(z)$  são boas aproximações para  $\bar{L}$  e  $\bar{F}$ .

**Lema 3.5** *Se  $\bar{z}$  é um ponto estacionário de  $(P')$  e vale a complementaridade estrita, então existe uma vizinhança de  $\bar{z}$  tal que para todo  $z$  nesta vizinhança,*

$$L(z) = \bar{L} \quad \text{e} \quad F(z) = \bar{F}.$$

*Demonstração.* [46, Teo. 3.1] □

Considere  $z^k \in \Omega$ . A ideia geral do método proposto em [46, 48] consiste em estimar quais restrições são ativas na solução e definir as respectivas componentes da

direção de busca. Assim, caso seja estimado que a  $i$ -ésima componente da solução esteja na fronteira, é atribuído  $d_i^k = -z_i^k$  e, desta forma na próxima iteração  $z_i^{k+1} = 0$  e estará na fronteira, desde que o tamanho do passo seja um. Depois de fazer isso, as demais componentes são obtidas resolvendo subproblemas estritamente convexos no espaço das variáveis livres, ou seja, as variáveis cujos índices não foram estimados como sendo ativos.

Seguindo as ideias discutidas em [46, 48] apresentamos a seguir a nossa proposta. Esta usa basicamente a ideia das direções sugeridas naqueles trabalhos, considerando um novo conjunto  $F$ . Verificamos que deste modo as direções obtidas naqueles trabalhos com este novo conjunto se reduzem ao algoritmo PPG com uma matriz de condicionamento específica.

Dados  $\rho > 0$  e  $z^k \in \Omega$ , considere os conjuntos de índices  $L^k = L(z^k)$  e  $F^k = F(z^k)$  com  $a_i(z^k) = \rho$ , para todo  $i \in \mathcal{N}$ . Desta maneira,

$$L^k = \{i \mid z_i^k \leq \rho \nabla_i f(z^k)\} \quad \text{e} \quad F^k = \mathcal{N} \setminus L^k. \quad (3.28)$$

O processo para obter a direção de busca  $d^k$  envolve a Hessiana  $M := \nabla^2 f(z^k)$  de  $f$  em  $z^k$  e consiste de dois passos. No primeiro passo são computadas as componentes  $L^k$  da direção  $d^k$  por

$$d_i^k = -z_i^k \quad \text{para} \quad i \in L^k. \quad (3.29)$$

Em seguida, são obtidas as demais componentes de  $d^k$ . Para tanto, o conjunto  $F^k$  é decomposto em dois conjuntos  $F_1$  e  $F_2$  tais que  $M_{F_1} \in SPD$  e  $F_2 = F^k \setminus F_1$ . Caso  $M \in SPD$ , basta considerar  $F_1 = F^k$  e  $F_2 = \emptyset$ . Apresentaremos mais adiante um processo para obter os conjuntos  $F_1$  e  $F_2$  no caso em que  $M \notin SPD$ . Determinados os conjuntos  $F_1$  e  $F_2$ , as demais componentes de  $d^k$  são obtidas como sendo as soluções  $d_{F_1}^k$  e  $d_{F_2}^k$  dos seguintes subproblemas:

$$(SP_1) \quad \begin{cases} \text{minimizar}_{d_{F_1} \in \mathbb{R}^{|F_1|}} & [\nabla f(z^k)]_{F_1}^t d_{F_1} + \frac{1}{2} d_{F_1}^t M_{F_1} d_{F_1} \\ \text{sujeito a} & d_{F_1} \geq -z_{F_1}^k, \end{cases}$$

e

$$(SP_2) \quad \begin{cases} \text{minimizar}_{d_{F_2} \in \mathbb{R}^{|F_2|}} & [\nabla f(z^k)]_{F_2}^t d_{F_2} + \frac{1}{2\gamma} d_{F_2}^t d_{F_2} \\ \text{sujeito a} & d_{F_2} \geq -z_{F_2}^k, \end{cases}$$

onde  $\gamma > 0$  é dado. Considere  $y_F \in \mathbb{R}^{|F|}$  dado por

$$y_F = z_F^k + d_F^k. \quad (3.30)$$

Assim, podemos reescrever estes dois subproblemas por

$$(SP_3) \quad \begin{cases} \text{minimizar}_{y_{F_1} \in \mathbb{R}^{|F_1|}} & ([\nabla f(z^k)]_{F_1} - M_{F_1} z_{F_1}^k)^t y_{F_1} + \frac{1}{2} y_{F_1}^t M_{F_1} y_{F_1} \\ \text{sujeito a} & y_{F_1} \geq 0 \end{cases}$$

e

$$(SP_4) \quad \begin{cases} \text{minimizar}_{y_{F_2} \in \mathbb{R}^{|F_2|}} & \left( [\nabla f(z^k)]_{F_2} - \frac{1}{\gamma} z_{F_2}^k \right)^t y_{F_2} + \frac{1}{2} y_{F_2}^t \left( \frac{1}{\gamma} I_{F_2} \right) y_{F_2} \\ \text{sujeito a} & y_{F_2} \geq 0, \end{cases}$$

respectivamente. Como em [46, 48], o programa quadrático estritamente convexo  $(SP_3)$  pode ser resolvido por um dos algoritmos de pivotamento em blocos principais. Além disso, a solução  $y_{F_2}$  de  $(SP_4)$  é dada por

$$[y_{F_2}]_i = \begin{cases} 0 & \text{se } z_i^k - \gamma [\nabla f(z^k)]_i \leq 0 \\ z_i^k - \gamma [\nabla f(z^k)]_i & \text{caso contrário.} \end{cases} \quad (3.31)$$

Obtido  $y_F = (y_{F_1}, y_{F_2})$ , então  $d_F$  é computado por

$$d_F^k = y_F - z_F^k. \quad (3.32)$$

Para a introdução da matriz de condicionamento  $B_k$ , como feito anteriormente, o vetor  $d_{L^k}^k$  dado em (3.29) pode ser reescrito por  $d_{L^k}^k = y_L - z_{L^k}^k$ , sendo  $y_L$  a única solução do problema

$$(SP_5) \quad \begin{cases} \text{minimizar}_{y \in \mathbb{R}^{|L|}} & \left( [\nabla f(z^k)]_L - \frac{1}{\rho} z_L^k \right)^t y + \frac{1}{2} y^t \left( \frac{1}{\rho} I_L \right) y \\ \text{sujeito a} & y \geq 0. \end{cases}$$

Portanto, a direção de busca  $d^k \in \mathbb{R}^n$  é dada por

$$d^k = y - z^k \quad (3.33)$$

sendo  $y = (y_F, y_L) \in \mathbb{R}^n$  a solução única de

$$(SP_6) \quad \begin{cases} \text{minimizar} & (\nabla f(z^k) - B_k z^k)^t y + \frac{1}{2} y^t B_k y \\ \text{sujeito a} & y \geq 0, \end{cases}$$

sendo  $B_k \in \mathbb{R}^{n \times n}$  a matriz *SPD* dada por

$$B_k = \begin{pmatrix} \frac{1}{\rho} I_{L^k} & 0 & 0 \\ 0 & M_{F_1} & 0 \\ 0 & 0 & \frac{1}{\gamma} I_{F_2} \end{pmatrix}. \quad (3.34)$$

As considerações apresentadas acima levam à nossa sugestão de usar a matriz  $B_k$  definida por (3.34) com  $L^k$  e  $F^k = F_1 \cup F_2$  como definidos anteriormente e  $M = \nabla^2 f(z^k)$ .

### Obtenção dos conjuntos $F_1$ e $F_2$

Diferentes escolhas de  $M$ ,  $F_1$  e  $F_2$ , geram diferentes matrizes de condicionamento  $B_k$ . Veremos a seguir uma proposta para escolha dos conjuntos  $F_1$  e  $F_2$ .

Considere, como anteriormente,  $M = \nabla^2 f(z^k)$ . Para construir os conjuntos  $F_i \subseteq F^k$ ,  $i = 1, 2$  tais que a submatriz  $M_{F_i} \in SPD$ , sugerimos uma modificação da decomposição  $LDL^t$ , como pode ser vista em [61], para a submatriz  $H = M_{F^k}$  associada ao conjunto  $F^k$ . Neste processo, os elementos de  $H$  são modificados de acordo com a regra de decomposição  $LDL^t$  para uma matriz simétrica  $PD$ . Entretanto, sempre que um elemento diagonal  $d_{ii}$  da matriz diagonal  $D$  não é suficientemente positivo ( $d_{ii} \leq \varepsilon$ , com  $\varepsilon$  um valor positivo pequeno), a linha e a coluna  $i$  são trocadas por  $\frac{1}{\gamma}e^i$ , sendo  $e^i$  o  $i$ -ésimo vetor canônico. Então no final deste processo obtemos os conjuntos  $F_1$  e  $F_2$ , com  $M_{F_1} = LDL^t$  tal que  $D = \text{diag}(d_{ii}) \in \mathbb{R}^{|F_1| \times |F_1|}$ , com  $d_{ii} > \varepsilon$  para todo  $i \in F_1$ , e  $L \in \mathbb{R}^{|F_1| \times |F_1|}$  sendo uma matriz triangular inferior com elementos diagonais unitários. Note que, desta forma,  $M_{F_1} \in SPD$  [61]. Os passos desses processos são apresentados a seguir.

#### Rotina 3 (Conjuntos $F_1$ e $F_2$ )

Dados:  $\varepsilon > 0$ ,  $\gamma > 0$ ,  $F = \{i_1, \dots, i_p\}$ ,  $F_1 = F_2 = \emptyset$  e  $H = M_F$

PARA  $k \in \{1, \dots, p\}$  faça

Se  $H_{kk} > \varepsilon$

$$F_1 = F_1 \cup \{i_k\}.$$

PARA  $i \in \{k+1, \dots, p\}$  faça

$$aux = H_{ik} \text{ e } H_{ik} = \frac{H_{ik}}{H_{kk}}$$

PARA  $j \in \{k+1, \dots, i\}$  faça

$$H_{ij} = H_{ij} - H_{jk} * aux$$

Senão

$$F_2 = F_2 \cup \{i_k\}$$

PARA  $j \in \{k+1, \dots, p\}$  faça

$$H_{jk} = 0$$

PARA  $j \in \{1, \dots, k-1\}$  faça

$$H_{kj} = 0$$

$$H_{kk} = \frac{1}{\gamma}$$

FIM

O parâmetro  $\gamma > 0$  pode ser tomado constante ou calculado, como o parâmetro espectral, pela Rotina 2.

Cabe salientar que a matriz de preconditionamento  $B_k$  utilizada no Algoritmo PPG é dada por (3.34) e, assim, a Rotina 3 é utilizada apenas para determinar os conjuntos de índices  $F_1$  e  $F_2$ . Deste modo, a matriz  $H$  determinada nesta rotina não é necessária para o Algoritmo PPG.

### 3.2.4 Experimentos numéricos

Para atestar a eficiência dos Algoritmos SPG e PPG apresentamos alguns resultados numéricos realizados na resolução de problemas quadráticos convexos com esses algoritmos.

Os testes foram realizados em um computador Dell Vostro 5470, Core(TM) i7-4510U, CPU 2.00GHz, com 8GB RAM, usando MATLAB R2012b (8.0.0.783).

#### Algoritmos testados

Foram considerados os Algoritmos SPG e PPG com matrizes de preconditionamento dadas por (3.34), com  $F_1$  e  $F_2$  obtidos pela Rotina 3. Nos casos quadráticos e convexos foi também utilizada a diagonal da Hessiana da função objetivo como matriz de preconditionamento [15]. Também utilizamos uma versão híbrida dos Algoritmos SPG e PPG. Nesta versão, utilizamos o Algoritmo SPG até que uma precisão de  $\varepsilon_t > 0$  seja atingida, a partir deste momento utilizamos o Algoritmo PPG com as matrizes de preconditionamento dadas por (3.34), com  $F_1$  e  $F_2$  obtidos pela Rotina 3. Assim, temos quatro versões a serem testadas:

- **F1F2**: Algoritmo PPG com matrizes de preconditionamento dadas por (3.34), com  $F_1$  e  $F_2$  obtidos pela Rotina 3.
- **diagH**: Algoritmo PPG com matrizes de preconditionamento dadas pela **diagonal** da matriz **Hessiana**, especialmente para funções quadráticas estritamente convexas.
- **SPG**: Algoritmo **SPG**.
- **SPG+F1F2**: Algoritmo **SPG** até obter uma precisão  $\varepsilon_t$  e em seguida o Algoritmo PPG com matrizes de preconditionamento dadas por (3.34), com  $F_1$  e  $F_2$  obtidos pela Rotina 3.

Como sugerido em [22], consideramos os seguintes parâmetros:

$$\eta_{\min} = 10^{-30}, \quad \eta_{\max} = 10^{30}, \quad \sigma_1 = 0.1, \quad \sigma_2 = 0.9, \quad \gamma = 10^{-4} \quad \text{e} \quad M = 10.$$

A tolerância necessária na Rotina 2 foi considerada como  $\varepsilon_e = 10^{-8}$ .

### Problemas teste

Os testes consistem em resolver problemas quadráticos convexos da forma

$$(PQ) \quad \begin{cases} \text{minimizar} & \frac{1}{2}x^t Q x + c^t x \\ \text{sujeito a} & x \geq 0, \end{cases}$$

com  $Q \in \mathbb{R}^{n \times n}$  definida positiva e  $c \in \mathbb{R}^n$ .

Dado  $J \in \mathbb{N}$ , é gerado inicialmente um vetor  $\lambda$  de autovalores positivos distribuídos em  $J + 1$  faixas, sendo até  $\lfloor n/J \rfloor$  autovalores da ordem  $10^0$ ,  $\lfloor n/J \rfloor$  autovalores da ordem  $10^1$  e assim por diante até  $\lfloor n/J \rfloor$  autovalores da ordem  $10^J$ . Em seguida, realizamos a decomposição QR de uma matriz randômica de ordem  $n$ . Então a matriz  $Q$  é gerada por um produto conveniente da matriz unitária resultante da decomposição, sua transposta e da matriz diagonal obtida pelo vetor de autovalores. Na Rotina 4 os passos para obtenção dessas matrizes são apresentados em detalhes.

### Rotina 4 (Matriz $Q \in \mathbb{R}^{n \times n}$ com módulo dos autovalores em $J$ faixas)

Dados:  $n \in \mathbb{N}$ ,  $J \in \mathbb{N}$ ,  $p \in (0, 1)$

Faça  $\lambda = (0, \dots, 0)$

PARA  $i \in \mathcal{N}$  faça

Tome  $r$  o resto da divisão de  $(i - 1)$  por  $(J + 1)$

$$\lambda(i) = 10^r$$

Tome  $\tau$  um vetor randômico de dimensão  $n$  com componentes no intervalo  $[0, 1]$

$$\lambda = \lambda + p(\lambda \circ \tau)$$

Considere  $A \in \mathbb{R}^{n \times n}$  uma matriz randômica

Faça  $[M, RR] = QR(A)$ , sendo  $QR$  a decomposição  $QR$

$$Q = M * \text{diag}(\lambda) * M^t.$$

Foram testados 20 problemas de dimensão 1000 da forma  $(QP)$ . O vetor  $c$  foi fixado com todas as componentes  $-1$ . As matrizes  $Q$  foram obtidas pela Rotina 4 considerando  $J = 4$  e  $p = 0.1$ . Os pontos iniciais  $x^0 \in \mathbb{R}^n$  foram gerados randomicamente

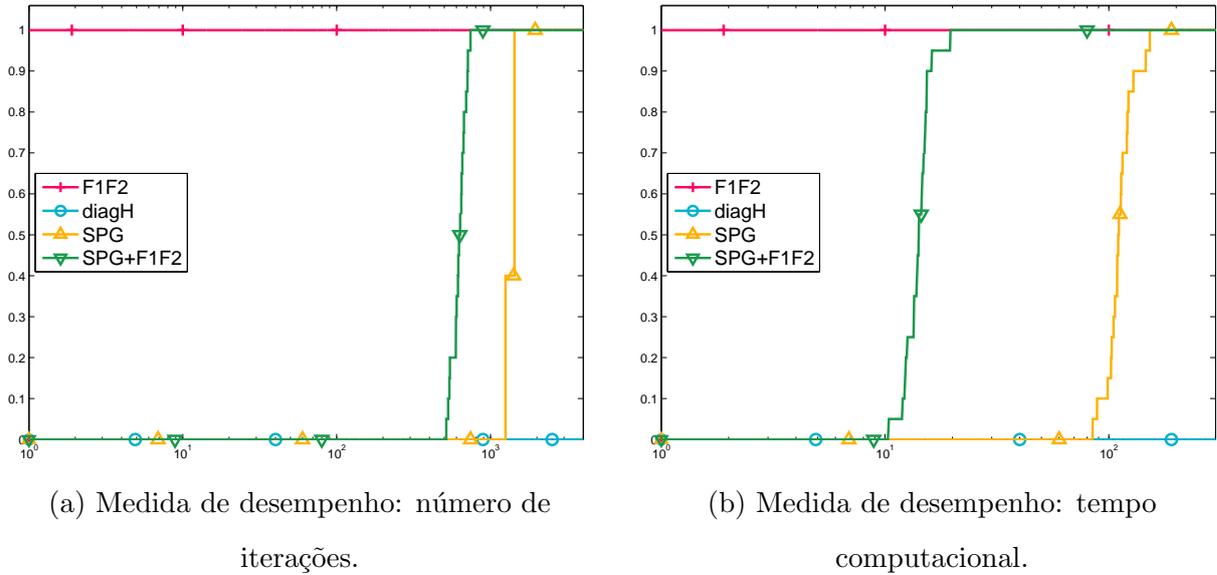
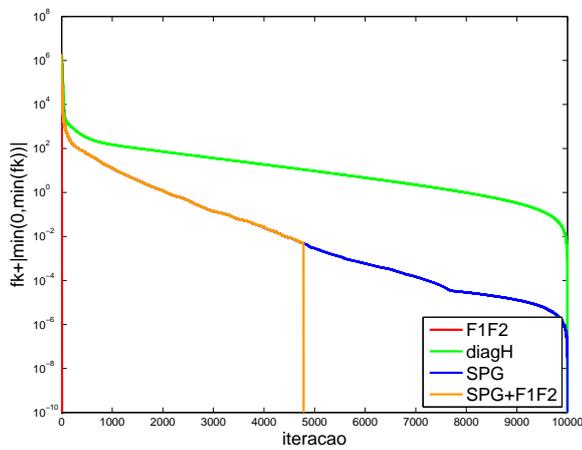


Figura 3.1: Gráfico de desempenho dos Algoritmos F1F2, diagH, SPG e SPG+F1F2.

