

---

Universidade Estadual de Campinas  
Instituto de Matemática, Estatística e Computação Científica  
Departamento de Matemática Aplicada

---

# Sobre o uso de regiões de confiança para minimização com restrições lineares

**Larissa Oliveira Xavier\***

Doutorado em Matemática Aplicada - Campinas - SP

**Orientador:** Profa. Dra. Sandra Augusta Santos

**Co-Orientador:** Prof. Dr. José Mario Martínez Pérez

Campinas, 2011.

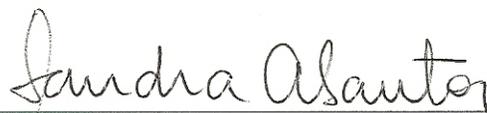
\* Este trabalho teve apoio financeiro da FAPESP, Processo 06/04477-2



# Sobre o uso de regiões de confiança para minimização com restrições lineares \*

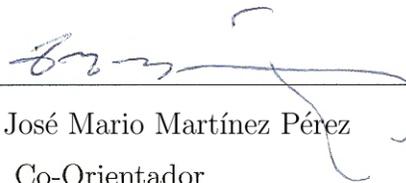
Este exemplar corresponde à redação final da tese devidamente corrigida e defendida por **Larissa Oliveira Xavier** e aprovada pela comissão julgadora.

Campinas, 9 de Novembro de 2011



---

Profa. Dra. Sandra A. Santos  
Orientadora



---

Prof. Dr. José Mario Martínez Pérez  
Co-Orientador

## Banca Examinadora

1. Prof. Dr. Nelson Maculan Filho (COPPE - UFRJ)
2. Prof. Dr. Ademir Alves Ribeiro (Depto. de Matemática - UFPR)
3. Profa. Dra. Márcia Aparecida Gomes Ruggiero (IMECC - Unicamp)
4. Prof. Dr. Roberto Andreani (IMECC - Unicamp)
5. Profa. Dra. Sandra Augusta Santos (IMECC - Unicamp)

Tese apresentada ao Instituto de Matemática, Estatística e Computação Científica, Unicamp, para obtenção do Título de DOUTOR em Matemática Aplicada.

\* Este trabalho teve apoio financeiro da FAPESP Processo 06/04477-2

FICHA CATALOGRÁFICA ELABORADA POR  
MARIA FABIANA BEZERRA MÜLLER - CRB8/6162  
BIBLIOTECA DO INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E  
COMPUTAÇÃO CIENTÍFICA - UNICAMP

X19s	<p>Xavier, Larissa Oliveira, 1983- Sobre o uso de regiões de confiança para minimização com restrições lineares / Larissa Oliveira Xavier. - Campinas, SP : [s.n.], 2011.</p> <p>Orientador: Sandra Augusta Santos. Coorientador: José Mario Martínez Pérez.</p> <p>Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.</p> <p>1. Otimização matemática. 2. Programação não- linear. 3. Método de região de confiança. 4. Métodos iterativos (Matemática). I. Santos, Sandra Augusta, 1964-. II. Martínez Pérez, José Mario, 1948- . III. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. IV. Título.</p>
------	--

Informações para Biblioteca Digital

**Título em inglês:** On trust-region algorithms for linearly constrained minimization

**Palavras-chave em inglês:**

Optimization (Mathematics)

Nonlinear programming

Trust-region methods

Iterative methods (Mathematics)

**Área de concentração:** Matemática Aplicada

**Titulação:** Doutor em Matemática Aplicada

**Banca examinadora:**

Sandra Augusta Santos [Orientador]

Márcia Aparecida Gomes Ruggiero

Roberto Andreani

Nelson Maculan Filho

Ademir Alves Ribeiro

**Data da defesa:** 09-11-2011

**Programa de Pós-Graduação:** Matemática Aplicada

**Tese de Doutorado defendida em 09 de novembro de 2011 e aprovada**

**Pela Banca Examinadora composta pelos Profs. Drs.**



---

**Prof(a). Dr(a). SANDRA AUGUSTA SANTOS**



---

**Prof(a). Dr(a). MÁRCIA APARECIDA GOMES RUGGIERO**



---

**Prof(a). Dr(a). ROBERTO ANDREANI**



---

**Prof(a). Dr(a). NELSON MACULAN FILHO**



---

**Prof(a). Dr(a). ADEMIR ALVES RIBEIRO**



# Agradecimentos

Agradeço:

Aos meus pais e a minha família por sempre me apoiarem em meus estudos e atividades.

À professora Sandra Santos, pelo apoio e a enorme contribuição no desenvolvimento deste projeto.

Ao professor José Mario Martínez pela co-orientação e pelas importantes sugestões para o desenvolvimento deste projeto.

Ao John Gardenghi pela ajuda e apoio durante as implementações computacionais.

Aos professores que fizeram parte da banca pelos comentários e sugestões.

Aos professores do IMECC com os quais pude aprender a maior parte do que sei hoje e pelas sugestões e incentivos durante estes anos de estudo.

Aos meus colegas e amigos pelo apoio e torcida.

Aos funcionários da secretaria e biblioteca do IMECC que sempre me ajudaram quando precisei.

À FAPESP pelo apoio financeiro.



# Resumo

Neste trabalho apresentamos o estudo de dois algoritmos baseados em regiões de confiança para minimização de problemas suaves com restrições lineares. O primeiro algoritmo proposto, com uma estratégia de restrições ativas, foi desenvolvido a partir do trabalho de Gay. O segundo algoritmo apresentado explora a técnica de pontos interiores presente nos métodos de barreira. Ambos são acompanhados de respectivos resultados de boa definição e de convergência global e local. Os dois algoritmos foram testados para a resolução de problemas de distribuição de pontos em polígonos, utilizando o algoritmo de Rojas, Santos e Sorensen, livre de fatorações de matrizes, para resolver os subproblemas internos de região de confiança. O problema dos pontos no polígono não foi encontrado na literatura para o teste de algoritmos de otimização e pode ser visto como uma modificação do problema de distribuição de pontos em caixas, sugerido por Powell. Embora possua estrutura favorável para a geração de problemas com dimensão variável, e potencialmente de grande porte, no contexto livre de fatorações, trata-se de um problema difícil e desafiador, com uma grande quantidade de minimizadores locais. Experimentos numéricos comparativos entre as propostas foram feitos e analisados, indicando que os algoritmos são efetivos na obtenção de pontos estacionários de segunda ordem, com ligeira vantagem para o desempenho do algoritmo baseado em restrições ativas, em termos do tempo computacional empregado.

**Palavras-chave:** minimização com restrições lineares; regiões de confiança; estratégia de restrições ativas; métodos de barreira; resultados de convergência; testes numéricos.



# Abstract

In this work two trust-region-based algorithms are analyzed for linearly constrained minimization. The first one is an active-set method, based on Gay's ideas. The second one uses interior-point techniques of barrier methods. Both algorithms are proved to be well defined and accompanied by the respective convergence results. The implementation was developed resting upon Rojas, Santos and Sorensen matrix-free algorithm for solving the inner trust-region subproblems. The family of adopted test-problems involves the distribution of points in a polygon, a modification of Powell's problem of distributing points in a square. Despite its favorable structure for generating instances with variable and potentially large dimension, in the matrix-free context, the problem is indeed hard and challenging, with many local minimizers. Comparative computational experiments illustrate the performance of the proposed algorithms, showing that both are effective to obtain second-order stationary points, with a slight advantage of the active-set-based algorithm when it comes to the CPU time spent.

**Keywords:** linearly constrained minimization; trust region; active-set strategy; barrier methods; convergence results; numerical tests.



# Notações utilizadas

- *Poliedro*: Dada uma matriz  $A \in \mathbb{R}^{m \times n}$  e um vetor  $b \in \mathbb{R}^m$ , a interseção de  $m$  semi-espacos, denotada por  $\{x \mid Ax \geq b\}$ , é denominada conjunto poliedral (cf. [10]).
- *Politopo*: É o fecho convexo de um número finito de pontos em  $\mathbb{R}^n$ . Um politopo é um poliedro limitado (cf. [10]).
- $\|x\|$ : Norma Euclidiana do vetor  $x$ .
- $\{x^k\}$ : Sequência de pontos  $x^0, x^1, x^2, \dots, x^k, \dots$
- $Nu(A)$ : Núcleo da matriz  $A$ .
- $Im(A)$ : Imagem da matriz  $A$ .
- $\mathcal{I}_A$ : Conjunto dos índices das linhas correspondentes às restrições ativas da matriz  $A$  para um determinado ponto.
- $A_{\mathcal{I}_A}$ : Submatriz de  $A$  correspondente às linhas com índices em  $\mathcal{I}_A$ .



# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Implementações conhecidas . . . . .	2
1.1.1	CONOPT . . . . .	3
1.1.2	CPLEX . . . . .	3
1.1.3	fmincon (MATLAB) . . . . .	4
1.1.4	GALAHAD . . . . .	5
1.1.5	IPOPT . . . . .	6
1.1.6	KNITRO . . . . .	6
1.1.7	LOQO . . . . .	8
1.1.8	MINOS . . . . .	10
1.1.9	MOSEK . . . . .	11
1.1.10	SNOPT . . . . .	11
1.1.11	TANGO . . . . .	12
1.2	Outros trabalhos relevantes . . . . .	14
1.3	Contribuições desta tese . . . . .	19
1.4	Organização do texto . . . . .	20
<b>2</b>	<b>Ferramentas e ingredientes básicos</b>	<b>23</b>
2.1	Problema e condições de otimalidade . . . . .	23
2.2	Fatoração $QR$ e base para o núcleo . . . . .	24
2.3	Algoritmo de Rojas, Santos e Sorensen . . . . .	28
<b>3</b>	<b>Algoritmo de restrições ativas e propriedades teóricas</b>	<b>31</b>
3.1	Modelo Algorítmico . . . . .	31
3.1.1	Ponto inicial viável . . . . .	32
3.1.2	Estimativa dos multiplicadores . . . . .	34
3.1.3	Resolução do subproblema . . . . .	35
3.2	Resultados de convergência . . . . .	41
<b>4</b>	<b>Algoritmo de barreira e propriedades teóricas</b>	<b>43</b>
4.1	Algoritmo com elementos de métodos de barreira . . . . .	45
4.2	Estimativa dos multiplicadores . . . . .	54
4.3	Resultados adicionais de convergência . . . . .	56

<b>5</b>	<b>O problema dos pontos em um polígono</b>	<b>67</b>
5.1	Abordagem com restrições de desigualdade . . . . .	69
5.1.1	Fatoração $QR$ para o cálculo de $Z$ . . . . .	70
5.2	Abordagem com método de barreiras . . . . .	74
5.2.1	Informações da matriz $Z$ . . . . .	75
<b>6</b>	<b>Experimentos computacionais</b>	<b>77</b>
6.1	Sobre o desempenho de LSTRS . . . . .	77
6.2	Geração dos problemas dos pontos no polígono . . . . .	84
6.2.1	Primeira geração dos pontos iniciais . . . . .	87
6.3	Testes preliminares . . . . .	87
6.3.1	Constantes e critérios de parada utilizados . . . . .	88
6.3.2	Conjunto inicial de problemas-teste . . . . .	92
6.3.3	Calibração do parâmetro $\rho_0$ para o método de barreira . . . . .	96
6.3.4	Outra estratégia para a escolha do ponto inicial . . . . .	101
6.3.5	Escolhas de $r_{min}$ e $r_{max}$ para a construção dos polígonos . . . . .	105
6.4	Testes com as novas escolhas . . . . .	106
	<b>Considerações finais</b>	<b>129</b>
	<b>Referências Bibliográficas</b>	<b>131</b>

# Capítulo 1

## Introdução

O problema considerado nesta tese é a minimização de uma função não linear em um conjunto poliedral. O principal objetivo é desenvolver estratégias livres de fatorações que permitam resolver problemas desta classe, estruturados e de grande porte, para os quais as derivadas de segunda ordem da função objetivo sejam conhecidas, de maneira que o produto da matriz Hessiana por um vetor seja implementável.

Para resolver problemas nos quais o conjunto viável é poliedral, e que, além disso, possuem função objetivo linear ou quadrática, estão disponíveis *solvers* bastante consolidados junto às comunidades acadêmica e comercial, como o CPLEX e o MINOS. Ambos ganharam notoriedade a partir da iniciativa pioneira de seus precursores, dos colaboradores que deram prosseguimento ao projeto, e da equipe que os mantém até os dias atuais. A seguir apresentaremos mais detalhes sobre estes *solvers*, juntamente com outros, para resolver problemas de otimização linear e não linear. Nossa intenção, ao preparar esse levantamento, foi colocar os diferentes *solvers* em perspectiva, para evidenciar as semelhanças e as diferenças com as abordagens propostas nesta tese. Não apresentaremos resultados numéricos baseados nos *solvers* descritos para os problemas-teste considerados neste trabalho.

Para facilitar o acompanhamento dos detalhes apresentados para os diversos códigos e programas disponíveis, vamos considerar a classificação dos problemas de acordo com a complexidade da função objetivo: Programação Linear (PL), Programação Quadrática (PQ) e Programação Não Linear (PNL). Também será conveniente subdividir os problemas em convexos e não-convexos, e ainda, em suaves e não-suaves, embora estes últimos estejam fora do escopo desta pesquisa.

Na resolução de problemas de minimização suave com restrições lineares podemos

destacar três paradigmas:

- i) explorar diretamente o conjunto viável, empregando estratégias de restrições ativas e projeções em conjuntos poliedrais;
- ii) trabalhar com pontos interiores e uma sequência de problemas irrestritos (ou de sistemas não lineares primais-duais);
- iii) trabalhar com pontos exteriores e uma sequência de problemas irrestritos, usando penalização exata ou Lagrangiano aumentado.

Quando se opta por um destes paradigmas, ou equivalentemente, quando se escolhe um determinado programa ou código para resolver um dado problema, tal escolha estará intimamente vinculada à natureza da não linearidade da função objetivo do problema em consideração. Esse aspecto ficará evidenciado no levantamento que apresentamos na próxima seção, com os principais códigos e programas disponíveis para a comunidade científica, em ordem alfabética: CONOPT, CPLEX, fmincon, GALAHAD, IPOPT, KNITRO, LOQO, MINOS, MOSEK, SNOPT, TANGO. Exceto pela rotina fmincon do MATLAB e pelos materiais da biblioteca GALAHAD, do código LOQO e do projeto TANGO, os demais *solvers* compõem a biblioteca da plataforma *AIMMS – Advanced Integrated Multidimensional Modeling Software*, uma das mais robustas disponíveis no mercado para o desenvolvimento de sistemas de decisão baseados em modelos de otimização.<sup>1</sup> Os *solvers* CONOPT, IPOPT, KNITRO, LANCELOT, LOQO, MINOS, MOSEK e SNOPT também fazem parte do conjunto disponível na ferramenta de submissão NEOS Server (cf. <http://www.neos-server.org/neos/> [43]) para PNL.

## 1.1 Implementações conhecidas

Apresentamos a seguir as principais características dos métodos empregados pelos diferentes *solvers*, bem como as referências relacionadas.

---

<sup>1</sup>O *AIMMS* é desenvolvido e comercializado pela *Paragon Decision Technology (Paragon)*, que oferece às universidades uma licença completa (versão educacional) para atividades de ensino e pesquisa sem fins lucrativos. Nessa versão, o *AIMMS* vem disponível com um conjunto padrão de *solvers*, incluindo o CONOPT, e versões em tamanho limitado do KNITRO. A versão educacional pode ser ampliada com a adição de versões completas (e tamanho ilimitado) do CPLEX, KNITRO, MINOS e SNOPT. Tal licença está associada ao domínio IP, de maneira a só funcionar nas redes de computadores da universidade em que foi registrada.

### 1.1.1 CONOPT

Na linha do primeiro paradigma destacado anteriormente, mas desenvolvido para PNL com restrições não lineares gerais, o código CONOPT consiste em uma implementação do método do Gradiente Reduzido Generalizado (GRG) para problemas de grande porte, cf. [46]. Foi desenvolvido pela *ARKI Consulting and Development A/S*, Bagsvaerd, Dinamarca, a partir do trabalho [45]. É o método padrão do *AIMMS* para PNL e PQ, sendo oferecido nessa plataforma como uma alternativa complementar para KNITRO e SNOPT, possivelmente pelo caráter restaurador da abordagem que emprega com relação ao conjunto viável. É recomendado para problemas com restrições com alto grau de não linearidade, com um bom desempenho para modelos com poucos graus de liberdade. Duas outras situações favoráveis para CONOPT ocorrem com modelos que possuem muitas equações, que podem ser resolvidas uma a uma, ou então quando variáveis intermediárias podem ser usadas para definir termos na função objetivo, sendo eliminadas do modelo, de forma que as restrições passem a integrar a função objetivo.

Quando usado no sistema de submissão NEOS Server, CONOPT admite dados de entrada nos formatos AMPL (*Modeling Language for Mathematical Programming*, cf. [53]) ou GAMS (*General Algebraic Modeling System*, cf. <http://www.gams.com/>). O manual do CONOPT pode ser obtido em <http://www.gams.com/dd/docs/solvers/conopt.pdf>.

### 1.1.2 CPLEX

O CPLEX, disponível em <http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/>, está atualmente na versão 12.2, denominada IBM ILOG CPLEX Optimization Studio. Independentemente da plataforma *AIMMS*, a *IBM Academic Initiative*<sup>2</sup> também permite que membros da comunidade acadêmica se registrem e tenham acesso gratuito à versão completa de todos os seus produtos.

O conjunto de ferramentas que compõem o IBM ILOG CPLEX Optimization Studio permite não só resolver problemas de PL e PQ, como também problemas com variáveis inteiras, variáveis inteiras mistas e problemas com restrições quadráticas. Possui ainda um sistema integrado para modelagem, e ferramentas para análise dos resultados. Da equipe de idealizadores originais do CPLEX destacamos os pesquisadores Robert Bixby e Irv Lustig.

---

<sup>2</sup><http://www-01.ibm.com/software/websphere/products/optimization/academic-initiative/index.html>

A implementação de alto desempenho do CPLEX para PL e PQ é baseada nos algoritmos primal simplex, dual simplex (*default*) e de barreira, e possui estratégias sofisticadas de pré-processamento, gerenciamento de memória, além de uma série de opções algorítmicas para controle de parâmetros. Mais detalhes podem ser obtidos no manual da versão 12, disponível em <http://www.gams.com/dd/docs/solvers/cplex.pdf>.

Para uma avaliação do desempenho do CPLEX em versão mais antiga, comparativamente a métodos primais duais de pontos interiores para PL ver [81].

Conforme dados apresentados em <http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/cplex-optimizer/cplex-performance/>, o desempenho comparativo da versão 12.2 do CPLEX relativamente às versões 12.1 e 12.0 anteriores se mostrou significativamente superior.

### 1.1.3 fmincon (MATLAB)

A rotina `fmincon` integra o *Optimization Toolbox* do MATLAB e foi desenvolvida para resolver PNL com restrições gerais, tanto lineares quanto não lineares. Possui duas implementações, específicas para problemas de médio e grande porte. Os métodos para grande porte são os *defaults*.

No caso de problemas com restrições lineares de igualdade (PNLI), que podem ser escritos como  $\min f(x)$  s.a  $Ax = b$ , em que  $A$  é uma matriz  $m \times n$  com  $m \leq n$ , de acordo com o *help online* da versão 7.0.0.19920 (R14), o *Optimization Toolbox* efetua um pré-processamento para remover possíveis dependências lineares usando uma técnica baseada na fatoração  $LU$  de  $A^T$  (cf. [37]). Assumindo-se então que  $A$  possui posto  $m$ , o método para o problema PNLI possui duas diferenças essenciais com a abordagem para minimização irrestrita. Primeiro, um ponto inicial viável  $x_0$  é computado, usando um passo de quadrados mínimos esparsos, de tal forma que  $Ax_0 = b$ . A seguir, um algoritmo de gradientes conjugados preconditionados reduzido (RPCG) é empregado para computar uma aproximação para o passo de Newton reduzido (ou para uma direção de curvatura negativa no espaço nulo da matriz  $A$ ). Para mais detalhes, ver [37].

Para PNLs em caixas, o algoritmo padrão é o método de Newton reflexivo de Coleman & Li (cf. [32, 33, 34]). Para PNLs gerais, o algoritmo implementado é de regiões de confiança em subespaços e baseia-se no método para o problema em uma caixa: cada iteração envolve a solução de um sistema linear de grande porte usando gradientes

conjugados preconditionados (PCG).

A rotina `fmincon` na versão para médio porte emprega um algoritmo de Programação Quadrática Sequencial (PQS), em que os PQs são resolvidos por meio da implementação de um método de restrições ativas nos moldes de [59, 62]. A matriz Hessiana do Lagrangiano é aproximada pela fórmula BFGS (cf. [42]).

As principais limitações de `fmincon` são: todas as funções envolvidas precisam ser contínuas; as variáveis devem ser reais; as soluções encontradas têm caráter local; para problemas com região viável vazia, `fmincon` busca minimizar a máxima violação das restrições; o método para problemas de grande porte não permite que os limitantes inferior e superior para a caixa coincidam (para modelar essa ocorrência é preciso ativar a versão para médio porte).

Vale dizer que para PL, o algoritmo *default* para grande porte implementado na rotina `linprog` do MATLAB é primal dual, uma variante do preditor corretor proposto por Mehrotra (cf. [85]). Em `linprog` há também uma implementação do simplex, para problemas de médio porte.

#### 1.1.4 GALAHAD

A biblioteca GALAHAD, descrita em [66], e acessível em <http://www.galahad.rl.ac.uk/galahad-www/>, contém um conjunto de rotinas para resolver problemas de programação quadrática, com implementações para algoritmos de pontos interiores e de restrições ativas, bem como ferramentas de pré-processamento para os problemas antes da etapa da resolução. Também inclui uma versão atualizada do *solver* LANCELOT (*Release B*), uma implementação de um algoritmo de Lagrangianos aumentados concebida para PNL de grande porte com restrições de igualdade lineares ou não lineares, e canalizações nas variáveis. Assume-se que todas as funções envolvidas sejam parcialmente separáveis. A primeira versão de LANCELOT está descrita em [38]. As mudanças introduzidas da versão A para a B incluem uma estratégia de decréscimo não monótono, projeções no sentido de Moré e Toraldo (cf. [88]), uso opcional do preconditionador de Lin e Moré [78], regiões de confiança estruturadas (ver [39]), entre outros. Resultados numéricos com o desempenho de LANCELOT B estão disponíveis em [65].

Gould e Toint propuseram, em 2002 [67], um método para resolução de problemas de programação quadrática resolvendo, a cada iteração, subproblemas com restrições de igualdade relacionadas ao conjunto das restrições ativas. Para a resolução dos subproblemas, um método de gradientes conjugados com preconditionamento é utilizado. Ou-

tro trabalho relacionado, dos mesmos autores, também desenvolvido para programação quadrática não convexa e de grande porte [68] propõe dois novos métodos, um de restrições ativas, outro de pontos interiores e regiões de confiança, com a solução de um subproblema com restrições de igualdade em comum. Apresenta resultados numéricos comparativos.

### 1.1.5 IPOPT

A versão original do IPOPT (em Fortran) foi resultado da pesquisa de doutorado de Andreas Wächter (cf. [118]). Combina a técnica de pontos interiores globalizada via busca linear e filtros para resolver PNL com restrições gerais, de grande porte. O programa foi preparado no formato código aberto, e distribuído por iniciativa da *COIN-OR* (*COmputational INfrastructure for Operations Research*), uma cooperação sem fins lucrativos. Desde 2002, IPOPT tem sido ativamente desenvolvido pela *COIN-OR*.

Visando prosseguir em extensões naturais do código e permitir a incorporação de novas características, a *IBM Research* investiu na implementação do IPOPT em C++. O código foi refeito com a ajuda de Carl Laird, que integrou o Departamento de Ciências Matemáticas da *IBM Research* nos verões de 2004 e 2005 durante seu doutorado. A nova versão C++ do *solver* de otimização IPOPT (atualmente 3.0.0) é patrocinada pela *IBM Research* e permanece parte da iniciativa *COIN-OR*. A documentação referente a instalação e uso está disponível em <http://www.coin-or.org/Ipopt/documentation/>. Embora o desenvolvimento da versão em Fortran tenha cessado, o código ainda pode ser obtido em <http://www.coin-or.org/download/source/Ipopt-Fortran/>.

As referências associadas aos aspectos teóricos e práticos do IPOPT são, respectivamente, [120] e [121]. Em [40], o desempenho de IPOPT original é comparado com um novo método, também de pontos interiores, mas com passos obtidos de maneira inexata. Para mais detalhes sobre a implementação desse novo método e a comparação feita, ver [41].

### 1.1.6 KNITRO

O *solver* KNITRO para PNL geral, atualmente na versão 7.0, foi desenvolvido e é mantido pela empresa *Ziena Optimization Inc.*, de modelagem, consultoria e *software*. Visando ampliar o espectro de aplicações de potencial resolução, KNITRO oferece três opções de algoritmos consolidados e implementados de maneira robusta, que são: 1) um

método de barreira com fatoração direta para a matriz do sistema não linear das condições de Karush-Kuhn-Tucker (KKT), com bom desempenho em problemas mal condicionados; 2) um método de barreira com gradientes conjugados, uma alternativa para o método direto quando a fatoração é inviável ou impraticável; 3) um método de restrições ativas, que combina os princípios clássicos com um novo subproblema de PL para identificar rapidamente o conjunto das restrições ativas. Além disso, as soluções aproximadas obtidas pelos métodos de barreira podem ser refinadas mediante uma passagem para o algoritmo de restrições ativas. Todos os algoritmos empregam derivadas segundas exatas, e adotam um escalamento eficiente de modo a resolver problemas com dezenas ou centenas de milhares de variáveis e restrições.

KNITRO foi planejado para resolver PL, PQ, e PNL suave, tanto os casos de PQ e PNL convexos quanto não convexos. Também é efetivo para regressões não lineares, problemas com restrições de complementaridade (MPCCs ou MPECs), e programação inteira-mista, particularmente problemas convexos inteiros mistos e não lineares (MINLP).

Em problemas não convexos, podem existir várias soluções localmente ótimas. Em geral, KNITRO retorna a primeira solução ótima encontrada, mas oferece um algoritmo *multi-start* que busca por uma melhor solução, recomeçando KNITRO a partir de diferentes pontos iniciais.

As principais referências técnicas para KNITRO são o capítulo [31] e a bibliografia contida no Manual do Usuário [122]. Os algoritmos de pontos interiores estão descritos em [27], e os aspectos teóricos correspondentes podem ser encontrados em [28, 30]. Para o método de restrições ativas, KNITRO implementa o algoritmo de programação linear sequencial (SLQP) descrito em [29], em que os subproblemas de PL são usados para estimar o conjunto ativo.

O desempenho do KNITRO para problemas não lineares de grande porte é descrito em [96], no qual aplicações com mais de cem mil variáveis e restrições são resolvidas, usando a versão 5.0 em uma estação de trabalho padrão. Inclui ainda discussão de técnicas simples de calibragem para melhora do desempenho em problemas grandes. Um estudo numérico comparativo do desempenho de KNITRO, SNOPT e LOQO foi feito em [12] (ver detalhes a seguir).

### 1.1.7 LOQO

O *solver* LOQO, para PNL geral, foi desenvolvido e é mantido por Robert Vanderbei e Hande Y. Benson (<http://www.princeton.edu/~rvdb/>). Implementa um método primal dual de pontos inactiváveis e seguidor de caminho. Lançado em 1991, inicialmente foi concebido para resolver PQ (cf. [115]), sendo posteriormente estendido para PNL geral ([106, 117]).

Vanderbei descreve em [115] a versão originalmente proposta do *solver*, implementada para PL e PQ. Especial ênfase é feita aos detalhes algorítmicos, particularmente às vantagens de se trabalhar com métodos de pontos interiores que preservem a simetria primal-dual. É mostrado que o formato padrão MPS (*Mathematical Programming System*) se ajusta adequadamente ao paradigma da formulação simétrica e inclui um conjunto de resultados computacionais para as duas classes de problemas.

No artigo [117], Vanderbei e Shanno descrevem as extensões feitas em LOQO para permitir a solução de PNLs gerais. As principais foram a inclusão de uma função de mérito e uma modificação na direção de busca para garantir descenso em tal função. A robustez do método ficou evidenciada em experimentos numéricos preliminares. Comparações com MINOS e LANCELOT ofereceram indicativos de eficiência, no sentido de grande redução no tempo de CPU demandado para a solução de algumas classes de modelos.

Shanno e Vanderbei, em [106], prosseguiram a análise do desempenho de LOQO para PNL geral. O principal aspecto pesquisado foi a comparação entre os ordenamentos primal e dual em métodos de ordem superior, nos quais, em geral, cada fatoração da matriz Hessiana é empregada mais de uma vez para melhorar a eficiência. Os resultados mostraram que, contrariamente ao que ocorre nos casos de PL e PQ convexa, as correções de ordem superior para a trajetória central não são computacionalmente eficientes para PNL geral. A melhora obtida no caso geral se deu com uma variante do método preditor corretor de Mehrotra [85].

Para o uso de LOQO, a única hipótese sobre as funções envolvidas é serem suaves. Há duas maneiras de trabalhar com o *solver*: usando uma interface AMPL, a qual gerencia as ferramentas para avaliar funções, gradientes e Hessianas, ou como subrotina de uma biblioteca de programas. Para PL, pode ser empregada a entrada de dados via arquivos MPS.

O manual da versão atual (4.05) [116], entre outros detalhes, oferece opções para ajustes de alguns parâmetros do programa, de maneira a deixá-lo mais eficiente para a

classe de problemas de interesse.