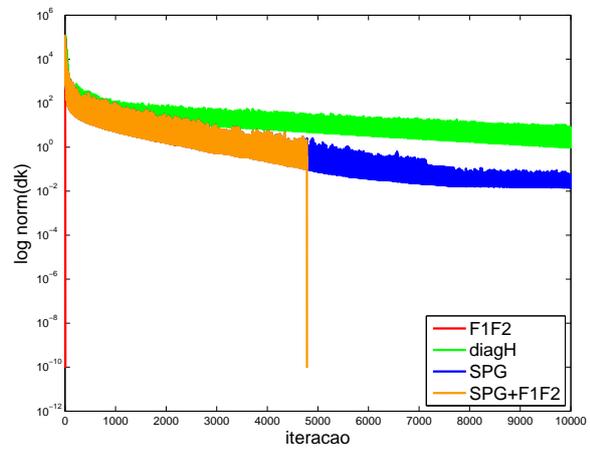
com componentes no intervalo  $[0, 5]$ . O número máximo de iterações foi fixado em 10000. Para a instância SPG+F1F2 foi considerado o parâmetro  $\varepsilon_t = 10^{-1}$ .

Consideramos o conceito de perfil de desempenho [42] para realizar a comparação dos algoritmos. Adotamos o número de iterações e o tempo computacional gasto (CPU) como medidas de desempenho nas comparações dos algoritmos. Na Figura 3.1 apresentamos os gráficos com os perfis de desempenho das quatro rotinas F1F2, diagH, SPG e SPG+F1F2 com relação a essas medidas. Podemos perceber nesta figura que o Algoritmo PPG com as matrizes de condicionamento dadas por (3.34), com  $F_1$  e  $F_2$  obtidos pela Rotina 3, obteve o melhor desempenho tanto com relação ao número de iterações como com relação ao tempo computacional gasto. O pior desempenho foi obtido pelo método PPG com a matriz de condicionamento sendo a diagonal da matriz  $Q$ , o qual não conseguiu resolver nenhum dos problemas em 10000 iterações.

A Figura 3.2 mostra a variação na escala logarítmica do valor da função objetivo (figura da esquerda) e a variação da medida de estacionariedade do problema (figura da direita) ao longo das iterações para as quatro instâncias na resolução de um dos problemas. Esta figura ilustra a dificuldade que o Algoritmo SPG tem para aumentar a precisão ao longo das iterações. Podemos perceber ainda a eficiência do Algoritmo SPG+F1F2 para obter a solução com uma precisão alta.



(a)  $k \times f(x^k) + |\min\{0, \min\{f(x^k)\}\}|$



(b)  $k \times \left\| \mathcal{P}_\Omega(x^k - \nabla f(x^k)) - x^k \right\|$

Figura 3.2: Comportamento dos Algoritmos F1F2, diagH, SPG e SPG+F1F2 ao longo das iterações.

# Conclusões

Nesta tese propusemos, inicialmente, um algoritmo de filtro com restauração inexata para problemas de programação não linear restritos sem fazer uso das derivadas da função objetivo. O algoritmo é dividido em duas etapas: uma fase de viabilidade e uma fase de otimalidade. Estas fases são independentes e o acoplamento entre elas é feito pelo filtro, onde pode ser usado tanto o critério de filtro original [55] quanto o inclinado [26]. A fase de viabilidade reduz a medida de inviabilidade  $h$  sem avaliar a função objetivo e pode ser computada por qualquer algoritmo que reduza suficientemente  $h$ , como por exemplo o proposto em [93]. Já a fase de otimalidade abrange técnicas de região de confiança sem derivadas. Seus modelos podem ser lineares ou quadráticos e ser obtidos por qualquer técnica que aproxime adequadamente a função objetivo. A principal contribuição da tese neste contexto foi provar que, sob hipóteses razoáveis, o passo completo satisfaz uma condição de eficiência [105, 106] que diz essencialmente que perto de um ponto viável não estacionário, o decréscimo da função objetivo é grande. E então a convergência global do algoritmo é herdada dos resultados provados em [78, 106]. O algoritmo foi implementado em MATLAB e testado para um conjunto de problemas selecionados da coleção CUTEST. Nesta implementação os modelos foram construídos por interpolação polinomial. Os testes mostram que a atualização do conjunto interpolador merece atenção em trabalhos futuros, visto que a falta de um bom controle do posicionamento deste conjunto gera erros numéricos na construção dos modelos. Excluindo os problemas em que este tipo de erro ocorreu, o algoritmo teve um bom desempenho comparado com o algoritmo de filtro com restauração inexata com derivadas proposto em [106]. Sugerimos também, como trabalho futuro, a extensão do algoritmo para problemas em que as derivadas das restrições também não estejam disponíveis.

Um segundo tema abordado neste trabalho foi o problema de complementaridade linear. Neste contexto fizemos uma revisão de algoritmos de pivotamento em blocos

principais reescrevendo-os numa linguagem unificada. Exibimos exemplos de problemas para os quais estes algoritmos entram em ciclo, e portanto não convergem. Uma revisão de resultados de convergência é apresentada, ou para uma subclasse restritiva de problemas ou para versões modificadas e mais lentas do algoritmo. Nossa conjectura é de que seja possível provar a convergência do algoritmo, sem modificações, quando o problema tem matriz- $P$  e diagonalmente dominante por colunas. Esta conjectura é acentuada através de alguns experimentos numéricos que mostram um excelente desempenho do algoritmo nestes casos.

Foram apresentadas também duas aplicações para os algoritmos de pivotamento em blocos principais. A primeira delas consiste na decomposição não negativa de uma matriz. A segunda no cálculo da projeção necessária no algoritmo de gradiente projetado preconditionado com norma elíptica. Os algoritmos preconditionados visam melhorar a precisão da solução obtida por um algoritmo sem preconditionamento. Como contribuição apresentamos uma matriz de preconditionamento com alguns resultados numéricos promissores.

# Referências Bibliográficas

- [1] R. Andreani, E.G. Birgin e J.M. Martínez. On augmented Lagrangian methods with general lower-level constraints. *SIAM J. Optim.*, 18:1286–1309, 2007.
- [2] R. Andreani, G. Haeser, J. M. Martínez e J. Yuan. Spectral projected gradient and variable metric methods for optimization with linear inequalities. *IMA J. Numer. Anal.*, 25:221–252, 2005.
- [3] R. Andreani, G. Haeser, M. L. Schuverdt e P. J. S. Silva. A relaxed constant positive linear dependence constraint qualification and applications. *Math. Program.*, 135:255–273, 2012.
- [4] R. Andreani, J. J. Júdice, J. M. Martínez e T. Martini. Feasibility problems with complementarity constraints. *Eur. J. Oper. Res.*, 249:41–54, 2016.
- [5] R. Andreani e P. J. S. Silva. Constant rank constraint qualification: A geometric introduction. *Pesquisa Operacional*, 34:481–494, 2014.
- [6] M. Anitescu. On solving mathematical programs with complementarity constraints as nonlinear programs. *SIAM J. Optim.*, 15:1203–1236, 2005.
- [7] M. B. Arouxét, N. Echebest e E. A. Pilotta. Active-set strategy in Powell’s method for optimization without derivatives. *Comput. Appl. Math.*, 30(1):171–196, 2011.
- [8] M. B. Arouxét, N. Echebest e E. A. Pilotta. Inexact restoration method for nonlinear optimization without derivatives. *Comput. Appl. Math.*, 290:26–43, 2015.
- [9] C. Audet e J. E. Dennis Jr. A pattern search filter method for nonlinear programming without derivatives. *SIAM J. Optim.*, 14(4):980–1010, 2004.
- [10] C. Audet e J. E. Dennis Jr. Mesh adaptive direct search algorithms for constrained optimization. *SIAM J. Optim.*, 17:188–217, 2006.

- [11] C. Audet e J. E. Dennis Jr. A progressive barrier for derivative-free nonlinear programming. *SIAM J. Optim.*, 20:445–472, 2009.
- [12] D. Azofeifa, N. Clark e W. Vargas. Optical and electrical properties of terbium films as a function of hydrogen concentration. *Phys. Status Solidi B*, 242:2005–2009, 2005.
- [13] J. Barzilai e J. M. Borwein. Two point step size gradient methods. *IMA J. Numer. Anal.*, 8:141–148, 1988.
- [14] M. Bazaraa, H. Sherali e C. Shetty. *Nonlinear Programming: Theory and Algorithms*. Athena Scientific, Wiley, Nova Iorque, 2<sup>a</sup> edição, 2006.
- [15] L. Bello e M. Raydan. Preconditioned spectral projected gradient method on convex sets. *J. Comput. Math.*, 23:225–232, 2005.
- [16] M. Berry, M. Browne, A. Langville, V. Pauca e R. Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Comput. Stat. Data An.*, 52:155–173, 2007.
- [17] D. P. Bertsekas. *Constrained Optimization and Lagrange Multipliers*. Academic Press, Nova Iorque, 1982.
- [18] E. G. Birgin, I. E. Chambouleyron e J. M. Martínez. Optimization problems in the estimation of parameters of thin films and the elimination of the influence of the substrate. *J. Comput. Appl. Math.*, 152:35–50, 2003.
- [19] E. G. Birgin, J. M. Martínez e W. F. Mascarenhas. Method of sentinels for packing items within arbitrary convex regions. *J. Oper. Res. Soc.*, 57:735–746, 2006.
- [20] E. G. Birgin, J. M. Martínez, F. H. Nishihara e D. P. Ronconi. Orthogonal packing of rectangular items within arbitrary convex regions by nonlinear optimization. *Comput. Oper. Res.*, 33:3535–3548, 2006.
- [21] E. G. Birgin, J. M. Martínez e M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM J. Optimiz.*, 10:1196–1211, 2000.
- [22] E. G. Birgin, J. M. Martínez e M. Raydan. SPG - Software for convex-constrained optimization. *ACM T. Math. Software*, 27:340–349, 2001.

- [23] E. G. Birgin, J. M. Martínez e M. Raydan. Inexact spectral projected gradient methods on convex sets. *IMA J. Numer. Anal.*, 23(4):539–559, 2003.
- [24] E. G. Birgin, J. M. Martínez e D. P. Ronconi. Optimizing the packing of cylinders into a rectangular container: A nonlinear approach. *Eur. J. Oper. Res.*, 160:19–33, 2005.
- [25] L. F. Bueno, A. Friedlander, J. M. Martínez e F. N. C. Sobral. Inexact restoration method for derivative-free optimization with smooth constraints. *SIAM J. Optim.*, 23(2):1189–1213, 2013.
- [26] C. M. Chin e R. Fletcher. On the global convergence of an SLP-filter algorithm that takes EQP steps. *Math. Program.*, 96(1):161–177, 2003.
- [27] P. G. Ciarlet e P. A. Raviart. General Lagrange and Hermite interpolation in  $\mathbb{R}^n$  with applications to finite elements methods. *Arch. Ration. Mech. An.*, 46:324–330, 1972.
- [28] P. D. Conejo, E. W. Karas, L. G. Pedroso, A. A. Ribeiro e M. Sachine. Global convergence of trust-region algorithms for convex constrained minimization without derivatives. *Appl. Math. Comput.*, 220:324–330, 2013.
- [29] A. R. Conn, N. I. M. Gould e Ph. L. Toint. *Trust-Region Methods*. MPS-SIAM Series on Optimization, SIAM, Filadélfia, 2000.
- [30] A. R. Conn, N. I. M. Gould e L. N. Vicente. Global convergence of general derivative-free trust-region algorithms to first and second order critical points. *SIAM J. Numer. Anal.*, 20:387–415, 2009.
- [31] A. R. Conn, K. Scheinberg e Ph. L. Toint. On the convergence of derivative-free methods for unconstrained optimization. Em A. Iserles e M. Buhmann, editores, *Approximation Theory and Optimization*, Tributo a M. J. D. Powell, pág. 83–108. 1997.
- [32] A. R. Conn, K. Scheinberg e Ph. L. Toint. A derivative free optimization algorithm in practice. Conferência Proceedings of the AIAA, 1998.
- [33] A. R. Conn, K. Scheinberg e L. N. Vicente. Geometry of interpolation sets in derivative free optimization. *Math. Program.*, 111:141–172, 2008.

- [34] A. R. Conn, K. Scheinberg e L. N. Vicente. *Introduction to Derivative-Free Optimization*. MPS-SIAM Series on Optimization, SIAM, Filadélfia, 2009.
- [35] A. R. Conn e Ph. L. Toint. An algorithm using quadratic interpolation for unconstrained derivative free optimization. Em G. D. Pillo e F. Gianessi, editores, *Nonlinear Optimization and Applications*, pág. 27–47. Springer US, Nova Iorque, 1996.
- [36] R. W. Cottle, Z. Csizmadia, L. M. Fernandes e J. J. Júdice. Notes on block pivoting algorithms for the linear complementarity problem. Relatório técnico, Universidade de Coimbra, 2008.
- [37] R. W. Cottle, J. S. Pang e R. E. Stone. *The linear complementarity problem*. Academic Press, Boston, 1992.
- [38] F. Curiel, W. E. Vargas e R. G. Barrera. Visible spectral dependence of the scattering and absorption coefficients of pigmented coatings from inversion of diffuse reflectance spectra. *Applied Optics*, 41:5969–5978, 2002.
- [39] F. E. Curtis, Z. Han e D. P. Robinson. A globally convergent primal-dual active-set framework for large-scale convex quadratic optimization. *Comput. Optim. Appl.*, 60:311–341, 2015.
- [40] P. J. Davis. *Interpolation and approximation*. Courier Corporation, Estados Unidos, 2014.
- [41] M. A. Diniz-Ehrhardt, J. M. Martínez e L. G. Pedroso. Derivative-free methods for nonlinear programming with general lower-level constraints. *Comput. Appl. Math.*, 30:19–52, 2011.
- [42] E.D. Dolan e J.J. Moré. Benchmarking optimization software with performance profiles. *Math. Program.*, 91:201–213, 2002.
- [43] K. L. Dunlap e M. M. Kostreva. Solving more linear complementarity problems with Murty’s Bard-type algorithm. *J. Optimiz. Theory App.*, 77:497–522, 1993.
- [44] N. Echebest, M. L. Schuverdt e R. P. Vignau. A derivative-free method for solving box-constrained underdetermined nonlinear systems of equations. *Appl. Math. Comput.*, 219:3198–3208, 2012.

- [45] N. Echebest, M. L. Schuverdt e R. P. Vignau. An inexact restoration derivative-free filter method for nonlinear programming. *Comput. Appl. Math.*, 2015. DOI 10.1007/s40314-015-0253-0.
- [46] F. Facchinei, J. Júdice e J. Soares. An active set Newton algorithm for large-scale nonlinear programs with box constraints. *SIAM J. Optimiz.*, 8:158–186, 1998.
- [47] G. Fasano, J. L. Morales e J. Nocedal. On the geometry phase in model-based algorithms for derivative-free optimization. *Optim. Method Softw.*, 24:145–154, 2009.
- [48] L. Fernandes, A. Fischer, J. Júdice, C. Requejo e J. Soares. A block active set algorithm for large-scale quadratic programming with box constraints. *Ann. Oper. Res.*, 81:75–95, 1998.
- [49] L. Fernandes, A. Friedlander, M. Guedes e J. Júdice. Solution of a general linear complementarity problem using smooth optimization and its application to bilinear programming and LCP. *Appl. Math. Opt.*, 43:1–19, 2001.
- [50] L. Fernandes, J. Júdice e J. Patrício. An investigation of interior-point and block pivoting algorithms for large-scale symmetric monotone linear complementarity problems. *Comput. Optim. Appl.*, 5:49–77, 1995.
- [51] P. S. Ferreira, E. W. Karas e M. Sachine. A globally convergent trust-region algorithm for unconstrained derivative-free optimization. *Comput. Appl. Math.*, 34:1075–1103, 2015.
- [52] C. Févotte e J. Idier. Algorithms for nonnegative matrix factorization with the  $\beta$ -divergence. *Neural Comput.*, 23:2421–2456, 2011.
- [53] A. Fischer e A. Friedlander. A new line search inexact restoration approach for nonlinear programming. *Comput. Optim. Appl.*, 46:333–346, 2010.
- [54] R. Fletcher, N. I. M. Gould, S. Leyffer, Ph. L. Toint e A. Wachter. Global convergence of trust-region SQP-filter algorithm for general nonlinear programming. *SIAM J. Optim.*, 13:635–659, 2002.
- [55] R. Fletcher e S. Leyffer. Nonlinear programming without a penalty function. *Math. Program.*, 91:239–269, 2002.

- [56] R. Fletcher e S. Leyffer. Filter-type algorithms for solving systems of algebraic equations and inequalities. Em G. di Pillo e A. Murli, editores, *Advances in Optimization and Numerical Analysis*, pág. 259–278. High Performance Algorithms and Software for Nonlinear Optimization, Kluwer, 2003.
- [57] R. Fletcher, S. Leyffer e Ph. L. Toint. On the global convergence of a filter-SQP algorithm. *SIAM J. Optim.*, 13(1):44–59, 2002.
- [58] J. B. Francisco, J. M. Martínez e L. Martínez. Density-based globally convergent trust-region methods for self-consistent field electronic structure calculations. *J. Math. Chem.*, 40:349–377, 2006.
- [59] E. G. Birgin e J. M. Gentil. Evaluating bound-constrained minimization software. *Comput. Optim. Appl.*, 53(2):347–373, 2012.
- [60] E. G. Birgin, J. M. Martínez e M. Raydan. Spectral projected gradient methods: Review and perspectives. *J. Stat. Softw.*, 60(3):539–559, 2014.
- [61] G. Golub e V. Loan. *Matrix Computations*. The John Hopkins University Press, Baltimore, 3<sup>a</sup> edição, 1996.
- [62] C. C. Gonzaga, E. W. Karas e M. Vanti. A globally convergent filter method for nonlinear programming. *SIAM J. Optimiz.*, 14(3):646–669, 2003.
- [63] N. I. M. Gould, S. Leyffer e Ph. L. Toint. A multidimensional filter algorithm for nonlinear equations and nonlinear least-squares. *SIAM J. Optimiz.*, 15(1):17–38, 2004.
- [64] N. I. M. Gould, D. Orban e Ph. L. Toint. CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization. *Comput. Optim. Appl.*, 60:545–557, 2015.
- [65] N. I. M. Gould e Ph. L. Toint. FILTRANE, a Fortran 95 filter-trust-region package for solving least-squares and nonlinear feasibility problems. *ACM T. Math. Software*, 33(1):3–25, 2007.
- [66] S. Gratton, Ph. L. Toint e A. Tröltzsch. An active set trust-region method for derivative-free nonlinear bound-constrained optimization. *Optim. Method Softw.*, 26:873–894, 2011.

- [67] C. Gu e D. Zhu. A secant algorithm with line search filter method for nonlinear optimization. *Appl. Math. Model.*, 35(2):879–894, 2011.
- [68] E. A. E. Gumma, M. H. A. Hashim e M. Montaz Ali. A derivative-free algorithm for linearly constrained optimization problems. *Comput. Optim. Appl.*, 57:599–621, 2014.
- [69] P. T. Harker e J. S Pang. Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications. *Math. Program.*, 48:161–220, 1990.
- [70] R. A. Horn e C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Nova Iorque, EUA, 1986.
- [71] P. Hungerlander, 2015. Comentário pessoal.
- [72] P. Hungerlander e F. Rendl. An infeasible active set method with step size control for bound constrained convex problems. Relatório técnico, Alpen–Adria Universitat Klagenfurt, 2014.
- [73] G. Isac. *Complementarity Problems*. Springer, Alemanha, 1991.
- [74] A. Izmailov e M. Solodov. *Otimização volume 2 - métodos computacionais*. IMPA, Rio de Janeiro, Brasil, 2ª edição, 2012.
- [75] J. J. Júdice e F. M. Pires. Direct methods for convex quadratic programs subject to box constraints. *Invest. Opl.*, 9:23–56, 1989.
- [76] J. J. Júdice e F. M. Pires. A block principal pivoting algorithm for large-scale strictly monotone linear complementarity problems. *Computers. Ops. Res.*, 21(5):587–596, 1994.
- [77] J. J. Júdice, M. Raydan, S. S. Rosa e S. A. Santos. On the solution of the symmetric eigenvalue complementarity problem by the spectral projected gradient algorithm. *Numer. Algorithms*, 45:391–407, 2008.
- [78] E. W. Karas, A. P. Oening e A. A. Ribeiro. Global convergence of slanting filter methods for nonlinear programming. *Appl. Math. Comput.*, 200:486–500, 2008.

- [79] E. W. Karas, L. G. Pedroso, K. Scheinberg e A. Verdério. On the construction of quadratic models for derivative-free trust-region algorithms. Relatório técnico, Universidade Federal do Paraná, 2016.
- [80] E. W. Karas, A. A. Ribeiro, C. Sagastizábal e M. Solodov. A bundle-filter method for nonsmooth convex constrained optimization. *Math. Program.*, 116:297–320, 2009.
- [81] D. Kim, S. Sra e I. Dhillon. Toward faster nonnegative matrix factorization: A new algorithm and comparisons. pág. 353–362. Eighth IEEE International Conference on ICDM'08, 2008.
- [82] T. G. Kolda, R. M. Lewis e V. Torczon. A generating set direct search augmented Lagrangian algorithm for optimization with a combination of general and linear constraints. Relatório Técnico 5315, Sandia National Laboratories, 2006.
- [83] K. Kunisch e F. Rendl. An infeasible active set method for quadratic problems with simple bounds. *SIAM J. Optimiz.*, 14(1):35–52, 2003.
- [84] D. D. Lee e H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.
- [85] C. E. Lemke. Bimatrix equilibrium points and mathematical programming. *Nature*, 11:681–689, 1965.
- [86] R. M. Lewis e V. Torczon. Pattern search algorithms for bound constrained minimization. *SIAM J. Optim.*, 9:1082–1099, 1999.
- [87] R. M. Lewis e V. Torczon. Pattern search algorithms for linearly constrained minimization. *SIAM J. Optim.*, 10:917–941, 2000.
- [88] C. J. Lin. Projected gradient methods for nonnegative matrix factorization. *Neural Comput.*, 19:2756–2779, 2007.
- [89] J. Long, C. Ma e P. Nie. A new filter method for solving nonlinear complementarity problems. *Appl. Math. Comput.*, 185(1):705–718, 2007.
- [90] J. Long e S. Zeng. A new Filter-Levenberg-Marquart method with disturbance for solving nonlinear complementarity problems. *Appl. Math. Comput.*, 216(2):677–688, 2010.

- [91] S. Lucidi, M. Sciandrone e P. Tseng. Objective-derivative-free methods for constrained optimization. *Math. Program.*, 92:37–59, 2002.
- [92] F. Luengo, M. Raydan, W. Glunt e T. L. Hayden. Preconditioned spectral gradient method. *Numer. Algorithms*, 30:241–258, 2002.
- [93] J. M. Martínez. Inexact-restoration method with Lagrangian tangent decrease and new merit function for nonlinear programming. *J. Optim. Theory Appl.*, 111:39–58, 2001.
- [94] J. M. Martínez e E. A. Pilotta. Inexact restoration algorithm for constrained optimization. *J. Optim. Theory Appl.*, 104:135–163, 2000.
- [95] J. M. Martínez e E. A. Pilotta. Inexact restoration methods for nonlinear programming: Advances and perspectives. Em L. Q. Qi, K. L. Teo e X. Q. Yang, editores, *Optimization and Control with Applications*, Kluwer Academic. 2001.
- [96] J.J. Moré e S.M. Wild. Benchmarking derivative-free optimization algorithms. *SIAM J. Optimiz.*, 20:172–191, 2009.
- [97] K. G. Murty. Note on a Bard-type scheme for solving the complementarity problem. *Opsearch*, 11:123–130, 1974.
- [98] K. G. Murty. *Linear Complementarity, Linear and Nonlinear Programming*. Heldermann, Berlin, 1988.
- [99] I.X.M. Nascimento. Otimização sem derivadas: sobre a construção e qualidade de modelos quadráticos na solução de problemas irrestritos. Dissertação de mestrado em matemática, UNICAMP–IMECC, 2014.
- [100] J. Nocedal e S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer-Verlag, EUA, 2ª edição, 2006.
- [101] Matrix Market. Japanese National Institute of Advanced Industrial Science e Technology. <http://math.nist.gov/MatrixMarket/>. Acessado em: maio de 2015.
- [102] P. Paatero e U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5:111–126, 1994.

- [103] Y. Pei e D. Zhu. A trust-region algorithm combining line search filter technique for nonlinear constrained optimization. *Int. J. Comput. Math.*, 91(8):1817–1839, 2014.
- [104] Y. Peng, H. Feng e Q. Li. A filter-variable-metric method for nonsmooth convex constrained optimization. *Appl. Math. Comput.*, 208(1):119–128, 2009.
- [105] G. A. Peričaro. *Algoritmos de filtro globalmente convergentes: teoria, implementação e aplicação*. Tese, Programa de Pós-Graduação em Métodos Numéricos da Universidade Federal do Paraná, Brasil, 2011.
- [106] G. A. Peričaro, A. A. Ribeiro e E. W. Karas. Global convergence of a general filter algorithm based on an efficiency condition of the step. *Appl. Math. Comput.*, 219:9581–9597, 2013.
- [107] L. F. Portugal, J. J. Júdice e L. N. Vicente. A comparison of block pivoting and interior-point algorithms for linear least squares problems with nonnegative variables. *Math. Comput.*, 63:625–643, 1994.
- [108] M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Computer Journal*, 7:155–162, 1964.
- [109] M. J. D. Powell. UOBYQA: unconstrained optimization by quadratic approximation. *Math. Program.*, 92:555–582, 2002.
- [110] M. J. D. Powell. The NEWUOA software for unconstrained optimization without derivatives. Em G. Di Pillo e M. Roma, editores, *Optimization, Large-Scale Nonlinear*, pág. 255–297. Springer, 2006.
- [111] M. J. D. Powell. Developments of NEWUOA for minimization without derivatives. *IMA J. Numer. Anal.*, 28:649–664, 2008.
- [112] M. J. D. Powell. The BOBYQA algorithm for bound constrained optimization without derivatives. Relatório técnico, Department of Applied Mathematics and Theoretical Physics, 2009. DAMTP 2009/NA06.
- [113] M. J. D. Powell. On derivative-free optimization with linear constraints. 21st ISMP, 2012.
- [114] M. J. D. Powell. On the convergence of trust region algorithms for unconstrained minimization without derivatives. *Comput. Optim. Appl.*, 53:527–555, 2012.

- [115] A. Ramirez-Porras e W. E. Vargas-Castro. Transmission of visible light through oxidized copper films: feasibility of using a spectral projected gradient method. *Applied Optics*, 43:1508–1514, 2004.
- [116] A. A. Ribeiro e E. W. Karas. *Otimização Contínua: aspectos teóricos e computacionais*. Cengage Learning, São Paulo, Brasil, 2013.
- [117] A. A. Ribeiro, E. W. Karas e C. C. Gonzaga. Global convergence of filter methods for nonlinear programming. *SIAM J. Optim.*, 19(3):1231–1249, 2008.
- [118] L. M. Rios e N. V. Sahinidis. Derivative-free optimization: a review of algorithms and comparison of software implementations. *J. Global Optim.*, 56:1247–1293, 2013.
- [119] J. B. Rosen. The gradient projection method for nonlinear programming. part i. linear constraints. *J. Soc. Ind. Appl. Math.*, 8:181–217, 1960.
- [120] J. B. Rosen. The gradient projection method for nonlinear programming. part ii. nonlinear constraints. *J. Soc. Ind. Appl. Math.*, 9(4):514–532, 1961.
- [121] S. A. Santos. Trust-region-based methods for nonlinear programming: recent advances and perspectives. *Pesquisa Operacional*, 34:447–462, 2014.
- [122] K. Scheinberg e Ph. L. Toint. Self-correcting geometry in model-based algorithms for derivative-free unconstrained optimization. *SIAM J. Optim.*, 20(6):3512–3532, 2010.
- [123] A. Stickney e L. Watson. Digraph models of Bard-type algorithms for the linear complementarity problem. *Math. Oper. Res.*, 3:322–333, 1978.
- [124] S. Takriti e K. G. Murty. On the convergence of the block principal pivotal algorithm for the LCP. *Eur. J. Oper. Res.*, 102:657–666, 1997.
- [125] M. Ulbrich, S. Ulbrich e L. N. Vicente. A globally convergent primal-dual interior-point filter method for nonlinear programming. *Math. Program.*, 100(2):379–410, 2004.
- [126] W. E. Vargas, D. E. Azoifeifa e N. Clark. Retrieved optical properties of thin films on absorbing substrates from transmittance measurements by application of a spectral projected gradient method. *Thin Solid Films*, 425:1–8, 2003.

- [127] A. Wächter e L. T. Biegler. Line search filter methods for nonlinear programming: motivation and global convergence. *SIAM J. Optimiz.*, 16(1):1–31, 2005.
- [128] X. Wang, Z. Zhu, S. Zuo e Q. Huang. An SQP-filter method for inequality constrained optimization and its global convergence. *Appl. Math. Comput.*, 217(24):10224–10230, 2011.
- [129] Y. X. Yuan. Recent advances in trust region algorithms. *Math. Program.*, 151(1):249–281, 2015.

# APÊNDICIES

## PARTE 1



# Apêndice A

## Resultados auxiliares – Capítulo 1

Neste apêndice discutimos as provas dos Teoremas 1.13 e 1.14, os quais garantem a convergência global do Algoritmo 1.1. Cabe ressaltar que este algoritmo se encaixa nos algoritmos gerais de filtro propostos em [106, Alg. 1] e [78, Alg. 2.1]. Para tanto, vamos assumir que o algoritmo gerou sequências infinitas  $(x^k)$  e  $(z^k)$  e que as Hipóteses H1-H5 estão satisfeitas.

O primeiro resultado é decorrente exclusivamente do Algoritmo 1.1 e garante que se este teve terminação finita, então um ponto estacionário com precisão  $\varepsilon$  foi obtido.

**Lema A.1** *Se o algoritmo parou de forma finita, então um ponto estacionário com precisão  $\varepsilon > 0$  foi obtido.*

*Demonstração.* Considere  $k$  a última iteração realizada e  $\varepsilon$  suficientemente pequeno. Pela desigualdade triangular e propriedade de contração da projeção, temos que

$$\begin{aligned} \|\mathcal{P}_{\mathcal{L}(z^k)}(z^k - \nabla f(z^k)) - z^k\| &\leq \|\mathcal{P}_{\mathcal{L}(z^k)}(z^k - g_{k_j}) - z^k\| + \\ &\quad + \|\mathcal{P}_{\mathcal{L}(z^k)}(z^k - \nabla f(z^k)) - \mathcal{P}_{\mathcal{L}(z^k)}(z^k - g_{k_j})\| \\ &\leq \|\mathcal{P}_{\mathcal{L}(z^k)}(z^k - g_{k_j}) - z^k\| + \|\nabla f(z^k) - g_{k_j}\|. \end{aligned}$$

Usando a definição de  $\pi_{k_j}$  dada em (1.9) e a Hipótese 4, segue que

$$\|\mathcal{P}_{\mathcal{L}(z^k)}(-\nabla f(z^k))\| \leq \pi_{k_j}(z^k) + \sigma \delta_{k_j}.$$

Com isso e o fato que  $k$  foi a última iteração realizada, temos que

$$\|\mathcal{P}_{\mathcal{L}(z^k)}(-\nabla f(z^k))\| \leq \left(\frac{1}{\mu} + \sigma\right) \varepsilon$$

e um ponto estacionário com precisão  $\varepsilon$  foi obtido. □

Primeiramente vamos nos concentrar nos resultados garantidos para o algoritmo geral de filtro proposto em [106, Alg. 1], os quais são herdados pelo nosso algoritmo e pelo [78, Alg. 2.1]. No lema a seguir, verificamos que, se um ponto corrente é não estacionário, então um novo ponto fora do filtro temporário pode ser calculado e nenhum par do filtro corresponde a um ponto viável.

**Lema A.2** [106, Lemma 2.1] *Para todo  $k \in \mathbb{N}$  tal que  $x^k \in \mathbb{R}^n$  é não estacionário valem os seguintes fatos:*

(i)  $h^j > 0$  para todo  $j \in \mathbb{N}$  tal que  $(f^j, h^j) \in F_k$ ;

(ii) existe o ponto  $x^{k+1} \notin \widehat{\mathcal{F}}_k$ .