Em [12], Benson, Shanno e Vanderbei fizeram um estudo comparativo entre os desempenhos dos *solvers* LOQO, KNITRO e SNOPT para PNL de grande porte. Um conjunto extenso de resultados numéricos é apresentado para diferentes classes de problemas, sendo evidenciadas as características de cada *solver* que o tornam eficiente, ou não, para cada uma das classes examinadas. Destacaram-se os seguintes aspectos:

- Empregar fatorações esparsas para a matriz reduzida do sistema KKT foi mais eficiente que utilizar gradientes conjugados para PNL esparsos.
- Comparado com fatorações densas, gradientes conjugados para resolver as equações de Newton podem ser eficientes quando a matriz é suficientemente bem condicionada.
- Para PQ apenas com restrições lineares de igualdade, os problemas devem ser resolvidos da maneira original, evitando a conversão em um problema com restrições de desigualdades para se adequar ao formato dos métodos de pontos interiores, como faz LOQO usualmente.
- Para PQ somente com restrições lineares de desigualdade, por outro lado, o melhor desempenho foi obtido com o método de pontos interiores implementado em LOQO.
- Para PQ com restrições lineares mistas, tanto de igualdade quanto de desigualdade, não houve evidências a favor de nenhum dos *solvers*, e os experimentos foram inconclusivos.
- Para PNLs restritos, o desacoplamento que LOQO faz das variáveis livres se mostrou eficiente para melhorar o condicionamento da matriz reduzida do sistema KKT ([106]).
- A maneira com que SNOPT trabalha em um espaço reduzido de variáveis torna-o eficiente para problemas com poucos graus de liberdade. Por usar apenas informações de primeira ordem para estimar a Hessiana reduzida (via BFGS com memória limitada), para problemas com muitos graus de liberdade, o método quase-Newton empregado por SNOPT não é competitivo com uma estratégia Newtoniana.

Uma investigação mais aprofundada das questões que podem levar LOQO a falhar são analisadas em [13], incluindo a avaliação de um exemplo, devido a Wächter e Biegler [119], no qual o algoritmo padrão primal dual de pontos interiores converge a um ponto da fronteira do conjunto viável que não satisfaz as condições de otimalidade de primeira ordem. Benson e Shanno, em [11], propuseram ainda uma abordagem de penalização exata no contexto dos métodos de pontos interiores primais duais para PNL. Esta modificação foi incluída no *solver* LOQO e seu desempenho é analisado em um extenso conjunto de testes numéricos para classes distintas de problemas.

### 1.1.8 MINOS

O *solver* MINOS, para PNL suave geral, atualmente na versão 5.5, foi desenvolvido e é mantido pela *Stanford Business Software, Inc.* ([http://sbsi-sol-optimize.com/asp/sol\\_product\\_minos.htm](http://sbsi-sol-optimize.com/asp/sol_product_minos.htm)).

A implementação do MINOS segue o desenvolvimento de [89], em que o algoritmo do gradiente reduzido de Wolfe [124] é combinado com uma estratégia quase Newton [42]. Desenvolvido para solucionar problemas tanto de pequeno quanto de grande porte, o MINOS permite resolver PLs, problemas irrestritos, otimização com restrições lineares e PNLs gerais. Não resolve programação inteira nem possui uma implementação do método dual simplex.

É particularmente efetivo para problemas com função objetivo linear ou não linear e restrições lineares esparsas (p. ex. PL e PQ). As funções não lineares envolvidas precisam ser suaves, mas não necessariamente convexas. MINOS é capaz de lidar com um grande número de restrições não lineares, e pode ser mais eficiente que CONOPT e SNOPT se a função objetivo e seu gradiente tiverem avaliação barata. Vale dizer que MINOS utiliza apenas informações de primeira ordem.

Um dos principais aspectos que destaca MINOS dos demais *solvers* são os algoritmos numericamente estáveis que o compõem, por exemplo a rotina para computar eficientemente uma base LU esparsa (cf. [60]). Também são fundamentais para seu bom desempenho o fato da implementação contemplar estratégias de bons recomeços (*warm starts*) e procedimentos práticos anti-degeneração, bem como escalamento automático das restrições lineares e não lineares. Permite entrada de dados no formato MPS. Mais detalhes podem ser obtidos no manual da versão MINOS 5.5 (cf. [90]).

### 1.1.9 MOSEK

O *solver* MOSEK é o principal produto da empresa dinamarquesa *MOSEK ApS*, especializada em *software* para otimização matemática geral. Trata-se de um conjunto de programas para problemas de otimização de grande porte esparsa das classes de PL, PQ convexos e programação inteira mista. MOSEK também pode resolver PLs generalizados envolvendo restrições cônicas e PNL convexos. Foi lançado em 1999 e atualmente está na versão 6.

MOSEK é particularmente adequado para resolver PL de grande porte pois, além de versões do primal e do dual Simplex, também possui uma implementação extremamente eficiente de um algoritmo de pontos interiores (cf. [1]), cuja base foi proposta por Mizuno, Todd e Ye ([127]). O algoritmo de pontos interiores implementado, particularmente, possui diversas opções intrínsecas (e complexas!) que o usuário pode especificar para fazer um ajuste fino do *solver* ao modelo específico que deseja resolver.

Os relatórios técnicos [2] e [3], disponíveis em <http://www.mosek.com/>, oferecem mais detalhes sobre os *solvers* implementados, a evolução das versões, e apresentam resultados numéricos comparativos.

### 1.1.10 SNOPT

O *solver* SNOPT, para PNL suave geral, atualmente na versão 7, foi desenvolvido por Philip Gill, Walter Murray e Michael Saunders e é mantido pela *Stanford Business Software, Inc.* ([http://sbsi-sol-optimize.com/asp/sol\\_product\\_snopt.htm](http://sbsi-sol-optimize.com/asp/sol_product_snopt.htm)). Baseia-se em um método de programação quadrática sequencial implementado para restrições de desigualdade gerais (lineares e não lineares). As derivadas de primeira ordem devem estar disponíveis e, por hipótese, os gradientes das restrições devem ser esparsos. A função de mérito usada em SNOPT para assegurar convergência global é um Lagrangiano aumentado suave, e existe uma estratégia explícita (variáveis elásticas) para detectar infactibilidade do problema original e dos subproblemas QPs. O subproblema quadrático (semidefinido) baseia-se em uma aproximação quase-Newton com memória limitada para a Hessiana do Lagrangiano e usa um algoritmo com Hessiana reduzida (SQOPT) para resolver os subproblemas QPs. Os aspectos teóricos e mais detalhes sobre a implementação podem ser vistos em [57], que inclui resultados numéricos para a maioria dos problemas das coleções CUTE e COPS.

Embora as funções envolvidas precisem ser suaves, o uso de SNOPT não demanda qualquer hipótese de convexidade para os problemas. Além disso, descontinuidades nos

gradientes são aceitáveis desde que não ocorram muito próximas a um ponto ótimo. Gradientes não fornecidos são estimados por diferenças finitas. Por demandar um número relativamente pequeno de avaliações das funções envolvidas, **SNOPT** é especialmente eficiente para problemas não lineares em que as funções e gradientes possuem avaliação cara. Em problemas de grande porte, **SNOPT** é mais eficiente para problemas com um grande número de restrições lineares esparsas (e possivelmente canalizações nas variáveis), se apenas algumas das variáveis atuam de forma não linear, ou se há um número relativamente pequeno de graus de liberdade na solução (i.e., se muitas restrições são ativas).

**SNOPT** está implementado em **Fortran77** e é distribuído como código fonte. Pode ser usado em qualquer máquina que possua memória suficiente e um compilador **Fortran**. Pode ser chamado como subrotina a partir de um *driver* (em **Fortran**, **C** ou **MATLAB**), ou usado diretamente como programa principal, nesse caso com leitura dos dados no formato **MPS**. Uma interface para a coleção **CUTE** está disponível.

Um estudo numérico comparativo do desempenho de **SNOPT**, **LOQO** e **KNITRO** foi feito em [12] (conforme detalhado em 1.1.7). Particularidades sobre a versão 7 e os avanços recentes de **SNOPT** podem ser obtidos no manual [58].

### 1.1.11 TANGO

O projeto **TANGO**, do acrônimo *Trustable Algorithms for Nonlinear General Optimization*, disponível em <http://www.ime.usp.br/~egbirgin/tango/>, é um conjunto de rotinas para otimização implementadas em **Fortran**, desenvolvidas conjuntamente nos Departamentos de Matemática Aplicada da UNICAMP e no Departamento de Ciências da Computação da USP (SP), sob a coordenação dos professores José Mario Martínez e Ernesto Birgin. Os programas foram preparados na modalidade *software* livre, podendo ser redistribuídos ou modificados, sob os termos da licença pública geral GNU.

Os principais códigos disponíveis são **ALGENCAN** e **SPG**. O primeiro se destina ao problema PNL com restrições gerais e não usa qualquer manipulação de matrizes, podendo resolver problemas extremamente grandes em tempo computacional moderado. Trata-se da implementação de um algoritmo de Lagrangiano aumentado (cf. [6, 15, 19]) no qual os subproblemas são resolvidos usando **GENCAN** (cf. [18]). O código **GENCAN** minimiza uma função suave com (potencialmente muitas) variáveis canalizadas. O segundo código principal, **SPG**, minimiza uma função suave com restrições convexas e, possivelmente, um número grande de variáveis (cf. [21, 22, 23]).

Outros códigos que compõem o projeto TANGO e usam ALGENCAN ou SPG como subalgoritmos são os seguintes, acompanhados das referências com a descrição completa e resultados numéricos:

- **ALABB**: Algoritmo de Lagrangiano aumentado para otimização global, que usa o método  $\alpha$ BB para obter a solução global de subproblemas com restrições lineares (cf. [16]);
- **ALBETRA**: Algoritmo de Lagrangiano aumentado que usa BETRA para resolver os subproblemas de minimização em caixas. BETRA é um método de restrições ativas para problemas com variáveis canalizadas, que dentro das faces emprega regiões de confiança Euclidianas clássicas e utiliza a direção do gradiente espectral projetado para abandonar as faces (cf. [8]);
- **ALGENCAN-NEWTON**: Desenvolvido para melhorar a convergência local de ALGENCAN, nesta versão o método de Newton é usado para resolver o sistema KKT identificado pelo algoritmo de Lagrangiano aumentado (cf. [20]);
- **ALSPG**: Algoritmo de Lagrangiano aumentado que usa o método SPG para resolver subproblemas convexos restritos (cf. [5]);
- **IVM**: Método de métrica variável inexato para minimização convexa restrita (cf. [7]);
- **SCG**: Método de gradiente conjugado espectral para minimização irrestrita (cf. [17]);
- **GENLIN**: Método de gradiente espectral projetado parcial com estratégia de restrições ativas para minimização com restrições lineares (cf. [9]);
- **ALGENCAN-OTR**: Método de região de confiança exterior para otimização com restrições (cf. [14]).

Três conjuntos de aplicações integram o projeto TANGO: o código PUMA, em C, para estimar a espessura e as constantes ópticas de filmes finos, baseado em otimização irrestrita, e que usa Gradiente Espectral; o código PACKMOL, em Fortran, para criar configurações iniciais para dinâmica molecular, no qual o problema é modelado via minimização em caixas e resolvido com GENCAN; e um conjunto de modelos não lineares para empacotamento. Além disso, está disponível uma coleção de problemas restritos, de natureza geométrica e dimensão variável, denominada La Cumparsita, com todas as funções envolvidas e (pelo menos) suas derivadas de primeira ordem codificadas em Fortran77.

## 1.2 Outros trabalhos relevantes

A minimização de quadráticas sujeitas a uma restrição elipsoidal, conhecido subproblema dos algoritmos de regiões de confiança para minimização irrestrita, é um problema com uma estrutura própria. Se fatorações de Cholesky de matrizes da forma  $H + \lambda I$  forem computáveis, em que  $H$  é a Hessiana da quadrática e  $\lambda > 0$  é um escalar, então o método de Moré e Sorensen (cf. [87]) é indicado para resolver o problema, por meio do método de Newton aplicado à equação secular na variável  $\lambda$ . Em problemas de grande porte, no entanto, pode ser impraticável fatorar, e até mesmo formar a matriz  $H$ , sendo apenas possível conhecer seu efeito sobre um vetor:  $Hu = v$ . Assim, entram em cena métodos *matrix free*, ou seja, livre de fatorações de matrizes, nos quais apenas se necessita saber computar o produto da matriz por um dado vetor.

Foi nesse contexto que Rojas, Santos e Sorensen propuseram o algoritmo LSTRS, do inglês *Large Scale Trust Region Subproblem*, para resolver o subproblema de região de confiança de grande porte [102]. A estratégia adotada consiste em reformular o problema original como um problema de autovalores parametrizado e computar o valor ótimo desse parâmetro. A solução do subproblema de região de confiança é então calculada a partir dos autovetores associados a dois dos menores autovalores do problema parametrizado correspondente ao parâmetro ótimo. No contexto *matrix free* o problema de autovalores e autovetores é resolvido usando o ARPACK [77]. O algoritmo LSTRS possui um esquema interpolante que inclui naturalmente o chamado *hard case*. Em [102] são apresentadas a boa definição e a convergência superlinear do algoritmo. Experimentos computacionais ilustram tais propriedades e a robustez do método. Em [103], a implementação do algoritmo LSTRS em MATLAB 6.0 é descrita, incluindo comparações com outras técnicas e aplicações do método em problemas de regularização (ver também [101]), bem como um guia para o uso do *software* e exemplos.

O algoritmo LSTRS tem sido bastante usado para resolver problemas de quadrados mínimos lineares com restrição elipsoidal, em que a matriz dos coeficientes vem da discretização de um operador em um problema mal-posto, e o lado direito do sistema é contaminado por ruído [101, 47]. Neste caso, as técnicas de regularização são essenciais para tratar do problema e controlar o nível de ruído que estará presente na solução. Em [101], Rojas e Sorensen utilizaram LSTRS para resolver problemas do *Regularization Toolbox* do MATLAB, um problema de interpolação inversa com dados reais e um modelo de inversão sísmica, também com dados reais. Em [47], a minimização de funcionais lineares definidos como soluções de problemas discretos mal-postos é consi-

derada. Este problema aparece na determinação de intervalos de confiança para tais soluções. Um algoritmo anteriormente proposto por Eldén para este mesmo problema é revisitado, agora na versão para grande porte MLFIP, empregando LSTRS para resolver os subproblemas. Experimentos numéricos com um problema inverso de condução de calor ilustram o uso do algoritmo. Recentemente, Lampe, Rojas, Sorensen e Voss [75] propuseram a substituição do método de Lanczos com recomeços implícitos, da implementação original de LSTRS (cf. [102]), pelo método de Arnoldi não linear. Isso permite reciclar grande parte do trabalho efetuado nas iterações anteriores, e acelerar substancialmente LSTRS, conforme evidenciado pelo conjunto de testes efetuado em [75].

Voltando ao subproblema de região de confiança de grande porte, a abordagem de Steihaug–Toint [110, 113] emprega o algoritmo de gradientes conjugados para minimizar a quadrática em uma sequência de subespaços de dimensão gradativamente crescente (1, 2, etc.) até que o iterando convirja a uma solução no interior da região de confiança, ou atravesse sua fronteira. A vantagem desta abordagem é que uma solução aproximada pode ser obtida com pouco esforço computacional e baixo requerimento de memória. No entanto, não permite especificar a tolerância com que a solução restrita será calculada. Recentemente, Erway, Gill e Griffin, em [48], propuseram uma extensão do método de Steihaug–Toint na qual a solução pode ser calculada dentro de qualquer precisão pré-estabelecida. Caso o ponto de Steihaug–Toint esteja na fronteira, um problema restrito é resolvido em uma sequência subespaços de dimensão crescente. Cada subespaço inclui uma direção aceleradora obtida a partir de uma regularização do método de Newton aplicada ao problema restrito. Uma propriedade crucial desta direção é poder ser computada pela aplicação do método de gradientes conjugados a um sistema com matriz definida positiva nas variáveis primais e duais do problema restrito. O método inclui um parâmetro que permite ao usuário controlar o número total de avaliações de função e de produtos matriz-vetor correspondentes ao método de região de confiança associado. Uma solução pouco precisa é comparável ao ponto de Steihaug–Toint, enquanto uma solução bastante precisa, e que minimize o número total de avaliações de função efetuadas pode ser obtida desde que se faça uma quantidade maior de produtos matriz-vetor.

Para problemas quadráticos em que o espectro da Hessiana está restrito a um determinado intervalo positivo, Dostál propõe, em seu livro [44], algoritmos baseados em Lagrangianos aumentados com controle de precisão adaptativo (SMALBE) e número de iterações uniformemente limitado e, no caso em que a Hessiana possui representação

esparsa, proporcional ao número de variáveis do problema. Nesse livro, além de reunir os pré-requisitos e ingredientes teóricos para propor os algoritmos e apresentar seus respectivos resultados de convergência, Dostál também inclui seções com experimentos numéricos, discussões dos resultados, comentários e referências específicas ao final dos capítulos.

Kearsley [74] apresentou um algoritmo livre de fatorações de matrizes para minimizar problemas quadráticos de grande porte com restrições lineares de desigualdade e uma restrição elipsoidal. Para lidar com as restrições de desigualdade lineares, um método de barreiras é utilizado. Os subproblemas de barreira, por sua vez, são tratados via regiões de confiança. Experimentos numéricos empregando LSTRS como *solver* interno são apresentados. No levantamento bibliográfico feito, este é o único trabalho encontrado na literatura que utiliza LSTRS no contexto iterativo.

Forsgren e Murray [51] abordaram a resolução de problemas de grande porte com restrições lineares de igualdade via método de Newton (com busca linear) sem efetivamente computar a Hessiana reduzida da função objetivo. Os mesmos autores, em [52], sistematizaram suas idéias para problemas com restrições lineares de desigualdade.

Em [82], Martínez, Pilotta e Raydan propuseram um método para problemas com restrições lineares de igualdade e canalizações. As desigualdades simples são incorporadas na função objetivo por meio de uma penalização exponencial modificada, enquanto as restrições lineares de igualdade são tratadas por um método de gradiente espectral preconditionado, que pode ser visto como um método quase-Newton no qual as aproximações para as Hessianas satisfazem uma equação secante fraca. A escolha espectral do passo está embutida na aproximação para a matriz Hessiana, e o processo é monitorado por uma estratégia de busca linear não monótona. Experimentos numéricos comparativos com LANCELOT ilustram a viabilidade da proposta em uma família de problemas provenientes da localização ótima de nós em um esquema para resolver equações diferenciais ordinárias com valores de contorno.

Outro trabalho recente para problemas com restrições lineares é o artigo de Wang e Zhu [123], no qual é apresentado um método baseado na técnica afim escala de pontos interiores com busca linear curvilínea. Inclui análise teórica de convergência global, resultado de convergência local superlinear e experimentos computacionais que ilustram a viabilidade da proposta. Também de Zhu são os artigos [129] e [130]. No primeiro, o autor modifica a proposta de Bonnans e Pola [25] usando uma estratégia híbrida de região de confiança e busca linear, que adota *backtracking* quando o passo tentativo produzido pelo subproblema de região de confiança não é aceito. Convergência global

e análise de convergência local são apresentadas sob hipóteses razoáveis. Um critério não monótono é usado para acelerar a convergência em alguns casos mal condicionados. Experimentos numéricos ilustram a viabilidade das idéias propostas. No segundo artigo [130], Zhu combina a mesma abordagem de [129] com um método afim escala que emprega gradientes conjugados preconditionados para resolver o sistema reduzido. Resultados de convergência global e taxa local superlinear são apresentados.

Também na linha de algoritmos afim escala destacamos os seguintes trabalhos: Ye e Tse [128] estenderam o algoritmo para PL, proposto por Karmarkar, para a família mais geral de PQs convexos, combinando a aplicação iterativa de um aumento na função objetivo com uma transformação projetiva, seguida pela otimização restrita a um elipsóide inscrito no conjunto viável e centrado no ponto corrente. Esse esquema cria uma sequência de pontos interiores que converge à solução ótima do problema em  $\mathcal{O}(Ln)$  iterações, sendo cada iteração computada em  $\mathcal{O}(Ln^3)$  operações aritméticas, sendo  $n$  o número de variáveis e  $L$  o número de bits na entrada de dados. Os autores destacaram as propriedades de convergência do método, sua eficiência prática e como se relaciona ao método do elipsóide.

Bonnans e Bouthou [24] analisaram, tanto do ponto de vista teórico quanto numérico, o algoritmo de pontos interiores para PQ usando regiões de confiança proposto por Ye e Tse. Sob hipóteses de não degeneração, mostraram que a sequência de pontos converge para um ponto estacionário com taxa linear. Provaram ainda convergência assintótica linear para a função custo, dependente apenas da dimensão do problema. Mediante algumas modificações no algoritmo original, obtiveram comportamento numérico favorável nos experimentos efetuados.

Bonnans e Pola [25] estenderam para PNL com restrições lineares os trabalhos prévios de Ye e Tse para PQ convexa, e de Bonnans e Bouthou para PQ. Para contemporar a não linearidade dos custos, usaram uma busca linear para reduzir o passo, caso necessário. Sob hipóteses adequadas, mostraram que o algoritmo converge a um ponto estacionário de primeira ordem. As condições a serem cumpridas pelo passo completo para ser assintoticamente aceito foram analisadas.

Monteiro e Wang [86] estudaram um algoritmo afim escala usando regiões de confiança para problemas convexos com restrições lineares. Provaram convergência global para pontos estacionários de primeira ordem assumindo não degeneração primal. Para uma classe especial de funções convexas apresentaram resultado de convergência local (R-linear e Q-linear), bem como convergência global a pontos estacionários de segunda ordem.

Coleman e Li [35] propuseram o método **TRAM** para PNL (não convexa) com restrições lineares de desigualdade que combina regiões de confiança para lidar com a não linearidade dos custos e a técnica afim escala, para o conjunto viável. A função objetivo é reduzida de maneira monótona na sequência gerada pelo método, e o passo de Newton é derivado diretamente das condições de complementaridade. O subproblema de região de confiança que é formado gera, como solução assintótica, um passo de Newton aproximado. A função objetivo do modelo quadrático é a aproximação quadrática para a função objetivo original somada a um termo de aumento quadrático convexo, semelhante ao da função Lagrangiano aumentado, que adiciona curvatura positiva ao espaço gerado pelos gradientes das restrições.

Em [36], Coleman e Li estabeleceram as propriedades de convergência global e local de **TRAM**. Mostraram que as três condições de decréscimo adotadas são suficientes para garantir complementaridade, viabilidade dual e cumprimento das condições necessárias de segunda ordem, respectivamente. Assumindo válidas as condições suficientes de segunda ordem e a complementaridade estrita na solução, também provaram que o subproblema de região de confiança admite assintoticamente uma solução no interior da região de confiança, e a sequência gerada pode atingir convergência quadrática, mesmo com uma redução conveniente para produzir o passo aceitável.

Recentemente, Hager, Mair e Zhang [71] desenvolveram um algoritmo afim escala para minimização em caixas no qual cada iterando é definido pelo gradiente escalado cíclico de Barzilai-Borwein (CBB) no interior do conjunto viável. Mostraram convergência global com busca linear não monótona, e taxa local R-linear em minimizadores locais não degenerados nos quais se cumprem as condições suficientes de segunda ordem. Nos experimentos numéricos verificaram que a velocidade de convergência é insensível ao condicionamento do problema, e que o algoritmo é bastante adequado para problemas de restauração de imagens de tomografia (PET) nos quais a função custo é ilimitada na fronteira do conjunto viável.

Também para o problema de minimização em caixas, Hager e Zhang [72] propuseram o algoritmo de restrições ativas **ASA**, que consiste de um passo de gradiente projetado não monótono, um passo de minimização irrestrita e um conjunto de regras para combinar estes dois passos. Provaram a convergência global a um ponto estacionário. Mostraram que o algoritmo se reduz à minimização irrestrita sem recomeços, tanto para pontos estacionários não degenerados, quanto para pontos estacionários degenerados nos quais valha uma condição suficiente de segunda ordem forte. Descrevem em [72] uma implementação específica de **ASA** que explora o algoritmo de Barzilai-Borwein

cíclico (CBB) para o passo de gradiente projetado, e o algoritmo CG DESCENT para a minimização irrestrita. Inclui experimentos numéricos com problemas das coleções CUTer e MINPACK-2.

Ainda para minimização em caixas, mas na linha da otimização sem derivadas, Gratton, Toint e Tröltzsch [69] combinaram restrições ativas e regiões de confiança, com o objetivo de explorar eficientemente a geometria do problema na construção dos conjuntos de interpolação necessários para determinar os modelos. Os resultados numéricos apresentados foram competitivos com os algoritmos BOBYQA e NEWUOA de Powell [99, 100].

Para PNL geral, Paul Tseng [114] propôs um método de pontos interiores e região de confiança primal-dual infactível baseado na função barreira logarítmica para as variáveis de folga, que são atualizadas usando uma correção de segunda ordem. Sob hipóteses adequadas, provou convergência global a pontos estacionários de primeira e segunda ordens. Apresentou resultados numéricos preliminares e discutiu uma possível extensão da proposta para PNL semidefinida.

### 1.3 Contribuições desta tese

O objetivo deste trabalho é minimizar uma função não linear suave em um polítopo. Como estratégia de convergência global optamos pela técnica de regiões de confiança. Neste sentido, visando a solução de problemas de grande porte, o algoritmo adotado para a solução (aproximada) dos subproblemas é o proposto por Rojas, Santos e Sorensen LSTRS, que é livre de fatorações, baseado apenas no produto matriz-vetor da Hessiana do modelo quadrático.

Nosso ponto de partida foi a estratégia de restrições ativas de Gay [55], que resolveu os subproblemas usando *dogleg* (cf. [42, Sec.6.4.2], e só apresentou resultados numéricos para problemas com restrições de canalização. Tendo como alvo problemas de grande porte, optamos pela resolução (aproximada) do subproblema de região de confiança reduzido à face corrente usando o algoritmo LSTRS. Para problemas com restrições poliedrais gerais, embora estejam disponíveis técnicas para atualizar a matriz cujas colunas geram o núcleo da matriz associada às restrições ativas correntes (ver, por exemplo [76] ou ainda [56] e [89]), este enfoque não é compatível com a solução de problemas efetivamente de grande porte na filosofia livre de fatorações. Isto por que a obtenção numericamente estável de tal matriz é feita por meio de uma fatoração  $QR$

da transposta da matriz associada às restrições ativas correntes. De qualquer maneira, para problemas com restrições suficientemente estruturadas esta abordagem é implementável na linha livre de fatorações, como foi feito neste trabalho. Consideramos uma família de problemas de pontos em polígonos, inspirada por um problema de pontos no quadrado sugerido por Powell (cf. [98, 99]), com a qual obtivemos resultados promissores utilizando uma estratégia de restrições ativas, baseada em uma modificação do algoritmo de Rosen para o abandono da face corrente. Em nosso caso, como lidar com um número grande de variáveis não é um impeditivo, e queremos efetivamente trabalhar com as derivadas segundas da função objetivo, optamos por uma abordagem alternativa, que se ajusta perfeitamente a tais características: acrescentamos variáveis de folga não negativas às restrições originais, introduzimos um termo com a barreira logarítmica para tais variáveis e resolvemos uma sequência de problemas com restrições lineares de igualdade. Supondo conhecido um ponto inicial estritamente viável para o problema original, teremos um ponto inicial com as variáveis de folga estritamente positivas. Para o problema dos pontos no polígono tal inicialização é perfeitamente factível. Com essa abordagem, não só asseguramos a convergência teórica do algoritmo, como também obtivemos resultados computacionais comparáveis para a família de problemas estruturados escolhida.

## 1.4 Organização do texto

No Capítulo 2 resumimos os principais elementos que integrarão o corpo dos algoritmos propostos. Inicialmente explicitamos a notação e o formato para o problema abordado, com as hipóteses básicas e as condições de otimalidade. A seguir detalhamos aspectos relativos à obtenção de uma base ortogonal para o núcleo da Jacobiana das restrições, e encerramos com uma breve apresentação do algoritmo de Rojas, Santos e Sorensen, para minimizar uma quadrática sujeita a uma restrição elipsoidal.

No Capítulo 3 descrevemos um algoritmo de restrições ativas e regiões de confiança para o problema de interesse, nos moldes da proposta feita em [55], acompanhado dos resultados de convergência global e local.

No Capítulo 4 apresentamos um algoritmo de barreira e regiões de confiança para resolver o problema restrito a um politopo, juntamente com as propriedades teóricas de convergência global.

Descrevemos a família de problemas dos pontos em um polígono no Capítulo 5,

detalhando os elementos utilizados na implementação livre de fatorações empregada na solução dos subproblemas em ambos algoritmos.

Apresentamos e analisamos os experimentos computacionais no Capítulo 6, que começa com uma breve análise do desempenho do algoritmo **LSTRS** para problemas quadráticos indefinidos com restrições lineares não estruturadas em uma bola. Em seguida, apresentamos os detalhes da implementação desenvolvida para o problema da distribuição de pontos em um polígono, e um conjunto de testes preliminares. Nesses primeiros testes calibramos e ajustamos alguns elementos essenciais, tanto do problema e da escolha dos pontos iniciais, quanto de parâmetros algorítmicos. Finalmente, apresentamos um novo conjunto de testes, levando em conta todos os ajustes efetuados. Finalizamos o texto com nossas conclusões e as questões que ficaram em aberto no desenvolvimento deste trabalho.



# Capítulo 2

## Ferramentas e ingredientes básicos

### 2.1 Problema e condições de otimalidade

O problema a ser abordado neste trabalho é um problema de minimização com restrições lineares que será representado da seguinte maneira:

$$\begin{aligned} &\text{minimizar} && f(x) \\ &\text{suj. a} && Ax \geq b \end{aligned} \tag{2.1}$$

onde  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  é uma função suave,  $x \in \mathbb{R}^n$ ,  $A \in \mathbb{R}^{p \times n}$  e  $b \in \mathbb{R}^p$ .

Este tipo de problema aparece com frequência na literatura e pode ser representado de diversas formas. As hipóteses específicas sobre a função  $f$  e sobre o conjunto viável serão assumidas juntamente com os resultados de convergência para os algoritmos propostos.

Nesta notação incorporamos as restrições de caixas  $l \leq x \leq u$  às restrições do tipo  $Ax \geq b$  para facilitar o desenvolvimento e apresentação da teoria. Porém, podemos perceber que em alguns casos podemos aproveitar a estrutura especial das restrições do tipo caixa para a implementação computacional. Na resolução do problema pelo método de restrições ativas proposto, a cada iteração poderíamos fixar as restrições ativas do tipo caixa, reformulando e reduzindo a dimensão do subproblema, o que não será detalhado neste trabalho, entretanto.

Vale também ressaltar que as restrições gerais  $Ax \geq b$  incluem restrições de igualdade que serão consideradas sempre ativas durante as iterações. Para a implementação computacional, as primeiras linhas da matriz  $A$  correspondem às restrições de igualdade e seus índices são tratados diferenciadamente dos demais índices das restrições ativas,

levando em consideração que nunca abandonam o conjunto ativo. Porém, por simplicidade de notação, não diferenciamos as restrições de igualdade das demais restrições ativas no texto.