*Demonstração.* Vamos provar o resultado por indução. Considere  $k = 0$ . O item (i) é trivial, uma vez que,  $F_0 = \emptyset$ . Além disso,  $\widehat{F}_0 = \{(f^0, h^0)\}$ . Se  $h^0 = 0$ , como  $x^0$  não é solução do problema (pois é não estacionário), então basta tomar  $x^1$  viável tal que  $f^1 < f^0$ , pois assim  $f^1 + \alpha h^1 = f^1 < f^0 = f^0 - \alpha h^0$ . Caso contrário, escolha  $x^1$  viável e conseqüentemente  $0 = h^1 < (1 - \alpha)h^0$ . Portanto,  $x^1 \notin \widehat{\mathcal{F}}_0$  em ambos os casos e (ii) vale para  $k = 0$ .

Suponha como hipótese de indução que os resultados são válidos para algum  $k - 1 \geq 0$ . Vamos verificar inicialmente o item (i). Caso  $k - 1$  seja uma iteração do tipo  $f$ , temos  $F_k = F_{k-1}$  e usando a hipótese de indução obtemos o resultado. Caso contrário,  $k - 1$  é uma iteração do tipo  $h$  e  $F_k = F_{k-1} \cup \{(f^{k-1}, h^{k-1})\}$ . Assim, novamente pela hipótese de indução, basta verificar que  $h^{k-1} > 0$ . Suponha, por contradição, que  $h^{k-1} = 0$ . Pela hipótese de indução, existe  $x^k \notin \widehat{\mathcal{F}}_{k-1}$ , logo por (1.13) e (1.14) teríamos que  $f^k < f^{k-1}$  e  $k - 1$  seria uma iteração do tipo  $f$ , o que é uma contradição, provando o item (i).

Para o segundo item temos, pela hipótese de indução, que existe  $x^k \notin \widehat{\mathcal{F}}_{k-1}$ . Se  $h^k > 0$ , como  $h^j > 0$  para todo  $(f^j, h^j) \in F_{k-1}$  então basta tomar  $x^{k+1}$  como qualquer ponto viável. Por outro lado, se  $h^k = 0$ , então como  $x^k$  não é solução do problema podemos obter um ponto  $x^{k+1}$  viável, tal que

$$f^{k+1} + \alpha h^{k+1} = f^{k+1} < f^k = f^k - \alpha h^k,$$

o que prova (1.13) e (1.14) e conclui a demonstração.  $\square$

O resultado a seguir apresenta algumas propriedades decorrentes diretamente do mecanismo do Algoritmo 1 de [106], e consequentemente, do Algoritmo 1.1 e do Algoritmo 2.1 de [78].

**Lema A.3** *Dado  $k \in \mathbb{N}$ , temos*

(i)  $\mathcal{F}_k \subset \mathcal{F}_j$ , para todo  $j > k$ ;

(ii)  $x^{k+p} \notin \mathcal{F}_{k+1}$ , para todo  $p \geq 1$ ;

(iii) *pelo menos uma das duas situações ocorre:*

(a)  $h^{k+1} < (1 - \alpha)h^k$ , ou

(b)  $f^{k+1} < f^k - \alpha h^k$ , caso seja usado o filtro original ou  
 $f^{k+1} + \alpha h^{k+1} < f^k$ , caso seja usado o filtro inclinado.

*Demonstração.* Considere uma iteração  $k$ . De acordo com o mecanismo do Algoritmo 1.1 temos

$$\mathcal{F}_{k+1} = \begin{cases} \mathcal{F}_k, & \text{quando } k \text{ é uma iteração } f \\ \mathcal{F}_k \cup \mathcal{R}_k, & \text{quando } k \text{ é uma iteração } h, \end{cases}$$

logo  $\mathcal{F}_k \subset \mathcal{F}_{k+1}$ . Usando este fato recursivamente verificamos o primeiro item. Para garantir o próximo tópico, considere  $p \geq 1$ . Usando o item anterior e a definição do filtro temporário temos  $\mathcal{F}_{k+1} \subset \mathcal{F}_{k+p} \subset \widehat{\mathcal{F}}_{k+p-1}$ . Como  $x^{k+p} \notin \widehat{\mathcal{F}}_{k+p-1}$ , temos que  $x^{k+p} \notin \mathcal{F}_{k+1}$ , verificando a condição (ii). O terceiro e último item decorre diretamente da definição de  $\widehat{\mathcal{F}}_k$  e da exigência do algoritmo de que  $x^{k+1} \notin \widehat{\mathcal{F}}_k$ .  $\square$

Considere o conjunto

$$\mathcal{K}_h = \{k \in \mathbb{N} \mid \text{o par } (f^k, h^k) \text{ entra no filtro}\}, \quad (\text{A.1})$$

constituído pelos índices das iterações do tipo  $h$ .

**Lema A.4** [106, Lema 1.2] *Se  $\mathcal{K}_h$  é um conjunto infinito, então  $h^k \xrightarrow{\mathcal{K}_h} 0$ .*

*Demonstração.* Suponha, por contradição, que existe  $\delta > 0$  tal que o conjunto  $\mathcal{K}$  definido por

$$\mathcal{K} = \{k \in \mathcal{K}_h \mid h^k \geq \delta\}$$

seja infinito. Por H1 temos que  $x \mapsto (f(x), h(x))$  é contínua e por H2 temos que a sequência  $\{x^k\}$  está contida em um conjunto compacto. Deste modo, existe um conjunto

$\mathcal{K}_1 \subset \mathcal{K}$  tal que a sequência  $((f^k, h^k))_{k \in \mathcal{K}_1}$  é convergente. Portanto, podemos tomar índices  $j, k \in \mathcal{K}_1$  com  $j < k$  tais que para  $\alpha \in (0, 1)$

$$\|(f^k, h^k) - (f^j, h^j)\| < \alpha\delta. \quad (\text{A.2})$$

Além disso, note que se  $x \in \widehat{\mathcal{F}}_j$ , então  $h(x) - h^j \geq -\alpha h^j$ . Ademais, temos que  $f(x) - f^j \geq -\alpha h^j$  caso o filtro considerado seja o original e  $f(x) - f^j \geq -\alpha h(x)$  caso seja o inclinado. Como  $k \in \mathcal{K}$ , temos  $\delta \leq \min\{h^k, h^j\}$ . Usando isto em (A.2), temos que  $f^j - f^k < \alpha \min\{h^j, h^k\}$  e  $h^j - h^k < \alpha h^j$ . Deste modo, por (1.13) e (1.14) temos que o ponto  $x^k \in \widehat{\mathcal{F}}_j$ . Como  $j \in \mathcal{K}_h$ , temos  $\widehat{\mathcal{F}}_j = \mathcal{F}_{j+1}$ . Portanto,  $x^k \in \mathcal{F}_{j+1}$ . Por outro lado, como  $j < k$  temos, pelo Lema A.3 item (ii), que  $x^k \notin \mathcal{F}_{j+1}$ , o que é uma contradição e o lema está demonstrado.  $\square$

Um resultado semelhante ao que acabamos de verificar é provado por Chin e Fletcher em [26] e também em Fletcher et al. em [57], em que os autores também assumem que a lista de pares que entram no filtro é infinita. Em [106], os autores verificam um resultado mais forte, provando que existe pelo menos um ponto de acumulação estacionário sem exigir esta condição. Já Karas et. al em [78] estabelecem que, se a regra de filtro inclinado é usada, todo ponto de acumulação é viável, sem nenhuma hipótese adicional.

O próximo lema garante que pelo menos um ponto de acumulação da sequência gerada pelo algoritmo é viável.

**Lema A.5** [106, Lema 1.3] *Existe um conjunto infinito  $\mathbb{N}' \subset \mathbb{N}$  tal que*

$$h^k \xrightarrow{\mathbb{N}'} 0.$$

*Demonstração.* Se o conjunto  $\mathcal{K}_h$  definido em (A.1) é infinito, o resultado decorre do lema anterior, com  $\mathbb{N}' = \mathcal{K}_h$ . Suponha agora que o conjunto  $\mathcal{K}_h$  é finito. Como consequência imediata, temos que existe  $k_0 \in \mathbb{N}$  tal que toda iteração  $k > k_0$  é do tipo  $f$ . Sendo assim,  $(f^k)_{k \geq k_0}$  é uma sequência decrescente. Usando isto e os fatos de  $f$  ser contínua por H1 e limitada por H2, temos que

$$f(x^k) - f(x^{k+1}) \rightarrow 0. \quad (\text{A.3})$$

Considere  $\mathcal{K}_2$  o conjunto dos índices das iterações que satisfazem a condição (iii)(b) do Lema A.3, isto é, o conjunto

$$\mathcal{K}_2 = \{k \in \mathbb{N} \mid \alpha h^k < f^k - f^{k+1}\}$$

para o filtro original ou o conjunto

$$\mathcal{K}_2 = \{k \in \mathbb{N} \mid \alpha h^{k+1} < f^k - f^{k+1}\}$$

para o filtro inclinado. Se o conjunto  $\mathcal{K}_2$  for infinito, temos diretamente de (A.3) que  $h^k \xrightarrow{\mathbb{N}'} 0$ , com  $\mathbb{N}' = \mathcal{K}_2$  para o filtro original e  $\mathbb{N}' = \{k + 1 \mid k \in \mathcal{K}_2\}$  para o filtro inclinado. Caso contrário, pelo Lema A.3 (iii) temos que existe  $k_1 \in \mathbb{N}$  tal que para todo  $k \geq k_1$

$$h^{k+1} < (1 - \alpha)h^k.$$

Logo

$$h^k - h^{k+1} > \alpha h^k \geq 0, \tag{A.4}$$

e conseqüentemente a seqüência  $(h^k)_{k \geq k_1}$  é decrescente. Assim, como  $h$  é uma função contínua e limitada, temos que  $h^k - h^{k+1} \rightarrow 0$ . Deste modo, por (A.4), temos que  $h^k \rightarrow 0$  quando  $\mathcal{K}_2$  é finito. Portanto, em ambos os casos temos que a seqüência  $(x^k)$  possui pelo menos um ponto de acumulação viável.  $\square$

No próximo lema verificamos um resultado mais forte que o lema anterior, apresentando que todo ponto de acumulação da seqüência gerada pelo Algoritmo 1.1 é viável caso o filtro utilizado seja o inclinado.

**Lema A.6** [78, Teorema 2.3] *Todo ponto de acumulação de uma seqüência gerada pelo Algoritmo 1.1 segundo o filtro inclinado é viável, isto é,*

$$\lim_{k \rightarrow \infty} h(x^k) = 0.$$

*Demonstração.* Seja o conjunto  $\mathcal{K}$  definido por

$$\mathcal{K} = \{k \in \mathbb{N} \mid f^j \geq f^k, \quad \forall j \in \mathbb{N} \text{ com } j > k\}.$$

Considere uma iteração  $k \in \mathcal{K}$ . Como  $k \in \mathcal{K}$ , temos  $f^{k+1} \geq f^k$  e, conseqüentemente,  $k$  é uma iteração do tipo  $h$ . Além disso, temos que, para todo  $j > k$ ,

$$f^j + \alpha h^j \geq f^j \geq f^k. \tag{A.5}$$

Por outro lado, pelo mecanismo do algoritmo e o Lema A.3 (i) temos que

$$x^j \notin \widehat{\mathcal{F}}_{j-1} \supset \mathcal{F}_j \supset \mathcal{F}_{k+1}.$$

Além disso, pelo fato de  $k$  ser uma iteração do tipo  $h$  temos  $\mathcal{F}_{k+1} = \widehat{\mathcal{F}}_k$ . Assim,  $x^j \notin \widehat{\mathcal{F}}_k$ . Deste modo, usando isto, o fato de  $(f^k, h^k) \in \widehat{F}_k$  em (1.14) e por (A.5) temos, para todo  $j > k$ ,

$$h^j < (1 - \alpha)h^k. \quad (\text{A.6})$$

Vamos dividir a demonstração em duas partes. Primeiramente, suponha que o conjunto  $\mathcal{K}$  é infinito. Defina

$$\mathcal{K} = \{\kappa_0, \kappa_1, \dots, \kappa_i, \dots\}.$$

Assim, usando (A.6) recursivamente para os índices em  $\mathcal{K}$  temos que,

$$0 \leq h^{\kappa_i} < (1 - \alpha)^i h^{\kappa_0}.$$

Deste modo,

$$h^k \xrightarrow{\mathcal{K}} 0.$$

Se o complementar de  $\mathcal{K}$  for finito, concluímos que  $h^k \rightarrow 0$ . Suponha agora que  $\mathbb{N} \setminus \mathcal{K}$  é infinito. Considere  $j \notin \mathcal{K}$  com  $j > \kappa_0$ . Como  $\mathcal{K}$  é infinito, existe um índice  $i_j$  tal que  $\kappa_{i_j} < j < \kappa_{i_j+1}$ . Usando novamente (A.6), temos

$$0 \leq h^j < (1 - \alpha)h^{\kappa_{i_j}} < (1 - \alpha)^{(i_j+1)}h^{\kappa_0}.$$

Portanto,  $h^k \rightarrow 0$  quando  $\mathcal{K}$  é um conjunto infinito.

Suponha agora que o conjunto  $\mathcal{K}$  é finito. Seja  $\bar{k} = \max\{k \in \mathcal{K}\}$  e o conjunto  $S = \{s_0, s_1, \dots, s_i, \dots\}$  definido por

$$s_i = \begin{cases} \bar{k} + 1, & \text{para } i = 0 \\ \min\{j > s_{i-1} \mid f^j < f^{s_{i-1}}\}, & \text{para } i > 0. \end{cases}$$

Note que a sequência  $(f^{s_i})_{i \in \mathbb{N}}$  é decrescente e o conjunto  $S$  deve ser infinito, pois do contrário como a sequência  $(x^k)_{k \in \mathbb{N}}$  é infinita, teríamos que o conjunto  $\mathcal{K}$  seria infinito. Se  $s_{i+1} = s_i + 1$ , então  $x^{s_{i+1}} \notin \widehat{\mathcal{F}}_{s_i}$ . Caso contrário, temos que  $s_{i+1} > s_i + 1$  e  $f^{s_{i+1}} \geq f^{s_i}$  e conseqüentemente  $s_i$  é uma iteração  $h$ . Deste modo,  $x^{s_{i+1}} \notin \mathcal{F}_{s_{i+1}} = \widehat{\mathcal{F}}_{s_i}$ . Assim, em ambos os casos temos  $x^{s_{i+1}} \notin \widehat{\mathcal{F}}_{s_i}$ . Portanto, ou temos que vale

$$0 \leq h^{s_{i+1}} < (1 - \alpha)h^{s_i} \quad (\text{A.7})$$

ou

$$f^{s_i} - f^{s_{i+1}} > \alpha h^{s_{i+1}} \geq 0. \quad (\text{A.8})$$

Primeiramente, vamos verificar que  $h^k \xrightarrow{S} 0$ . Se (A.8) vale para todo  $i$  suficientemente grande, o resultado é trivial. Suponha que existe um conjunto  $\mathbb{N}' \subset \mathbb{N}$  infinito tal que (A.8) vale para todo  $i \in \mathbb{N}'$ . Então, por H2 e como  $(f^{s_i})_{i \in \mathbb{N}}$  é decrescente, temos que  $f^{s_i} - f^{s_{i+1}} \rightarrow 0$ . Portanto, usando isto e (A.8), temos  $h^{s_{i+1}} \xrightarrow{\mathbb{N}'} 0$ . Deste modo, dado  $\varepsilon > 0$ , existe  $\ell \in \mathbb{N}$  tal que  $h^{s_{i+1}} < \varepsilon$  para todo  $i \geq \ell$  com  $i \in \mathbb{N}'$ . Vamos verificar por indução que  $h^{s_{\ell+j}} < \varepsilon$ , para todo  $j \in \mathbb{N}$ . Para  $j = 1$  é trivial. Suponha, como hipótese de indução, que esta condição vale para algum  $j \in \mathbb{N}$ . Considere  $\ell + j \notin \mathbb{N}'$ , pois do contrário não há mais nada a fazer. Logo, (A.7) vale em  $\ell + j$  e assim,  $h^{s_{\ell+j+1}} < (1 - \alpha)h^{s_{\ell+j}} < \varepsilon$ . Portanto,  $h^k \xrightarrow{S} 0$ .

Finalmente, vamos mostrar que, no caso em que  $\mathcal{K}$  é finito, todo ponto de acumulação é viável. Dado  $j \notin S$  com  $j > \bar{k}$ , existe  $i \in \mathbb{N}$  tal que  $s_i < j < s_{i+1}$ . Pela definição de  $S$  e como  $s_i + 1 < s_{i+1}$ , temos que  $s_i$  é uma iteração  $h$  e conseqüentemente,  $f^j \geq f^{s_i}$ . Assim,  $h^j < (1 - \alpha)h^{s_i}$  e portanto,  $h^k \rightarrow 0$  como desejado.  $\square$

No lema seguinte mostramos que próximo a um ponto não estacionário, todas as iterações suficientemente grandes reduzem a função objetivo.

**Lema A.7** [106, Lema 1.5] *Considere  $\bar{x} \in X$  não estacionário. Então existem  $\bar{k} \in \mathbb{N}$  e uma vizinhança  $V_5$  de  $\bar{x}$  tais que para todo  $k > \bar{k}$ , com  $x^k \in V_5$ , a iteração  $k$  será do tipo  $f$ , ou seja  $k \notin \mathcal{K}_h$ , sendo  $\mathcal{K}_h$  o conjunto dado em (A.1).*

*Demonstração.* Caso  $\bar{x}$  seja um ponto viável, temos pelo Lema 1.12 que existe uma constante  $C_8 > 0$  e uma vizinhança  $V_4$  de  $\bar{x}$  tais que para todo  $x^k \in V_4$ , temos

$$f(x^k) - f(x^{k+1}) \geq C_8 v^k.$$

Pelo Lema A.2, temos que  $v^k > 0$ . Assim,  $f(x^k) > f(x^{k+1})$  e  $k \notin \mathcal{K}_h$ .

Considere agora o caso em que  $\bar{x}$  é um ponto inviável. Suponha, por contradição, que existe  $\mathcal{K} \subset \mathcal{K}_h$  infinito tal que  $x^k \xrightarrow{\mathcal{K}} \bar{x}$ . Como  $h$  é uma função contínua, temos que  $h(x^k) \xrightarrow{\mathcal{K}} h(\bar{x})$ . Pelo Lema A.4 temos que  $h(\bar{x}) = 0$ , o que contradiz o fato de  $\bar{x}$  ser inviável e o lema está provado.  $\square$

Vamos verificar, no resultado a seguir, que pelo menos um ponto de acumulação da sequência gerada pelo Algoritmo 1.1 é estacionário.

**Teorema A.8** [106, Teorema 1.6] *Toda sequência  $(x^k)_{k \in \mathbb{N}}$  gerada pelo algoritmo possui um ponto de acumulação estacionário.*

*Demonstração.* Considere uma sequência  $(x^k)_{k \in \mathbb{N}}$  gerada pelo algoritmo. Suponha inicialmente que o conjunto  $\mathcal{K}_h$  dado em (A.1) é infinito. Pela Hipótese H2, temos que existe um conjunto  $\mathcal{K}_1 \subset \mathcal{K}_h$  e um ponto  $\bar{x} \in X$  tais que  $x^k \xrightarrow{\mathcal{K}_1} \bar{x}$ . Pela contrapositiva do lema anterior, temos que  $\bar{x}$  é um ponto estacionário.

Suponha agora que o conjunto  $\mathcal{K}_h$  seja finito. Assim, existe  $k_0 \in \mathbb{N}$  tal que toda iteração  $k > k_0$  é do tipo  $f$ . Portanto,  $\{f^k\}_{k \geq k_0}$  é uma sequência decrescente. Usando isto e os fatos de  $f$  ser contínua por H1 e limitada por H2, temos que

$$f(x^k) - f(x^{k+1}) \rightarrow 0. \quad (\text{A.9})$$

Do Lema A.5, temos que a sequência gerada pelo algoritmo possui pelo menos um ponto de acumulação viável. Assim, vamos verificar que o ponto de acumulação viável é também estacionário. Seja o conjunto  $\mathcal{K}$  tal que  $x^k \xrightarrow{\mathcal{K}} \bar{x}$  e suponha por contradição que  $\bar{x}$  não é estacionário. Pelo Lema 1.12, temos que existem  $k_2 \in \mathbb{N}$  e uma constante  $C_8 > 0$  tais que  $f^k - f^{k+1} \geq C_8 v^k$ , para todo  $k \in \mathcal{K}$  com  $k \geq k_2$ . Como  $v^k = v^{k_0} > 0$  para todo  $k \geq k_0$ , temos que existe uma constante  $\bar{c} > 0$  tal que

$$f(x^k) - f(x^{k+1}) \geq \bar{c},$$

para todo  $k \in \mathcal{K}$  com  $k \geq \max\{k_0, k_2\}$ , o que contradiz (A.9). Logo  $\bar{x}$  é um ponto estacionário e a demonstração está concluída.  $\square$

No resultado anterior verificamos uma convergência mais fraca do método, independentemente da regra do filtro adotada. Provamos que se a sequência gerada pelo algoritmo é infinita, então pelo menos um dos pontos de acumulação desta sequência é um ponto estacionário. Porém, quando o critério de filtro inclinado é considerado, conseguimos uma convergência mais forte. Vamos verificar nos próximos resultados que, se for usada a regra de filtro inclinado, então todo ponto de acumulação da sequência é viável e estacionário. Para tanto, consideramos para os demais resultados que o critério considerado é o de filtro inclinado.

**Lema A.9** [78, Lema 3.3] *Considere  $C_4 > 0$  a constante dada em (1.29). Então*

$$f(x^{k+1}) \leq f(x^k) + C_4 h(x^k).$$

*Demonstração.* Por (1.29) e pelo fato que  $f(x^{k+1}) \leq f(z^k)$ , temos que

$$f(x^{k+1}) \leq f(z^k) \leq f(x^k) + C_4 h(x^k)$$

e a prova está concluída.  $\square$

O lema a seguir garante que o valor de  $f$  não aumenta muito em uma sequência de iterações.

**Lema A.10** [78, Lema 3.4] *Sejam  $C_4 > 0$  a constante dada em (1.29) e  $\alpha \in (0, 1)$ . Considere um sequência finita de iterações  $I = \{\bar{k}, \bar{k} + 1, \dots, K\}$  tal que para todo  $k \in I$  segue que  $f^k \geq f^{\bar{k}}$ . Se a regra de filtro é segundo a região  $\mathcal{R}_k$  definida em (1.5), então*

$$f^K \leq f^{\bar{k}} + \frac{C_4}{\alpha} h^{\bar{k}}.$$

*Demonstração.* Vamos inicialmente definir os seguintes valores:

$$\begin{aligned} \Phi_0 &= f^{\bar{k}} \\ \Phi_1 &= \Phi_0 + C_4 h^{\bar{k}} \\ \Phi_2 &= \Phi_1 + C_4(1 - \alpha)h^{\bar{k}} = \Phi_0 + C_4[1 + (1 - \alpha)]h^{\bar{k}} \\ \Phi_j &= \Phi_0 + C_4 \left( \sum_{i=0}^{j-1} (1 - \alpha)^i \right) h^{\bar{k}} \leq \Phi_0 + C_4 h^{\bar{k}}. \end{aligned}$$

Vamos provar que existe um intervalo  $J$  com  $|J| \leq K - \bar{k}$  tal que a sequência neste intervalo teve pelo menos um elemento em cada intervalo  $[\Phi_j, \Phi_{j+1}]$  para  $j \in 0, 1, \dots, J$  e  $f^K \in [\Phi_J, \Phi_{J+1}]$ . Consequentemente,

$$f^K \leq \Phi_0 + \frac{C_4}{\alpha} h^{\bar{k}}.$$

*1º intervalo ( $j = 0$ ).* Como  $f^{\bar{k}+1} \geq f^{\bar{k}}$ , a iteração  $\bar{k}$  é do tipo  $h$ . Deste modo, o par  $(f^{\bar{k}}, h^{\bar{k}})$  entrou no filtro permanente e, portanto, para todo  $k > \bar{k}$  temos que

$$h^k \leq (1 - \alpha)h^{\bar{k}}. \quad (\text{A.10})$$

Pelo Lema A.9, temos que

$$f^{\bar{k}+1} \leq f^{\bar{k}} + C_4 h^{\bar{k}} = \Phi_1.$$

Considere  $k_0$  o maior índice  $k \in I$  tal que  $f^k \leq \Phi_1$ . Se  $k_0 = K$ , a prova está concluída.

Se não,

$$f^{k_0+1} > \Phi_1 \geq f^{k_0}. \quad (\text{A.11})$$

$\mathcal{I}$  intervalo ( $j = 1$ ). Por (A.11), a iteração  $k_0$  é do tipo  $h$  e, como para  $j = 0$ ,  $(f^{k_0}, h^{k_0})$  entrou no filtro permanente e, conseqüentemente, para todo  $k > k_0$

$$h^k \leq (1 - \alpha)^2 h^{\bar{k}}.$$

Por (A.11), Lema A.9 e (A.10), temos que

$$\Phi_1 < f^{k_0+1} \leq f^{k_0} + C_4 h^{k_0} \geq \Phi_1 + C_4 h^{k_0} \geq \Phi_1 + C_4 (1 - \alpha) h^{\bar{k}} = \Phi_2.$$

Seguindo o mesmo processo, tomamos  $k_1$  o maior índice neste intervalo tal que a iteração é do tipo  $h$ . Se  $k_1 = K$ , a prova está concluída. Se não,

$$f^{k_1+1} > \Phi_2 \geq f^{k_1},$$

e  $f^{k_1+1}$  estará no próximo intervalo, até que  $f^{k_j} = f^K$  seja obtido. Então

$$f^K \leq \Phi_0 + \frac{C_4}{\alpha} h^{\bar{k}},$$

e a prova está concluída. □

O próximo resultado garante que a seqüência  $(f^k)$  converge.

**Lema A.11** [78, Lema 3.5] *A seqüência  $(f(x^k))_{k \in \mathbb{N}}$  converge.*

*Demonstração.* Por H1 e H2, a seqüência  $(f^k)$  é limitada.

Nesta prova, vamos usar o fato de que, dada uma seqüência  $(f^k)$  tal que  $\limsup(f^k) > \liminf(f^k) + \chi$ , para  $\chi > 0$ , é possível extrair duas subsequências  $(f^k)_{\mathcal{K}}$  e  $(f^{k+j_k})_{\mathcal{K}}$  tal que para todo  $k \in \mathcal{K}$ ,

$$f^{k+j_k} \geq f^k + \chi \quad \text{e} \quad f^{k+r} \geq f^k, \quad \forall r \in \{1, 2, \dots, j_k\}.$$

Para tanto, é suficiente considerar uma subsequência convergindo para o  $\limsup(f^k)$  e associar a cada índice  $\ell$  o maior índice  $\ell - j_\ell$  tal que

$$f^{\ell-j_\ell} \leq f^\ell - \chi,$$

se existir. Como para  $\ell$  suficientemente grande esta definição é razoável, pois basta desconsiderar os primeiros índices da subsequência.

Suponha por contradição que  $\limsup(f^k) > \liminf(f^k) + \chi$ , para  $\chi > 0$ . Considere a subsequência  $(f^k)_{\mathcal{K}}$  como construída acima. Assim, concluímos que para todo  $k \in \mathcal{K}$  a iteração  $k$  é uma iteração- $h$  e pelo Lema A.10,

$$f^k + \chi \leq f^{k+j_k} \leq f^k + \frac{C_4}{\alpha} h^k. \tag{A.12}$$

Considerando uma subsequência se necessário, assuma que  $(x^k)_\mathcal{K}$  convirja para  $\bar{x}$ . Então  $\bar{x}$  é viável pelo Lema A.6. Isto contradiz (A.12) e a prova está completa.  $\square$

O próximo resultado garante que, perto de um ponto viável não estacionário, a redução na função objetivo é positiva, impedindo que haja um ponto de acumulação não estacionário.

**Lema A.12** [78, Lema 3.6] *Considere  $\mathcal{R}_k$  como definido em (1.5). Seja  $\bar{x} \in X$  um ponto viável não estacionário. Então existem uma vizinhança  $V_6$  de  $\bar{x}$  e uma constante  $\chi > 0$  tais que para todo  $x^k \in V_6$ , existe um índice  $\ell_k \in \mathbb{N}$  tais que*

$$f(x^k) - f(x^{k+\ell_k}) \geq \chi.$$

*Demonstração.* Pelo Lema 1.12, existem constantes  $C_8 > 0$  e  $C_9 > 0$  e uma vizinhança  $V_4$  de  $\bar{x}$  tais que para todo  $x^k \in V_4$ , (1.56) e (1.57) valem. Consequentemente  $k$  é uma iteração- $f$ . Considere  $\tau > 0$  tal que  $B(\bar{x}, \tau) \subset V_4$  e defina  $V_6 = B\left(\bar{x}, \frac{\tau}{2}\right)$ .

Considere  $x^k \in V_6$ . Enquanto  $x^{k+i} \in B(\bar{x}, \tau)$ , as iterações  $(k+i)$  são do tipo  $f$  e o filtro não muda. Consequentemente, por (1.56),  $f$  decresce por pelo menos um valor  $C_8 v_k > 0$ . Portanto, existe um  $\ell_k \in \mathbb{N}$  tal que  $x^{k+\ell_k} \notin B(\bar{x}, \tau)$  e  $x^{k+i} \in B(\bar{x}, \tau)$ , para  $i = 0, 1, \dots, \ell_k - 1$ . Como  $x^k \in V_6$ , temos que

$$\|x^k - x^{k+\ell_k}\| \geq \frac{\tau}{2}.$$

Usando (1.56) e (1.57) e a desigualdade triangular,

$$f(x^k) - f(x^{k+\ell_k}) \geq C_9 \sum_{i=0}^{\ell_k-1} \|x^{k+i} - x^{k+i+1}\| \geq C_9 \|x^k - x^{k+\ell_k}\| \geq \frac{\tau C_9}{2}.$$

Tomando  $\chi = \frac{\tau C_9}{2} > 0$ , completamos a prova.  $\square$

No resultado a seguir verificamos a convergência forte do algoritmo.

**Teorema A.13** [78, Teorema 3.7] *Considere  $\mathcal{R}_k$  como definido em (1.5). Então todo ponto de acumulação da sequência gerada pelo algoritmo é estacionário.*

*Demonstração.* Por contradição, suponha que exista um ponto  $\bar{x}$  não estacionário e um conjunto infinito  $\mathcal{K} \subset \mathbb{N}$  tal que  $x^k \xrightarrow{\mathcal{K}} \bar{x}$ . Pelo Lema A.6,  $\bar{x}$  é viável. Pelo Lema A.12, existe uma constante  $\chi > 0$  tal que, para todo  $k \in \mathcal{K}$  suficientemente grande, existe um índice  $\ell_k \in \mathbb{N}$  tal que

$$f(x^k) - f(x^{k+\ell_k}) \geq \chi.$$

Isto significa que a sequência  $(f(x^k))$  não é uma sequência de Cauchy, contradizendo o Lema A.11 e completando a prova. □

# Apêndice B

## Desempenho dos métodos de filtro para os problemas da coleção

### CUTE<sub>ST</sub>

Neste apêndice apresentamos, nas tabelas a seguir, os resultados numéricos obtidos pelos Algoritmos SD-FO, SD-FI, CD-FO e CD-FI, como discutidos na Seção 1.5. Os dados dos 196 problemas selecionados da coleção CUTE<sub>ST</sub> estão dispostos nas primeiras quatro colunas de acordo com a seguinte ordem: nome do problema na coleção, sua dimensão  $n$ , quantidade  $|\mathcal{E}|$  de restrições de igualdade e quantidade  $|\mathcal{I}|$  de restrições de desigualdade. As demais colunas apresentam os dados numéricos obtidos pelos algoritmos de acordo com: quantidade de iterações  $k$  realizadas, quantidade de avaliações da função objetivo, valor da função objetivo na solução obtida  $\bar{x}$ , valor da medida de inviabilidade em  $\bar{x}$  e o critério de saída que o algoritmo obteve. Os resultados obtidos dos quatro Algoritmos SD-FO, SD-FI, CD-FO e CD-FI são apresentados linha por linha para cada problema.