Para o problema (2.1) temos as condições necessárias de primeira ordem (Karush-Kuhn-Tucker) dadas a seguir.

O conjunto viável do problema (2.1) é qualificado por ser linear: as direções viáveis a partir de um dado ponto são perfeitamente caracterizadas pelo núcleo do Jacobiano das restrições ativas neste ponto. Como consequência, as condições necessárias de otimalidade serão sempre satisfeitas em minimizadores locais. Temos assim garantida a existência de multiplicadores de Lagrange associados a cada conjunto de restrições e correspondentes a um dado ponto viável.

Seja  $x^*$  um minimizador local de (2.1). Então existem os multiplicadores de Lagrange  $\mu^* \in \mathbb{R}^p$  tais que:

$$Ax^* \geq b \tag{2.2}$$

$$\nabla f(x^*) = A^T \mu^* \tag{2.3}$$

$$(b - Ax^*)^T \mu^* = 0, \quad \mu^* \geq 0 \tag{2.4}$$

Os algoritmos propostos neste trabalho visam a obtenção de uma aproximação para um minimizador  $x^*$ .

Antes de detalharmos os algoritmos propostos nos capítulos 3 e 4, apresentamos nas próximas seções alguns elementos importantes para o desenvolvimento destes algoritmos.

## 2.2 Fatoração $QR$ e base para o núcleo

Visando combinar uma opção numericamente estável e que não interfira na geometria das regiões de confiança, por não introduzir um re-escalamento da bola original, optamos pela fatoração  $QR$  e não pela fatoração  $LU$  para determinar as matrizes cujas colunas constituem uma base para o subespaço gerado pelos gradientes das restrições ativas no ponto corrente para os dois algoritmos desenvolvidos nesta tese. A implementação

desta escolha se mostrou compatível com a estrutura de esparsidade dos problemas-teste implementados neste trabalho, conforme detalhado no Capítulo 5, Seções 5.1.1 e 5.2.1.

Denotamos por  $Z$  a matriz cujas colunas geram o núcleo de uma matriz  $M$  ( $Im(Z) = Nu(M)$ ). Usando uma fatoração  $QR$  da matriz transposta  $M^T$ , podemos obter a matriz  $Z$  como descrito em maiores detalhes a seguir. Este estudo foi baseado no trabalho de Gill e Murray (1974) [56] bem como em Murtagh e Saunders [89].

Seja  $M \in \mathbb{R}^{m \times q}$ , com  $q \geq m$  e suponhamos que  $M$  tenha posto completo.

Podemos então construir uma matriz  $T \in \mathbb{R}^{q \times q}$  dada por

$$T = \begin{pmatrix} M \\ V^T \end{pmatrix},$$

onde  $V^T \in \mathbb{R}^{(q-m) \times q}$  é uma matriz arbitrária cujas linhas são linearmente independentes em relação às linhas de  $M$ .

Com esta construção, temos o teorema a seguir:

**Teorema 1** Seja  $t_j$  a  $j$ -ésima coluna de  $T^{-1}$ , então  $Mt_j = 0$  para  $j = m + 1, \dots, q$

**Prova.** Pela construção das colunas da matriz inversa  $T^{-1}$ , temos que

$$\begin{aligned} T \begin{pmatrix} t_1 & \dots & t_q \end{pmatrix} &= \begin{pmatrix} M \\ V^T \end{pmatrix} \begin{pmatrix} t_1 & \dots & t_q \end{pmatrix} = I \\ \Rightarrow \begin{pmatrix} Mt_1 & \dots & Mt_m & Mt_{m+1} & \dots & Mt_q \\ V^T t_1 & \dots & V^T t_m & V^T t_{m+1} & \dots & V^T t_q \end{pmatrix} &= I. \end{aligned}$$

Portanto  $Mt_{m+1} = \dots = Mt_q = 0$ . ■

Com este teorema vemos que a matriz  $Z$  pode ser obtida por meio das últimas  $q - m$  colunas da matriz inversa  $T^{-1}$ . A matriz  $T^{-1}$  não é explicitamente computada. O processo para o cálculo de  $Z$  é mostrado em detalhes nesta seção.

Antes, porém, destacamos que para cada escolha da matriz  $V^T$ , um método diferente para a obtenção de  $Z$  é caracterizado. Podemos citar alguns métodos clássicos apresentados na literatura.

Wolfe (1967) [125] sugere a utilização de restrições selecionadas do conjunto de restrições inativas para definir a matriz  $V^T$ . Com isso, a cada iteração, a matriz  $V^T$  é

pouco alterada.

Em 1970, McCormick [84] propõe a escolha de  $V^T$  baseada nas linhas da matriz identidade, reordenando as variáveis. Assim:

$$T = \begin{pmatrix} M & \\ 0 & I_{q-m} \end{pmatrix} = \begin{pmatrix} B & D \\ 0 & I_{q-m} \end{pmatrix},$$

e com isso

$$T^{-1} = \begin{pmatrix} B^{-1} & -B^{-1}D \\ 0 & I_{q-m} \end{pmatrix} \quad \text{e} \quad Z = \begin{pmatrix} -B^{-1}D \\ I_{q-m} \end{pmatrix}.$$

Esta técnica é utilizada especialmente na resolução de problemas de programação linear por meio de métodos clássicos para os quais as variáveis são reordenadas utilizando a nomenclatura *variáveis básicas* e *variáveis não básicas*, associadas às matrizes  $B$  e  $D$ .

Em problemas de programação não linear, uma escolha ruim de  $V^T$  pode gerar uma matriz  $T$  mal-condicionada e, conseqüentemente, o mau condicionamento de  $Z$ . Com isso, o valor da norma do gradiente reduzido ( $Z^T \nabla f(x)$ ) poderia ser grande mesmo que  $x$  estivesse próximo a um ponto estacionário.

Para evitar este problema, utilizaremos a técnica apresentada por Gill e Murray [56] para obter as informações da matriz  $T^{-1}$  e para a construir  $V^T$ .

Seja a fatoração  $LQ$  da matriz  $T$ :

$$T = LQ,$$

onde  $L$  é uma matriz triangular inferior e  $Q$  uma matriz ortogonal.

Particionando  $Q$  temos:

$$Q = \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} \quad \text{com } Q_1 \in \mathbb{R}^{m \times q} \text{ e } Q_2 \in \mathbb{R}^{(q-m) \times q}$$

Pós multiplicando  $T$  por  $Q_2^T$  temos que

$$TQ_2^T = L \begin{pmatrix} 0 \\ I_{q-m} \end{pmatrix},$$

o que implica  $MQ_2^T = 0$ .

Assim, uma possível escolha para  $Z$  é dada por  $Q_2^T$ , sem a necessidade de computar  $T^{-1}$ . Além disso, é possível mostrar que, com uma escolha adequada de  $V$ , não é preciso computar a fatoração completa da matriz  $T$ . Basta realizarmos a fatoração da matriz  $M$ , como mostramos a seguir.

Pela construção da matriz  $T$ , devemos escolher  $V^T$  de tal forma que suas linhas sejam linearmente independentes em relação às linhas de  $M$  (ou às colunas de  $M^T$ ). Como  $\text{Im}(M^T)$  é ortogonal ao  $\text{Nu}(M)$ , poderíamos tomar  $V = Z$ .

Assumindo  $V = Z$  e sabendo que  $Q_2$  pode ser expressa como  $Q_2 = Z^T$  temos, pela fatoração  $LQ$  de  $T$  que

$$T = \begin{pmatrix} M \\ Z^T \end{pmatrix} = L \begin{pmatrix} Q_1 \\ Z^T \end{pmatrix}.$$

Assim,

$$L = \begin{pmatrix} \bar{L} & 0 \\ 0 & I_{q-m} \end{pmatrix}$$

onde a fatoração  $LQ$  de  $M$  (ou, analogamente, a fatoração  $QR$  de  $M^T$ ) é dada por

$$M = \begin{pmatrix} \bar{L} & 0 \end{pmatrix} \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix}$$

com  $Z = Q_2^T$ .

Com este procedimento, não há problemas de mau condicionamento, tendo em vista que  $Z^T Z = I$ .

## 2.3 Algoritmo de Rojas, Santos e Sorensen

O algoritmo de Rojas, Santos e Sorensen [102] é utilizado para a resolução de problemas da forma

$$\begin{aligned} \min \quad & \psi(s) = g^T s + \frac{1}{2} s^T H s \\ \text{s. a} \quad & \|s\|_2 \leq \Delta \end{aligned} \tag{2.5}$$

onde  $g \in \mathbb{R}^n$ ,  $H \in \mathbb{R}^{n \times n}$  simétrica,  $\Delta > 0$ . Para simplificar a notação denotaremos ao longo deste texto a norma Euclidiana por  $\|\cdot\|_2 = \|\cdot\|$ .

Um resultado importante para este tipo de problema e que caracteriza completamente uma solução de (2.5) está enunciado a seguir, cuja prova pode ser encontrada em [109]:

**Lema 1** Um vetor viável  $s^*$  é solução de (2.5) com correspondente multiplicador de Lagrange  $\lambda^*$  se e somente se  $s^*$  e  $\lambda^*$  satisfazem  $(H - \lambda^* I)s = -g$  com  $H - \lambda^* I$  semidefinida positiva,  $\lambda^* \leq 0$  e  $\lambda^*(\Delta - \|s\|) = 0$ .

O algoritmo de Rojas, Santos e Sorensen é fortemente apoiado no Lema 1. A principal idéia deste algoritmo é reescrever o subproblema de região de confiança como um problema de autovalor parametrizado, encontrando um valor ótimo para o parâmetro com o auxílio de um esquema de interpolação. A solução do subproblema estará relacionada então aos autovetores associados aos menores autovalores de um problema de autovalor com o parâmetro ótimo.

Uma característica importante deste método é a utilização apenas de produtos de matriz por vetor, não sendo necessário o armazenamento explícito das matrizes envolvidas no processo.

Para deduzir o método proposto por Rojas, Santos e Sorensen primeiro consideramos a matriz  $B_\alpha = \begin{pmatrix} \alpha & g^T \\ g & H \end{pmatrix}$ .

Supondo  $\lambda$  autovalor de  $B_\alpha$  com autovetor associado  $(1, s^T)^T$  temos que

$$\alpha - \lambda = -g^T s \quad \text{e} \quad (H - \lambda I)s = -g.$$

Observando o Lema 1 vemos que, se  $H - \lambda I$  é semidefinida positiva,  $s$  pode ser um candidato à solução do subproblema (2.5), dependendo do sinal de  $\lambda$  e do produto  $\lambda(\Delta - \|s\|)$  ser zero.

Porém, pela propriedade de entrelaçamento de Cauchy (cf. [64], pag. 396) temos que, se  $\delta_1 \leq \dots \leq \delta_n$  são os autovalores de  $H$  e  $\lambda_1(\alpha) \leq \dots \leq \lambda_{n+1}(\alpha)$  são autovalores de  $B_\alpha$  então

$$\lambda_1(\alpha) \leq \delta_1 \leq \lambda_2(\alpha) \leq \dots \leq \lambda_n(\alpha) \leq \delta_n \leq \lambda_{n+1}(\alpha).$$

Assim  $\lambda_1(\alpha) \leq \delta_1$ , ou seja, os autovalores da matriz  $H - \lambda_1(\alpha)I$  são sempre maiores ou iguais a zero, fazendo com que esta matriz seja semidefinida positiva, independentemente do valor de  $\alpha$ .

Então, definindo

$$\phi(\lambda) = g^T (H - \lambda I)^\dagger g = -g^T s$$

temos

$$\phi'(\lambda) = g^T ((H - \lambda I)^\dagger)^2 g = s^T s,$$

onde  $\dagger$  denota a pseudoinversa da matriz. A idéia do algoritmo é ajustar  $\alpha$  para que  $\phi'(\lambda) = s^T s = \Delta^2$  e então encontrar o menor autovalor de  $B_\alpha$ , normalizando a primeira componente do autovetor correspondente e obtendo  $s$  através das demais componentes.

Se  $\lambda_1(\alpha) \leq 0$ , então  $s$  será uma solução para o subproblema (2.5).

Para estimar o menor autovalor de  $B_\alpha$  e um de seus autovetores é utilizado o método de Arnoldi/Lanczos, implementado na biblioteca ARPACK [77].

Uma situação que pode ocorrer é o fato dos autovetores associados ao menor autovalor de  $B_\alpha$  terem primeira componente sempre nula. Neste caso podemos não ter solução para  $\phi'(\lambda) = \Delta^2$  com  $\lambda \leq \delta_1$ , o que caracteriza o *hard case*.

No algoritmo, este caso é detectado através de salvaguardas baseadas em um conjunto de lemas e teoremas, que podem ser encontrados com mais detalhes em Rojas, Santos e Sorensen [102].



# Capítulo 3

## Algoritmo de restrições ativas e propriedades teóricas

Para resolver o problema (2.1) utilizamos um método local de restrições ativas com subproblemas de região de confiança apresentado nas próximas seções, cuja construção é baseada no trabalho de Gay [55] combinado com o algoritmo para solução dos subproblemas de Rojas, Santos e Sorensen.

### 3.1 Modelo Algorítmico

Em [55], Gay propôs um algoritmo com regiões de confiança e uma estratégia local de restrições ativas para resolver problemas de minimização com restrições lineares, desenvolvido para problemas suficientemente pequenos, aos quais são aplicáveis técnicas de matrizes densas e o trabalho explícito com a matriz Hessiana da função objetivo.

A seguir apresentamos um resumo do modelo algorítmico utilizado, incluindo as adaptações para o nosso trabalho. O detalhamento dos principais passos do algoritmo é apresentado nas próximas seções.

#### Modelo Algorítmico Externo

Dados  $\alpha \in (0, 1)$ , o raio da região de confiança inicial  $\Delta_0$  e tomando  $\ell = 0$ ,

1. Obter um ponto inicial viável  $x^0$  para (2.1).
2. Computar uma estimativa para os multiplicadores de Lagrange  $\mu^\ell$ ;

3. Enquanto os critérios de convergência não forem satisfeitos,

(a) Obter  $s^\ell$ , aproximação para a solução do subproblema

$$\begin{aligned} \min \quad & q(s) = \frac{1}{2}s^T \nabla^2 f(x^\ell)s + s^T \nabla f(x^\ell) \\ \text{s. a} \quad & s \in B(\Delta_\ell, x^\ell) \end{aligned}, \quad (3.1)$$

onde  $q(s)$  é um modelo quadrático que aproxima o decréscimo da função objetivo de (2.1) em torno de  $x^\ell$  e

$$B(\Delta_\ell, x^\ell) = \{s \in \mathbb{R}^n \mid \|s\| \leq \Delta_\ell, As \geq b - Ax^\ell\}.$$

(b) Comparar o decréscimo do modelo e da função objetivo:

$$\text{Se } f(x^\ell + s^\ell) - f(x^\ell) \leq \alpha q(s^\ell) \text{ então } x^{\ell+1} = x^\ell + s^\ell,$$

$$\text{senão } x^{\ell+1} = x^\ell.$$

(c) Atualizar  $\Delta_\ell$ .

(d)  $\ell = \ell + 1$ .

### 3.1.1 Ponto inicial viável

No passo 1 do algoritmo, é necessário obter um ponto viável  $x^0$ . Quando não se conhece inicialmente tal ponto, é usual trabalhar com um problema mais simples, cuja solução será um ponto inicial desejado. Este processo é conhecido em programação linear como *Fase I*.

O primeiro passo para a construção deste problema é reescrever as restrições originais na forma padrão, com apenas restrições de igualdade e caixa, acrescentando variáveis de folga  $u$  para as restrições do tipo  $Ax \geq b$ .

Assim temos as restrições reescritas da seguinte forma:

$$\begin{aligned} Ax - u = b \\ 0 \leq u \end{aligned} \quad \rightarrow \quad \begin{aligned} \tilde{A}\tilde{x} = b \\ \tilde{l} \leq \tilde{x} \end{aligned} \quad (3.2)$$

onde  $\tilde{A} = \begin{pmatrix} A & -I \end{pmatrix}$ ,  $\tilde{x} = \begin{pmatrix} x \\ u \end{pmatrix}$  e  $\tilde{l} = \begin{pmatrix} -\infty \\ 0 \end{pmatrix}$ .

Para sistematizar a apresentação admitiremos limitantes infinitos nas componentes da caixa. Computacionalmente, trabalharemos com valores suficientemente grandes, de acordo com a escala dos problemas-teste.

Para realizar a *Fase I* acrescentamos variáveis artificiais  $z$  ao conjunto de restrições na forma padrão e resolvemos o seguinte problema artificial:

$$\begin{aligned} \min \quad & \frac{1}{2}e^T z \\ \text{s. a} \quad & \tilde{A}\tilde{x} + z = b \\ & l \leq \tilde{x} \\ & z \geq 0 \end{aligned} \tag{3.3}$$

onde  $e = (1, 1, \dots, 1)^T \in \mathbb{R}^p$ .

Se o problema original (2.1) admite pontos viáveis, ao resolver o problema (3.3), obteremos como solução  $z^* = 0$  e  $x^*$  extraído de  $\tilde{x}^* = \begin{pmatrix} x^* \\ u^* \end{pmatrix}$  será utilizado como ponto inicial viável para o problema original.

Para inicializar o algoritmo que resolve o problema artificial (3.3), adotamos o ponto inicial  $(\tilde{x}^0, z^0)$  preparando  $b$  da seguinte maneira:

$$\tilde{x}^0 = 0$$

e

$$z_i^0 = b_i - \tilde{a}_i^T \tilde{x}^0, \quad \text{com } b_i - \tilde{a}_i^T \tilde{x}^0 \geq 0$$

onde  $\tilde{a}_i^T$  é a  $i$ -ésima linha de  $\tilde{A}$ .

Caso  $b_i - \tilde{a}_i^T \tilde{x}^0 < 0$  para algum índice  $i$  em (3.2), fazemos um pré-processamento multiplicando a  $i$ -ésima linha de  $\tilde{A}$  e a respectiva componente de  $b$  por  $-1$  antes de resolver o problema artificial, para garantir que  $z^0 \geq 0$ .

Uma observação importante é o fato das variáveis de folga e artificiais serem introduzidas apenas nesta etapa. Após a obtenção do ponto inicial, retomamos o problema (2.1) com as variáveis originais.

### 3.1.2 Estimativa dos multiplicadores

Seja o vetor dos multiplicadores de Lagrange, correspondente ao ponto  $x^\ell$ , denotado por  $\mu_{\mathcal{A}}^\ell$ .

A cada iteração do algoritmo, precisamos de uma estimativa destes multiplicadores utilizada na escolha das restrições a serem consideradas ativas na resolução dos subproblemas e também no critério de parada baseado nas condições de otimalidade de (2.1).

Para obter uma aproximação para  $\mu_{\mathcal{A}}^\ell$ , utilizamos as relações (2.2)-(2.4) baseadas nas condições de otimalidade, estimadas em  $x^\ell$ .

Pela condição de complementaridade (2.4), podemos estimar parte dos multiplicadores de Lagrange tomando iguais a zero as componentes de  $\mu_{\mathcal{A}}^\ell$  relativas às restrições não ativas em  $x^\ell$ .

Para obter uma estimativa dos multiplicadores correspondentes às restrições ativas, temos da relação (2.3) aplicada em  $x^\ell$ :

$$\nabla f(x^\ell) = A^T \mu_{\mathcal{A}}^\ell. \quad (3.4)$$

Desta maneira podemos obter uma estimativa para os multiplicadores de Lagrange resolvendo a relação

$$A_{\mathcal{I}_{\mathcal{A}}}^T \mu_{\mathcal{I}_{\mathcal{A}}}^\ell = \nabla f(x^\ell), \quad (3.5)$$

onde  $A_{\mathcal{I}_{\mathcal{A}}}$  é a submatriz de  $A$  correspondente às linhas cujos índices estão em  $\mathcal{I}_{\mathcal{A}}$ , o conjunto de índices das restrições ativas em  $x^\ell$ .

Como veremos com maiores detalhes nas próximas seções, a cada iteração temos informações sobre a fatoração  $QR$  desta submatriz,  $A_{\mathcal{I}_{\mathcal{A}}}^T = QR$ .

Então da relação (3.5), temos o sistema linear para a estimativa de  $\mu_{\mathcal{A}}^\ell$ :

$$R \mu_{\mathcal{I}_{\mathcal{A}}}^\ell = Q^T \nabla f(x^\ell), \quad (3.6)$$

que pode ser resolvido de forma simples pelo fato da matriz  $R$  ser triangular superior.

### 3.1.3 Resolução do subproblema

Para a resolução do subproblema do passo 3(a) do modelo algorítmico utilizamos o procedimento sintetizado a seguir, cujos passos serão detalhados posteriormente nesta subseção. Os índices  $\ell$  e  $k$  indicarão, respectivamente, a iteração externa para resolução do problema (2.1) e a iteração interna para a resolução do subproblema (3.1).

#### Modelo Algorítmico Interno

1. Iniciamos o processo com  $k = 1$  e tomamos  $s^{\ell,0} = 0$  se a norma do gradiente projetado  $\|g_p\|$  é suficientemente grande. Caso contrário,  $s^{\ell,0} = \zeta s^0$ , definido pelo escalamento da solução de um problema de Cauchy.
2. Seja  $x^c \equiv x^\ell + s^{\ell,k-1}$  o ponto corrente e  $\mathcal{I}(x^c)$  o conjunto de índices das restrições ativas em  $x^c$ . Se  $k = 1$ , escolhemos  $\mathcal{I}^1 \subset \mathcal{I}(x^c)$ . Caso contrário,  $\mathcal{I}^k \equiv \mathcal{I}(x^c)$ .
3. Seja  $A_{\mathcal{I}^k}$  a matriz correspondente às restrições com índices em  $\mathcal{I}^k$ . Determinamos a matriz  $Z \equiv Z_k$  tal que  $Im(Z_k) = Nu(A_{\mathcal{I}^k})$ .
4. A fim de obter uma aproximação para a solução do subproblema de restrições ativas

$$\begin{aligned} \min \quad & f(x) \\ \text{s. a} \quad & A_{\mathcal{I}^k} x = b_{\mathcal{I}^k} \end{aligned} \tag{3.7}$$

computamos  $\bar{w}$  solução do subproblema reduzido com região de confiança:

$$\begin{aligned} \min \quad & \bar{g}^T Z w + \frac{1}{2} w^T Z^T H Z w \\ \text{s. a} \quad & \|w\| \leq \Delta_\ell \end{aligned} \tag{3.8}$$

onde o vetor gradiente  $\bar{g}$  e a matriz Hessiana  $H$  usados para definir esta função objetivo quadrática serão detalhados adiante.

5. Retornamos para as variáveis originais do subproblema com  $Z\bar{w} = s^{\ell,k}$ .
6. Se  $s^{\ell,k}$  é viável para (3.1), então tomamos  $s^\ell = s^{\ell,k}$ . Caso contrário recalculamos  $s^{\ell,k}$  em  $B(\Delta_\ell, x^\ell)$ , tomamos  $k = k + 1$  e voltamos para o passo 2.

A seguir detalhamos os principais passos deste processo.

### O problema de Cauchy do Passo 1

As principais alterações do conjunto ativo ocorrem na iteração externa e na escolha de  $\mathcal{I}^1$ , já que, para  $k > 1$ , a definição do conjunto  $\mathcal{I}^k$  no passo 2 deste modelo algorítmico é feita de tal forma que  $\mathcal{I}^k \subset \mathcal{I}^{k+1}$ . Em outras palavras, com exceção da primeira, a cada iteração interna apenas novas restrições serão agregadas ao conjunto ativo, sem abandono das restrições que já estavam ativas. O abandono da face é uma etapa delicada dos métodos de restrições ativas, que merece uma estratégia cuidadosamente desenvolvida. Como para problemas com restrições lineares gerais a direção de Newton pode ser inviável, ou até mesmo de subida, não serve de base para a decisão sobre o abandono da face. A direção de máxima descida projetada, por sua vez, é uma opção seguramente de descida e viável.

Em nosso caso, utilizamos a matriz ortogonal  $Z$  tal que  $Im(Z) = Nu(A_{\mathcal{I}^0})$ , sendo  $\mathcal{I}^0$  o conjunto das restrições ativas no ponto corrente externo  $x^\ell$ , para definir o primeiro conjunto de índices. Assim, quando  $k = 1$ , uma vez determinada a matriz  $Z$ , pelo fato de suas colunas serem vetores ortonormais, calculamos a projeção do gradiente diretamente por meio da expressão

$$g_p^\ell \equiv ZZ^T \nabla f(x^\ell).$$

Se  $\|g_p^\ell\| \geq \theta_0 \|\nabla f(x^\ell)\|$ , com  $\theta_0 \in (0, 1]$ , mantemos  $s^{\ell,0} = 0$ , pois vale a pena permanecer na face corrente. Caso contrário, a relação indica que a face atual deve ser abandonada. Para tanto, redefinimos  $s^{\ell,0}$  como um passo de Cauchy escalado, isto é,  $s^{\ell,0} \equiv \zeta s^0$  em que  $s^0$  é a solução do problema de Cauchy modificado, cuja resolução é detalhada nesta seção:

$$\begin{aligned} & \text{minimizar} && (g^{\ell T})s \\ & \text{suj. a} && \|s\| \leq 1 \\ & && A_{\mathcal{I}^0}s \geq 0 \end{aligned} \tag{3.9}$$

onde  $g^\ell = \nabla f(x^\ell)$  e, dadas as constantes  $\theta_1, \theta_2 \in (0, 1]$ , o escalar  $\zeta$  é calculado da seguinte maneira:

$$\zeta = \min \left\{ \theta_1 \Delta_\ell, \max \{ \tau > 0 \mid \tau A s^0 \geq b - A x^\ell \}, \frac{-\theta_2 (s^0)^T \nabla f(x^\ell)}{(s^0)^T H s^0} \right\}. \tag{3.10}$$

Na expressão (3.10), o primeiro termo garante que  $x^\ell + \zeta s^0$  permanecerá na região de confiança corrente, o segundo termo assegura a manutenção de  $x^\ell + \zeta s^0$  no conjunto viável, e o terceiro termo corresponde a uma fração do passo da busca linear exata ao longo da direção  $s^0$ , e que deve ser substituído por  $+\infty$  se  $(s^0)^T H s^0 \leq 0$ . Com isso, redefinimos o ponto corrente como  $x^c \equiv x^\ell + \zeta s^0$ .

Para a resolução do problema de Cauchy modificado (3.9), primeiro observamos que trata-se de um problema convexo cujo ponto estacionário  $s^0$  é um minimizador global.

Reescrevendo a restrição  $\|s\| \leq 1$  de maneira equivalente e diferenciável  $s^T s \leq 1$ , as condições KKT de (3.9) são:

$$g^\ell + 2\lambda s - A_{\mathcal{I}^0}^T \mu = 0 \quad (3.11)$$

$$\lambda(s^T s - 1) = 0, \quad \lambda \geq 0, \quad s^T s \leq 1 \quad (3.12)$$

$$\mu_j (A_{\mathcal{I}^0} s)_j = 0, \quad j = 1, \dots, n_0 \quad (3.13)$$

$$\mu \geq 0, \quad A_{\mathcal{I}^0} s \geq 0. \quad (3.14)$$

em que  $n_0$  é a cardinalidade de  $\mathcal{I}^0$ ,  $\lambda \in \mathbb{R}$  e  $\mu \in \mathbb{R}^{n_0}$ .

Supondo  $\bar{s}$  tal que  $\|\bar{s}\| = 1$  e  $\lambda, \mu$  os multiplicadores associados que cumprem (3.11)-(3.14), ao pré-multiplicarmos (3.11) por  $(\bar{s})^T$ , por (3.13) obtemos

$$\lambda = \frac{-(\bar{s})^T g^\ell}{2}. \quad (3.15)$$

Assumindo que  $A_{\mathcal{I}^0}^T = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R \\ 0 \end{pmatrix}$ , e substituindo (3.15) em (3.11), obtemos

$$g^\ell - (\bar{s})^T g^\ell \bar{s} = QR\mu$$

e então os multiplicadores  $\mu$  são estimados resolvendo o sistema triangular

$$R\mu = Q_1^T (g^\ell - (\bar{s})^T g^\ell \bar{s}). \quad (3.16)$$

Para obter  $s^0$ , solução do problema (3.9), vamos utilizar, conforme sugerido por Gay [55], uma modificação do algoritmo do gradiente projetado de Rosen [104], de forma a incorporar a restrição quadrática  $s^T s \leq 1$ .

Dados um conjunto de índices  $\mathcal{I}_j$  e uma aproximação  $s^j$  tal que  $A_{\mathcal{I}_j}s^j = 0$ , o algoritmo calcula  $\mathcal{I}_{j+1}$  e  $s^{j+1}$ , uma melhor aproximação para  $s^0$ :

### Algoritmo de Rosen modificado para resolver (3.9)

Dados  $j = 1$ ,  $\mathcal{I}_1 = \{i \in \mathcal{I}^0 \mid \mu_i^\ell > 0\}$  e  $s^1 \equiv 0$ ,

1. Computar  $Z_j$  tal que  $Im(Z_j) = Nu(A_{\mathcal{I}_j})$ .

2. Computar

$$\bar{s}^j = \begin{cases} 0, & \text{se } Z_j \equiv 0 \text{ ou } Z_j^T g^\ell = 0 \\ \frac{-Z_j Z_j^T g^\ell}{\| -Z_j Z_j^T g^\ell \|} & \text{caso contrário.} \end{cases}$$

3. Se  $\bar{s}^j$  é tal que  $A_{\mathcal{I}^0} \bar{s}^j \geq 0$ , ou seja,  $\bar{s}^j$  é viável para (3.9), então vá para o Passo 4.

Caso contrário fazer

$$s^{j+1} = \theta \bar{s}^j + (1 - \theta) s^j,$$

com  $\theta \in [0, 1)$  o maior possível tal que  $A_{\mathcal{I}^0} s^{j+1} \geq 0$ . E então  $\mathcal{I}_{j+1} = \mathcal{I}_j \cup \{i\}$ , sendo  $i$  o índice cuja  $i$ -ésima restrição produziu o escalar  $\theta$ .