Nome	$n$	$ \mathcal{E} $	$ Z $	$k$	$\#f$	$f(\bar{x})$	$h(\bar{x})$	saída
AIRCRAFTA	8	5	0	2	47	0.0000e+00	2.3001E-07	1
				1	47	0.0000E+00	2.3001E-07	1
				3	4	0.0000E+00	2.2035E-07	1
				3	4	0.0000E+00	2.2035E-07	1
ALLINITC	4	1	0	6	130	3.0479E+01	3.7864E-07	1
				5	130	3.0479E+01	3.7864E-07	1
				9	14	3.0475E+01	5.9468E-07	1
				9	14	3.0475E+01	5.9468E-07	1
ALSOTAME	2	1	0	8	80	8.2085E-02	4.4409E-16	1
				7	80	8.2085E-02	4.4409E-16	1
				5	6	8.2085E-02	2.2204E-16	1
				5	6	8.2085E-02	2.2204E-16	1
AVGASA	8	0	10	3	93	-4.6319E+00	1.0478E-11	1
				2	93	-4.6319E+00	1.0478E-11	1
				13	24	-4.6319E+00	2.9412E-09	1
				13	24	-4.6319E+00	2.9412E-09	1
AVGASB	8	0	10	3	93	-4.4832E+00	5.6970E-14	1
				2	93	-4.4832E+00	5.6970E-14	1
				12	29	-4.4832E+00	6.5498E-09	1
				12	29	-4.4832E+00	6.5498E-09	1
BIGGSC4	4	0	7	6	52	-2.4500E+01	0.0000E+00	1
				5	52	-2.4500E+01	0.0000E+00	1
				1196	20280	-2.4500E+01	0.0000E+00	1
				1196	20280	-2.4500E+01	0.0000E+00	1
BT10	2	2	0	2	8	-1.0000E+00	4.4039E-09	1
				1	8	-1.0000E+00	4.4039E-09	1
				2	3	-1.0000E+00	4.4039E-09	1
				2	3	-1.0000E+00	4.4039E-09	1
BT11	5	3	0	15	278	8.2489E-01	6.9420E-07	1
				14	278	8.2489E-01	6.9420E-07	1
				10	20	8.2489E-01	2.7847E-09	1
				10	20	8.2489E-01	2.7847E-09	1
BT12	5	3	0	27	303	6.1881E+00	3.4108E-13	1
				26	303	6.1881E+00	3.4108E-13	1
				8	24	6.1881E+00	1.4894E-11	1
				8	24	6.1881E+00	1.4894E-11	1
BT2	3	1	0	14	73	3.2568E-02	3.2783E-08	1
				13	73	3.2568E-02	3.2783E-08	1
				10	19	3.2568E-02	3.8325E-09	1
				10	19	3.2568E-02	3.8325E-09	1

Nome	$n$	$ \mathcal{E} $	$ \mathcal{I} $	$k$	$\#f$	$f(\bar{x})$	$h(\bar{x})$	saída
BT3	5	3	0	6	48	4.0930E+00	4.4019E-15	1
				5	48	4.0930E+00	4.4019E-15	1
				9	10	4.0930E+00	1.0947E-08	1
				9	10	4.0930E+00	1.0947E-08	1
BT4	3	2	0	169	242	-4.0703E+01	8.4581E-07	-2
				125	211	-4.0765E+01	5.7480E-07	-2
				8	14	-4.5511E+01	8.5798E-13	1
				8	14	-4.5511E+01	8.5798E-13	1
BT5	3	2	0	29	168	9.6172E+02	3.2241E-07	1
				28	168	9.6172E+02	3.2241E-07	1
				6	10	9.6172E+02	1.4460E-11	1
				6	10	9.6172E+02	1.4460E-11	1
BT7	5	3	0	125	209	3.0650E+02	1.7146E-07	-4
				221	311	3.0650E+02	1.1017E-07	-4
				11	19	3.0650E+02	2.7711E-07	1
				11	19	3.0650E+02	2.7711E-07	1
BT8	5	2	0	176	426	1.5630E+00	9.0088E-07	-4
				126	359	1.0006E+00	5.7364E-07	-4
				3	5	1.0000E+00	3.3720E-07	1
				3	5	1.0000E+00	3.3720E-07	1
BT9	4	2	0	51	275	-1.0000E+00	8.6013E-07	1
				51	263	-1.0000E+00	1.9828E-07	1
				7	10	-1.0000E+00	8.3483E-09	1
				7	10	-1.0000E+00	8.3483E-09	1
CANTILVR	5	0	1	124	374	1.3408E+00	2.4134E-08	-4
				170	450	1.3401E+00	4.0187E-10	-4
				22	40	1.3400E+00	9.0666E-07	1
				22	40	1.3400E+00	9.0666E-07	1
CB2	3	0	3	40	185	1.9522E+00	3.8588E-08	1
				37	197	1.9522E+00	4.1106E-08	-4
				5	8	1.9522E+00	2.9364E-09	1
				5	8	1.9522E+00	2.9364E-09	1
CB3	3	0	3	2	12	2.0000E+00	7.1086E-13	1
				1	12	2.0000E+00	7.1086E-13	1
				2	39	2.0000E+00	1.4853E-10	1
				2	39	2.0000E+00	1.4853E-10	1
CHACONN1	3	0	3	26	153	1.9522E+00	9.6818E-07	-4
				25	153	1.9522E+00	9.6818E-07	-4
				6	9	1.9522E+00	2.7316E-07	1
				6	9	1.9522E+00	2.7316E-07	1

Nome	$n$	$ \mathcal{E} $	$ \mathcal{I} $	$k$	$\#f$	$f(\bar{x})$	$h(\bar{x})$	saída
CHACONN2	3	0	3	2	12	2.0000E+00	1.5821E-13	1
				1	12	2.0000E+00	1.5821E-13	1
				2	39	2.0000E+00	1.4843E-10	1
				2	39	2.0000E+00	1.4843E-10	1
CONGIGMZ	3	0	5	2	12	2.8000E+01	2.1202E-07	1
				1	12	2.8000E+01	2.1202E-07	1
				6	61	2.8000E+01	2.1250E-07	1
				6	61	2.8000E+01	2.1250E-07	1
COSHFUN	10	0	3	80	625	-6.6142E-01	1.5799E-10	1
				79	625	-6.6142E-01	1.5799E-10	1
				12	25	-6.6142E-01	1.1528E-08	1
				12	25	-6.6142E-01	1.1528E-08	1
CRESC4	6	0	8	35	142	2.7720E+00	4.6563E-08	-4
				5	97	7.2614E+00	6.3442E-09	-4
				42	145	8.7190E-01	7.3922E-07	1
				88	318	8.7190E-01	1.1713E-11	1
CSFI1	5	2	2	189	349	-1.3092E+01	2.2618E-07	-4
				337	633	-1.5137E+01	7.0549E-09	-4
				11	79	-4.9075E+01	1.0053E-09	1
				11	79	-4.9075E+01	1.0053E-09	1
CUBENE	2	2	0	2	8	0.0000E+00	4.4436E-14	1
				1	8	0.0000E+00	4.4436E-14	1
				2	39	0.0000E+00	5.8276E-10	1
				2	39	0.0000E+00	5.8276E-10	1
DEMYMALO	3	0	3	8	39	-3.0000E+00	9.6634E-13	1
				7	39	-3.0000E+00	9.6634E-13	1
				5	15	-3.0000E+00	2.1954E-09	1
				5	15	-3.0000E+00	2.1954E-09	1
DIPIGRI	7	0	4	21	91	6.9474E+02	2.1540E-04	-4
				20	91	6.9474E+02	2.1540E-04	-4
				23	59	6.8063E+02	3.0291E-07	1
				23	59	6.8063E+02	1.7616E-07	1
DUALC1	9	1	214	3	168	6.1553E+03	1.1935E-15	1
				2	168	6.1553E+03	1.1935E-15	1
				18	93	6.1552E+03	1.8372E-08	1
				18	93	6.1552E+03	1.8372E-08	1
DUALC2	7	1	228	3	147	3.5513E+03	2.7617E-15	1
				2	147	3.5513E+03	2.7617E-15	1
				18	82	3.5513E+03	5.6364E-09	1
				18	82	3.5513E+03	5.6364E-09	1

Nome	$n$	$ \mathcal{E} $	$ \mathcal{I} $	$k$	$\#f$	$f(\bar{x})$	$h(\bar{x})$	saída
DUALC5	8	1	277	3	93	4.2723E+02	0.0000E+00	1
				2	93	4.2723E+02	0.0000E+00	1
				27	130	4.2723E+02	2.2787E-14	1
				27	130	4.2723E+02	2.2787E-14	1
DUALC8	8	1	502	3	93	1.8309E+04	6.9389E-15	1
				2	93	1.8309E+04	6.9389E-15	1
				14	84	1.8309E+04	4.7241E-09	1
				14	84	1.8309E+04	4.7241E-09	1
EXPFITA	5	0	22	2	22	2.9986E+01	0.0000E+00	-4
				1	22	2.9986E+01	0.0000E+00	-4
				26	32	1.1366E-03	0.0000E+00	1
				26	32	1.1366E-03	0.0000E+00	1
EXPFITB	5	0	102	2	22	1.4990E+02	0.0000E+00	-4
				1	22	1.4990E+02	0.0000E+00	-4
				22	24	5.0194E-03	0.0000E+00	1
				22	24	5.0194E-03	0.0000E+00	1
EXPFITC	5	0	502	2	22	7.4952E+02	0.0000E+00	-4
				1	22	7.4952E+02	0.0000E+00	-4
				4786	271703	2.3303E-02	1.4442E-11	-1
				5001	283958	2.3303E-02	1.4442E-11	-1
EXTRASIM	2	1	0	3	21	1.0000E+00	1.1670E-10	1
				2	21	1.0000E+00	1.1670E-10	1
				3	4	1.0000E+00	8.4266E-13	1
				3	4	1.0000E+00	8.4266E-13	1
FCCU	19	8	0	50	479	1.1149E+01	5.7732E-15	1
				49	479	1.1149E+01	5.7732E-15	1
				29	65	1.1149E+01	4.2649E-09	1
				29	65	1.1149E+01	4.2649E-09	1
FLETCHER	4	1	3	18	171	1.9525E+01	7.9486E-07	1
				17	171	1.9525E+01	7.9486E-07	1
				7	12	1.9525E+01	1.2322E-09	1
				7	12	1.9525E+01	1.2322E-09	1
GENHS28	10	8	0	6	138	9.2717E-01	4.2597E-13	1
				5	138	9.2717E-01	4.2597E-13	1
				9	19	9.2717E-01	7.2899E-08	1
				9	19	9.2717E-01	7.2899E-08	1
GIGOMEZ1	3	0	3	12	43	-3.0000E+00	3.3946E-12	1
				11	43	-3.0000E+00	3.3946E-12	1
				5	11	-3.0000E+00	1.0580E-08	1
				5	11	-3.0000E+00	1.0580E-08	1

Nome	$n$	$ \mathcal{E} $	$ \mathcal{I} $	$k$	$\#f$	$f(\bar{x})$	$h(\bar{x})$	saída
GIGOMEZ2	3	0	3	57	194	1.9522E+00	1.0609E-09	-4
				57	180	1.9522E+00	7.3399E-07	1
				8	13	1.9522E+00	7.9515E-07	1
				8	13	1.9522E+00	7.9515E-07	1
GIGOMEZ3	3	0	3	3	34	2.0000E+00	5.6127E-07	1
				2	34	2.0000E+00	5.6127E-07	1
				3	6	2.0000E+00	5.6127E-07	1
				3	6	2.0000E+00	5.6127E-07	1
GOTTFR	2	2	0	2	8	0.0000E+00	4.5122E-07	1
				1	8	0.0000E+00	4.5122E-07	1
				2	3	0.0000E+00	4.5122E-07	1
				2	3	0.0000E+00	4.5122E-07	1
HAIFAS	13	0	9	189	603	-9.3536E-02	8.5375E-10	-4
				4	221	-3.1491E-02	2.3595E-03	-4
				9	16	-4.5000E-01	1.2553E-07	1
				2386	84675	-4.5000E-01	5.9194E-07	1
HALDMADS	6	0	42	4	258	1.2532E-04	2.2372E-12	1
				3	258	1.2532E-04	2.2372E-12	1
				10	100	1.2237E-04	1.6605E-09	1
				10	100	1.2237E-04	1.6605E-09	1
HATFLDH	4	0	7	21	168	-2.4376E+01	6.9053E-05	-4
				20	168	-2.4376E+01	6.9053E-05	-4
				11	21	-2.4500E+01	8.0002E-10	1
				11	21	-2.4500E+01	8.0002E-10	1
HEART6	6	6	0	2	30	0.0000E+00	3.7031E-08	1
				1	30	0.0000E+00	3.7031E-08	1
				2	3	0.0000E+00	3.7031E-08	1
				2	3	0.0000E+00	3.7031E-08	1
HEART8	8	8	0	2	47	0.0000E+00	4.2555E-11	1
				1	47	0.0000E+00	4.2555E-11	1
				2	3	0.0000E+00	4.2555E-11	1
				2	3	0.0000E+00	4.2555E-11	1
HIMMELBA	2	2	0	2	8	0.0000E+00	0.0000E+00	1
				1	8	0.0000E+00	0.0000E+00	1
				2	39	0.0000E+00	1.2000E-10	1
				2	39	0.0000E+00	1.2000E-10	1
HIMMELBC	2	2	0	2	8	0.0000E+00	9.3286E-08	1
				1	8	0.0000E+00	9.3286E-08	1
				2	5	0.0000E+00	9.3286E-08	1
				2	5	0.0000E+00	9.3286E-08	1

Nome	$n$	$ \mathcal{E} $	$ \mathcal{I} $	$k$	$\#f$	$f(\bar{x})$	$h(\bar{x})$	saída
HIMMELBE	3	3	0	2	12	0.0000E+00	0.0000E+00	1
				1	12	0.0000E+00	0.0000E+00	1
				2	39	0.0000E+00	5.0409E-11	1
				2	39	0.0000E+00	5.0409E-11	1
HIMMELP2	2	0	1	80	135	8.4848E+01	0.0000E+00	-4
				116	171	8.4848E+01	0.0000E+00	-4
				17	24	-6.2054E+01	0.0000E+00	1
				17	24	-6.2054E+01	0.0000E+00	1
HIMMELP3	2	0	2	10	21	-1.0288E+01	0.0000E+00	-2
				9	21	-1.0288E+01	0.0000E+00	-2
				8	17	-5.9013E+01	0.0000E+00	1
				8	17	-5.9013E+01	0.0000E+00	1
HIMMELP4	2	0	3	10	21	-1.0288E+01	0.0000E+00	-2
				9	21	-1.0288E+01	0.0000E+00	-2
				8	11	-5.9013E+01	0.0000E+00	1
				8	11	-5.9013E+01	0.0000E+00	1
HIMMELP5	2	0	3	13	19	-1.2530E+01	0.0000E+00	-2
				12	19	-1.2530E+01	0.0000E+00	-2
				14	24	-5.9013E+01	0.0000E+00	1
				14	24	-5.9013E+01	0.0000E+00	1
HIMMELP6	2	0	5	13	19	-1.2530E+01	0.0000E+00	-2
				12	19	-1.2530E+01	0.0000E+00	-2
				13	70	-5.9013E+01	0.0000E+00	1
				13	70	-5.9013E+01	0.0000E+00	1
HONG	4	1	0	18	75	2.2571E+01	4.9127E-14	1
				17	75	2.2571E+01	4.9127E-14	1
				15	39	2.2571E+01	0.0000E+00	1
				15	39	2.2571E+01	0.0000E+00	1
HS10	2	0	1	26	143	-1.0000E+00	1.3030E-08	-4
				24	154	-1.0000E+00	0.0000E+00	1
				7	13	-1.0000E+00	1.0547E-11	1
				7	13	-1.0000E+00	1.0547E-11	1
HS100	7	0	4	21	91	6.9474E+02	2.1540E-04	-4
				20	91	6.9474E+02	2.1540E-04	-4
				23	55	6.8063E+02	3.0154E-07	1
				23	59	6.8063E+02	1.7616E-07	1
HS104	8	0	5	2	47	3.6574E+00	4.3502E-01	-4
				1	47	3.6574E+00	4.3502E-01	-4
				16	43	3.9512E+00	2.1570E-07	1
				16	43	3.9512E+00	2.1570E-07	1

Nome	$n$	$ \mathcal{E} $	$ \mathcal{I} $	$k$	$\#f$	$f(\bar{x})$	$h(\bar{x})$	saída
HS105	8	0	1	2	47	1.2538E+03	5.0000E+00	-4
				1	47	1.2538E+03	5.0000E+00	-4
				43	91	1.0446E+03	4.5543E-13	1
				43	91	1.0446E+03	4.5543E-13	1
HS108	9	0	13	58	309	-7.6024E-01	5.1390E-08	-4
				57	309	-7.6024E-01	5.1390E-08	-4
				21	1118	-7.3903E-01	4.1745E-07	-3
				21	1118	-7.3903E-01	4.1745E-07	-3
HS11	2	0	1	14	106	-8.4985E+00	2.2302E-07	1
				13	106	-8.4985E+00	2.2302E-07	1
				6	10	-8.4985E+00	4.2285E-08	1
				6	10	-8.4985E+00	4.2285E-08	1
HS111	10	3	0	217	425	-4.4875E+01	9.9401E-07	-4
				283	520	-4.4875E+01	3.7638E-07	-4
				51	95	-4.7761E+01	3.2658E-08	1
				51	95	-4.7761E+01	3.2658E-08	1
HS112	10	3	0	2	68	-2.0960E+01	1.4491E+00	-4
				1	68	-2.0960E+01	1.4491E+00	-4
				35	156	-4.7761E+01	1.4831E-09	1
				35	156	-4.7761E+01	1.4831E-09	1
HS113	10	0	8	18	1028	2.4306E+01	7.3362E-11	1
				17	1028	2.4306E+01	7.3362E-11	1
				16	38	2.4306E+01	2.7740E-07	1
				15	34	2.4306E+01	2.4931E-08	1
HS114	10	3	8	203	590	-1.5527E+03	6.9254E-07	-4
				280	687	-1.5527E+03	7.9540E-07	-4
				30	58	-1.7688E+03	3.1114E-07	1
				30	58	-1.7688E+03	3.1114E-07	1
HS117	15	0	5	54	238	2.4040E+03	0.0000E+00	-4
				53	238	2.4040E+03	0.0000E+00	-4
				25	43	3.2349E+01	1.3376E-14	1
				25	43	3.2349E+01	1.3376E-14	1
HS118	15	0	17	5	160	8.6630E+02	0.0000E+00	-2
				4	160	8.6630E+02	0.0000E+00	-2
				21	21	6.6482E+02	0.0000E+00	1
				21	21	6.6482E+02	0.0000E+00	1
HS119	16	8	0	4	159	1.4886E+03	1.7550E-15	-2
				3	159	1.4886E+03	1.7550E-15	-2
				15	36	2.4490E+02	3.9513E-09	1
				15	36	2.4490E+02	3.9513E-09	1