4. Definir  $s^{j+1} = \bar{s}^j$  e verificar a otimalidade:

- Computar  $\mu$  resolvendo o sistema linear (3.16) com  $\bar{s} \equiv \bar{s}^j$ .
- Se  $\mu \geq 0$ , então  $s^0 = \bar{s}^j$  e a solução para o problema (3.9) foi encontrada.  
Caso contrário, definir  $\mathcal{I}_{j+1} = \mathcal{I}_j - \{j_{\min}\}$ , sendo  $j_{\min}$  o menor índice tal que  $\mu_{j_{\min}} < 0$  e voltar para o Passo 1.

Vale mencionar que, no Passo 4, o multiplicador  $\lambda$  associado à restrição  $s^T s \leq 1$  é não negativo por construção, bastando portanto computar e verificar o sinal do vetor  $\mu$  nas condições de otimalidade (3.11)-(3.14) para o problema (3.9).

### Escolha do conjunto ativo inicial do Passo 2

Para qualquer escolha do conjunto  $\mathcal{I}^1$  que cumpra  $\mathcal{I}^1 \subset \mathcal{I}(x^c)$  teremos assegurada a convergência do modelo algorítmico descrito anteriormente, conforme estabelecem os resultados da próxima seção. Gay [55] observa que seria permitido até mesmo tomar  $\mathcal{I}^1 = \emptyset$ , embora escolhas mais elaboradas certamente ajudarão a reduzir o número

de iterações internas. Nossa opção foi a de manter ativas as restrições cujo sinal dos multiplicadores é positivo, tomando

$$\mathcal{I}^1 = \{i \in \mathcal{I}(x^c) \mid \mu_i^c > 0\}.$$

Cabe mencionar que, combinada com o passo espectral, a direção de máxima descida projetada, não só preserva as propriedades de convergência, como também permite alterar significativamente o conjunto de restrições ativas, o que é interessante para problemas de grande porte. Esta é a base do algoritmo recentemente proposto por Andretta, Birgin e Martínez [9], que por sua vez se baseia nos trabalhos [8, 17, 18] para problemas em caixas. Em [9] o conjunto viável é ampliado pela eliminação de restrições, de maneira a baratear o custo das projeções no conjunto poliedral, que são feitas usando a subrotina QL, do método de Goldfarb-Idnani [63], implementada por Powell [97], e modificada por Schittkowski [105]. Outras opções para a estratégia de projeção no conjunto viável poliedral são o algoritmo de Dykstra (ref. [26], [95]), ou outro método iterativo, como o de Llanas e Moreno [79].

#### **Resolução do subproblema do Passo 4**

A função objetivo quadrática do subproblema reduzido (3.8) é construída baseada em uma aproximação em torno do ponto  $x^c$ .

Podemos construir este modelo quadrático utilizando a matriz Hessiana original avaliada em  $x^\ell$  e um fator de correção introduzido ao vetor  $g = \nabla f(x^\ell) + H s^{\ell, k-1}$ .

Assim para a construção do subproblema reduzido (3.8) utilizamos  $H$ , a matriz Hessiana avaliada em  $x^\ell$ , e  $\bar{g} = \nabla f(x^\ell) + H s^{\ell, k-1}$ .

Para a resolução do subproblema reduzido (3.8) utilizaremos o algoritmo LSTRS de Rojas, Santos e Sorensen.

#### **Obtenção de $s^{\ell, k}$ viável no Passo 6**

No passo 6, se  $s^{\ell, k}$  não é viável, devemos retrair o passo para a região viável. Quando isso ocorre, temos que uma ou mais restrições não ativas e, portanto, não consideradas para a construção do subproblema do passo 4, foram violadas. Isso indica que pelo menos uma destas restrições deve ser considerada ativa neste momento.

Assim realizamos a retração do passo tal que a primeira restrição violada fique ativa com o novo passo  $s^{\ell,k+1}$ . Este processo de retração pode ser comparado com o processo conhecido como *teste da razão*, utilizado com frequência em algoritmos de programação linear.

Uma observação importante sobre o passo 6 é o fato de alguns autores tomarem  $s^\ell = s^{\ell,k}$  após a retração, finalizando o processo em apenas uma iteração e retornando para o algoritmo externo. A vantagem de fazer isso é, obviamente, evitar várias iterações internas. Porém, com isso, é necessário introduzir algumas técnicas para evitar efeitos de *zigue-zague*.

Além disso, o algoritmo poderia exigir muito mais avaliações de função e gradiente, o que, dependendo da função, poderia ser caro computacionalmente.

Assim como sugerido por Gay em [55], utilizamos a técnica descrita no passo 5 do modelo algorítmico, utilizando um pouco mais de esforço para obter  $s^\ell$ , o que verificaremos nos testes computacionais se é uma boa estratégia.

### Considerações sobre problemas com restrições do tipo caixa

Apesar de neste trabalho não apresentarmos experimentos computacionais para problemas com restrições do tipo  $l \leq x \leq u$  explicitamente, gostaríamos de destacar algumas características importantes na resolução deste tipo de problemas.

A principal alteração na resolução dos subproblemas é a sua reformulação antes do passo 3 do algoritmo. Os valores das variáveis que estão ativas em  $x^c$  com relação às restrições de caixa são fixados, reduzindo assim a dimensão do subproblema.

Com essa reformulação, a matriz  $Z$  determinada no passo 4 é a matriz cujas colunas geram o núcleo da matriz  $A_{\mathcal{I}_A, \mathcal{F}_R}$ , uma submatriz da matriz  $A$  cujas linhas são relativas às restrições ativas do tipo  $Ax \geq b$  descartando as colunas correspondentes às variáveis fixas pelas restrições ativas de caixa.

Vale ressaltar que para o cálculo de  $Z$  utilizamos fatorações  $QR$  da matriz  $A_{\mathcal{I}_A, \mathcal{F}_R}^T$ , a matriz transposta de  $A_{\mathcal{I}_A, \mathcal{F}_R}$ . A cada iteração interna essa matriz pode sofrer acréscimo de colunas ou exclusão de linhas durante o processo de retração, com novas restrições ativas do tipo  $Ax \geq b$  ou do tipo caixa, respectivamente. Com isso podemos aproveitar as informações do cálculo da fatoração  $QR$  anterior utilizando rotações de Givens para o novo cálculo de  $Z$ , como mostrado em [111] (Cap. 4, Seção 1.3).

## 3.2 Resultados de convergência

Para o algoritmo proposto neste capítulo, os resultados de convergência são apresentados a seguir.

Estes resultados estão baseados no desenvolvimento realizado por Gay em [55] e [54].

O primeiro lema garante que uma sequência finita de iterandos é produzida pelo Modelo Algorítmico Interno da Seção 3.1, caracterizando a boa definição do algoritmo.

**Lema 3.2.1.** *Seja  $q(s) = s^T g + \frac{1}{2} s^T H s$  a função quadrática utilizada a cada iteração do processo e seja o subproblema resolvido pelos passos do Modelo Algorítmico Interno da Seção 3.1.3. Se  $f$  é continuamente diferenciável e  $x^\ell$  é viável mas não satisfaz as condições necessárias de primeira ordem (2.2)–(2.4), então para qualquer valor inicial  $\Delta_0 > 0$ , a condição  $f(x^\ell + s^\ell) - f(x^\ell) \leq \alpha q(s^\ell)$  do passo 3(b) é satisfeita após finitas iterações do Modelo Algorítmico Externo da Seção 3.1.*

**Prova.** Ver Lema 1 de [55], que por sua vez emprega o Lema 1 de [54]. ■

O próximo teorema estabelece a convergência do algoritmo para pontos estacionários, e é utilizado para demonstrar o Teorema 3.2.6, o qual assegura a convergência para minimizadores locais do problema.

Para o Teorema 3.2.4 assumimos as seguintes hipóteses sobre o problema.

**Hipótese 3.2.2.** *A função  $f(x)$  é limitada inferiormente.*

**Hipótese 3.2.3.** *A norma da Hessiana do modelo quadrático  $\|H\|$  é localmente limitada.*

**Teorema 3.2.4.** *Supondo a função  $f$  continuamente diferenciável e a sequência  $\{x^\ell\}$  gerada pelo Modelo Algorítmico Externo, temos que*

(a) *Se a Hipótese 3.2.2 é satisfeita, então ou*

$$\lim_{\ell \rightarrow \infty} \|x^{\ell+1} - x^\ell\| = 0 \quad \text{ou}$$

*para todos  $\epsilon_s > 0$  e  $\Delta_0 > 0$ ,  $\max\{q(s) \mid s \in B(\Delta_\ell, x^\ell)\} \leq \epsilon_s$  é satisfeito infinitas vezes.*

(b) *Além disso, se a Hipótese 3.2.3 é satisfeita, então todo ponto limite  $x^*$  da sequência  $\{x^\ell\}$  é um ponto crítico de  $f$  satisfazendo as condições necessárias de primeira ordem (2.2)–(2.4).*

**Prova.** Ver Teorema 2 de [55], que por sua vez é uma extensão do Teorema 2 de [54]. ■

Originalmente em [55], a Hipótese 3.2.2 é substituída pela hipótese de limitação da sequência dos valores funcionais correspondente à sequência gerada pelo algoritmo:  $\inf\{f(x^\ell)\} > -\infty$ . Isto, no entanto, é obtido diretamente se a Hipótese 3.2.2 sobre o problema é satisfeita.

A Hipótese 3.2.2 é frequentemente assumida na literatura, como por exemplo, no Teorema 2.2 de Shultz, Schnabel e Byrd [107] ou ainda, de uma forma indireta e mais restrita como em Moré e Sorensen [87] que pedem que o conjunto de nível  $\{x \mid f(x) \leq f(x_0)\}$  seja compacto. Há alguns casos ainda que a Hipótese 3.2.2 é obtida como tese de teoremas, como por exemplo em Dennis e Schnabel [42], Teorema 6.3.3.

O próximo teorema estabelece a convergência para um minimizador local do problema e sua ordem de convergência. Para este teorema assumimos a seguinte hipótese:

**Hipótese 3.2.5.** *A função  $\nabla^2 f(x)$  está bem definida e é Lipschitz.*

**Teorema 3.2.6.** *Suponha que  $x^*$  é um minimizador local do problema satisfazendo as condições necessárias de primeira ordem 2.2–2.4 e também as condições necessárias de segunda ordem  $d^T Z^T \nabla^2 f(x^*) Z d > 0$  para todo  $d \in \mathbb{R}^q$ , onde  $q$  é a cardinalidade do núcleo da submatriz gerada pelas restrições ativas em  $x^*$ , e a Hipótese 3.2.5 é satisfeita. Existe  $\zeta > 0$  tal que se  $\|x^0 - x^*\| < \zeta$  e a sequência  $\{x^\ell\}$  é gerada como no Lema 3.2.1 a partir de  $x^0$ , então a sequência tem convergência  $Q$ -quadrática para  $x^*$ .*

**Prova.** Ver [55], Teorema 3. ■

Este teorema garante que, ao utilizarmos informações de segunda ordem, a convergência para um minimizador local do problema  $x^*$  ocorrerá com ordem quadrática se o ponto inicial  $x^0$  estiver em uma vizinhança adequada do minimizador.

A seguir apresentamos outro método para a resolução do mesmo problema, com a abordagem baseada em elementos presentes em algoritmos de barreira.

## Capítulo 4

# Algoritmo de barreira e propriedades teóricas

Uma outra maneira para resolver o problema (2.1) é apresentada neste capítulo. Utilizamos elementos de métodos de barreira para esta segunda abordagem. Este estudo foi baseado nos trabalhos detalhados por Nocedal e Wright [92] e Luenberger [80], com algumas modificações específicas para as restrições lineares do nosso problema.

Métodos de barreira são uma classe de métodos para problemas com restrições de desigualdade que aparece com frequência na literatura (ver, por exemplo, os *surveys* [50, 126] e referências neles contidas) e tem como principal característica substituir a resolução do problema original pela solução de uma sequência de subproblemas irrestritos.

Essa característica também está presente em métodos de penalização e de Lagrangiano aumentado. A principal diferença destes métodos para os métodos de barreira é que este último pode ser considerado um método de pontos interiores enquanto os primeiros são métodos de pontos exteriores.

Os métodos de barreira apresentam um termo na função objetivo dos subproblemas irrestritos que previne, a cada iteração, os iterandos de se aproximarem da fronteira da região viável. Para o nosso trabalho utilizaremos um termo de barreira logarítmico.

Nossa proposta difere dos algoritmos tradicionais de barreira pois utilizamos subproblemas com restrições de igualdade. Para isso, primeiro introduzimos variáveis de folga para tratar as restrições de desigualdade originais do problema (2.1) e em seguida utilizamos a técnica de métodos com barreira para lidar com as restrições adicionais de desigualdade para estas novas variáveis.

Os subproblemas resultantes podem então ser resolvidos com uma pequena adaptação da técnica proposta no capítulo anterior com apenas uma iteração interna, devido ao fato do conjunto ativo permanecer constante em todas as iterações.

Embora o número de variáveis do novo problema com as folgas aumente em relação ao problema original, a princípio não temos impedimento para utilizar esta abordagem, já que no algoritmo de região de confiança empregado para resolver os subproblemas não há armazenamento explícito de matrizes.

Existem outras abordagens que também combinam regiões de confiança e barreira para problemas com restrições, dentre as quais destacamos os trabalhos de Coleman e Li [36], Kearsley [74], Monteiro e Wang [86] e Tseng [114], cujas principais características já foram resumidas na Seção 1.2.

O problema original (2.1) a ser resolvido é dado por:

$$\begin{aligned} \text{Min} \quad & f(x) \\ \text{suj. a} \quad & Ax \geq b. \end{aligned} \tag{4.1}$$

Com o objetivo de transformar as restrições originais em restrições de igualdade, acrescentamos variáveis de folga  $u \in \mathbb{R}^p$  da seguinte forma:

$$\begin{aligned} \text{Min} \quad & f(x) \\ \text{suj. a} \quad & Ax - u = b \\ & u \geq 0. \end{aligned} \tag{4.2}$$

Para lidar com as restrições de desigualdade adicionais decorrentes das variáveis de folga  $u \geq 0$ , utilizamos a técnica de métodos de barreira.

Modificamos então a função objetivo agregando um termo penalizador para inibir o iterando  $u$  de se aproximar da região  $\{u \in \mathbb{R}^p \mid u_i < 0, \text{ para algum } i = 1, \dots, p\}$ .

Assim temos uma sequência de problemas da seguinte forma a serem resolvidos:

$$\begin{aligned} \text{Min} \quad & f(x) + \frac{1}{\rho} B(u) \\ \text{suj. a} \quad & Ax - u = b, \end{aligned} \tag{4.3}$$

onde  $\rho > 0$  e  $B(u) : \mathbb{R}^p \rightarrow \mathbb{R}$  é uma função barreira com as seguintes propriedades:

1.  $B(u)$  é contínua e diferenciável para todo  $u > 0$ .
2.  $B(u)$  é infinita se  $u_i < 0$ , para algum  $i = 1, \dots, p$ .

3. Se  $\{u^k\}$ ,  $u^k > 0$  para todo  $k$  e  $\lim_{k \rightarrow \infty} u_i^k = 0$  para algum  $i \in \{1, \dots, p\}$ , então  $\lim_{k \rightarrow \infty} B(u^k) = \infty$ .

Após definida a função barreira  $B(u)$ , o processo de resolução do problema consiste em resolver uma sequência de problemas com restrições de igualdade (4.3), alterando o valor de  $\rho$ . A cada resolução adotamos  $\rho_k$  tal que  $0 < \rho_{k-1} \leq \rho_k$  e tal que a sequência  $\{\rho_k\}$  satisfaça  $\lim_{k \rightarrow \infty} \rho_k = \infty$ .

A seguir esquematizamos o algoritmo proposto neste capítulo e em seguida mostramos as propriedades de convergência.

## 4.1 Algoritmo com elementos de métodos de barreira

Nesta seção apresentamos um esquema do modelo algorítmico utilizado para a abordagem com o método de barreira.

Dados o ponto inicial  $x^0 \in \mathbb{R}^n$ , e suas respectivas folgas  $u^0 \in \mathbb{R}^p$ , construídas para que  $Ax^0 - u^0 = b$ , temos o seguinte esquema algorítmico:

- Armazenar informações da fatoração  $QR$  de  $A$ .
- Tomar  $\ell = 0$  e  $\rho_0 > 0$ .
- Enquanto o critério de parada não for satisfeito:
  1. Encontrar uma solução aproximada  $(x^{\ell+1}, u^{\ell+1})$  para o problema (4.3):

$$\begin{aligned} \min_{x,u} \quad & U(x, u, \rho) \\ \text{suj. a} \quad & Ax - u = b, \end{aligned} \tag{4.4}$$

onde  $U(x, u, \rho) = f(x) + \frac{1}{\rho}B(u)$ .

2. Atualizar o parâmetro  $\rho$  tal que  $\rho_{\ell+1} > \rho_\ell$ .

Para a resolução do passo 1 do modelo algorítmico, utilizamos o algoritmo apresentado no capítulo 3 de uma forma simplificada para restrições de igualdade como sintetizado a seguir. Vale ressaltar que utilizaremos a notação de vetores com subíndice

*exp* sempre que denotarmos um vetor expandido em relação às variáveis originais. Conforme a conveniência, utilizaremos tanto  $U(x, u, \rho)$  como  $U(x_{exp}, \rho)$ . O vetor gradiente e a matriz Hessiana da função  $U$ , avaliados em  $x_{exp}^{\ell, k}$  serão denotados por

$$g_{\ell, k} \equiv \nabla U(x_{exp}^{\ell, k}, \rho) \quad \text{e} \quad H_{\ell, k} \equiv \nabla^2 U(x_{exp}^{\ell, k}, \rho), \quad (4.5)$$

respectivamente, onde  $\ell$  é o contador das iterações do algoritmo de barreira e  $k$ , o contador do algoritmo interno de região de confiança. Pela separabilidade das variáveis  $x$  e  $u$  na função  $U$  temos

$$\nabla U(x, u, \rho) = \begin{pmatrix} \nabla f(x) \\ \frac{-1}{\rho} D_U^{-1} e \end{pmatrix} \quad (4.6)$$

com  $D_U^{-1}$  a matriz diagonal com elementos  $\frac{1}{u_i}$ ,  $i = 1, \dots, p$  e  $e = (1, \dots, 1)^T \in \mathbb{R}^p$ , e

$$\nabla^2 U(x, u, \rho) = \begin{pmatrix} \nabla^2 f(x) & 0 \\ 0 & \frac{1}{\rho} D_U^{-2} \end{pmatrix} \quad (4.7)$$

com  $D_U^{-2}$  a matriz diagonal com elementos  $\frac{1}{u_i^2}$ ,  $i = 1, \dots, p$ .

A matriz  $Z_B \in \mathbb{R}^{(n+p) \times n}$  é tal que  $Im(Z_B) = Nu([A \quad -I_p])$ .

Uma vez esclarecidos os elementos e a notação utilizada, vamos apresentar um algoritmo de região de confiança para obter  $(x^{k+1}, u^{k+1}) = x_{exp}^{k+1}$ , solução aproximada para o problema (4.3):

### Algoritmo de região de confiança com barreira (RCB)

Dadas as constantes  $\eta_0, \eta_1, \eta_2$  tais que  $0 < \eta_0 < \eta_1 < 1 < \eta_2$ ,  $c_1$  e  $c_2$  tais que  $0 < c_1 < 1 < c_2$ , o raio de confiança inicial  $\Delta_0 \geq \Delta_{min} > 0$  e o ponto inicial  $x_{exp}^{0, k} = (x^k, u^k)$  viável para (4.3), faça  $k = 0$ .

1. Enquanto os critérios de convergência <sup>1</sup> não forem satisfeitos,

(a) Computar  $\bar{w}$  minimizador global do modelo quadrático reduzido de região de confiança:

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & q_B(w) \equiv g_{\ell, k}^T Z_B w + \frac{1}{2} w^T Z_B^T H_{\ell, k} Z_B w \\ \text{s. a} \quad & \|w\| \leq \Delta_\ell \end{aligned} \quad (4.8)$$

---

<sup>1</sup>A norma do gradiente de  $U$  projetado no núcleo da matriz  $(A \quad -I)$  é suficientemente pequena

(b) Obter o passo viável para as variáveis originais do subproblema (4.3):  
 $s^k = Z_B \bar{w}$ .

(c) Decidir sobre a aceitação do ponto corrente, computando  $U(x_{exp}^{\ell,k} + s^k, \rho_\ell)$  e definindo

$$\alpha = \frac{U(x_{exp}^{\ell,k} + s^k, \rho_\ell) - U(x_{exp}^{\ell,k}, \rho_\ell)}{q_B(\bar{w})}$$

Se  $\alpha \geq \eta_0$ , então  $x_{exp}^{\ell,k+1} = x_{exp}^{\ell,k} + s^k$ , senão  $x_{exp}^{\ell,k+1} = x_{exp}^{\ell,k}$ .

(d) Atualizar o raio da região de confiança:

$$\Delta_{k+1} = \begin{cases} c_1 \min\{\|\bar{w}\|, \Delta_k\} & \text{se } \alpha < \eta_0 \\ \Delta_k & \text{se } \eta_0 \leq \alpha < \eta_1 \text{ ou } \alpha > \eta_2 \\ c_2 \min\{\|\bar{w}\|, \Delta_k\} & \text{se } \eta_1 \leq \alpha \leq \eta_2 \end{cases}$$

(e)  $k = k + 1$ .

2. Definir  $x_{exp}^{\ell+1} = x_{exp}^{\ell,k}$ , ponto que cumpriu o critério de convergência do Passo 1.

É essencial que o algoritmo que acabamos de propor não recuse indefinidamente os pontos tentativos, de tal maneira que  $x_{exp}^{\ell,k+1} \neq x_{exp}^{\ell,k}$  seja obtido, ainda que sob o preço da região de confiança ser reduzida. Além disso, é desejável que o algoritmo proposto obtenha assintoticamente pontos estacionários do problema (4.4). Como trabalhamos efetivamente com as derivadas segundas das funções envolvidas, obteremos convergência a pontos estacionários de segunda ordem. Essa análise é feita a seguir.

O Teorema 4.1.1 garante a boa definição da sequência de pontos  $x_{exp}^{\ell,k}$ , enquanto que o Teorema 4.1.2 diz respeito à convergência global de segunda ordem do algoritmo. As demonstrações foram baseadas nas encontradas em Martínez e Santos [83] (Teoremas 7.2.2 e 7.2.3, respectivamente), com as devidas adaptações para este trabalho.

**Teorema 4.1.1.** *(Boa definição)*

*Se o ponto  $x_{exp}^{\ell,k}$  não é estacionário de segunda ordem para o problema (4.4) então  $x_{exp}^{\ell,k+1} \neq x_{exp}^{\ell,k}$  está bem definido e*

$$U(x_{exp}^{\ell,k+1}, \rho_\ell) < U(x_{exp}^{\ell,k}, \rho_\ell). \quad (4.9)$$

**Prova.** Para facilitar a leitura, vamos utilizar  $x^k$  para denotar  $x_{exp}^{\ell,k}$  e omitir o parâmetro  $\rho_\ell$  na função  $U$ , e em suas derivadas, de tal forma que (4.9) é simplificada para  $U(x^{k+1}) < U(x^k)$ .

Se  $x^k$  não é estacionário de segunda ordem para (4.4) então

$$g_k \equiv Z_B^T g_{\ell,k} \neq 0 \quad (4.10)$$

ou

$$g_k = 0 \quad \text{mas } H_k \equiv Z_B^T H_{\ell,k} Z_B \text{ não é semidefinida positiva,} \quad (4.11)$$

onde a notação  $g_{\ell,k}$  e  $H_{\ell,k}$  foi definida em (4.5).

Suponhamos inicialmente que  $g_k \neq 0$ . Seja  $d \in \mathbb{R}^n$  tal que  $\|d\| = 1$  e

$$g_k^T d < 0. \quad (4.12)$$

Considere o modelo quadrático para a função objetivo de (4.4) em torno do ponto corrente  $x^k$ :

$$Q_{\ell,k}(x_{exp}) = U(x^k) + \nabla U(x^k)^T (x_{exp} - x^k) + \frac{1}{2} (x_{exp} - x^k)^T \nabla^2 U(x^k) (x_{exp} - x^k). \quad (4.13)$$

Logo, pela boa definição do modelo reduzido em (4.6) temos:

$$Q_{\ell,k}(x^k + Z_B w) = U(x^k) + \nabla U(x^k)^T Z_B w + \frac{1}{2} w^T Z_B^T \nabla^2 U(x^k) Z_B w = U(x^k) + q_B(w). \quad (4.14)$$

Usando a notação condensada de (4.10) e (4.11), o modelo quadrático do problema (4.8) fica:

$$q_B(w) = g_k^T w + \frac{1}{2} w^T H_k w. \quad (4.15)$$

Para  $\bar{w}(\Delta)$  minimizador global de  $q_B(w)$  sujeito a  $\|w\| \leq \Delta$ , como  $\|d\| = 1$ , segue

que  $\|\Delta d\| = \Delta$  e então usando (4.15) temos:

$$\begin{aligned}
q_B(\bar{w}(\Delta)) &\leq q_B(\Delta d) \\
&= g_k^T(\Delta d) + \frac{1}{2}(\Delta d)^T H_k \Delta d \\
&= \Delta g_k^T d + \frac{\Delta^2}{2} d^T H_k d \\
&\leq \Delta g_k^T d + \frac{\Delta^2}{2} \|H_k\| \\
\Rightarrow \frac{q_B(\bar{w}(\Delta))}{\Delta} &\leq g_k^T d + \frac{\Delta}{2} \|H_k\|
\end{aligned} \tag{4.16}$$

De (4.12) em (4.16) segue que existe  $\bar{\Delta} > 0$  tal que para  $\Delta \leq \bar{\Delta}$ ,

$$\frac{q_B(\bar{w}(\Delta))}{\Delta} \leq \frac{g_k^T d}{2} \equiv a < 0. \tag{4.17}$$

Definimos a razão entre a redução efetiva e a redução prevista pelo modelo com o passo  $\bar{w}(\Delta)$  como:

$$\alpha(\Delta) = \frac{U(x^k + Z_B \bar{w}(\Delta)) - U(x^k)}{q_B(\bar{w}(\Delta))}. \tag{4.18}$$

Assim, de (4.13), (4.14) e (4.17) segue que

$$\begin{aligned}
|\alpha(\Delta) - 1| &= \left| \frac{U(x^k + Z_B \bar{w}(\Delta)) - U(x^k) - (Q_{\ell,k}(x^k + Z_B \bar{w}(\Delta)) - U(x^k))}{q_B(\bar{w}(\Delta))} \right| \\
&= \left| \frac{U(x^k + Z_B \bar{w}(\Delta)) - Q_{\ell,k}(x^k + Z_B \bar{w}(\Delta))}{q_B(\bar{w}(\Delta))} \right| \\
&\leq \frac{o(\Delta^2)}{-a\Delta} \xrightarrow{\Delta \rightarrow 0} 0.
\end{aligned}$$

Logo,  $\lim_{\Delta \rightarrow 0} \alpha(\Delta) = 1$  e portanto existe  $\hat{\Delta} \in (0, \bar{\Delta}]$  tal que para  $\Delta \leq \hat{\Delta}$  o ponto  $x^k + Z_B \bar{w}(\Delta)$  satisfaz o teste do passo (c), isto é,

$$\frac{U(x^k + Z_B \bar{w}(\Delta)) - U(x^k)}{q_B(\bar{w}(\Delta))} \geq \eta_0 \tag{4.19}$$

e então  $x^{k+1} = x^k + Z_B \bar{w}(\Delta)$  está bem definido neste caso.

Suponhamos agora a situação (4.11). Então existe  $d \in \mathbb{R}^n$  tal que  $\|d\| = 1$  e

$$d^T H_k d < 0 \quad (4.20)$$

Como no caso anterior, seja  $\bar{w}(\Delta)$  minimizador global de  $q_B(w)$  sujeito a  $\|w\| \leq \Delta$ . Por (4.11) temos:

$$q_B(\bar{w}(\Delta)) \leq q_B(\Delta d) = \frac{1}{2}(\Delta d)^T H_k (\Delta d)$$

ou seja,

$$\frac{q_B(\bar{w}(\Delta))}{\Delta^2} \leq \frac{d^T H_k d}{2}. \quad (4.21)$$

Portanto, de (4.20) em (4.21) segue que existe  $\bar{\Delta} > 0$  tal que para  $\Delta \leq \bar{\Delta}$ ,

$$\frac{q_B(\bar{w}(\Delta))}{\Delta^2} \leq \frac{1}{4} d^T H_k d \equiv b < 0 \quad (4.22)$$

Assim,

$$\begin{aligned} |\alpha(\Delta) - 1| &= \left| \frac{U(x^k + Z_B \bar{w}(\Delta)) - Q_{\ell,k}(x^k + Z_B \bar{w}(\Delta))}{q_B(\bar{w}(\Delta))} \right| \\ &= \frac{o(\Delta^2)}{-b\Delta^2} \xrightarrow{\Delta \rightarrow 0} 0. \end{aligned}$$

Logo  $\lim_{\Delta \rightarrow 0} \alpha(\Delta) = 1$ . Dessa forma, para  $\Delta$  suficientemente pequeno (4.19) se verifica, o que completa a prova. ■

**Teorema 4.1.2.** (*Convergência global de segunda ordem*)

Seja  $\{x_{exp}^{\ell,k}\}_{k=0}^{\infty}$  uma sequência infinita gerada pelo algoritmo RCB. Se  $x_{exp}^*$  é um ponto limite de  $\{x_{exp}^{\ell,k}\}$  então  $Z_B^T \nabla U(x_{exp}^*) = 0$  e  $Z_B^T \nabla^2 U(x_{exp}^*) Z_B \geq 0$ .

**Prova.** Pelo teorema anterior podemos tomar uma subsequência  $\{x^k\} \equiv \{x_{exp}^{\ell,k}\}$  tal que  $x^k \neq x^{k+1}$ , para todo  $k \in \mathbb{N}_0 \subset \mathbb{N}$ .

Seja  $x^*$  um ponto limite desta subsequência, e seja  $\mathbb{N}_1 \subset \mathbb{N}_0$  um conjunto infinito de índices tal que

$$\lim_{k \in \mathbb{N}_1} x^k = x^*.$$

Devemos então considerar duas possibilidades:

$$\inf_{k \in \mathbb{N}_1} \Delta_k = 0 \quad (4.23)$$

ou

$$\inf_{k \in \mathcal{N}_1} \Delta_k > 0. \quad (4.24)$$

Assumindo inicialmente (4.23), então existe  $\mathcal{N}_2 \subset \mathcal{N}_1$  tal que

$$\lim_{k \in \mathcal{N}_2} \Delta_k = 0. \quad (4.25)$$

Desta forma existe  $k_2 \in \mathcal{N}$  tal que  $\Delta_k < \Delta_{min}$  para todo  $k \in \mathcal{N}_3$ , onde  $\mathcal{N}_3 \equiv \{k \in \mathcal{N}_2 | k \geq k_2\}$ . Mas, em cada iteração  $k$  tentamos inicialmente o raio  $\Delta \geq \Delta_{min}$ . Então, para todo  $k \in \mathcal{N}_3$ , existem  $\bar{\Delta}_k$  e  $\bar{w}(\bar{\Delta}_k)$  tais que  $\bar{w}(\bar{\Delta}_k)$  é solução global de:

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & q_B(w) \\ \text{s. a} \quad & \|w\| \leq \bar{\Delta}_k \end{aligned} \quad (4.26)$$

mas

$$U(x^k + Z_B \bar{w}, \rho_\ell) > U(x^k, \rho_\ell) + \eta_0 q_B(\bar{w}). \quad (4.27)$$

Pela atualização do raio de confiança do Passo 1(d) do algoritmo *RCB*, temos

$$\Delta_k > c_1 \|Z_B \bar{w}\|. \quad (4.28)$$

Logo, por (4.25) e (4.28) segue que

$$\lim_{k \in \mathcal{N}_3} \|Z_B \bar{w}\| = 0. \quad (4.29)$$

Suponhamos que  $x^*$  não seja um minimizador local de (4.3). Então

$$g(x^*) \equiv \nabla U(x^*, \rho_\ell) \neq 0 \quad (4.30)$$

ou

$$g(x^*) = 0 \quad \text{mas} \quad \nabla^2 U(x^*, \rho_\ell) \not\geq 0. \quad (4.31)$$

Se ocorre (4.30), então existe  $d \in \mathbb{R}^n$  tal que  $\|d\| = 1$  e

$$g(x^*)^T d < 0. \quad (4.32)$$

Então, para  $k \in \mathcal{N}_3$ ,

$$q_B(\bar{w}) \leq q_B(\bar{\Delta}_k d) = \bar{\Delta}_k g_k^T d + \frac{\bar{\Delta}_k^2}{2} d^T H_k d.$$

Logo,

$$\frac{q_B(\bar{w})}{\bar{\Delta}_k} \leq g_k^T d + \frac{\|H_k\|}{2} \bar{\Delta}_k.$$

Portanto, existe  $k_3 \in \mathbb{N}$  tal que para  $k \in \mathbb{N}_4 \equiv \{k \in \mathbb{N}_3 | k \geq k_3\}$ ,

$$\frac{q_B(\bar{w})}{\bar{\Delta}_k} \leq \frac{(g^*)^T d}{2} \equiv a_1 < 0. \quad (4.33)$$

Definimos

$$\alpha_k = \frac{U(x^k + Z_B \bar{w}, \rho_\ell) - U(x^k, \rho_\ell)}{q_B(\bar{w})}. \quad (4.34)$$

Então

$$|\alpha_k - 1| = \left| \frac{U(x^k + Z_B \bar{w}, \rho_\ell) - U(x^k, \rho_\ell) - q_B(\bar{w})}{q_B(\bar{w})} \right|$$

E com o mesmo desenvolvimento feito para a demonstração do teorema anterior, temos que:

$$|\alpha_k - 1| = \frac{o(\bar{\Delta}_k^2)}{-a_1 \bar{\Delta}_k}.$$

Portanto,

$$\lim_{k \in \mathbb{N}_4} \alpha_k = 1$$

o que contradiz o fato de que os raios  $\bar{\Delta}_k$  eram rejeitados. Logo  $\nabla U(x^*, \rho_\ell) = 0$ .

Vamos agora assumir a validade de (4.31). Então existe  $d \in \mathbb{R}^n$  tal que  $\|d\| = 1$  e

$$d^T H(x^*) d < 0. \quad (4.35)$$

Para  $k \in \mathbb{N}_3$ , definimos  $d_k = \bar{\Delta}_k d$  se  $g_k^T d \leq 0$  e  $d_k = -\bar{\Delta}_k d$  se  $g_k^T d > 0$ .

Então,

$$q_B(\bar{w}) \leq q_B(d_k) \leq \frac{\bar{\Delta}_k^2}{2} d^T H_k d,$$

logo,

$$\frac{q_B(\bar{w})}{\bar{\Delta}_k^2} \leq \frac{1}{2} d^T H_k d.$$

Portanto, existe  $k_4 \in \mathbb{N}$  tal que para  $k \in \mathbb{N}_5 \equiv \{k \in \mathbb{N}_3 | k \geq k_4\}$ ,

$$\frac{q_B(\bar{w})}{\bar{\Delta}_k^2} \leq \frac{1}{4} d^T H^* d \equiv a_2 < 0.$$

Assim, usando de novo a aproximação de Taylor de segunda ordem, temos:

$$|\alpha_k - 1| \leq \frac{1}{|a_2|} \frac{o(\|\bar{w}\|^2)}{\bar{\Delta}_k^2}.$$

Portanto,  $\lim_{k \in \mathcal{N}_5} \alpha_k = 1$ , o que contradiz o fato de  $\bar{\Delta}_k$  ser um raio rejeitado. Assim,  $H^* \geq 0$ , o que conclui a prova quando vale (4.23).

Vamos agora considerar a possibilidade (4.24). Como  $\lim_{k \in \mathcal{N}_1} x_k = x^*$  e  $\{U(x_k, \rho_\ell)\}_{k \in \mathcal{N}_0}$  é monotonicamente decrescente, da continuidade das funções envolvidas a sequência  $\{U(x_k, \rho_\ell)\}_{k \in \mathcal{N}_0}$  é limitada inferiormente por  $U(x^*, \rho_\ell)$  e então temos

$$\lim_{k \in \mathcal{N}_1} (U(x_{k+1}, \rho_\ell) - U(x_k, \rho_\ell)) = 0. \quad (4.36)$$

Mas, pelo passo 1(c) do algoritmo *RCB*,

$$U(x_{k+1}, \rho_\ell) \leq U(x_k, \rho_\ell) + \eta_0 q_B(\bar{w}). \quad (4.37)$$

Então, por (4.36) e (4.37), segue que

$$\lim_{k \in \mathcal{N}_1} q_B(\bar{w}) = 0. \quad (4.38)$$

Definimos  $\hat{\Delta} = \inf_{k \in \mathcal{N}_1} \Delta_k > 0$  e chamamos  $\hat{x}$  a solução global de

$$\begin{aligned} \min \quad & (g^*)^T(x - x^*) + \frac{1}{2}(x - x^*)^T H^*(x - x^*) \\ \text{sujeito a} \quad & \|x - x^*\| \leq \frac{\hat{\Delta}}{2}. \end{aligned} \quad (4.39)$$

Seja  $k_5 \in \mathcal{N}$  tal que

$$\|x_k - x^*\| \leq \frac{\hat{\Delta}}{2} \quad (4.40)$$

para todo  $k \in \mathcal{N}_6 \equiv \{k \in \mathcal{N}_1 | k \geq k_5\}$ .

Para  $k \in \mathcal{N}_6$ , por (4.39) e (4.40), temos

$$\|\hat{x} - x_k\| \leq \hat{\Delta} \leq \Delta_k, \quad (4.41)$$

ou seja,  $\hat{w} = (\hat{x} - x_k)$  é viável para o subproblema (4.8) do algoritmo *RCB*.

Então, pelo fato de  $\bar{w}$  ser minimizador global de  $q_B(w)$  em  $\|w\| \leq \Delta_k$ , segue que

$$q_B(\bar{w}) \leq q_B(\hat{w}) = g_k^T(\hat{x} - x_k) + \frac{1}{2}(\hat{x} - x_k)^T H_k(\hat{x} - x_k). \quad (4.42)$$

Por (4.38), passando (4.42) ao limite para  $k \in \mathcal{N}_6$ , obtemos:

$$0 \leq (g^*)^T(\hat{x} - x^*) + \frac{1}{2}(\hat{x} - x^*)^T H^*(\hat{x} - x^*),$$

portanto  $x^*$  é minimizador de (4.39) com a restrição  $\|(x - x^*)\| \leq \frac{\hat{\Delta}}{2}$  inativa.

Logo  $g^* = 0$  e  $H^* \geq 0$  pelas condições necessárias de otimalidade de segunda ordem para minimização sem restrições, completando a prova. ■

A seguir apresentamos um detalhamento teórico do cálculo dos multiplicadores de Lagrange para os subproblemas adotados.

## 4.2 Estimativa dos multiplicadores

Nesta seção mostramos as relações entre os multiplicadores do problema original com variáveis de folga (4.2) e dos subproblemas irrestritos com o termo de barreira (4.3).

Seja  $(x^*, u^*)$  um minimizador do problema (4.2) com multiplicadores  $\mu_I^*$  e  $\mu_D^*$  associados às restrições de igualdade  $Ax - u = b$  e de desigualdade  $u \geq 0$ , respectivamente. Então, pelas condições necessárias de primeira ordem temos que:

$$\begin{pmatrix} \nabla f(x^*) \\ 0 \end{pmatrix} = \begin{pmatrix} A^T \\ -I \end{pmatrix} \mu_I^* + \begin{pmatrix} 0 \\ \mu_D^* \end{pmatrix} \quad (4.43)$$

$$\mu_D \geq 0 \quad (4.44)$$

$$[\mu_D^*]_i u_i^* = 0 \quad (4.45)$$

$$u^* \geq 0 \quad (4.46)$$

$$\begin{pmatrix} A & -I \end{pmatrix} \begin{pmatrix} x^* \\ u^* \end{pmatrix} = b. \quad (4.47)$$

De forma análoga podemos escrever as condições necessárias para o subproblema

(4.3), dados  $(\hat{x}, \hat{u})$  e  $\hat{\mu}_I$ , um minimizador e seu respectivo multiplicador associado às restrições de igualdade:

$$\begin{pmatrix} \nabla f(\hat{x}) \\ 0 \end{pmatrix} = \begin{pmatrix} A^T \\ -I \end{pmatrix} \hat{\mu}_I + \begin{pmatrix} 0 \\ -\frac{1}{\rho} \nabla B(\hat{u}) \end{pmatrix} \quad (4.48)$$

$$\begin{pmatrix} A & -I \end{pmatrix} \begin{pmatrix} \hat{x} \\ \hat{u} \end{pmatrix} = b. \quad (4.49)$$

Observando a relação (4.48) vemos que uma escolha intuitiva para uma estimativa do multiplicador  $\mu_I^{k,j}$  a cada iteração interna  $j$  pode ser dada por:

$$\mu_I^{k,j} = -\frac{1}{\rho_k} \nabla B(u^j). \quad (4.50)$$

Assim um critério de parada do algoritmo interno poderia ser atingido quando

$$\nabla f(x^j) \approx A^T \mu_I^{k,j}.$$

Note que este critério é equivalente a pararmos o algoritmo quando

$$Z_B^T \begin{pmatrix} \nabla f(x^j) \\ -\frac{1}{\rho_k} \nabla B(u^j) \end{pmatrix} \approx 0,$$

com  $Z_B$  a matriz cujas colunas geram o núcleo de  $\begin{pmatrix} A & -I \end{pmatrix}$ .

No nosso algoritmo utilizamos o segundo critério de parada, aproveitando o fato de que já temos informações sobre a matriz  $Z_B$ .

Outra consequência desta construção é a relação entre  $\mu_I^{k,j}$  e  $\mu_I^* = \mu_D^*$ . Como mostrado em Luenberger [80], temos que

$$\lim_{k \rightarrow \infty} -\frac{1}{\rho_k} \nabla B(u) = \mu_D^*.$$

Assim, ao adotar  $\mu_I^{k,j} = -\frac{1}{\rho_k} \nabla B(u^j)$ , temos também uma estimativa para os multiplicadores do problema (4.2).

A seguir apresentamos alguns resultados de convergência para este algoritmo com método de barreira.

### 4.3 Resultados adicionais de convergência

Para este algoritmo com a incorporação de elementos de métodos com barreira, temos resultados de convergência específicos, os quais trataremos nesta seção.

Estes resultados teóricos foram obtidos a partir dos estudos apresentados por Nocedal e Wright [92], Wright [126] e Fiacco e McCormick [49], com as devidas adaptações para o algoritmo proposto neste trabalho.

Nesta seção assumiremos que o conjunto poliedral que define as restrições do problema original é limitado, como ocorre com os problemas teste que serão considerados. Por isso fazemos a seguinte hipótese:

**Hipótese 4.3.1.** *O conjunto viável  $\{x \mid Ax \geq b\}$  é compacto.*

Temos assim garantida a existência de minimizadores para o problema original e para cada subproblema do método com barreira.

Os resultados e definições a seguir são preliminares para a análise de convergência desta seção.

**Lema 4.3.2.** *(Weierstrass) Seja  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  contínua em um conjunto compacto  $\mathcal{V}$ , então existe  $v^*$  finito e um ponto  $x^* \in \mathcal{V}$  tal que  $f(x^*) = v^* = \min_{\mathcal{V}} f(x)$ .*

**Prova.** Ver [73], Teorema 1.2.1. ■

**Definição 4.3.3.** *O conjunto  $\mathcal{R}$  é dado por*

$$\mathcal{R} = \{x_{exp} \equiv (x, u) \in \mathbb{R}^{n+p} \mid Ax - u = b \quad e \quad u \geq 0\} \quad (4.51)$$

*e seu interior relativo é dado por*

$$\mathcal{R}^0 = \{x_{exp} \equiv (x, u) \in \mathbb{R}^{n+p} \mid Ax - u = b \quad e \quad u > 0\}. \quad (4.52)$$

**Lema 4.3.4.** *Sob a Hipótese 4.3.1 o conjunto  $\mathcal{R}$  definido em 4.3.3 é compacto.*

**Prova.** Para provar a compacidade de  $\mathcal{R}$ , basta mostrar que este conjunto é limitado. Pela Hipótese 4.3.1 temos que  $\{x \mid Ax \geq b\}$  é compacto e, portanto, limitado. Então  $\{(x, u) \mid u = Ax - b, u \geq 0\}$  é também limitado, o que completa a demonstração. ■

**Lema 4.3.5.** A função  $U(x, u, \rho_k)$  dada em (4.4) é contínua no interior relativo  $\mathcal{R}^0$  do conjunto compacto  $\mathcal{R}$ .

**Prova.** A continuidade segue diretamente da continuidade das funções  $f(x)$  e  $\log(u)$  no conjunto viável  $\{x \mid Ax \geq b\}$  e em  $u > 0$ , respectivamente. ■

**Definição 4.3.6.** Uma função  $\varphi : D \rightarrow \mathbb{R}$  é coerciva em um conjunto limitado  $D$  se para toda sequência  $\{x^k\} \in D^0$  tal que  $\{x^k\} \rightarrow x \in D - D^0$ , temos que  $\lim_{k \rightarrow \infty} \varphi(x^k) = +\infty$ .

**Lema 4.3.7.** A função  $U(x, u, \rho_k) = f(x) - \frac{1}{\rho_k} \sum \log(u_i)$  é coerciva em  $\mathcal{R}$ .

**Prova.** Seja  $\{x_{exp}^j\} \equiv \{(x^j, u^j)\} \subset \mathcal{R}^0$  uma sequência tal que

$$\lim_{j \rightarrow \infty} x_{exp}^j = \hat{x}_{exp} \equiv (\hat{x}, \hat{u}) \quad \text{com } \hat{x}_{exp} \in \mathcal{R} - \mathcal{R}^0.$$

Pela continuidade da função  $f$  e da compacidade de  $\mathcal{R}$ , temos que  $f$  é limitada em  $\mathcal{R}$ , ou seja, existem  $f_{min}$  e  $f_{max}$  tais que  $f_{min} \leq f(x) \leq f_{max}$  para  $(x, u) \in \mathcal{R}$ . Portanto:

$$U(x^j, u^j, \rho_k) = f(x^j) - \frac{1}{\rho_k} \sum_i \log(u_i^j) \geq f_{min} - \frac{1}{\rho_k} \sum_i \log(u_i^j) \text{ para todo } j.$$

Assim, pelo fato de que  $\lim_{j \rightarrow \infty} (x^j, u^j) = (\hat{x}, \hat{u})$  com  $\hat{u}_i = 0$  para algum  $i$  e  $\lim_{u \rightarrow 0} -\log(u) = +\infty$ , temos que:

$$\lim_{j \rightarrow \infty} U(x^j, u^j, \rho_k) = +\infty,$$

completando a demonstração. ■

Estas hipóteses e resultados são importantes para garantir a convergência dos minimizadores dos subproblemas para minimizadores do problema original, como é evidenciado nos próximos teoremas, extraídos de [92] e [126], com as devidas adaptações para o nosso algoritmo.

Os Teoremas 4.3.10 e 4.3.13 dependem da limitação do conjunto de minimizadores do problema original, que é diretamente obtida assumindo a compacidade do conjunto viável. Em [92], uma hipótese alternativa sobre a função é dada: para garantir a limitação do conjunto de minimizadores é pedida a convexidade da função objetivo.

Contudo, os resultados apresentados a seguir podem ser obtidos para funções mais gerais, com hipóteses menos restritivas. No nosso caso a limitação do conjunto de minimizadores será consequência da Hipótese 4.3.1, válida para a classe de problemas-teste considerada neste trabalho.

**Lema 4.3.8.** *O conjunto  $\mathcal{M}$  de minimizadores globais do problema (4.2), dado por:*

$$\mathcal{M} = \{(x, u) \in \mathcal{R} \mid f(x) = f^*\},$$

*é não vazio e limitado, onde  $f^*$  é o valor mínimo de  $f(x)$  em  $\mathcal{R}$ .*

**Prova.** Segue diretamente da compacidade de  $\mathcal{R}$  e da continuidade da função objetivo  $f$  do problema original. ■

O Teorema 4.3.10 apresenta resultados que descrevem a convergência dos subproblemas de barreira (4.3) e sua relação com os minimizadores do problema (4.2). Mas antes de enunciar e provar este teorema, temos o lema a seguir, adaptado do Lema 1 de [126].

**Lema 4.3.9.** *Assuma a região  $\mathcal{R}^0$  dada na Definição 4.3.3 não vazia e seja  $\{y_{exp}^k\} \equiv \{x^k, u^k\}$  uma sequência convergente em  $\mathcal{R}^0$  tal que seu ponto limite  $\hat{y}_{exp}$  esteja na fronteira de  $\mathcal{R}^0$ :*

$$\lim_{k \rightarrow \infty} y_{exp}^k = \hat{y}_{exp} \quad \text{tal que } \hat{y}_{exp} \in \mathcal{R} - \mathcal{R}^0. \quad (4.53)$$

*Suponha  $\varphi$  uma função contínua em  $\mathcal{R}^0$  tal que*

$$\lim_{k \rightarrow \infty} \varphi(y_{exp}^k) = +\infty, \quad (4.54)$$

*para toda sequência  $\{y_{exp}^k\}$  satisfazendo (4.53).*

*Então o mínimo global  $\varphi^*$  de  $\varphi$  em  $\mathcal{R}^0$  é finito e realizado para algum ponto  $x_{exp}^* \equiv (x^*, u^*) \in \mathcal{R}^0$ :*

$$\min\{\varphi(x_{exp}) \mid x_{exp} \in \mathcal{R}^0\} = \varphi(x_{exp}^*) \equiv \varphi^*.$$

**Prova.** Dado um ponto arbitrário  $\hat{x}_{exp} \equiv (\hat{x}, \hat{u}) \in \mathcal{R}^0$ , definimos o conjunto de nível associado  $\hat{\mathcal{W}}$  como:

$$\hat{\mathcal{W}} = \{x_{exp} \in \mathcal{R}^0 \mid \varphi(x_{exp}) \leq \hat{\varphi}\}$$

onde  $\hat{\varphi} \equiv \varphi(\hat{x}_{exp})$ .

Como o conjunto  $\mathcal{R}$  é compacto e  $\mathcal{R}^0 \subset \mathcal{R}$ , temos que  $\hat{\mathcal{W}}$  é limitado.

Para mostrar que  $\hat{\mathcal{W}}$  é compacto, basta mostrar que é um conjunto fechado. Faremos isso mostrando que  $\hat{\mathcal{W}}$  contém todos os seus pontos de acumulação.

Seja  $\{x_{exp}^k\}$  uma sequência convergente com ponto limite  $\bar{x}_{exp}$  tal que  $x_{exp}^k \in \hat{\mathcal{W}}$  para todo  $k$ .

Sabemos, pela compacidade de  $\mathcal{R}$  e pelo fato de  $\mathcal{R}^0 \subset \mathcal{R}$ , que  $\bar{x}_{exp} \in \mathcal{R}$ .

Supondo por absurdo que  $\bar{x}_{exp} \in \mathcal{R} - \mathcal{R}^0$ , a sequência  $\{x_{exp}^k\}$  satisfaria (4.53) e (4.54).

Sendo  $\hat{\varphi}$  um limitante superior para  $\varphi(x_{exp})$  com  $x_{exp} \in \hat{\mathcal{W}}$ , para  $k$  suficientemente grande teríamos que  $x_{exp}^k \notin \hat{\mathcal{W}}$ , o que contradiz a hipótese da construção da sequência  $\{x_{exp}^k\}$ . Portanto provamos por absurdo que  $\bar{x}_{exp} \in \mathcal{R}^0$ .

Como  $x_{exp}^k \in \hat{\mathcal{W}}$  para todo  $k$  e pela continuidade de  $\varphi$  em  $\mathcal{R}^0$ , temos que  $\varphi(\bar{x}_{exp}) \leq \hat{\varphi}$  e, conseqüentemente,  $\bar{x}_{exp} \in \hat{\mathcal{W}}$ .

Pelo fato de  $\{x_{exp}^k\}$  ser uma sequência convergente arbitrária, segue que  $\hat{\mathcal{W}}$  é fechado. Logo temos que  $\hat{\mathcal{W}}$  é compacto.

Como  $\varphi$  é contínua no conjunto compacto  $\hat{\mathcal{W}}$ , pelo Lema de Weierstrass temos que  $\varphi$  atinge o mínimo global em  $\hat{\mathcal{W}}$  para algum ponto  $x_{exp}^*$ .

Pela definição de  $\hat{\mathcal{W}}$ , o valor de  $\varphi$  para qualquer ponto em  $\mathcal{R}^0 - \hat{\mathcal{W}}$  precisa ser estritamente maior que o menor valor em  $\hat{\mathcal{W}}$ . Então  $x_{exp}^*$  é minimizador global de  $\varphi$  em  $\mathcal{R}^0$ . ■

No resultado a seguir a convergência da sequência gerada pelo método de barreira é estabelecida. Foi adaptado do Teorema 7 de [126], com as simplificações decorrentes da compacidade assumida para o conjunto  $\mathcal{R}$ . O teorema a seguir descreve condições para que a sequência de minimizadores do problema com barreira convirja para minimizadores do problema (4.2).

**Teorema 4.3.10.** *Considere o problema (4.2), a região viável  $\mathcal{R}$  da Definição 4.3.3 e o conjunto  $\mathcal{M}$  do Lema 4.3.8. Seja  $\{\rho_k\}$  uma sequência crescente de parâmetros de barreira tal que  $\lim_{k \rightarrow \infty} \rho_k = \infty$ . Então:*

(i) *O problema com barreira (4.4) admite pelo menos um minimizador global em  $\mathcal{R}^0$ , ou seja, tal que  $u > 0$ . E, além disso, qualquer sequência de minimizadores globais em  $\mathcal{R}^0$  possui pelo menos uma subsequência convergente.*

(ii) *Seja  $\{x_{exp}^k\}$  uma subsequência convergente de minimizadores globais de  $U(x_{exp}, \rho_k)$  em  $\mathcal{R}^0$ . Então o ponto limite de  $\{x_{exp}^k\}$  está em  $\mathcal{M}$ .*

(iii)

$$\lim_{k \rightarrow \infty} f(x^k) = f^* = \lim_{k \rightarrow \infty} U(x^k, u^k, \rho_k).$$

**Prova.** Pela continuidade (Lema 4.3.5) e coercividade (Lema 4.3.7) da função  $U(x, u, \rho_k)$ , temos as hipóteses do Lema 4.3.9 satisfeitas. Portanto para cada  $k$  a função  $U(x, u, \rho_k)$  possui um mínimo global para algum ponto em  $\mathcal{R}^0 \subset \mathcal{R}$ .

Seja  $\{y_{exp}^k\} \equiv \{x^k, u^k\}$  a sequência dos pontos que realizam estes mínimos para cada  $\rho_k$ .

Pela compacidade de  $\mathcal{R}$ , temos que  $\{y_{exp}^k\}$  é limitada e, portanto, admite pelo menos um ponto limite  $\hat{x}_{exp} \equiv (\hat{x}, \hat{u}) \in \mathcal{R}$ , o que prova (i).

Para provar (ii), devemos mostrar que  $\hat{x}_{exp} \in \mathcal{M}$ , ou seja,  $f(\hat{x}) = f^*$ .

Provaremos este resultado por contradição, assumindo que  $\hat{x}_{exp} \notin \mathcal{M}$ .

Como  $\hat{x}_{exp} \in \mathcal{R}$ , assumir  $\hat{x}_{exp} \notin \mathcal{M}$  implica  $f(\hat{x}) > f^*$ . E então podemos mostrar que existe  $x_{exp}^{int} \equiv (x^{int}, u^{int}) \in \mathcal{R}^0$  tal que  $f(\hat{x}) > f(x^{int})$ .

Seja  $x_{exp}^* \equiv (x^*, u^*) \in \mathcal{M}$ . Se  $x_{exp}^* \in \mathcal{R}^0$ , basta tomar  $x_{exp}^{int} = x_{exp}^*$ . Caso contrário, temos que  $x_{exp}^* \in \mathcal{R} - \mathcal{R}^0$ , ou seja,  $Ax^* - u^* = b$  com  $u_i^* = 0$  para algum  $i$ . Assim toda vizinhança de  $x_{exp}^*$  contém pontos em  $\mathcal{R}^0$ . Pela continuidade de  $f$  e por  $x_{exp}^* \in \mathcal{M}$ , temos que existe  $x_{exp}^{int} \in \mathcal{R}^0$  em uma vizinhança de  $x_{exp}^*$  tal que  $f(x^{int}) < f(\hat{x})$ .

Agora, seja  $\{x_{exp}^k\}$  uma subsequência convergente de  $\{y_{exp}^k\}$  com ponto limite  $\hat{x}_{exp}$ . A relação  $f(\hat{x}) > f(x^{int})$  implicaria que, para  $k$  suficientemente grande, temos:

$$f(x^k) > f(x^{int}). \quad (4.55)$$

Como  $x_{exp}^{int} \in \mathcal{R}^0$  e  $x_{exp}^k$  é minimizador global de  $U(x_{exp}, \rho_k)$  em  $\mathcal{R}^0$ , temos que:

$$f(x^k) - \frac{1}{\rho_k} \sum \log(u_i^k) \leq f(x^{int}) - \frac{1}{\rho_k} \sum \log(u_i^{int}). \quad (4.56)$$

Pela viabilidade estrita de  $x_{exp}^{int}$ , o termo da barreira envolvendo  $u^{int}$  em (4.56) é finito e

$$\lim_{k \rightarrow \infty} U(x_{exp}^{int}, \rho_k) = f(x^{int}).$$

Suponha que o ponto limite  $\hat{x}_{exp}$  é também estritamente viável, ou seja,  $\hat{x}_{exp} \in \mathcal{R}^0$ . Então

$$\lim_{k \rightarrow \infty} U(x_{exp}^k, \rho_k) = f(\hat{x}).$$

Logo, se  $k \rightarrow \infty$  em (4.56), temos a desigualdade  $f(x^k) \leq f(x^{int})$  para  $k$  suficiente-

mente grande, contradizendo (4.55).

Por outro lado, suponha que  $\hat{x}_{exp} \notin \mathcal{R}^0$ . Adicionando o termo da barreira nos dois lados da desigualdade (4.55) temos:

$$f(x^{int}) - \frac{1}{\rho_k} \sum \log(u_i^{int}) < f(x^k) - \frac{1}{\rho_k} \sum \log(u_i^{int}). \quad (4.57)$$

Assim, de (4.56) e (4.57) temos que

$$\begin{aligned} f(x^k) - \frac{1}{\rho_k} \sum \log(u_i^k) &< f(x^k) - \frac{1}{\rho_k} \sum \log(u_i^{int}) \\ \Rightarrow - \sum \log(u_i^k) &< - \sum \log(u_i^{int}). \end{aligned} \quad (4.58)$$

Como  $u^{int} > 0$  desde que  $x_{exp}^{int} \in \mathcal{R}^0$ , temos que o lado direito de (4.58) é finito. Porém, como  $\hat{x}_{exp} \in \mathcal{R} - \mathcal{R}^0$ , o lado esquerdo da desigualdade é ilimitado quando  $k \rightarrow \infty$  o que novamente gera uma contradição.

A contradição vem do fato de assumirmos  $\hat{x}_{exp} \notin \mathcal{M}$ . Portanto,  $\hat{x}_{exp} \in \mathcal{M}$  e  $f(\hat{x}) = f^*$ .

Como  $\hat{x}_{exp}$  é um ponto limite arbitrário de  $\{y_{exp}^k\}$ , então todo ponto limite de uma subsequência de minimizadores globais do problema com barreira em  $\mathcal{R}^0$  é minimizador do problema (4.2) com valor de função objetivo  $f^*$ . ■

O próximo teorema caracteriza, sob determinadas condições, a trajetória dos minimizadores de barreira. Para obter os resultados do Teorema 4.3.13, o minimizador local  $x_{exp}^*$  do problema (4.2) deve ser um ponto isolado, o que é assegurado com as hipóteses a seguir.