Nome	$n$	$ \mathcal{E} $	$ \mathcal{I} $	$k$	$\#f$	$f(\bar{x})$	$h(\bar{x})$	saída
HS20	2	0	3	2	58	9.0900E+02	1.8028E+00	-4
				1	58	9.0900E+02	1.8028E+00	-4
				5	6	4.0199E+01	0.0000E+00	1
				5	6	4.0199E+01	0.0000E+00	1
HS21	2	0	1	3	51	-9.9960E+01	0.0000E+00	1
				2	51	-9.9960E+01	0.0000E+00	1
				4	13	-9.9960E+01	0.0000E+00	1
				4	13	-9.9960E+01	0.0000E+00	1
HS22	2	0	2	3	52	1.0000E+00	8.3991E-11	1
				2	52	1.0000E+00	8.3991E-11	1
				4	6	1.0000E+00	8.6324E-11	1
				4	6	1.0000E+00	8.6324E-11	1
HS23	2	0	5	7	91	2.0000E+00	0.0000E+00	1
				6	91	2.0000E+00	0.0000E+00	1
				7	8	2.0000E+00	0.0000E+00	1
				7	8	2.0000E+00	0.0000E+00	1
HS24	2	0	3	148	213	-6.9796E-01	3.3475E-07	-4
				273	367	-8.0890E-01	6.4477E-07	-4
				6	8	-1.0000E+00	5.5511E-14	1
				6	8	-1.0000E+00	5.5511E-14	1
HS26	3	1	0	7	61	1.4315E-09	2.8598E-08	1
				6	61	1.4315E-09	2.8598E-08	1
				21	41	2.7528E-10	3.8970E-06	1
				20	45	4.9202E-10	5.2241E-06	1
HS27	3	1	0	120	258	4.0006E-02	1.5578E-07	-4
				104	191	4.0000E-02	4.5667E-07	-2
				9	14	4.0000E-02	2.3626E-07	1
				9	14	4.0000E-02	2.3626E-07	1
HS28	3	1	0	7	36	5.1428E-21	4.4409E-16	1
				6	36	5.1428E-21	4.4409E-16	1
				10	10	1.1981E-18	6.1888E-09	1
				10	10	1.1981E-18	6.1888E-09	1
HS29	3	0	1	14	23	-1.0648E+01	0.0000E+00	-2
				13	23	-1.0648E+01	0.0000E+00	-2
				10	23	-2.2627E+01	5.4611E-08	1
				10	23	-2.2627E+01	5.4611E-08	1
HS30	3	0	1	20	156	1.0000E+00	0.0000E+00	1
				19	156	1.0000E+00	0.0000E+00	1
				21	46	1.0000E+00	0.0000E+00	1
				21	46	1.0000E+00	0.0000E+00	1

Nome	$n$	$ \mathcal{E} $	$ \mathcal{I} $	$k$	$\#f$	$f(\bar{x})$	$h(\bar{x})$	saída
HS31	3	0	1	30	163	6.0000E+00	1.6606E-08	-4
				29	163	6.0000E+00	1.6606E-08	-4
				8	20	6.0000E+00	4.2318E-09	1
				8	20	6.0000E+00	4.2318E-09	1
HS32	3	1	1	3	22	1.0000E+00	0.0000E+00	1
				2	22	1.0000E+00	0.0000E+00	1
				5001	179886	1.0000E+00	4.6024E-07	-1
				5001	179886	1.0000E+00	4.6024E-07	-1
HS33	3	0	2	2	30	-3.0000E+00	0.0000E+00	-4
				1	30	-3.0000E+00	0.0000E+00	-4
				13	15	-4.5858E+00	5.2750E-09	1
				13	20	-4.5858E+00	5.4510E-10	1
HS34	3	0	2	5000	5136	-6.6517E-01	2.5798E-07	-1
				5000	5169	-7.0825E-01	0.0000E+00	-1
				6	19	-8.3403E-01	5.5650E-07	1
				6	19	-8.3403E-01	5.5650E-07	1
HS35	3	0	1	3	12	1.1111E-01	0.0000E+00	1
				2	12	1.1111E-01	0.0000E+00	1
				8	9	1.1111E-01	0.0000E+00	1
				8	9	1.1111E-01	0.0000E+00	1
HS36	3	0	1	18	27	-1.3701E+03	0.0000E+00	-4
				17	27	-1.3701E+03	0.0000E+00	-4
				5	5	-3.3000E+03	0.0000E+00	1
				5	5	-3.3000E+03	0.0000E+00	1
HS37	3	0	2	14	23	-1.4049E+03	0.0000E+00	-4
				13	23	-1.4049E+03	0.0000E+00	-4
				5	35	-3.4560E+03	1.7077E-08	1
				5	35	-3.4560E+03	1.7077E-08	1
HS39	4	2	0	51	275	-1.0000E+00	8.6013E-07	1
				51	263	-1.0000E+00	1.9828E-07	1
				7	10	-1.0000E+00	8.3483E-09	1
				7	10	-1.0000E+00	8.3483E-09	1
HS40	4	3	0	13	175	-2.5000E-01	5.6446E-07	1
				12	175	-2.5000E-01	5.6446E-07	1
				6	10	-2.5000E-01	2.6440E-07	1
				6	10	-2.5000E-01	2.6440E-07	1
HS41	4	1	0	5	201	1.9259E+00	6.6613E-16	1
				4	201	1.9259E+00	6.6613E-16	1
				6	12	1.9259E+00	4.4409E-16	1
				6	12	1.9259E+00	4.4409E-16	1

Nome	$n$	$ \mathcal{E} $	$ \mathcal{I} $	$k$	$\#f$	$f(\bar{x})$	$h(\bar{x})$	saída
HS42	4	2	0	39	142	1.3856E+01	7.7358E-04	-4
				32	144	1.3858E+01	5.9674E-06	-4
				9	17	1.3858E+01	6.8457E-09	1
				9	17	1.3858E+01	6.8457E-09	1
HS43	4	0	3	124	234	-4.4000E+01	8.9853E-07	-4
				97	195	-4.4000E+01	6.9615E-07	-4
				12	24	-4.4000E+01	1.9178E-07	1
				12	24	-4.4000E+01	1.9178E-07	1
HS44	4	0	6	105	179	-3.5158E+00	8.2546E-07	-4
				146	255	-3.5142E+00	1.0083E-06	-4
				7	7	-1.5000E+01	3.7218E-09	1
				7	7	-1.5000E+01	3.7218E-09	1
HS46	5	2	0	2	41	3.3376E+00	5.2764E-10	-4
				1	41	3.3376E+00	5.2764E-10	-4
				25	75	7.0355E-11	9.9386E-07	1
				227	1525	1.8473E-08	1.0832E-07	1
HS47	5	3	0	100	194	5.1018E-01	2.7950E-08	-4
				90	164	2.3776E+00	5.0739E-07	-4
				22	44	2.9947E-11	3.8947E-08	1
				22	44	2.9947E-11	3.8947E-08	1
HS48	5	2	0	7	279	4.2212E-18	1.2799E-08	1
				6	279	4.2212E-18	1.2799E-08	1
				10	13	2.8671E-18	1.1356E-08	1
				10	13	2.8671E-18	1.1356E-08	1
HS49	5	2	0	108	209	5.2975E+01	8.8184E-07	-2
				115	231	5.2975E+01	3.1278E-07	-4
				24	24	1.4044E-09	2.0659E-08	1
				24	24	1.4044E-09	2.0659E-08	1
HS50	5	3	0	129	200	2.6540E+03	2.5705E-07	1
				76	139	2.8918E+03	2.7688E-07	1
				14	14	3.1627E-23	5.8516E-11	1
				14	14	3.1627E-23	5.8516E-11	1
HS51	5	3	0	5	298	2.4128E-17	8.8749E-09	1
				4	298	2.4128E-17	8.8749E-09	1
				9	12	1.7927E-18	2.6470E-09	1
				9	12	1.7927E-18	2.6470E-09	1
HS52	5	3	0	3	24	5.3266E+00	2.2034E-15	1
				2	24	5.3266E+00	2.2034E-15	1
				10	14	5.3266E+00	2.2961E-15	1
				10	14	5.3266E+00	2.2961E-15	1

Nome	$n$	$ \mathcal{E} $	$ \mathcal{I} $	$k$	$\#f$	$f(\bar{x})$	$h(\bar{x})$	saída
HS53	5	3	0	3	24	4.0930E+00	4.1592E-12	1
				2	24	4.0930E+00	4.1592E-12	1
				9	12	4.0930E+00	1.5052E-08	1
				9	12	4.0930E+00	1.5052E-08	1
HS54	6	1	0	2	43	-7.6510E-01	5.6000E+03	-4
				1	43	-7.6510E-01	5.6000E+03	-4
				32	95	-8.6741E-01	9.0949E-13	1
				32	95	-8.6741E-01	9.0949E-13	1
HS55	6	6	0	2	30	6.6667E+00	1.2467E-12	1
				1	30	6.6667E+00	1.2467E-12	1
				2	3	6.6667E+00	1.2467E-12	1
				2	3	6.6667E+00	1.2467E-12	1
HS56	7	4	0	5000	5183	-3.4093E+00	4.4057E-08	-1
				5000	5194	-3.4107E+00	6.2868E-07	-1
				10	19	-3.4560E+00	1.3943E-07	1
				10	19	-3.4560E+00	1.3943E-07	1
HS57	2	0	1	3	38	3.0647E-02	0.0000E+00	1
				2	38	3.0647E-02	0.0000E+00	1
				17	30	2.8460E-02	0.0000E+00	1
				17	30	2.8460E-02	0.0000E+00	1
HS59	2	0	3	20	29	1.1809E+01	0.0000E+00	-2
				19	29	1.1809E+01	0.0000E+00	-2
				18	31	-6.7495E+00	0.0000E+00	1
				18	31	-6.7495E+00	0.0000E+00	1
HS6	2	1	0	10	80	9.3436E-14	9.1704E-13	1
				9	80	9.3436E-14	9.1704E-13	1
				8	16	2.1124E-17	6.2029E-09	1
				8	17	2.5029E-17	1.1785E-08	1
HS60	3	1	0	6	110	3.2568E-02	3.3951E-13	1
				5	110	3.2568E-02	3.3951E-13	1
				10	26	3.2568E-02	6.8020E-07	1
				10	26	3.2568E-02	6.8020E-07	1
HS61	3	2	0	12	159	-1.4365E+02	5.0681E-07	1
				11	159	-1.4365E+02	5.0681E-07	1
				7	53	-1.4365E+02	2.1826E-09	1
				7	53	-1.4365E+02	2.1826E-09	1
HS62	3	1	0	2	11	-2.5698E+04	2.7756E-17	-4
				1	11	-2.5698E+04	2.7756E-17	-4
				5001	299302	-2.6273E+04	6.2450E-17	-1
				5001	299302	-2.6273E+04	6.2450E-17	-1

Nome	$n$	$ \mathcal{E} $	$ \mathcal{I} $	$k$	$\#f$	$f(\bar{x})$	$h(\bar{x})$	saída
HS63	3	2	0	55	202	9.6172E+02	8.9935E-07	-4
				49	182	9.6172E+02	6.4047E-15	1
				6	15	9.6172E+02	5.0298E-10	1
				6	15	9.6172E+02	5.0298E-10	1
HS65	3	0	1	14	138	9.5353E-01	2.6357E-07	1
				13	138	9.5353E-01	2.6357E-07	1
				11	26	9.5353E-01	3.1251E-09	1
				11	26	9.5353E-01	3.1251E-09	1
HS66	3	0	2	85	230	5.1822E-01	7.9615E-09	-4
				88	255	5.1816E-01	2.3322E-08	-4
				7	11	5.1816E-01	2.7062E-07	1
				7	11	5.1816E-01	2.7062E-07	1
HS67	3	0	14	74	130	-8.7241E+02	0.0000E+00	-4
				63	123	-8.7241E+02	4.2278E-09	-4
				23	36	-1.1621E+03	2.2737E-13	1
				23	36	-1.1621E+03	2.2737E-13	1
HS68	4	2	0	2	17	-2.6184E-01	6.8269E-01	-4
				1	17	-2.6184E-01	6.8269E-01	-4
				17	46	-9.2043E-01	5.5044E-07	1
				17	46	-9.2043E-01	5.5044E-07	1
HS7	2	1	0	17	108	-1.7321E+00	8.3489E-14	1
				16	108	-1.7321E+00	8.3489E-14	1
				7	12	-1.7321E+00	7.5146E-10	1
				7	12	-1.7321E+00	7.5146E-10	1
HS70	4	0	1	2	30	9.8186E-01	0.0000E+00	-4
				1	30	9.8186E-01	0.0000E+00	-4
				38	64	7.4985E-03	0.0000E+00	1
				38	64	7.4985E-03	0.0000E+00	1
HS71	4	1	1	2	42	1.7994E+01	4.1150E-10	1
				1	42	1.7994E+01	4.1150E-10	1
				7	16	1.7014E+01	1.7805E-07	1
				7	16	1.7014E+01	1.7805E-07	1
HS73	4	1	2	3	64	2.9894E+01	6.9364E-08	1
				2	64	2.9894E+01	6.9364E-08	1
				4	7	2.9894E+01	9.0386E-10	1
				4	7	2.9894E+01	9.0386E-10	1
HS74	4	3	2	87	176	5.2343E+03	8.2381E-07	-2
				155	265	5.2272E+03	6.6272E-07	-2
				11	20	5.1265E+03	6.4941E-08	1
				11	20	5.1265E+03	6.4941E-08	1

Nome	$n$	$ \mathcal{E} $	$ \mathcal{I} $	$k$	$\#f$	$f(\bar{x})$	$h(\bar{x})$	saída
HS76	4	0	3	4	33	-4.6818E+00	0.0000E+00	1
				3	33	-4.6818E+00	0.0000E+00	1
				8	9	-4.6818E+00	0.0000E+00	1
				8	9	-4.6818E+00	0.0000E+00	1
HS77	5	2	0	2	42	4.0000E+00	5.6822E+01	-4
				1	42	4.0000E+00	5.6822E+01	-4
				14	40	2.4151E-01	1.1320E-07	1
				14	40	2.4151E-01	1.1320E-07	1
HS78	5	3	0	16	321	-2.9197E+00	4.9262E-07	1
				15	321	-2.9197E+00	4.9262E-07	1
				6	13	-2.9197E+00	4.6392E-07	1
				6	13	-2.9197E+00	4.6392E-07	1
HS79	5	3	0	47	112	8.1252E-02	8.0422E-07	-4
				46	112	8.1252E-02	8.0422E-07	-4
				9	22	7.8777E-02	5.1863E-08	1
				9	22	7.8777E-02	5.1863E-08	1
HS8	2	2	0	2	8	-1.0000E+00	7.8872E-11	1
				1	8	-1.0000E+00	7.8872E-11	1
				2	4	-1.0000E+00	7.8872E-11	1
				2	4	-1.0000E+00	7.8872E-11	1
HS81	5	3	0	5	33	5.3317E-02	1.1766E-01	-2
				4	33	5.3317E-02	1.1766E-01	-2
				6	14	5.3950E-02	4.4046E-09	1
				6	14	5.3950E-02	4.4046E-09	1
HS84	5	0	3	19	42	-2.7656E+06	0.0000E+00	-2
				18	42	-2.7656E+06	0.0000E+00	-2
				7	8	-5.2803E+06	3.4925E-10	1
				7	8	-5.2803E+06	3.4925E-10	1
HS85	5	0	21	94	248	-1.2985E+00	0.0000E+00	-4
				143	303	-1.2986E+00	0.0000E+00	-4
				34	48	-2.2156E+00	0.0000E+00	1
				35	75	-2.2156E+00	0.0000E+00	1
HS86	5	0	10	9	199	-3.2349E+01	2.5462E-08	1
				8	199	-3.2349E+01	2.5462E-08	1
				7	14	-3.2349E+01	2.2204E-16	1
				7	14	-3.2349E+01	2.2204E-16	1
HS9	2	1	0	170	210	-3.9417E-01	8.7386E-11	-2
				301	342	-4.2886E-01	7.0019E-11	-2
				8	10	-5.0000E-01	1.0414E-09	1
				8	10	-5.0000E-01	1.0414E-09	1