**Hipótese 4.3.11.** *O minimizador local  $x_{exp}^* \equiv (x^*, u^*)$  do problema (4.2), que cumpre as condições necessárias de primeira ordem (4.43)-(4.47), é um ponto que satisfaz as condições suficientes de segunda ordem para os multiplicadores de Lagrange  $\mu^* \equiv \mu_J^* = \mu_D^*$ , nos quais vale a complementaridade estrita:*

- (a)  $\mu_i^* u_i^* = 0$  com  $\mu_i^* > 0$  se  $u_i^* = 0$ ;
- (b) Existe  $\alpha_H > 0$  tal que  $x_{exp}^T Z_J^T H^* Z_J x_{exp} \geq \alpha_H \|x_{exp}\|^2$  para todo  $x_{exp} = (x, u)$ , onde  $Z_J$  é uma matriz cujas colunas geram o núcleo do Jacobiano  $J$  das restrições

ativas em  $x_{exp}^*$ , dado por

$$J = \begin{pmatrix} A & -I \\ 0 & I_{\mathcal{I}_u, i} \end{pmatrix},$$

em que  $I_{\mathcal{I}_u, i} \in \mathbb{R}^{n_u \times p}$  denota a submatriz da matriz identidade com apenas  $n_u$  linhas correspondentes aos índices  $i$  das restrições ativas  $u_i^* = 0$ , e  $H^*$  é a matriz Hessiana da função objetivo do problema (4.2) em relação às variáveis  $x$  e  $u$ , avaliada em  $x_{exp}^*$ :

$$H^* = \begin{pmatrix} \nabla^2 f(x^*) & 0 \\ 0 & 0 \end{pmatrix}.$$

**Hipótese 4.3.12.** O ponto  $x_{exp}^*$  é não degenerado para as restrições do problema (4.2) (ou (4.4)), ou seja, os gradientes das restrições ativas em  $x_{exp}^*$  são linearmente independentes.

**Teorema 4.3.13.** Seja o problema (4.2) com a região viável  $\mathcal{R}$  e o conjunto  $\mathcal{R}^0$  da Definição 4.3.3 não vazios. Seja  $x_{exp}^* \equiv (x^*, u^*)$  uma solução local do problema (4.2) satisfazendo as Hipóteses 4.3.11 e 4.3.12. Então, se o algoritmo de barreira é aplicado com uma sequência monótona  $\{\rho_k\}$  tal que  $\lim_{k \rightarrow \infty} \rho_k = +\infty$ :

- (i) Existe uma subsequência convergente de minimizadores globais  $x_{exp}(\rho_k) \equiv (x(\rho_k), u(\rho_k))$  para o problema (4.3) tal que  $\lim_{\rho_k \rightarrow \infty} x_{exp}(\rho_k) = x_{exp}^*$ .
- (ii) Para a subsequência convergente  $\{x_{exp}^k\}$  do item (i), as estimativas dos multiplicadores de Lagrange  $\mu^k$  (cf. Seção 4.2) convergem para  $\mu^*$  da Hipótese 4.3.11:

$$\lim_{k \rightarrow \infty} \frac{1}{\rho_k u_i^k} = \mu_i^*, i = 1, \dots, p.$$

- (iii) A Hessiana reduzida  $Z_J^T \nabla^2 U(x_{exp}, \rho_k) Z_J$  é definida positiva para todo  $\rho_k$  suficientemente grande.
- (iv) Existe uma única função diferenciável  $x_{exp}(\rho)$  de minimizadores globais de  $U(x, u, \rho)$  tais que  $Ax - u = b$  para todo  $\rho$  suficientemente grande.
- (v)  $\lim_{\rho \rightarrow \infty} x_{exp}(\rho) = x_{exp}^*$ .

**Prova.** A prova de (i) segue do Teorema 4.3.10.

Para provar (ii), supomos a sequência convergente  $\{x_{exp}^k\}$  com apropriada redefinição de  $k$ , tal que:

$$\lim_{k \rightarrow \infty} x_{exp}^k = (x^*, u^*), \quad (4.59)$$

onde  $x_{exp}^k \equiv (x^k, u^k)$  é um minimizador global do problema (4.4) com  $\rho = \rho_k$ .

Então, usando a notação definida em (4.6),

$$\nabla U(x^k, u^k, \rho_k) = \begin{pmatrix} A^T \\ -I \end{pmatrix} \mu_k. \quad (4.60)$$

De (4.60) temos a seguinte relação para as restrições inativas com  $u_i^k > 0$ , que também já foi discutida na Seção 4.2:

$$\mu_i^k = \frac{1}{\rho_k u_i^k} > 0 \text{ para qualquer } \rho_k > 0. \quad (4.61)$$

Supondo que a  $i$ -ésima restrição é inativa em  $x_{exp}^*$ , ou seja,  $u_i^* > 0$ , temos que

$$\lim_{k \rightarrow \infty} \mu_i^k = \mu_i^* > 0.$$

Então, para  $k$  suficientemente grande, temos que  $u_i^k > 0$  e por (4.61), desde que  $\lim_{k \rightarrow \infty} \frac{1}{\rho_k} = 0$ , temos:

$$\lim_{k \rightarrow \infty} \mu_i^k = 0 = \mu_i^*.$$

Logo, se nenhuma restrição é ativa na solução, (ii) é verificado.

Caso contrário, seja  $\mathcal{I}_u$  o conjunto de índices das restrições ativas em  $x_{exp}^*$ , ou seja, se  $i \in \mathcal{I}_u$ , então  $u_i^* = 0$ . Definimos os escalares positivos  $s_k$  e  $v_i^k$ , para  $i \in \mathcal{I}_u$ , da seguinte forma:

$$s_k = \sum_{i=1}^p \mu_i^k \quad \text{e} \quad (4.62)$$

$$v_i^k = \frac{\mu_i^k}{s_k}. \quad (4.63)$$

Então  $v_i^k > 0$  e  $\sum_{i=1}^p v_i^k = 1$ . Em outras palavras,  $v_i^k$  é limitado,  $0 < v_i^k \leq 1$ ,  $i \in \mathcal{I}_u$  e  $0 \leq v_i^k \leq 1$ ,  $i = 1, \dots, p$  pois  $\mu_j^k = 0$  se  $j \notin \mathcal{I}_u$  pela Hipótese 4.3.11.

Como a relação (4.60) é válida para todo  $x_{exp}^k$ , então temos:

$$\frac{1}{s_k} \nabla f(x^k) - \frac{\sum_{i=1}^p A_i^T \mu_i^k}{s_k} = 0,$$

onde  $A_i$  denota a  $i$ -ésima linha da matriz  $A$ .

Simplificando esta relação temos:

$$\frac{1}{s_k} \nabla f(x^k) - \sum_{i=1}^p A_i^T v_i^k = 0. \quad (4.64)$$

Definindo a sequência  $\{v_i^k\}$  com  $k \rightarrow \infty$ , pela limitação de  $v_i^k$ , temos que a sequência admite uma subsequência convergente.

Agora, por contradição vejamos que  $\lim_{k \rightarrow \infty} s_k = \hat{s}$  é finito. Supondo que este limite não fosse finito, para um conjunto  $\{\hat{v}_i\}$  de pontos limite de  $\{v_i^k\}$ , teríamos de (4.64):

$$\sum_{i=1}^p A_i^T \hat{v}_i = 0 \quad \text{com } \hat{v}_i \geq 0 \text{ e } \sum_{i=1}^p \hat{v}_i = 1.$$

Em outras palavras, uma combinação não trivial dos gradientes das restrições ativas em  $x_{exp}^*$  é zero, o que contradiz a hipótese de independência linear.

Portanto, provamos que  $\hat{s}$  é finito. Isso implica que  $\mu_i^k$ ,  $i = 1, \dots, p$ , é limitado para todo  $k$  e, conseqüentemente,  $\{\mu^k\}$  tem pelo menos um ponto de acumulação, digamos  $\hat{\mu}$ . Então de (4.60) temos que:

$$\nabla f(x^*) = A^T \hat{\mu}.$$

Como as linhas de  $A_{\mathcal{I}_u}$  são linearmente independentes por hipótese, há um único  $\hat{\mu}$  satisfazendo esta equação. Portanto,  $\hat{\mu} = \mu^*$ , o que completa a prova de (ii).

Para demonstrar (iii) basta provar que a matriz  $\nabla^2 U(x^k, u^k, \rho_k)$  dada em (4.7) é positiva definida para vetores no núcleo de  $J$  com  $k$  suficientemente grande.

Agora seja  $\begin{pmatrix} px \\ pu \end{pmatrix}$  um vetor no núcleo de  $J$ . Temos que

$$\begin{pmatrix} px^T & pu^T \end{pmatrix} \nabla^2 U(x^k, u^k, \rho_k) \begin{pmatrix} px \\ pu \end{pmatrix} = px^T \nabla^2 f(x_k) px + \frac{1}{\rho_k} \sum \frac{1}{(u_i^k)^2} pu_i^2.$$

Pela Hipótese 4.3.11 (b), temos que existe  $\alpha_H$  tal que  $px^T \nabla^2 f(x^*) px \geq \alpha_H \|px\|^2$ . Assim, pela convergência da sequência  $\{x^k, u^k\}$ , temos que para  $k$  suficientemente grande,  $px^T \nabla^2 f(x_k) px + \frac{1}{\rho_k} \sum \frac{1}{(u_i^k)^2} pu_i^2 > 0$ , o que completa a demonstração de (iii).

Para mostrar (iv), primeiro observamos que o seguinte sistema de equações tem

solução:

$$\Phi(x, u, \rho) = \nabla f(x) - \frac{1}{\rho} \sum \frac{A_i^T}{u_i} = 0.$$

O Jacobiano de  $\Phi$  com respeito a  $(x, u)$  é a Hessiana da função barreira que foi mostrada ser positiva definida no item anterior para  $k$  suficientemente grande.

Pelo teorema da função implícita (Ver Ortega e Rheinboldt [94]), existe uma única função diferenciável  $x_{exp}(\rho)$  passando por  $x_{exp}(\rho_k)$  tal que  $\Phi(x, u, \rho) = 0$  para todo  $\rho$  tal que  $\frac{1}{\rho}$  esteja na vizinhança de  $\frac{1}{\rho_k}$ .

Usando então um argumento de continuidade, existe  $x_{exp}(\rho)$  para todo  $\rho$  tal que  $0 < \frac{1}{\rho} < \frac{1}{\rho_k}$  para  $k$  suficientemente grande. Em outras palavras, existe  $x_{exp}(\rho)$  para todo  $\rho$  tal que  $\rho_k < \rho$ , provando assim (iv).

Pela unicidade local de  $x_{exp}(\rho)$  e por (i), temos  $\lim_{\rho \rightarrow \infty} x_{exp}(\rho) = x_{exp}^*$ , o que completa a demonstração. ■



## Capítulo 5

# O problema dos pontos em um polígono

Uma das vantagens do algoritmo apresentado neste trabalho é a possibilidade de resolução do problema utilizando apenas produtos matriz por vetor, sem a utilização de fatorações explícitas para matrizes de grande porte. Isso se deve ao fato de incorporarmos o algoritmo de Rojas, Santos e Sorensen como resolvidor interno, que é considerado um algoritmo do tipo *matrix-free*.

Em relação às fatorações  $QR$  para o cálculo da matriz  $Z$ , podemos aproveitar a estrutura dos problemas e utilizar apenas funções que calculam os produtos das matrizes  $Z$  e  $Z^T$  por vetores, sem ter o armazenamento explícito da matriz  $Z$ .

Para explorar estes tipos de resultados trabalhamos com uma família de problemas com uma estrutura particular que foi utilizada de forma simplificada em trabalhos de Powell [98] e [99].

Originalmente Powell apenas considerou restrições do tipo caixa, o que resulta em problemas com uma estrutura específica mais simples. Porém, neste trabalho optamos por introduzir novas restrições gerais do tipo  $Ax \geq b$  ao problema testado por Powell, gerando problemas mais complexos do ponto de vista da resolução.

A seguir apresentamos o problema a ser resolvido e, nas próximas seções, duas abordagens para resolver a mesma família de problemas, as quais são comparadas nos testes computacionais.

O problema dos pontos no quadrado apresentado por Powell consiste em, dados  $np$  pontos  $P^i = (P_1^i, P_2^i)^T \in [0, 1] \times [0, 1] \subset \mathbb{R}^2$ ,  $i = 1, \dots, np$ , distribuí-los no quadrado

unitário de tal forma que fiquem o mais distante possível uns dos outros.

No nosso caso utilizaremos polígonos ao invés de caixas, tornando o problema mais complexo, mas com a vantagem de possuir uma estrutura bastante favorável para ser explorada em um contexto de grande porte e livre de fatorações.

As nossas restrições consistem em, dado um polígono de  $nl$  lados, queremos que todos os pontos  $P^i$ , com  $i = 1, \dots, np$ , estejam limitados por este polígono:

$$\bar{A}P^i \geq \bar{b} \quad , \quad i = 1, \dots, np,$$

no qual a matriz  $\bar{A} \in \mathbb{R}^{nl \times 2}$  e o vetor  $\bar{b} \in \mathbb{R}^{nl}$  estão relacionados com as expressões que descrevem os lados do polígono.

Modificamos também a função objetivo utilizada por Powell, que originalmente apresentava descontinuidades e foi proposta no contexto de algoritmos para problemas com funções sem derivadas. A função objetivo adotada neste trabalho é:

$$f(P) = \sum_{j=i+1}^{np} \sum_{i=1}^{np} \frac{1}{(\|P^i - P^j\|^2 + \xi)^{\frac{1}{2}}}. \quad (5.1)$$

A principal diferença entre a nossa função objetivo e a de Powell é o acréscimo da constante  $\xi$ . Escolhemos  $\xi$  suficientemente pequeno de forma a preservar as características da função objetivo originalmente proposta em [98] e assegurar a diferenciabilidade da função quando temos  $P^i = P^j$ .

Para modelar este problema, inicialmente reescrevemos as coordenadas dos vetores  $P^i$ , com  $i = 1, \dots, np$ , em um único vetor  $x \in \mathbb{R}^n$ , com  $n = 2np$ . Desta forma temos que  $(x_{2i-1}, x_{2i})^T = P^i$  e o nosso problema fica expresso como:

$$\begin{aligned} \text{Min} \quad & f(x) = \sum_{j=i+1}^{np} \sum_{i=1}^{np} ((x_{2i-1} - x_{2j-1})^2 + (x_{2i} - x_{2j})^2 + \xi)^{\frac{-1}{2}} \\ \text{suj. a} \quad & Ax \geq b \end{aligned} \quad (5.2)$$

onde  $A \in \mathbb{R}^{np*nl \times 2np}$  e  $b \in \mathbb{R}^{nl*np}$  são dados por

$$A = \begin{pmatrix} \bar{A} & 0 & 0 & \dots & 0 \\ 0 & \bar{A} & 0 & \dots & 0 \\ 0 & 0 & \bar{A} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \bar{A} \end{pmatrix} \quad \text{e} \quad b = \begin{pmatrix} \bar{b} \\ \bar{b} \\ \bar{b} \\ \vdots \\ \bar{b} \end{pmatrix},$$

em que  $\bar{A} \in \mathbb{R}^{nl \times 2}$  e  $\bar{b} \in \mathbb{R}^{nl}$ .

Podemos notar que as restrições apresentam uma estrutura esparsa particular, originando o desacoplamento para cada duas componentes do vetor  $x$ . Este fato repercute na obtenção das informações para o cálculo da matriz  $Z$  posteriormente. A única informação explícita que precisaremos será a fatoração  $QR$  da matriz  $\bar{A}^T \in \mathbb{R}^{2 \times nl}$ , que sempre será uma matriz de pequeno porte, tendo em vista que o número de lados do polígono não aumentará com o acréscimo de pontos, e conseqüentemente com o aumento da dimensão do problema.

Outro aspecto importante está relacionado à função objetivo que, apesar de não apresentar esparsidade na estrutura do vetor gradiente e da matriz Hessiana, apresenta uma regra de formação para suas derivadas, o que nos possibilita calcular produtos da matriz Hessiana por vetores sem a necessidade de armazenarmos a matriz explicitamente.

A seguir apresentamos as duas abordagens diferentes para resolver este problema. A primeira resolve o problema com as restrições de desigualdade diretamente com o algoritmo apresentado no capítulo 3. A segunda maneira utiliza a idéia de métodos de barreira, resolvendo a cada iteração um subproblema de igualdade com o algoritmo proposto no capítulo 4.

## 5.1 Abordagem com restrições de desigualdade

A primeira maneira adotada para resolver o problema dos pontos em um polígono é abordar diretamente o problema (5.2) com as restrições de desigualdade utilizando o algoritmo proposto no capítulo 3.

O ponto mais importante nesta resolução é o aproveitamento da estrutura do problema sem o armazenamento explícito da matriz expandida  $A$  e da matriz  $Z$  cujas colunas geram o núcleo da submatriz  $A_{\mathcal{I}_A} \in \mathbb{R}^{na \times 2np}$  com os índices relativos às restrições ativas, onde  $na < 2np < nl * np$  é o número total das restrições ativas no ponto corrente.

A seguir detalhamos o cálculo do produto das matrizes  $Z$ ,  $Z^T$  e da Hessiana  $H$  por um vetor.

### 5.1.1 Fatoração $QR$ para o cálculo de $Z$

Como citamos anteriormente, não necessitamos armazenar a matriz  $Z$  e nem realizar a fatoração  $QR$  explícita da matriz  $A_{\mathcal{I}_A}$ . Porém nesta seção mostraremos as propriedades desta fatoração e como utilizamos suas informações para calcular os produtos das matrizes  $Z$  e  $Z^T$  por vetores.

Para a montagem do subproblema reduzido a cada iteração do algoritmo apresentado na seção 3.1.3, devemos ter informações sobre a matriz  $Z$  cujas colunas geram o núcleo da submatriz  $A_{\mathcal{I}_A}$ , submatriz de  $A$  apenas com as linhas referentes às restrições ativas.

Como descrito na seção 2.2, para obter a matriz  $Z$ , bastaria realizarmos uma fatoração  $QR$  da matriz  $A_{\mathcal{I}_A}^T$ . Neste caso teríamos  $\bar{Q}\bar{R} = A_{\mathcal{I}_A}^T$  com  $\bar{Q} \in \mathbb{R}^{2np \times 2np}$  e  $\bar{R} \in \mathbb{R}^{2np \times na}$ . A matriz  $Z$  seria então dada pelas últimas  $(2np - na)$  colunas de  $\bar{Q}$ . Porém não iremos calcular a matriz  $Z$  de forma explícita.

Para obter informações do cálculo da matriz  $\bar{Q}$ , devemos considerar o fato que tal matriz possui uma estrutura de esparsidade, assim como a matriz  $A_{\mathcal{I}_A}$ . Portanto trabalharemos primeiramente com os blocos desta matriz.

Sejam  $\mathcal{I}_i$ , para  $i = 1, \dots, np$ , os conjuntos de índices das restrições ativas para cada ponto  $P^i = (x_{2i-1}, x_{2i})^T$ . A cardinalidade  $n_i$  dos conjuntos  $\mathcal{I}_i$  varia entre 0 (ponto no interior do polígono, com nenhuma restrição ativa) e 2 (ponto em um vértice do polígono, com duas restrições ativas), e a soma de todas as cardinalidades é igual a  $na$ .

Vamos supor, a princípio, que temos a fatoração  $QR$  da transposta da matriz  $\bar{A}_{\mathcal{I}_i}$ ,  $Q^i R^i = \bar{A}_{\mathcal{I}_i}^T$ , para todo  $i = 1, \dots, np$ . Neste caso  $Q^i \in \mathbb{R}^{2 \times 2}$  para todo  $i$ , e  $R^i$  é uma matriz triangular superior com a mesma dimensão da matriz  $\bar{A}_{\mathcal{I}_i}^T$  correspondente, com duas linhas e no máximo duas colunas.

Podemos então representar as matrizes  $\bar{Q}$  e  $\bar{R}$  da seguinte forma:

$$\bar{R} = \begin{pmatrix} R^1 & 0 & \dots & 0 \\ 0 & R^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & R^{np} \\ & & & 0 \end{pmatrix}$$

e

$$\bar{Q} = \left( Q^1 \mid Q^2 \right), \quad (5.3)$$

onde

$$Q^1 = \begin{pmatrix} Q_1^1 & 0 & \dots & 0 \\ 0 & Q_2^1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Q_{np}^1 \end{pmatrix} \quad \text{e} \quad (5.4)$$

$$Q^2 = \begin{pmatrix} Q_1^2 & 0 & \dots & 0 \\ 0 & Q_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Q_{np}^2 \end{pmatrix}. \quad (5.5)$$

Assim, por (5.3) temos que

$$Z = Q^2 \quad (5.6)$$

Nesta notação, as matrizes  $Q_i = \begin{pmatrix} Q_i^1 & Q_i^2 \end{pmatrix} \in \mathbb{R}^{2 \times 2}$ ,  $i = 1, \dots, np$ , são as matrizes formadas a partir das fatorações  $QR$  das submatrizes  $\bar{A}_{I_i}^T$  que serão detalhadas a seguir.

O número de colunas de cada submatriz  $Q_i^1$  e  $Q_i^2$  varia para cada  $i = 1, \dots, np$  de acordo com o número de restrições ativas no respectivo ponto  $P^i$ . Com isso temos três casos distintos:

- *Ponto  $P^i$  é vértice do polígono, com número de restrições ativas  $n_i = 2$*

Neste caso  $Q_i^1 = Q_i$  com duas colunas. Já a matriz  $Q_i^2$  é vazia e, portanto, a coluna correspondente da matriz  $\bar{Q}$  não existirá.

- *Ponto  $P^i$  em um lado do polígono, com número de restrições ativas  $n_i = 1$*

Neste caso ambas matrizes  $Q_i^1$  e  $Q_i^2$  apresentam apenas uma coluna cada uma.

- *Ponto  $P^i$  é ponto interior do polígono, com número de restrições ativas  $n_i = 0$*

Neste caso  $Q_i^1$  é vazia e, portanto, a coluna correspondente da matriz  $\bar{Q}$  não existirá. Para a construção da matriz  $Q_i^2$  poderíamos utilizar  $Q_i^2 = Q_i$ , porém, como a fatoração  $\bar{Q}\bar{R}$  não é única, neste caso podemos fazer uma escolha mais adequada.

Podemos substituir a submatriz  $Q_i^2$  pela matriz identidade em  $\mathbb{R}^{2 \times 2}$ . Isso se deve ao fato de que a matriz  $Q^2$  é construída para ser o complemento ortogonal das primeiras colunas  $Q^1$  da matriz  $\bar{Q}$  de tal forma que a fatoração  $QR$  para  $A_{I_A}^T$  seja

satisfeita:

$$A_{\mathcal{I}_A}^T = \left( Q^1 \mid Q^2 \right) \bar{R}.$$

A escolha de  $Q^2$  não influencia a multiplicação  $\bar{Q}\bar{R}$ , desde que as últimas linhas da matriz  $R$  são nulas. Assim, devemos apenas assegurar a ortogonalidade de  $\bar{Q}$ . Pela estrutura de esparsidade desta matriz e pelo fato da submatriz  $Q_i^1$  ser vazia, basta tomar  $Q_i^2$  uma matriz ortogonal arbitrária em  $\mathbb{R}^{2 \times 2}$ , que pode ser a matriz identidade.

A seguir apresentamos em maiores detalhes como obter informações das matrizes  $Q_i$  e como utilizar estas informações para o cálculo do produto das matrizes  $Z$  e  $Z^T$  por um vetor.

### Informações sobre $Q_i$

As matrizes  $Q_i \in \mathbb{R}^{2 \times 2}$ ,  $i = 1, \dots, np$ , são originadas da fatoração  $QR$  das submatrizes  $\bar{A}_{\mathcal{I}_i}^T \in \mathbb{R}^{2 \times n_i}$ . Essas matrizes são facilmente obtidas através de apenas uma rotação de Givens responsável por zerar o segundo elemento da primeira coluna de  $\bar{A}_{\mathcal{I}_i}^T$ :

$$Q_i = \begin{pmatrix} \cos(\theta_i) & -\text{sen}(\theta_i) \\ \text{sen}(\theta_i) & \cos(\theta_i) \end{pmatrix},$$

onde  $\theta_i$  é o ângulo da rotação de Givens correspondente.

Vale ressaltar que  $\theta_i$  está diretamente relacionado ao índice da primeira restrição ativa do conjunto  $\mathcal{I}_i$ . E por esta razão podemos ter  $Q_i = Q_j$  para  $i \neq j$ . Assim, ao implementar o algoritmo não precisamos calcular  $\theta_i$  a cada iteração.

Na implementação apenas armazenamos uma única vez as informações na matriz

$$G = \begin{pmatrix} \cos(\theta_1) & \cos(\theta_2) & \dots & \cos(\theta_{nl}) \\ \text{sen}(\theta_1) & \text{sen}(\theta_2) & \dots & \text{sen}(\theta_{nl}) \end{pmatrix}.$$

Para cada ponto  $P^i$  teremos  $\theta_i$  correspondente ao índice da primeira restrição ativa do ponto.

## Produtos da matriz $Z$ por um vetor

Com as informações armazenadas na matriz  $G$  e a estrutura de esparsidade de  $Z$ , podemos calcular os produtos desta matriz por um vetor sem o armazenamento explícito de  $Z$ .

Para ilustrar este cálculo utilizaremos um exemplo. Suponhamos que nosso problema tem  $np = 5$  pontos, o polígono tem  $nl = 3$  lados e em uma determinada iteração temos:  $\mathcal{I}_1 = \{1\}$ ,  $\mathcal{I}_2 = \{2, 3\}$ ,  $\mathcal{I}_3 = \{\}$ ,  $\mathcal{I}_4 = \{2\}$  e  $\mathcal{I}_5 = \{1, 2\}$ .

Então a matriz  $Z$  é dada por:

$$Z = \begin{pmatrix} Z_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & Z_3 & 0 & \\ 0 & 0 & 0 & Z_4 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

onde

$$Z_1 = \begin{pmatrix} -\text{sen}(\theta_1) \\ \text{cos}(\theta_1) \end{pmatrix} \quad Z_3 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{e} \quad Z_4 = \begin{pmatrix} -\text{sen}(\theta_2) \\ \text{cos}(\theta_2) \end{pmatrix}.$$

Note que  $Q_2^2$  e  $Q_5^2$  são vazias, por isso não existem as submatrizes  $Z_2$  e  $Z_5$  correspondentes.

Neste exemplo temos então a multiplicação de  $Z$  por um vetor  $v = (v_1, v_2, v_3, v_4)^T \in \mathbb{R}^4$  dada por:

$$Zv = \begin{pmatrix} -\text{sen}(\theta_1)v_1 \\ \text{cos}(\theta_1)v_1 \\ 0 \\ 0 \\ v_2 \\ v_3 \\ -\text{sen}(\theta_2)v_4 \\ \text{cos}(\theta_2)v_4 \\ 0 \\ 0 \end{pmatrix}.$$

De forma análoga também implementamos uma rotina para o cálculo dos produtos da matriz  $Z^T$  por um vetor utilizando apenas as informações dos ângulos  $\theta_i$  armazenados

inicialmente na matriz  $G$ .

A seguir apresentamos a segunda abordagem para a resolução do mesmo problema utilizando o método de barreiras.

## 5.2 Abordagem com método de barreiras

A segunda maneira para resolver o problema dos pontos no polígono é utilizando o algoritmo apresentado na seção 4.1.

A cada iteração interna resolvemos então o subproblema:

$$\begin{aligned} \text{Min} \quad & f(x) + \frac{1}{\rho} B(u) \\ \text{suj. a} \quad & Ax - u = b, \end{aligned} \tag{5.7}$$

onde  $u \in \mathbb{R}^{nl*np}$  e  $B(u) = -\sum_{i=1}^{nl*np} \log(u_i)$ .

Para melhorar o desempenho computacional, reordenamos as variáveis e definimos uma única variável expandida  $x_{exp}$  da seguinte forma:

$$x_{exp} = \begin{pmatrix} x_1 \\ x_2 \\ u_1 \\ \vdots \\ u_{nl} \\ x_3 \\ x_4 \\ u_{nl+1} \\ \vdots \\ u_{2nl} \\ \vdots \\ x_{2np-1} \\ x_{np} \\ u_{(nl-1)np+1} \\ \vdots \\ u_{nl*np} \end{pmatrix}.$$

Com esta reordenação, o problema (5.7) pode ser escrito da seguinte forma:

$$\begin{aligned} \text{Min} \quad & \bar{f}(x_{exp}) \\ \text{suj. a} \quad & Ax_{exp} = b, \end{aligned} \tag{5.8}$$

onde  $A \in \mathbb{R}^{nl*np \times (2+nl)*np}$  é expressa por

$$A = \begin{pmatrix} A_{exp} & 0 & 0 & \dots & 0 \\ 0 & A_{exp} & 0 & \dots & 0 \\ 0 & 0 & A_{exp} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & A_{exp} \end{pmatrix} \tag{5.9}$$

e  $A_{exp} \in \mathbb{R}^{nl \times (2+nl)}$  é dada por

$$A_{exp} = \begin{pmatrix} \bar{A} & -I \end{pmatrix},$$

com  $I$  a matriz identidade em  $\mathbb{R}^{nl \times nl}$ .

Com esta reordenação não precisamos da fatoração  $QR$  explícita da matriz  $A$ , apenas da matriz reduzida  $A_{exp}$  cuja dimensão depende apenas da quantidade de lados  $nl$  do polígono, independentemente da quantidade de pontos  $np$  do problema.

Na próxima seção mostramos como obter as informações necessárias para o cálculo dos produtos da matriz  $Z$  por vetores, utilizando apenas a decomposição  $QR$  da matriz  $A_{exp}$ .

### 5.2.1 Informações da matriz $Z$

A vantagem desta abordagem em comparação à anterior é o fato de termos restrições de igualdade e, portanto, todas as restrições são consideradas ativas. Isso faz com que não precisamos armazenar informações sobre os índices das restrições ativas repercutindo também na construção da matriz  $Z$ , que se torna mais simples.

Primeiro vamos supor que temos uma fatoração  $QR$  da transposta da matriz  $A_{exp} = \begin{pmatrix} \bar{A} & -I \end{pmatrix}$  com

$$Q \begin{pmatrix} R \\ 0 \end{pmatrix} = A_{exp}^T,$$

onde  $Q \in \mathbb{R}^{(nl+2) \times (nl+2)}$  é uma matriz ortogonal e  $R \in \mathbb{R}^{nl \times nl}$  é uma matriz triangular superior.

A decomposição  $QR$  da transposta da matriz  $A$  seria então dada por  $A^T = \bar{Q}\bar{R}$  com

$$\bar{R} = \begin{pmatrix} R & 0 & \dots & 0 \\ 0 & R & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & R \\ & & & 0 \end{pmatrix}$$

e

$$\bar{Q} = \left( Q^1 \mid Q^2 \right), \quad (5.10)$$

onde

$$Q^1 = \begin{pmatrix} Q(1 : nl) & 0 & \dots & 0 \\ 0 & Q(1 : nl) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Q(1 : nl) \end{pmatrix} \quad e \quad (5.11)$$

$$Q^2 = \begin{pmatrix} Q(nl+1 : nl+2) & 0 & \dots & 0 \\ 0 & Q(nl+1 : nl+2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Q(nl+1 : nl+2) \end{pmatrix}. \quad (5.12)$$

Nesta notação  $Q(1 : nl)$  representa as primeiras  $nl$  colunas da matriz  $Q$  e  $Q(nl+1 : nl+2)$  representa as duas últimas colunas de  $Q$ .

Desta forma, a matriz  $Z$  é formada utilizando apenas informações das duas últimas colunas da matriz  $Q$ , que comporão a matriz  $Q^2$ :

$$Z \equiv Q^2. \quad (5.13)$$

Lembramos novamente que não armazenamos a matriz  $Z$  explicitamente, apenas utilizamos essas informações para o cálculo de produtos das matrizes  $Z$  e  $Z^T$  por vetores. Portanto, apenas precisamos armazenar explicitamente as duas últimas colunas da matriz  $Q$ .

# Capítulo 6

## Experimentos computacionais

Para a análise dos algoritmos propostos neste trabalho, utilizamos uma família de problemas de pontos em polígonos gerados em torno da origem de maneira aleatória porém em uma região pré-definida de forma a preservar a convexidade dos polígonos.

Uma explicação mais detalhada da geração destes problemas é dada na seção 6.2.

Antes de comparar o desempenho dos algoritmos para os problemas gerados, analisamos o desempenho do algoritmo de Rojas, Santos e Sorensen na resolução de subproblemas quadráticos com matriz Hessiana indefinida e restrições lineares sem uma estrutura específica, em uma bola.

Embora a principal família de problemas-teste considerada neste trabalho possua uma estrutura esparsa especial para as restrições lineares, a análise do algoritmo de Rojas, Santos e Sorensen para a resolução de subproblemas com restrições lineares não estruturadas oferece um indicativo do desempenho deste algoritmo em situações mais gerais. A comparação é feita com o algoritmo clássico de Moré e Sorensen [87] para a resolução deste tipo de problemas, como descrito na próxima seção.

### 6.1 Sobre o desempenho de LSTRS

Para avaliar o desempenho do algoritmo de Rojas, Santos e Sorensen na solução de problemas quadráticos indefinidos com restrições lineares não estruturadas em uma bola, preparamos inicialmente um conjunto de experimentos com problemas dessa natureza, com dados gerados aleatoriamente.

Estes experimentos foram realizados em MATLAB, versão 7.0.0.19920 (R14), em um Pentium (R) Intel (R) com 3.40 GHz de CPU e 1,99 GB de RAM.

Na primeira família de testes foram geradas matrizes indefinidas  $H \in \mathbb{R}^{n \times n}$  para as Hessianas das quadráticas e matrizes  $A \in \mathbb{R}^{m \times n}$  com esparsidade variável para formar as restrições lineares  $As = 0$ . Os problemas considerados têm a seguinte forma:

$$\begin{aligned} \min \quad & \frac{1}{2}s^T H s + g^T s \\ \text{s.a} \quad & A s = 0 \\ & \|s\| \leq \Delta. \end{aligned} \tag{6.1}$$

O controle da esparsidade foi feito fixando-se o número de elementos não nulos por linha nas duas matrizes do problema. Na Hessiana, a simetria foi assegurada fazendo-se  $H = (H + H^T)/2$ .

Uma vez determinada uma matriz ortogonal  $Z \in \mathbb{R}^{n \times n-m}$  tal que  $Im(Z) = Nu(A)$ , obtemos o problema reduzido:

$$\begin{aligned} \min \quad & \frac{1}{2}w^T Z^T H Z w + g^T Z w \\ \text{s.a} \quad & \|w\| \leq \Delta. \end{aligned} \tag{6.2}$$

Para podermos avaliar o desempenho de LSTRS na solução de problemas da forma (6.2), originados de (6.1) após o cálculo da matriz  $Z$ , geramos inicialmente uma família de testes com a matriz  $H$  esparsa (cerca de 2% de elementos não nulos por linha), com três possíveis dimensões:  $n \in \{1000, 2000, 3000\}$ , matrizes  $A$  de dimensões e densidades variáveis ( $m \in \{50, 100, 200, 500\}$  com 2, 20 ou 200 elementos diferentes de zero por linha, com os quais quantificamos o custo da solução em termos da dimensão e dos demais elementos envolvidos. Vale dizer que LSTRS foi acionado com todos os critérios de parada sugeridos como *default*, e na grande maioria dos testes a convergência se deu para uma solução quase-ótima (**Info** = 2, *quasi-optimal solution*).

Para cada escolha das dimensões  $n$  e  $m$  e do número de elementos não nulos em  $A$  (denotado por NZA), foram gerados e resolvidos cinco problemas (6.2) usando o algoritmo LSTRS. As médias do número de iterações executadas, produtos matriz-vetor e tempo de CPU em segundos estão expressas nas colunas **iter**, **MVP** e **CPU** das Tabelas 6.1, 6.2 e 6.3.

Com base nos resultados das Tabelas 6.1, 6.2 e 6.3 notamos que nem o tamanho dos problemas nem a densidade da matriz  $A$  parecem interferir de maneira significativa no desempenho do algoritmo LSTRS. Embora o tempo computacional dispendido aumente com o tamanho do problema, os números de iterações e de produtos matriz-vetor empregados mantiveram-se em uma mesma faixa de valores para este conjunto

$m$	NZA	iter	MVP	CPU
50	100	6.6	272.0	2.6840
	1000	6.4	313.8	2.9936
	10000	6.2	272.0	2.6216
100	200	6.0	267.4	2.4624
	2000	6.8	272.6	2.4996
	20000	6.2	281.4	2.5504
200	400	6.4	270.6	2.2690
	4000	6.4	309.4	2.5870
	40000	6.2	273.4	2.2996
500	1000	8.0	319.2	2.0676
	10000	6.0	252.6	1.6614
	100000	6.2	243.8	1.5958

Tabela 6.1: Resultados obtidos para os problemas em que  $H$  tem dimensão  $n = 1000$ .

$m$	NZA	iter	MVP	CPU
50	100	6.2	349.6	13.334
	1000	6.0	268.4	10.118
	10000	6.4	284.4	10.793
100	200	6.0	326.4	12.120
	2000	6.2	317.2	11.707
	20000	6.2	335.0	12.350
200	400	6.4	367.0	13.067
	4000	6.2	317.8	11.326
	40000	6.0	278.6	9.9438
500	1000	6.2	341.6	10.854
	10000	6.2	352.8	11.312
	100000	6.2	335.0	10.634

Tabela 6.2: Resultados obtidos para os problemas em que  $H$  tem dimensão  $n = 2000$ .

de testes.

Assumindo disponível a decomposição espectral da matriz  $Z^T H Z$ , consideramos o algoritmo que emprega a solução da equação secular associada ao problema (6.2), cf. [87], o qual denominamos de algoritmo espectral-secular (ES). Para contextualizar o desempenho de LSTRS, no segundo conjunto de experimentos comparamos os esforços dos algoritmos ES e LSTRS, em termos do tempo do CPU demandado para resolver o mesmo problema, também com matriz  $H$  indefinida. Embora a estratégia de Steihaug & Toint [110, 113] seja uma opção livre de fatorações e decomposições para tratar o

$m$	NZA	iter	MVP	CPU
50	100	6.0	328.6	28.299
	1000	6.2	332.0	27.564
	10000	6.0	311.8	25.853
100	200	6.2	339.2	27.847
	2000	6.0	317.2	26.010
	20000	6.0	328.2	26.883
200	400	6.0	326.0	26.089
	4000	6.0	301.2	24.471
	40000	6.0	319.4	25.592
500	1000	6.2	328.8	24.726
	10000	6.0	305.8	22.668
	100000	6.0	305.8	22.668

Tabela 6.3: Resultados obtidos para os problemas em que  $H$  tem dimensão  $n = 3000$ .

subproblema de região de confiança, ela determina apenas uma solução aproximada para esse problema. Por essa razão, preferimos adotar a estratégia de Moré & Sorensen [87] como *benchmark* para comparação, que, assim como LSTRS, também obtém uma solução exata para o subproblema, dentro da tolerância pré-estabelecida.

Além disso, como a estratégia empregada em LSTRS se fundamenta em um problema de autovalores parametrizado, optamos pela comparação deste com uma versão espectral do algoritmo de Moré-Sorensen, que ao invés de usar fatorações de Cholesky, constrói a função secular com as informações da decomposição espectral da matriz Hessiana do problema.

Notamos que uma única decomposição espectral de  $Z^T H Z$  produz os elementos necessários para resolver o problema (6.2) eficientemente, sem a necessidade de fatorações adicionais.

Para os testes comparativos entre ES e LSTRS consideramos a matriz Hessiana  $H$  densa e de dimensão fixa  $n = 1000$ , com elementos aleatoriamente gerados no intervalo  $[-10, 10]$ . A matriz  $A$  das restrições lineares possui dimensão  $m$  variável no conjunto

$$\mathcal{M} = \{10, 100, 200, 300, 400, 500, 600, 700, 800, 900, 990\} \quad (6.3)$$

e esparsidade também variável, controlada pelo número de elementos não nulos em cada linha, que tomamos como 10, 100 ou 1000, caso em que  $A$  também é densa. Seus elementos também foram gerados aleatoriamente em  $[-10, 10]$ . Como  $n = 1000$ , a

dimensão  $n - m$  dos problemas (6.2) considerados variou no mesmo conjunto  $\mathcal{M}$  dado em (6.3). Fixamos o raio  $\Delta = 5$  para todos os problemas.

Para cada escolha da dimensão  $m$  e do número de elementos não nulos em  $A$  (10, 100 ou 1000 em cada linha) foram gerados e resolvidos cinco problemas (6.2) usando os algoritmos ES e LSTRS. As médias dos tempos de CPU empregados pelos algoritmos, em segundos, estão expressas nas colunas  $\text{CPU}_{\text{ES}}$  e  $\text{CPU}_{\text{LSTRS}}$  das Tabelas 6.4–6.6, com os dados correspondentes ilustrados nas Figuras 6.1–6.3, respectivamente.

Nestes resultados não há diferenças sensíveis em termos do grau de esparsidade da matriz  $A$ . De fato, os três conjuntos de dados e de gráficos são bastante semelhantes.

Como esperado, o algoritmo ES demanda menos tempo de CPU que LSTRS para problemas com dimensões menores, o que pode ser acompanhado pelos gráficos da direita nas Figuras 6.1–6.3, mas o esforço por ele requerido cresce sensivelmente à medida que a dimensão do problema aumenta. Já o tempo dispendido pelo algoritmo LSTRS varia de maneira quase independente da dimensão, conforme indica a poligonal com os pontos # nos gráficos da esquerda nas Figuras 6.1–6.3.

$n - m$	$\text{CPU}_{\text{ES}}$	$\text{CPU}_{\text{LSTRS}}$
10	0.0160	0.1686
100	0.0470	0.5002
200	0.3120	0.7754
300	0.7190	1.009
400	1.657	0.4252
500	3.359	1.050
600	6.297	1.538
700	9.843	0.4810
800	14.42	1.222
900	22.09	2.000
990	33.39	2.350

Tabela 6.4: Resultados comparativos para problemas gerados com 10 elementos não nulos em cada linha da matriz  $A$ .

Para completar esta análise e ilustrar nossa exposição, na Tabela 6.7 apresentamos os tempos médios de CPU gastos para efetuar uma decomposição espectral densa (rotina `eig`) da matriz  $H$ , de dimensão  $n$ , considerando a média de 5 rodadas, com 5 diferentes matrizes de mesma dimensão, em que  $n \in \{1000, 2000, 3000, 4000, 5000\}$ . O custo teórico esperado  $\mathcal{O}(n^3)$  se confirma (cf. Anderson et al [4], Stewart [112])). Um modelo de quadrados mínimos para os dados da tabela produz o ajuste  $T(n) = 1.691 \times 10^{-9}n^3$ ,

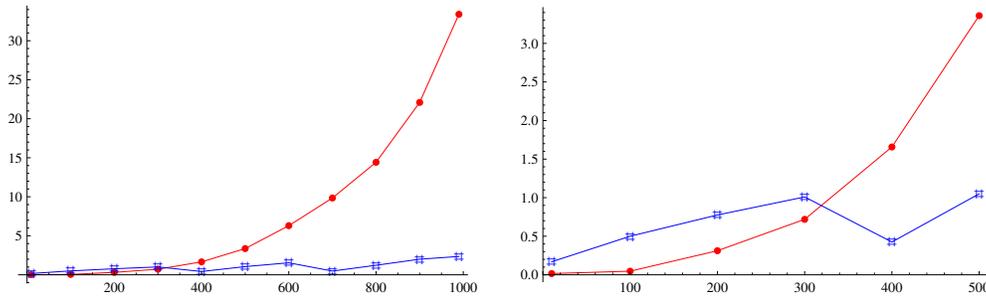


Figura 6.1: Comparação dos tempos médios de CPU gastos para resolver os problemas com 10 elementos não nulos em cada linha de  $A$ : algoritmo ES (pontos  $\bullet$ ) *versus* algoritmo LSTRS (pontos  $\#$ ). O gráfico da direita amplia o primeiro conjunto de pontos, para melhor visualização. Os dados correspondentes estão na Tabela 6.4.

$n - m$	$\text{CPU}_{\text{ES}}$	$\text{CPU}_{\text{LSTRS}}$
10	0.0150	0.1776
100	0.0460	0.5920
200	0.2490	0.5206
300	0.7320	0.8134
400	1.651	0.5892
500	3.491	1.829
600	5.967	1.313
700	10.13	1.022
800	14.78	0.6320
900	24.21	1.838
990	30.01	2.199

Tabela 6.5: Resultados comparativos para problemas gerados com 100 elementos não nulos em cada linha da matriz  $A$ .

ilustrado na Figura (6.4).

Concluimos que o desempenho do algoritmo LSTRS para os subproblemas quadráticos (6.1), tratados sob a forma (6.2), no conjunto de testes analisados, não se mostrou fortemente sensível ao aumento da dimensão dos problemas, como ocorre com algoritmos baseados em fatorações ou decomposições. Este comportamento favorece o uso de LSTRS como *solver* em métodos de região de confiança para resolver problemas de dimensão potencialmente alta.

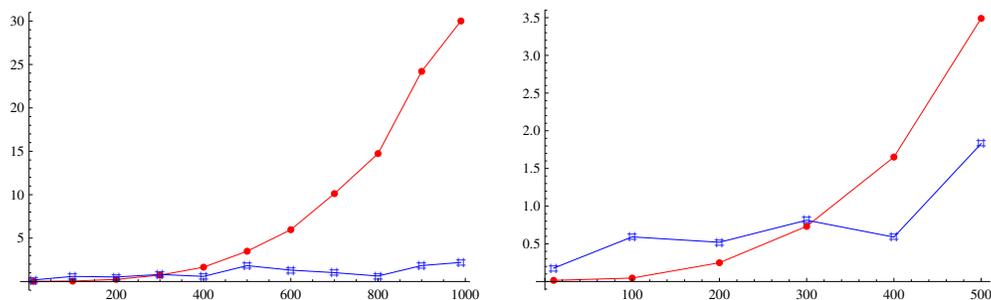


Figura 6.2: Comparação dos tempos médios de CPU gastos para resolver os problemas com 100 elementos não nulos em cada linha de  $A$ : algoritmo ES (pontos ●) *versus* algoritmo LSTRS (pontos #). O gráfico da direita amplia o primeiro conjunto de pontos, para melhor visualização. Os dados correspondentes estão na Tabela 6.5.

$n - m$	$\text{CPU}_{\text{ES}}$	$\text{CPU}_{\text{LSTRS}}$
10	0.0160	0.1594
100	0.0620	0.6938
200	0.2340	0.619
300	0.7500	0.5876
400	1.891	1.4782
500	3.891	2.0064
600	5.860	1.1468
700	10.06	2.3908
800	14.94	2.2919
900	21.58	1.4094
990	29.56	1.4688

Tabela 6.6: Resultados comparativos para problemas gerados com a matriz  $A$  densa.

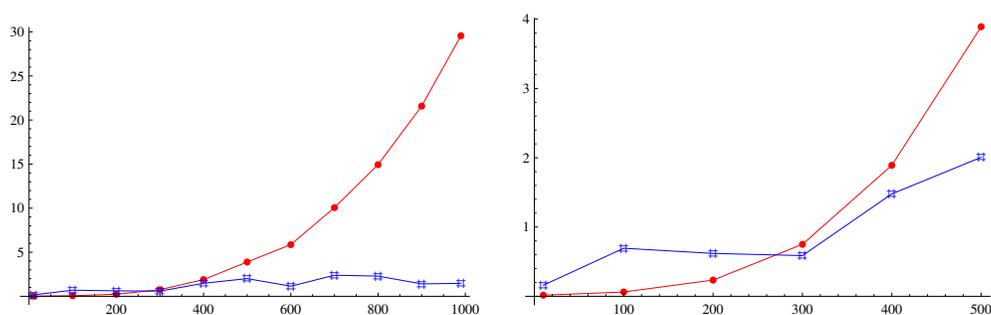


Figura 6.3: Comparação dos tempos médios de CPU gastos para resolver os problemas com matrizes  $A$  densas: algoritmo ES (pontos ●) *versus* algoritmo LSTRS (pontos #). O gráfico da direita amplia o primeiro conjunto de pontos, para melhor visualização. Os dados correspondentes estão na Tabela 6.6.

Dimensão ( $n$ )	Tempo de CPU (seg.)
1000	1.446
2000	11.68
3000	40.55
4000	101.3
5000	216.2

Tabela 6.7: Tempos médios de CPU para efetuar uma decomposição espectral densa (rotina `eig`) da matriz  $H$ , de dimensão  $n$ .

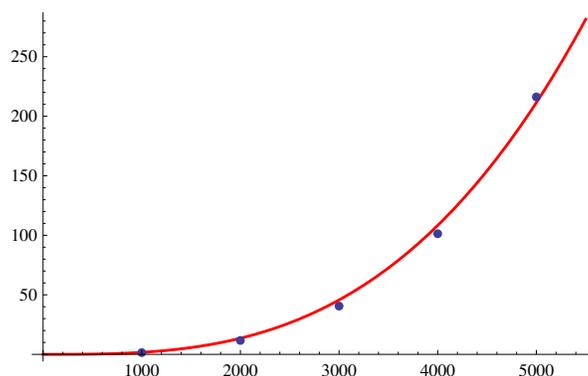


Figura 6.4: Gráfico da função  $T(n) = 1.691 \times 10^{-9} n^3$  que ajusta, via quadrados mínimos, os dados da Tabela 6.7 (pontos destacados). O eixo das abcissas contém as dimensões testadas e, nas ordenadas, aparece o tempo gasto de CPU, em segundos.

## 6.2 Geração dos problemas dos pontos no polígono

Para os testes dos algoritmos apresentados neste trabalho utilizamos o problema dos pontos em um polígono com a geração aleatória de seus lados em uma região pré-determinada e pontos iniciais viáveis no interior do polígono.

Vale ressaltar que a geração dos vértices do polígono deve ser feita de maneira cuidadosa para que consigamos um polígono convexo.

Inicialmente fixamos o número de lados do polígono, denotado por  $nl$ , e geramos  $nl$  vértices de maneira ordenada em uma coroa circular em volta da origem. Para facilitar a geração dos vértices, trabalhamos inicialmente com coordenadas polares e depois convertemos para coordenadas cartesianas.

Sejam  $r_{min}$  e  $r_{max}$  o menor e o maior raio, respectivamente, que descrevem a coroa circular dada por  $C = \{x = (r, \theta) \mid r_{min} \leq r \leq r_{max}\}$ .

Para a geração dos vértices  $v^i = (r_i, \theta_i)$ ,  $i = 1, \dots, nl$ , dividimos esta coroa em  $nl$  partes iguais e fixamos aleatoriamente cada vértice em uma partição.

Para particionar a coroa em  $nl$  partes, inicialmente obtemos o ângulo máximo de cada partição dado por  $\omega = \frac{2\pi}{nl}$ . Para garantir a unicidade dos vértices e uma distância mínima entre eles, construímos as partições da coroa com uma folga de  $\Delta_\omega$  do ângulo máximo, como ilustrado na Figura 6.5. Para os testes computacionais utilizamos  $\Delta_\omega = \frac{\omega}{10}$  para os casos em que  $nl = 4$  e  $nl = 5$ . Para o caso em que  $nl = 3$ , utilizamos  $\Delta_\omega = \frac{\pi}{6}$ , garantindo assim a origem no interior do polígono. A importância da origem no interior do polígono é justificada pela escolha dos sinais na construção das restrições que descrevem os lados dos polígonos, descrita em mais detalhes adiante.

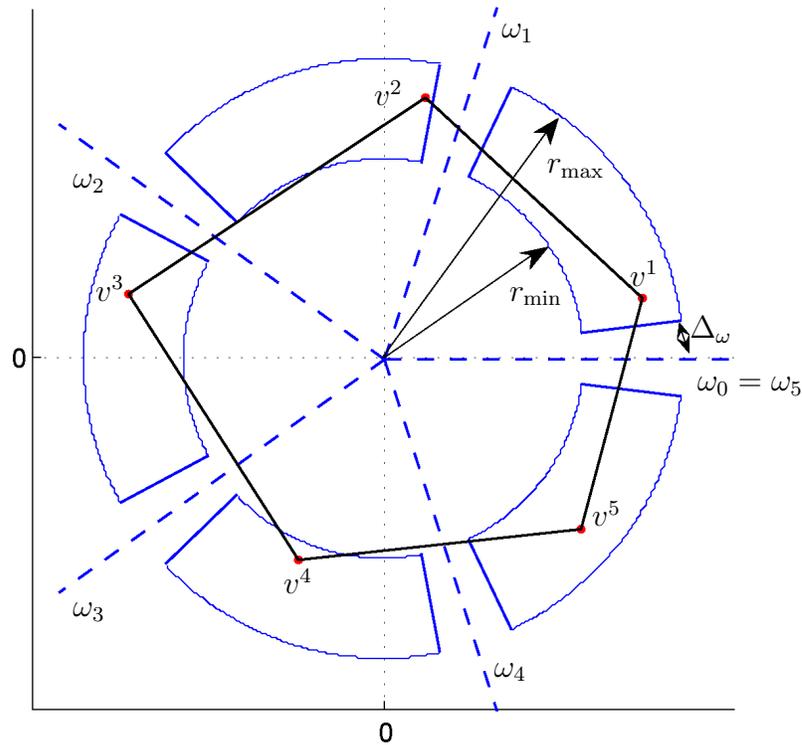


Figura 6.5: Exemplo da geração dos vértices de um polígono.

Notamos que a convexidade do polígono está associada à escolha de  $r_{\min}$  e  $r_{\max}$ . Quanto mais distante da origem  $r_{\max} > r_{\min} \gg 0$  e menor a diferença  $r_{\max} - r_{\min}$ , menor é a probabilidade de gerarmos uma estrutura não convexa. Na Figura 6.6 ilustramos a influência das escolhas de  $r_{\min}$  e  $r_{\max}$ .

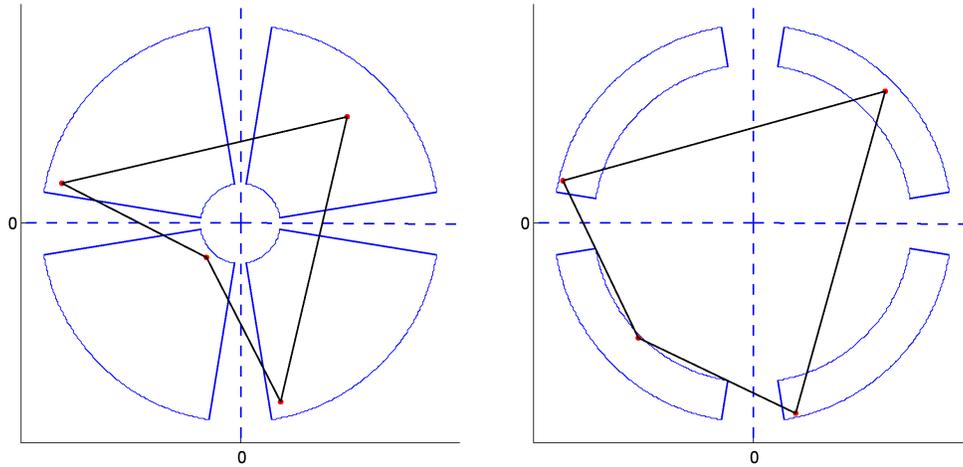


Figura 6.6: Exemplos da influência da escolha de  $r_{min}$  e  $r_{max}$  na geração dos polígonos: (a) geração de um polígono não convexo, (b) geração de um polígono convexo.

Fixados os valores de  $r_{min}$ ,  $r_{max}$  e  $\Delta\omega$ , geramos os vértices  $v^i = (r_i, \theta_i)$ ,  $i = 1, \dots, nl$ , de tal forma que  $r_i \in [r_{min}, r_{max}]$  e  $\theta_i \in [\omega_{i-1} + \Delta\omega, \omega_i - \Delta\omega]$ , onde  $\omega_i = i * \omega$ .

Em coordenadas cartesianas temos:

$$v^i = \begin{pmatrix} r_i \cos(\theta_i) \\ r_i \sin(\theta_i) \end{pmatrix}. \quad (6.4)$$

Com os vértices gerados temos a determinação das restrições lineares  $Ap^i \geq b$  por meio das retas que unem estes pontos.

Seja  $y_i = \tilde{a}x + \tilde{b}$  a reta que une os vértices  $v^i$  e  $v^{i+1}$ , com  $\tilde{a} = \frac{v_2^{i+1} - v_2^i}{v_1^{i+1} - v_1^i}$  e  $\tilde{b} = v_2^i - \tilde{a}v_1^i$ . E no caso em que  $v_1^{i+1} = v_1^i$ , seja a reta  $x = v_1^i$ .

Para a construção das restrições de desigualdade  $\begin{pmatrix} A_{i1} & A_{i2} \end{pmatrix} p \geq b_i$ , consideramos o fato de que a origem  $p = (0, 0)^T$  está no conjunto viável. Assim podemos tomar:

$$\begin{aligned} A_{i1} &= -\tilde{a}, & A_{i2} &= 1 & \text{e } b_i &= \bar{b}, & \text{se } \tilde{b} &\leq 0 \text{ e } v_1^i \neq v_1^{i+1}, \\ A_{i1} &= \tilde{a}, & A_{i2} &= -1 & \text{e } b_i &= -\bar{b}, & \text{se } \tilde{b} &> 0 \text{ e } v_1^i \neq v_1^{i+1}, \\ A_{i1} &= \tilde{a}, & A_{i2} &= 0 & \text{e } b_i &= \bar{b}, & \text{se } \tilde{b} &\leq 0 \text{ e } v_1^i = v_1^{i+1} \text{ e} \\ A_{i1} &= -\tilde{a}, & A_{i2} &= 0 & \text{e } b_i &= -\bar{b}, & \text{se } \tilde{b} &> 0 \text{ e } v_1^i = v_1^{i+1}. \end{aligned}$$

### 6.2.1 Primeira geração dos pontos iniciais

Para a geração dos  $np$  pontos iniciais, consideramos todos os pontos no interior do polígono.

A idéia inicial é construir um polígono auxiliar estritamente interior ao original com vértices  $v_{int}^i = 0.8v^i$ ,  $i = 1, \dots, nl$ . E então os pontos são gerados aleatoriamente em uma combinação convexa destes vértices. Na Figura 6.7 ilustramos a geração de pontos no interior de polígonos de quatro e cinco lados.

Poderíamos também utilizar uma combinação convexa dos vértices originais  $v^i$ , porém neste caso poderíamos ter pontos na face e nos próprios vértices. Como queremos tomar o mesmo ponto inicial para testar os dois algoritmos, o de restrições ativas e o de barreira, optamos por inicializar o processo com pontos estritamente no interior do polígono, utilizando o polígono auxiliar originado dos vértices  $v_{int}^i$  para a geração da combinação convexa.

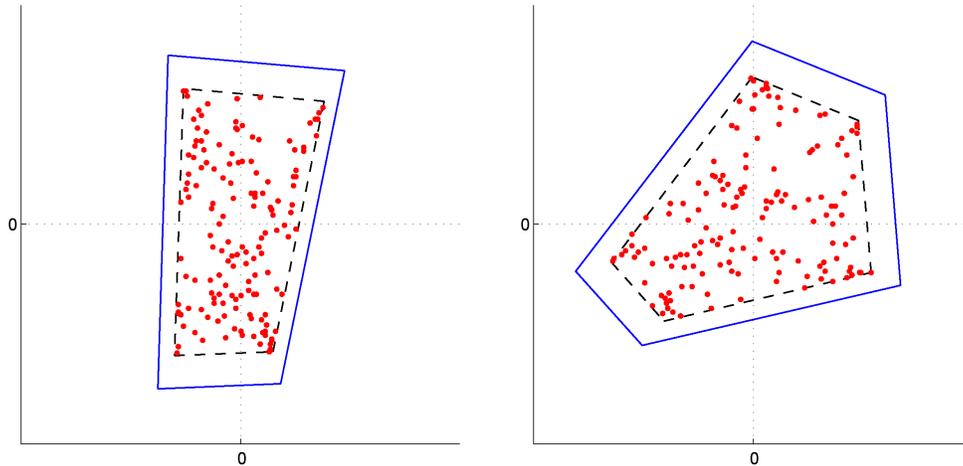


Figura 6.7: Exemplos para a geração dos pontos iniciais interiores em polígonos de 4 e 5 lados.