Nome	$n$	$ \mathcal{E} $	$ \mathcal{I} $	$k$	$\#f$	$f(\bar{x})$	$h(\bar{x})$	saída
HS93	6	0	2	3	56	1.3707E+02	0.0000E+00	-2
				2	56	1.3707E+02	0.0000E+00	-2
				14	59	1.3508E+02	4.2807E-07	1
				14	59	1.3508E+02	4.2807E-07	1
HS95	6	0	4	3	311	1.5669E-02	0.0000E+00	1
				2	311	1.5669E-02	0.0000E+00	1
				4	5	1.5620E-02	2.3927E-11	1
				4	5	1.5620E-02	2.3927E-11	1
HS96	6	0	4	3	311	1.5669E-02	1.1177E-14	1
				2	311	1.5669E-02	1.1177E-14	1
				4	15	1.5620E-02	6.6721E-10	1
				4	15	1.5620E-02	6.6721E-10	1
HS97	6	0	4	5	259	3.1360E+00	0.0000E+00	1
				4	259	3.1360E+00	0.0000E+00	1
				6	9	3.1358E+00	5.9330E-11	1
				6	9	3.1358E+00	5.9330E-11	1
HS98	6	0	4	5	315	3.1358E+00	0.0000E+00	1
				4	315	3.1358E+00	0.0000E+00	1
				6	9	3.1358E+00	1.4589E-10	1
				6	9	3.1358E+00	1.4589E-10	1
HUBFIT	2	0	1	3	14	1.6893E-02	0.0000E+00	1
				2	14	1.6893E-02	0.0000E+00	1
				8	9	1.6893E-02	0.0000E+00	1
				8	9	1.6893E-02	0.0000E+00	1
HYPCIR	2	2	0	2	8	0.0000E+00	2.8677E-07	1
				1	8	0.0000E+00	2.8677E-07	1
				2	3	0.0000E+00	2.8677E-07	1
				2	3	0.0000E+00	2.8677E-07	1
KIWCRES	3	0	2	14	145	-2.8691E-07	4.6812E-07	1
				13	145	-2.8691E-07	4.6812E-07	1
				7	14	-5.8303E-11	3.5545E-10	1
				7	14	-5.8303E-11	3.5545E-10	1
LOTSCHD	12	7	0	18	200	2.3984E+03	1.7036E-08	1
				17	200	2.3984E+03	1.7036E-08	1
				7	8	2.3984E+03	7.9667E-11	1
				7	8	2.3984E+03	7.9667E-11	1
LSQFIT	5	4	0	16	49	5.1177E+02	3.8215E-01	-2
				15	49	5.1177E+02	3.8215E-01	-2
				6	7	1.2311E+02	5.4654E-13	1
				6	7	1.2311E+02	5.4654E-13	1

Nome	$n$	$ \mathcal{E} $	$ \mathcal{I} $	$k$	$\#f$	$f(\bar{x})$	$h(\bar{x})$	saída
LSNNODOC	2	0	1	3	14	3.3787E-02	0.0000E+00	1
				2	14	3.3787E-02	0.0000E+00	1
				7	8	3.3787E-02	0.0000E+00	1
				7	8	3.3787E-02	0.0000E+00	1
MADSEN	3	0	6	24	193	6.1643E-01	2.9787E-07	1
				23	193	6.1643E-01	2.9787E-07	1
				9	15	6.1643E-01	4.6817E-07	1
				9	15	6.1643E-01	4.6817E-07	1
MAKELA1	3	0	2	85	252	-1.4082E+00	1.0260E-09	-2
				107	276	-1.4082E+00	0.0000E+00	-2
				5	8	-1.4142E+00	1.4426E-12	1
				5	8	-1.4142E+00	1.4426E-12	1
MAKELA2	3	0	3	11	109	7.2000E+00	8.4279E-07	1
				10	109	7.2000E+00	8.4279E-07	1
				6	10	7.2000E+00	4.3036E-08	1
				6	10	7.2000E+00	4.3036E-08	1
MARATOS	2	1	0	26	116	-1.0000E+00	4.4761E-08	1
				25	122	-1.0000E+00	4.6544E-08	1
				5	7	-1.0000E+00	1.3830E-07	1
				5	7	-1.0000E+00	1.3830E-07	1
MATRIX2	6	0	2	17	410	9.9678E-08	0.0000E+00	1
				16	410	9.9678E-08	0.0000E+00	1
				5001	178813	1.1000E-09	8.9372E-09	-1
				5001	178813	1.1000E-09	8.9372E-09	-1
MIFFLIN1	3	0	2	13	123	-1.0000E+00	9.1025E-07	1
				12	123	-1.0000E+00	9.1025E-07	1
				7	16	-1.0000E+00	3.4326E-09	1
				7	16	-1.0000E+00	3.4326E-09	1
MIFFLIN2	3	0	2	22	163	-1.0000E+00	1.8117E-10	-4
				16	151	-1.0000E+00	2.6248E-10	-4
				7	14	-1.0000E+00	2.2269E-10	1
				7	14	-1.0000E+00	2.2269E-10	1
MINMAXBD	5	0	20	158	203	7.6556E+02	0.0000E+00	-4
				231	276	7.5560E+02	0.0000E+00	-4
				16	28	1.1571E+02	2.9428E-07	1
				16	28	1.1571E+02	2.9428E-07	1
MINMAXRB	3	0	4	11	167	-9.3070E-11	4.7527E-08	1
				10	167	-9.3070E-11	4.7527E-08	1
				7	17	-1.4821E-12	1.2847E-10	1
				9	18	5.9481E-11	4.3933E-08	1

Nome	$n$	$ \mathcal{E} $	$ \mathcal{I} $	$k$	$\#f$	$f(\bar{x})$	$h(\bar{x})$	saída
MISTAKE	9	0	13	25	777	-1.0000E+00	8.2623E-07	-4
				24	777	-1.0000E+00	8.2623E-07	-4
				14	192	-1.0000E+00	1.9649E-07	1
				14	192	-1.0000E+00	1.9649E-07	1
MWRIGHT	5	3	0	77	182	2.4979E+01	3.3412E-07	1
				98	234	2.4979E+01	5.1289E-08	1
				10	20	2.4979E+01	3.3601E-07	1
				10	20	2.4979E+01	3.3601E-07	1
OET1	10	6	0	37	333	-2.3246E+03	1.6922E-09	-4
				36	333	-2.3246E+03	1.6922E-09	-4
				39	40	-2.3800E+03	2.0196E-09	1
				39	40	-2.3800E+03	2.0196E-09	1
OET2	3	0	1002	14	158	5.3824E-01	2.7213E-09	-4
				13	158	5.3824E-01	2.7213E-09	-4
				8	16	5.3824E-01	5.5511E-15	1
				8	16	5.3824E-01	5.5511E-15	1
OET3	3	0	1002	10	51	8.7160E-02	1.4052E-08	1
				9	51	8.7160E-02	1.4052E-08	1
				8	22	8.7160E-02	1.0397E-08	1
				8	22	8.7160E-02	1.0397E-08	1
ODFITS	4	0	1002	3	93	4.5051E-03	5.0538E-12	1
				2	93	4.5051E-03	5.0538E-12	1
				7	18	4.5051E-03	0.0000E+00	1
				7	18	4.5051E-03	0.0000E+00	1
PENTAGON	6	0	15	80	299	1.3442E-03	0.0000E+00	-4
				5000	5219	1.3442E-03	0.0000E+00	-1
				41	63	1.3653E-04	0.0000E+00	1
				41	63	1.3653E-04	0.0000E+00	1
PFIT1	3	3	0	2	12	0.0000E+00	3.3145E-07	1
				1	12	0.0000E+00	3.3145E-07	1
				2	3	0.0000E+00	3.3145E-07	1
				2	3	0.0000E+00	3.3145E-07	1
PFIT2	3	3	0	2	12	0.0000E+00	4.7807E-09	1
				1	12	0.0000E+00	4.7807E-09	1
				2	3	0.0000E+00	4.7807E-09	1
				2	3	0.0000E+00	4.7807E-09	1
PFIT3	3	3	0	2	12	0.0000E+00	1.0370E-07	1
				1	12	0.0000E+00	1.0370E-07	1
				2	3	0.0000E+00	1.0370E-07	1
				2	3	0.0000E+00	1.0370E-07	1

Nome	$n$	$ \mathcal{E} $	$ \mathcal{I} $	$k$	$\#f$	$f(\bar{x})$	$h(\bar{x})$	saída
PFIT4	3	3	0	2	12	0.0000E+00	2.8414E-09	1
				1	12	0.0000E+00	2.8414E-09	1
				2	3	0.0000E+00	2.8414E-09	1
				2	3	0.0000E+00	2.8414E-09	1
POLAK1	3	0	2	59	157	7.3432E+00	6.4512E-08	-4
				248	359	7.3155E+00	6.5181E-07	-4
				12	26	2.7183E+00	1.2433E-08	1
				12	26	2.7183E+00	1.2433E-08	1
POLAK3	12	0	10	43	814	5.9330E+00	7.5405E-07	-4
				42	814	5.9330E+00	7.5405E-07	-4
				16	28	5.9330E+00	8.9360E-07	1
				16	28	5.9330E+00	8.9360E-07	1
POLAK4	3	0	3	19	162	-9.1850E-09	6.1598E-08	1
				18	162	-9.1850E-09	6.1598E-08	1
				16	28	-2.8179E-08	4.4474E-08	1
				16	28	-2.8179E-08	4.4474E-08	1
POLAK5	3	0	2	2	12	5.0000E+01	0.0000E+00	1
				1	12	5.0000E+01	0.0000E+00	1
				11	46	5.0000E+01	5.7906E-06	1
				5001	100196	5.0000E+01	9.5319E-07	-1
POLAK6	5	0	4	136	201	-3.0468E+01	0.0000E+00	-4
				164	233	-3.5401E+01	0.0000E+00	-4
				12	35	-4.4000E+01	2.1802E-07	1
				12	35	-4.4000E+01	2.1802E-07	1
POWELL20	5	0	4	136	201	-3.0468E+01	0.0000E+00	-4
				164	233	-3.5401E+01	0.0000E+00	-4
				12	35	-4.4000E+01	2.1802E-07	1
				12	35	-4.4000E+01	2.1802E-07	1
PT	2	0	501	3	33	1.7839E-01	0.0000E+00	1
				2	33	1.7839E-01	0.0000E+00	1
				7	13	1.7839E-01	0.0000E+00	1
				7	13	1.7839E-01	0.0000E+00	1
QC	9	0	4	2	169	-8.0985E+02	0.0000E+00	-4
				1	169	-8.0985E+02	0.0000E+00	-4
				3	3	-9.5654E+02	0.0000E+00	1
				3	3	-9.5654E+02	0.0000E+00	1
RES	20	12	2	2	2352	0.0000E+00	1.3106E-14	-4
				1	2352	0.0000E+00	1.3106E-14	-4
				6	6	0.0000E+00	1.3106E-14	1
				6	6	0.0000E+00	1.3106E-14	1

Nome	$n$	$ \mathcal{E} $	$ \mathcal{I} $	$k$	$\#f$	$f(\bar{x})$	$h(\bar{x})$	saída
RK23	17	11	0	4	1545	8.3382E-02	2.1667E-11	1
				3	1545	8.3382E-02	2.1667E-11	1
				7	13	8.3333E-02	2.6751E-10	1
				7	13	8.3333E-02	2.6751E-10	1
ROBOT	14	2	0	2	122	6.5933E+00	6.0550E-08	1
				1	122	6.5933E+00	6.0550E-08	1
				2	3	6.5933E+00	6.0550E-08	1
				2	3	6.5933E+00	6.0550E-08	1
ROSENMMX	5	0	4	114	180	-3.0479E+01	0.0000E+00	-4
				181	248	-3.5439E+01	0.0000E+00	-4
				12	30	-4.4000E+01	5.5936E-08	1
				12	30	-4.4000E+01	5.5936E-08	1
RSNBRNE	2	2	0	2	8	0.0000E+00	0.0000E+00	1
				1	8	0.0000E+00	0.0000E+00	1
				2	39	0.0000E+00	2.9249E-10	1
				2	39	0.0000E+00	2.9249E-10	1
S268	5	0	5	33	179	6.0830E-03	0.0000E+00	-4
				32	179	6.0830E-03	0.0000E+00	-4
				23	58	1.1911E-08	0.0000E+00	1
				23	58	1.1911E-08	0.0000E+00	1
SIMPLLPA	2	0	2	3	21	1.0000E+00	0.0000E+00	1
				2	21	1.0000E+00	0.0000E+00	1
				4	5	1.0000E+00	0.0000E+00	1
				4	5	1.0000E+00	0.0000E+00	1
SIMPLLPB	2	0	3	3	21	1.1000E+00	0.0000E+00	1
				2	21	1.1000E+00	0.0000E+00	1
				4	12	1.1000E+00	0.0000E+00	1
				4	12	1.1000E+00	0.0000E+00	1
SINVALNE	2	2	0	2	8	0.0000E+00	1.1033E-15	1
				1	8	0.0000E+00	1.1033E-15	1
				2	39	0.0000E+00	1.4552E-11	1
				2	39	0.0000E+00	1.4552E-11	1
SNAKE	2	0	2	7	88	-5.9426E-04	5.9399E-08	1
				6	88	-5.9426E-04	5.9399E-08	1
				6	11	-1.2902E-09	1.2902E-13	1
				6	11	-1.2902E-09	1.2902E-13	1
SUPERSIM	2	2	0	2	8	6.6667E-01	2.4825E-16	1
				1	8	6.6667E-01	2.4825E-16	1
				2	7	6.6667E-01	2.4825E-16	1
				2	7	6.6667E-01	2.4825E-16	1

Nome	$n$	$ \mathcal{E} $	$ \mathcal{I} $	$k$	$\#f$	$f(\bar{x})$	$h(\bar{x})$	saída
SYNTHES1	6	0	6	2	29	1.0000E+01	0.0000E+00	-4
				1	29	1.0000E+01	0.0000E+00	-4
				10	38	7.5928E-01	4.6797E-07	1
				10	38	7.5928E-01	4.6797E-07	1
SYNTHES2	11	1	13	2	80	1.4200E+02	1.0000E+00	-4
				1	80	1.4200E+02	1.0000E+00	-4
				6	10	-5.5442E-01	6.5390E-07	1
				6	10	-5.5442E-01	6.5390E-07	1
SYNTHES3	17	2	21	2	173	1.2200E+02	1.0000E+00	-4
				1	173	1.2200E+02	1.0000E+00	-4
				12	19	1.5082E+01	2.8587E-10	1
				12	19	1.5082E+01	2.8587E-10	1
TAME	2	1	0	2	8	0.0000E+00	0.0000E+00	1
				1	8	0.0000E+00	0.0000E+00	1
				2	4	0.0000E+00	0.0000E+00	1
				2	4	0.0000E+00	0.0000E+00	1
TFI1	3	0	101	158	233	6.0856E+00	0.0000E+00	-4
				115	208	6.0366E+00	0.0000E+00	-4
				604	4909	5.3347E+00	1.1183E-07	1
				687	5650	5.3347E+00	1.1011E-07	1
TFI2	3	0	101	3	83	6.4903E-01	0.0000E+00	1
				2	83	6.4903E-01	0.0000E+00	1
				7	15	6.4903E-01	0.0000E+00	1
				7	15	6.4903E-01	0.0000E+00	1
TFI3	3	0	101	4	63	4.3012E+00	0.0000E+00	1
				3	63	4.3012E+00	0.0000E+00	1
				6	7	4.3012E+00	0.0000E+00	1
				6	7	4.3012E+00	0.0000E+00	1
TRUSPYR1	11	3	1	138	757	1.1461E+01	1.3906E-08	-4
				137	759	1.1461E+01	1.3224E-07	-4
				23	45	1.1229E+01	4.1374E-07	1
				23	45	1.1229E+01	4.1374E-07	1
TRUSPYR2	11	3	8	54	638	1.1451E+01	7.1753E-07	-4
				53	638	1.1451E+01	7.1753E-07	-4
				14	16	1.1229E+01	1.4000E-07	1
				14	16	1.1229E+01	1.4000E-07	1
TRY-B	2	1	0	2	8	1.0000E+00	8.6543E-07	1
				1	8	1.0000E+00	8.6543E-07	1
				2	3	1.0000E+00	8.6543E-07	1
				2	3	1.0000E+00	8.6543E-07	1

Nome	$n$	$ \mathcal{E} $	$ \mathcal{I} $	$k$	$\#f$	$f(\bar{x})$	$h(\bar{x})$	saída
TWOBARS	2	0	2	25	105	1.5087E+00	1.1993E-06	-4
				17	116	1.5087E+00	2.0652E-07	-4
				9	23	1.5087E+00	4.3637E-08	1
				9	23	1.5087E+00	4.3637E-08	1
WOMFLET	3	0	3	33	192	1.4656E-06	1.0050E-06	-4
				32	193	1.4656E-06	1.0050E-06	-4
				21	55	-5.5297E-08	1.0950E-07	1
				31	202	4.7172E-06	1.6054E-05	1
ZANGWIL3	3	3	0	2	12	0.0000E+00	0.0000E+00	1
				1	12	0.0000E+00	0.0000E+00	1
				2	7	0.0000E+00	2.0630E-33	1
				2	7	0.0000E+00	2.0630E-33	1
ZECEVIC2	2	0	2	4	16	-4.1250E+00	0.0000E+00	1
				3	16	-4.1250E+00	0.0000E+00	1
				5	10	-4.1250E+00	7.0599E-10	1
				5	10	-4.1250E+00	7.0599E-10	1
ZECEVIC4	2	0	2	9	102	7.5575E+00	0.0000E+00	1
				8	102	7.5575E+00	0.0000E+00	1
				9	11	7.5575E+00	8.8818E-16	1
				9	11	7.5575E+00	8.8818E-16	1
ZY2	3	0	2	2	33	4.1410E+00	0.0000E+00	-2
				1	33	4.1410E+00	0.0000E+00	-2
				6	6	2.0000E+00	0.0000E+00	1
				6	6	2.0000E+00	0.0000E+00	1