## 6.3 Testes preliminares

Para analisar o desempenho dos algoritmos para o conjunto de testes proposto, realizamos experimentos em MATLAB, versão 7.0.1.24704(R14), em um Intel Core i7 870 com 2.93 GHz de CPU e 8 GB de RAM. Infelizmente, o desempenho da máquina não foi o máximo possível, devido ao fato de utilizarmos uma versão 32 bits do MATLAB, com-

patível para o uso dos *mexfiles* do LSTRS, originalmente desenvolvido para o MATLAB 6.0.

Primeiro definimos as constantes e critérios de parada utilizados para cada algoritmo proposto e, em seguida, definimos a classe de problemas-teste analisados.

### 6.3.1 Constantes e critérios de parada utilizados

Para os dois algoritmos, o raio da região de confiança inicial foi definido como:

$$\Delta_0 = \frac{\|g_0\|}{10n},$$

onde  $n$  é a dimensão do problema e  $g_0$  denota o gradiente calculado no ponto inicial para a realização da primeira iteração do algoritmo de região de confiança.

Para a atualização do raio da região de confiança, baseada na comparação entre o decréscimo do modelo quadrático e da função objetivo, utilizamos as constantes com valores:  $\eta_0 = 0.1$ ,  $\eta_1 = 0.9$ ,  $\eta_2 = 1.1$  e  $\eta_3 = 1.5$ .

Outro parâmetro importante a ser atualizado é o parâmetro  $\rho_k$  do algoritmo de barreira. Ele contribui para a penalização do subproblema nos pontos não viáveis do problema original e quanto maior o seu valor, mais precisa é a aproximação da solução do subproblema com barreira para a solução do problema original. Porém, o aumento rápido deste parâmetro também pode contribuir para a produção de subproblemas mal condicionados, o que dificulta sua resolução.

Para os primeiros testes realizados, tomamos como parâmetro inicial  $\rho_0 = \frac{n}{2}$  e a sua atualização dada por

$$\rho_{k+1} = 2\rho_k.$$

A função barreira logarítmica escolhida é dada por:

$$B(u) = - \sum_{i=1}^p \text{lag}(u_i),$$

onde  $\text{lag}(u_i)$  é a função a qual denominamos *Lagarítmo*, criada com o objetivo de evitar, computacionalmente, cálculos envolvendo aritmética com números complexos, e é dada por:

$$lag(x) = \begin{cases} \log x & , \text{ se } x > 0 \\ -\infty & , \text{ se } x \leq 0. \end{cases}$$

Também são essenciais as escolhas dos parâmetros dos critérios de parada. Para compatibilizar os critérios dos dois algoritmos propostos, utilizamos a mesma filosofia para a parada dos algoritmos externos.

O critério relativo utilizado se baseia na norma do gradiente projetado no ponto corrente comparada com uma fração da norma do gradiente projetado calculado no ponto inicial. Assim, consideramos que o algoritmo convergiu quando

$$\|Z^T g\| \leq ep_{ext} \equiv 10^{-4} \max\{1, \|Z_0^T g_0\|\}. \quad (6.5)$$

Aqui vale ressaltar que para o algoritmo de barreira utilizamos os gradientes projetados dos vetores expandidos  $Z^T g \equiv Z_{exp}^T g_{exp}$ , com  $Z_{exp}$  a matriz cujas colunas geram o núcleo da matriz  $A_{exp}$  dada por (5.9).

Já para o algoritmo com restrições ativas, temos a projeção dos gradientes com a dimensão do problema original, e a matriz  $Z$  é a matriz cujas colunas geram o núcleo da matriz  $A_{\mathcal{I}_A}$ . Para este algoritmo, é preciso ainda considerar o sinal dos multiplicadores associados às restrições ativas, terminando o algoritmo quando estes multiplicadores tem sinal não negativo e o critério (6.5) for atingido.

Devemos ser cuidadosos ao escolher os parâmetros para o critério de parada do algoritmo de região de confiança interno para a resolução dos subproblemas de barreira.

O critério de parada escolhido foi também baseado nas normas dos gradientes projetados, o algoritmo interno parou quando:

$$\|Z^T g_k\| \leq ep_{int}, \quad (6.6)$$

lembrando que  $k$  representa a  $k$ -ésima iteração externa do método de barreira.

A escolha de  $ep_{int}$  é fundamental para o desempenho do algoritmo. Inicialmente realizamos testes utilizando  $ep_{int} \equiv ep_{ssF} = 0.1\|Z^T g_{k-1}\|$ . A desvantagem desta escolha é quando temos a norma do gradiente projetado inicial  $\|Z^T g_{k-1}\|$  grande, o que faz com que o algoritmo realize menos iterações internas para a resolução do primeiro subproblema, que apresenta o parâmetro  $\rho$  ainda pequeno.

Para forçar que o algoritmo atinja uma precisão maior ainda nas primeiras iterações,

introduzimos uma sequência forçante, utilizando

$$ep_{cSF} = \min\{0.1\|Z^T g_{k-1}\|, 4^{-k}\}. \quad (6.7)$$

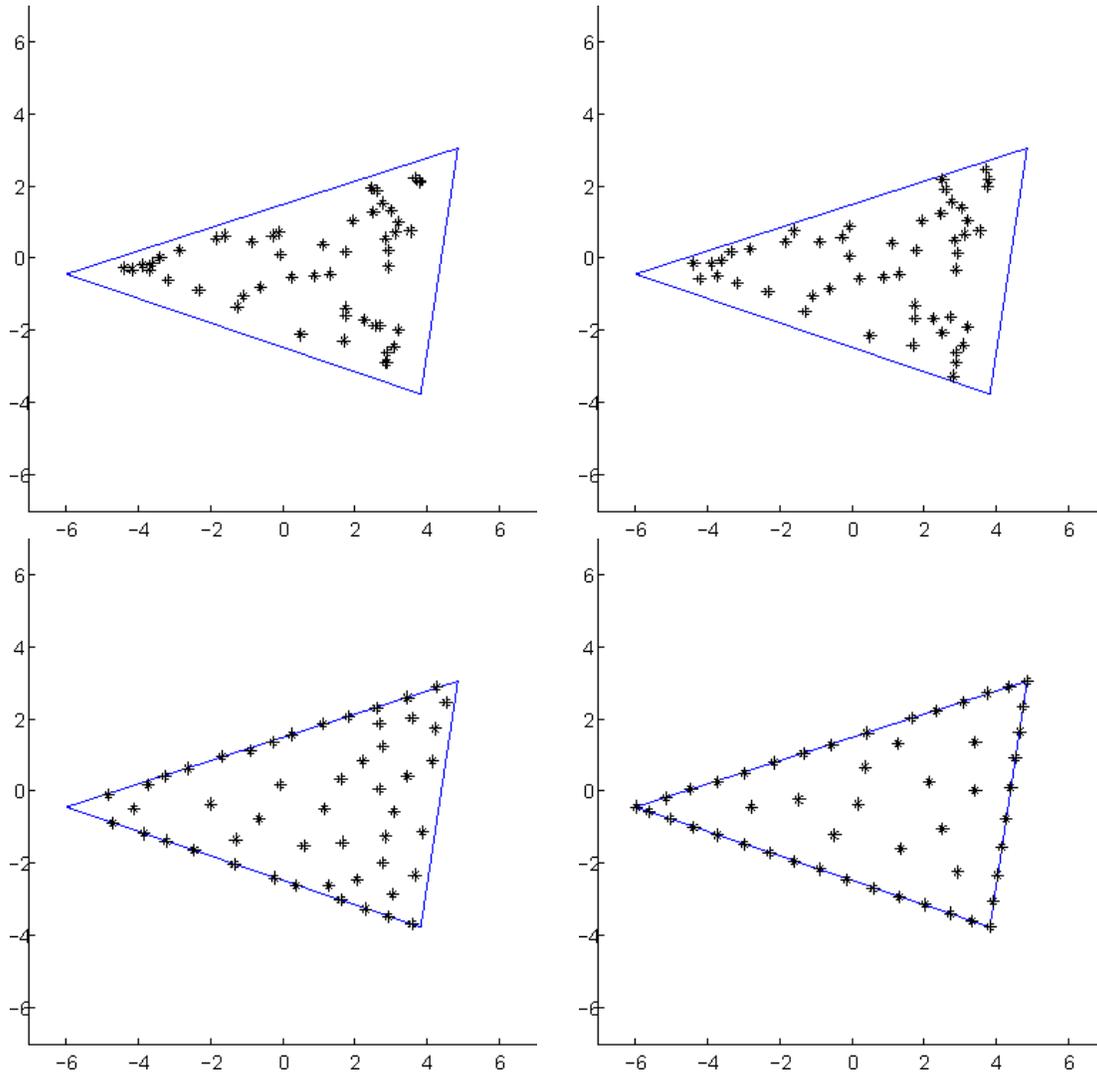


Figura 6.8: Resultados para um problema com um polígono de  $nl = 3$  lados e  $np = 50$  pontos iniciais com o algoritmo de barreira sem a introdução da sequência forçante para a parada interna. A primeira figura representa a configuração inicial dos pontos, seguida das saídas obtidas nas iterações externas subsequentes.

Os exemplos das Figuras 6.8 e 6.9 mostram o comportamento para as duas escolhas de  $ep_{int}$  para o mesmo problema. Na Figura 6.8 vemos a configuração inicial dos pontos e as três primeiras iterações do algoritmo sem a sequência forçante. Notamos uma

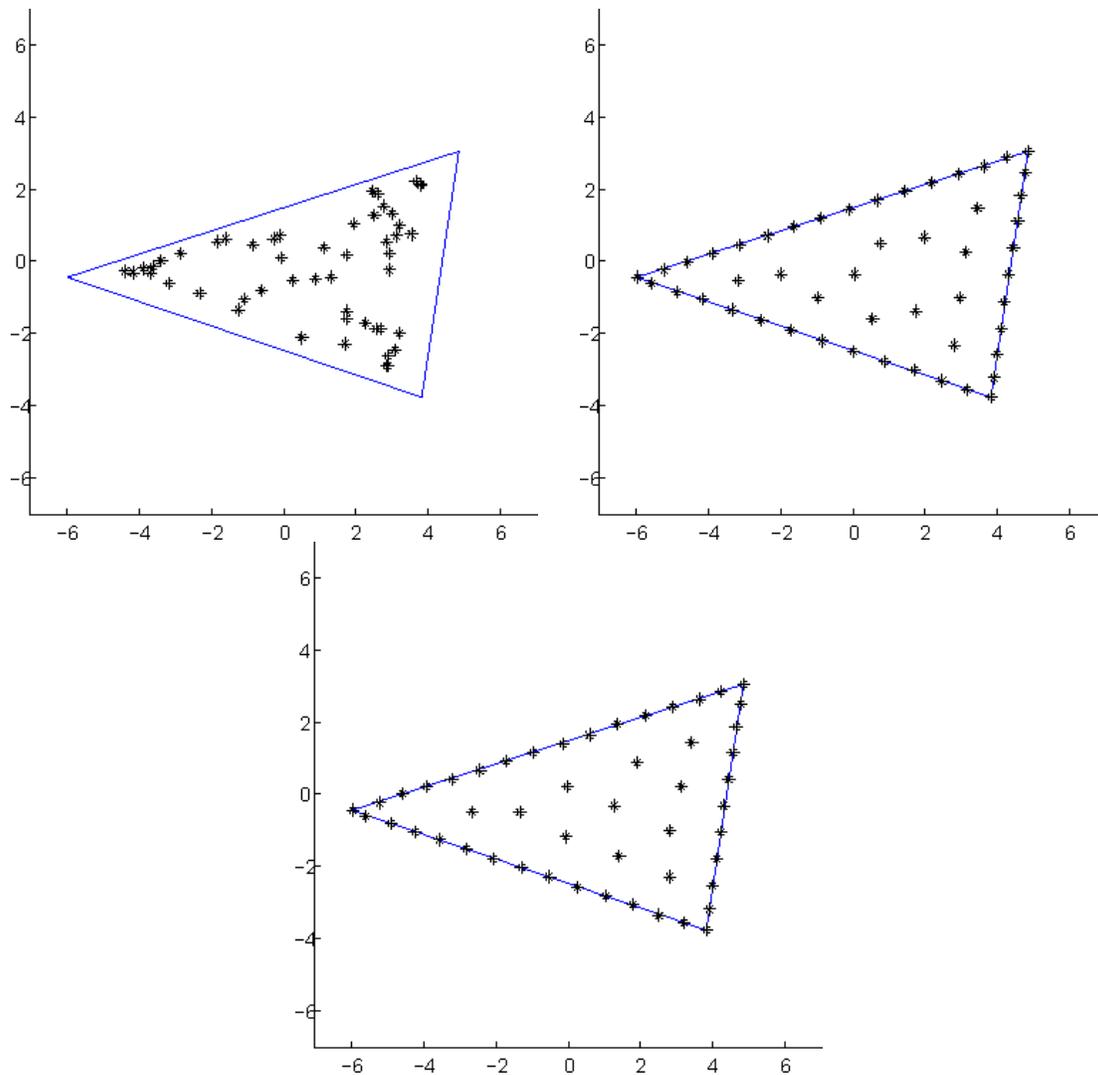


Figura 6.9: Resultados para um problema com um polígono de  $nl = 3$  lados e  $np = 50$  pontos iniciais com o algoritmo de barreira com a introdução da sequência forçante para a parada interna. A primeira figura representa a configuração inicial dos pontos, seguida das saídas obtidas nas iterações externas subsequentes.

parada prematura na primeira iteração, o que acarreta um maior esforço computacional posterior com o aumento de  $\rho_k$  nas iterações seguintes, originando problemas mais mal condicionados.

Este comportamento é evitado com a introdução da sequência forçante como vemos no exemplo da Figura 6.9. Neste caso, o algoritmo trabalha um pouco mais na primeira iteração, atingindo uma precisão maior, o que faz a convergência do algoritmo externo ocorrer em apenas 2 iterações, evitando o mau condicionamento dos problemas com o

aumento do parâmetro  $\rho_k$ .

Assim, o parâmetro para o critério de parada interna (6.6) foi tomado como:

$$ep_{int} = \max\{ep_{ext}, ep_{cSF}\},$$

com  $ep_{ext}$  dado em (6.5) e  $ep_{cSF}$  dado por (6.7).

### 6.3.2 Conjunto inicial de problemas-teste

Para os testes computacionais comparativos utilizamos o problema dos pontos em um polígono apresentado em (5.2). A constante para suavizar a função objetivo em (5.1) utilizada foi  $\xi = 10^{-4}$ .

O conjunto de testes inicial, cujos resultados são apresentados nas Tabelas 6.8–6.13, consiste em 18 tipos de problemas com polígonos de 3, 4 e 5 lados e diferentes configurações para os pontos iniciais. Para a geração destes polígonos utilizamos  $r_{min} = 5$ ,  $r_{max} = 6$ ,  $\Delta_\omega = \frac{\pi}{6}$  para os polígonos com 3 lados e  $\Delta_\omega = \frac{\omega}{10}$  para os polígonos com 4 ou 5 lados.

Os problemas de números 1–6 são de polígonos de três lados. Para a geração dos polígonos apresentados nas tabelas, utilizamos duas sementes distintas, sendo que os problemas 1, 3 e 5 e 2, 4 e 6 representam o mesmo polígono, respectivamente. Para cada polígono, temos três quantidades de pontos distribuídos: 20 (problemas 1 e 2), 40 (problemas 3 e 4) e 60 pontos (problemas 5 e 6). Além disso, utilizamos pontos iniciais diferentes para cada problema, cujos resultados estão expressos em cada linha do respectivo problema. As mesmas configurações iniciais foram utilizadas para ambos os métodos e os resultados estão mostrados nas Tabelas 6.8 (método de restrições ativas) e 6.9 (método de barreira).

De forma análoga, as Tabelas 6.10–6.11 e 6.12–6.13 apresentam os resultados para o problema com polígonos de 4 e 5 lados, respectivamente.

As tabelas contêm o número de pontos  $np$  no polígono, a dimensão  $Dim.$  do problema, o número de restrições  $Rest.$  e o tempo total de execução  $t(s)$  em segundos.

Para as Tabelas 6.8, 6.10 e 6.12 com os resultados do método de restrições ativas, temos: o número de iterações externas do algoritmo de região de confiança ( $it_{RC}$ ), o número total de modelos quadráticos construídos com alteração da matriz Hessiana ( $cont_H$ ), o total de iterações internas ( $it_{int}$ ), o número de restrições ativas na solução ( $ativ$ ), o valor da função objetivo no ponto final ( $f$ ), a norma do gradiente projetado

no ponto final ( $\|Z^T g\|$ ) e o número total de produtos de matriz por vetor ( $MVP_{total}$ ).

Nas Tabelas 6.9, 6.11 e 6.13 com os resultados do método com barreira, mostramos: o número de iterações externas do método de barreira ( $it$ ), o valor médio de iterações do algoritmo de região de confiança por iteração externa ( $\overline{it_{RC}}$ ), o valor médio da quantidade de modelos quadráticos construídos por iteração externa ( $\overline{cont_H}$ ), o valor da função objetivo no ponto final ( $f$ ), a norma do gradiente projetado do vetor expandido no ponto final ( $\|Z^T g_{exp}\|$ ) e o valor médio de produtos de matriz por vetor por iteração externa ( $\overline{MVP}$ ). Para alguns problemas o algoritmo de barreira não convergiu, sendo interrompidos com um critério extra de parada pelo tempo máximo ( $it_{max} = 12500s$ ) atingido no decorrer de uma iteração externa. Estes casos estão representados pelo símbolo \* ao lado do número de iterações externas. Nos demais casos, houve convergência a um ponto que satisfaz as condições de otimalidade de primeira ordem para o problema.

Na grande maioria dos testes, no ponto  $\bar{x}$  obtido como solução, foi verificado que a matriz Hessiana reduzida  $Z^T \nabla f(\bar{x}) Z$  é positiva definida, em que  $Im(Z) = Nu(A_{\mathcal{I}_A})$ , sendo  $\mathcal{I}_A$  o conjunto das restrições ativas em  $\bar{x}$ . As exceções estão marcadas por \* ao lado do valor da função objetivo em cada uma das Tabelas 6.8, 6.10 e 6.12.

<i>Prob.</i>	<i>np</i>	<i>Dim.</i>	<i>Rest.</i>	<i>it<sub>RC</sub></i>	<i>cont<sub>H</sub></i>	<i>itint</i>	<i>ativ</i>	<i>f</i>	$\ Z^T g\ $	<i>MVP<sub>tot</sub></i>	<i>t(s)</i>
1	20	40	60	20	15	44	21	50.61	0.0061	5807	28.5
				22	17	45	21	50.52	0.0027	5311	26.4
2	20	40	60	19	13	49	21	50.38	0.0195	6016	29.5
				19	14	47	21	50.32	0.0053	6077	29.8
3	40	80	120	27	21	58	34	239.14	0.0284	7528	137.2
				31	22	67	35	238.47	0.0060	11644	206.8
4	40	80	120	31	22	68	33	238.50	0.0355	8138	146.1
				29	23	61	33	238.53	1.6e-004	7597	137.8
5	60	120	180	30	21	88	46	576.77	0.0561	23418	909.8
				39	29	96	45	577.17	0.0226	16391	642.8
6	60	120	180	31	21	102	43	576.39	0.0824	16429	646.3
				34	26	81	43	576.41	0.0257	13657	538.1

Tabela 6.8: Resultados dos testes iniciais para polígonos de  $nl = 3$  lados utilizando o método de restrições ativas do capítulo 3.

Analisando as tabelas vemos um melhor desempenho para o método proposto com restrições ativas em todas as comparações. Este método resolveu todos os problemas em um tempo menor de execução, com menos iterações de região de confiança e com

<i>Prob.</i>	<i>np</i>	<i>Dim.</i>	<i>Rest.</i>	<i>it</i>	$\overline{It_{RC}}$	$\overline{cont_H}$	<i>f</i>	$\ Z^T g_{exp}\ $	$\overline{MVP}$	<i>t(s)</i>
1	20	40	60	2	44.5	30.5	51.28	0.0022	9642	89.7
				2	43	29	51.27	0.0013	9233.5	85.9
2	20	40	60	2	29	18.5	50.94	0.0031	9414	88.2
				3	26	18	50.58	0.0006	5709.3	79.7
3	40	80	120	2	137.5	96	238.91	0.0328	34608	1188.2
				3	110	83	239.80	0.0049	32508.7	1674.4
4	40	80	120	2	100	67.5	239.01	0.0380	28169.5	960.4
				3	52.3	33.3	238.20	0.0048	15760.7	819.9
5	60	120	180	2	470.5	290.5	576.44	0.0341	196970.5	14975.7
				2*	664	423.5	577.08	3.1725	278400.5	21133.0
6	60	120	180	2	181	106.5	574.98	0.0528	63278	4809.5
				2*	502.5	337	574.88	0.2508	248736	18816

Tabela 6.9: Resultados dos testes iniciais para polígonos de  $nl = 3$  lados utilizando o método com barreira do capítulo 4.

<i>Prob.</i>	<i>np</i>	<i>Dim.</i>	<i>Rest.</i>	<i>it<sub>RC</sub></i>	<i>cont<sub>H</sub></i>	<i>itint</i>	<i>ativ</i>	<i>f</i>	$\ Z^T g\ $	$\overline{MVP}_{tot}$	<i>t(s)</i>
7	20	40	80	20	14	39	24	44.73	0.0361	5249	25.9
				17	15	38	21	44.46	2.2e-004	4964	24.5
8	20	40	80	19	15	45	22	46.98	5.1e-004	5026	24.8
				16	15	31	24	47.21	6.3e-004	3338	16.9
9	40	80	160	25	19	55	32	210.38	0.0408	9877	177.4
				26	16	76	33	210.12 *	0.1438	12002	213.4
10	40	80	160	26	17	69	34	221.35	0.0255	10127	181.2
				29	19	67	34	221.43 *	0.1565	9865	178.3
11	60	120	240	28	20	88	44	505.89	0.0359	18121	704.5
				31	22	97	42	506.76	0.0576	14212	563.2
12	60	120	240	33	24	80	44	533.84 *	0.0608	10312	408.2
				30	20	80	44	534.00	0.1521	14354	566.7

Tabela 6.10: Resultados dos testes iniciais para polígonos de  $nl = 4$  lados utilizando o método de restrições ativas do capítulo 3.

menos avaliações de produtos matriz por vetor.

Outro fato que merece destaque é o número de iterações do algoritmo de região de confiança para os dois métodos. No caso do método com restrições ativas notamos um aumento pequeno no número de iterações com o aumento da dimensão do problema, como acontece, por exemplo, nos problemas 2, 4 e 6 da Tabela 6.8, com dimensões 40, 80 e 120, respectivamente. Para o primeiro ponto inicial gerado, temos o aumento do

<i>Prob.</i>	<i>np</i>	<i>Dim.</i>	<i>Rest.</i>	<i>it</i>	$\overline{It_{RC}}$	$\overline{cont_H}$	<i>f</i>	$\ Z^T g_{exp}\ $	$\overline{MVP}$	<i>t(s)</i>
7	20	40	80	2	30.5	19.5	45.08	0.0153	6768	63.2
				3	31.3	24.6	44.82	0.0001	6315.3	88.9
8	20	40	80	4	31	26.5	47.26	0.0001	6988.5	131.3
				3	44	36	47.36	0.0005	8225	116.7
9	40	80	160	2	74.5	46	210.63	0.0153	15516.5	535.8
				2	79.5	48.5	210.57	0.0532	16633.5	573.4
10	40	80	160	2	203	156.5	222.00	0.0076	38036.5	1324.8
				2	115	84	222.68	0.0839	23825	828.6
11	60	120	240	2	367	248.5	505.47	0.0130	117559.5	8956.9
				2	139	89	506.05	0.0329	37947	2896.2
12	60	120	240	2	306	209.5	534.10	0.0124	87320.5	6608.4
				2	346	219.5	535.21	0.0852	115521	8806.2

Tabela 6.11: Resultados dos testes iniciais para polígonos de  $nl = 4$  lados utilizando o método com barreira do capítulo 4.

<i>Prob.</i>	<i>np</i>	<i>Dim.</i>	<i>Rest.</i>	<i>it<sub>RC</sub></i>	<i>cont<sub>H</sub></i>	<i>itint</i>	<i>ativ</i>	<i>f</i>	$\ Z^T g\ $	$MVP_{tot}$	<i>t(s)</i>
13	20	40	100	17	15	39	22	40.09	0.0024	6691	32.4
				20	22	45	16	40.33	0.0025	6025	29.3
14	20	40	100	21	18	45	21	42.25	7.2e-004	6930	33.7
				16	14	40	22	42.02	7.7e-005	6128	30.2
15	40	80	200	28	18	72	32	188.49 *	0.1337	11725	208.3
				32	23	73	33	188.04	0.0075	9183	165.9
16	40	80	200	24	17	81	33	197.51 *	0.0279	14444	257.1
				27	20	75	34	197.11	0.0096	11597	206.6
17	60	120	300	38	25	110	42	452.61	0.1701	20891	808.4
				33	22	81	42	453.11 *	0.1074	10237	401.9
18	60	120	300	39	25	119	43	475.34	0.1870	21527	840.5
				30	24	75	42	475.91 *	0.0162	12045	476.8

Tabela 6.12: Resultados dos testes iniciais para polígonos de  $nl = 5$  lados utilizando o método de restrições ativas do capítulo 3.

número de iterações de 19 para 31 e então se mantém.

O mesmo comportamento não acontece para o método com barreira, para o qual temos por padrão o aumento significativo da média de iterações de região de confiança por iteração externa com o aumento da dimensão do problema. Em alguns casos temos um aumento de mais de dez vezes o número de iterações, como por exemplo nos problemas 7 e 11 da Tabela 6.11, com o aumento de  $\overline{It_{RC}}$  de 30.5 para 367 nos problemas

<i>Prob.</i>	<i>np</i>	<i>Dim.</i>	<i>Rest.</i>	<i>it</i>	$\overline{It_{RC}}$	$\overline{cont_H}$	<i>f</i>	$\ Z^T g_{exp}\ $	$\overline{MVP}$	<i>t(s)</i>
13	20	40	100	3	113.3	108.6	40.36	0.0001	20392	287.1
				3	124.7	117.4	40.36	0.0012	19819	279.2
14	20	40	100	4	42.5	34.2	42.42	0.0005	8327	155.7
				4	26.5	21.2	42.38	0.0001	5228.8	98.1
15	40	80	200	2	297.5	244	190.14	0.0371	53309	1847.4
				3	268.7	217.7	188.83	0.0035	50895	2644.7
16	40	80	200	3	82.7	60.4	198.05	0.0062	21505.3	1111.9
				3	94.7	73.4	197.32	0.0028	25952	1338.6
17	60	120	300	2*	810	570	454.01	0.7392	233734.5	17795
				2*	981	687.5	453.53	0.1146	252992.5	19186
18	60	120	300	2	295	210.5	475.36	0.0335	76107	5826.4
				2	582	409	474.81	0.0092	170180	12919

Tabela 6.13: Resultados dos testes iniciais para polígonos de  $nl = 5$  lados utilizando o método com barreira do capítulo 4.

de dimensão 40 e 120, respectivamente.

Podemos observar que o aumento do número de produtos matriz por vetor também tem uma característica semelhante ao compararmos os dois métodos, com um aumento mais significativo para o método com barreira.

Este comportamento desfavorável do método com barreira pode estar relacionado ao fato de que os seus subproblemas se tornam cada vez mais mal condicionados a cada iteração externa com o aumento do parâmetro de barreira  $\rho_k$ , o que contribui para a elevação do valor médio do número de iterações internas e produtos matriz por vetor.

Visando adequar as escolhas feitas para o método de barreira, apresentamos o estudo da próxima subseção.

### 6.3.3 Calibração do parâmetro $\rho_0$ para o método de barreira

Para analisar a sensibilidade da rapidez de convergência em relação à escolha do parâmetro de barreira, fizemos um breve estudo para alguns valores diferentes de  $\rho_0$ .

Comparamos a escolha original de  $\rho_0$  com este valor dividido por 2, 4, 8, 10, 12 e 16 com o objetivo de calibrar este parâmetro para a resolução de problemas com maior dimensão.

Alguns resultados estão mostrados nas Tabelas 6.14, 6.15 e 6.16 para problemas com 40, 60 e 80 pontos, respectivamente.

Nas tabelas estão apresentados o valor obtido para a função objetivo ( $f$ ), o número total de iterações de região de confiança ( $it_{total}$ ), o total de produtos matriz-vetor ( $MVP_{total}$ ), o tempo de execução ( $t(s)$ ) e as médias de produtos matriz-vetor por iteração ( $MVP/it$ ) e tempo por iteração ( $t/it$ ).

Observando as tabelas podemos notar que o tempo de execução total diminui com a diminuição de  $\rho_0$  de forma geral. Porém, a precisão da solução também é afetada com o relaxamento do parâmetro inicial. Por exemplo, para o problema com  $np = 40$  pontos e  $nl = 3$  lados obtivemos uma variação de 239.01 para 245.39 no valor da função objetivo na solução aproximada quando inicializamos com  $\rho_0$  originalmente proposto e  $\rho_0/16$ .

Portanto, apesar de aparentes melhores resultados para  $\rho_0$  pequenos, não podemos diminuí-lo tanto.

Analisando nas tabelas os tempos de execução e as médias  $MVP/it$  e  $t/it$ , notamos os melhores resultados para  $\rho_0/8$  e  $\rho_0/10$ , com uma ligeira vantagem para  $\rho_0/8$ .

Uma boa calibração de  $\rho_0$  é também fortemente dependente do problema, sendo influenciada não apenas pela sua dimensão, mas também pelo número de restrições.

O *solver* LOQO adota estratégias adaptativas para ajustar o parâmetro da barreira [117]. Recentemente, Nocedal, Wächter e Waltz [93] também propuseram estratégias adaptativas para ajustar o parâmetro de barreira em métodos de pontos interiores para PNL, as quais foram testadas nos ambientes algorítmicos de IPOPT e KNITRO.

Neste trabalho fixaremos a escolha  $\rho_0/8$  como parâmetro inicial para os testes maiores apresentados nas seções subsequentes.

A seguir vamos apresentar uma outra maneira de gerar os pontos iniciais para comparar os dois algoritmos propostos.

$nl$	$Rest.$	$\rho_0$	$f$	$it_{total}$	$MVP_{total}$	t(s)	$MVP/it$	$t/it$
3	120	$\rho_0$	239.01	200	56339	960	282	4.8
		$\rho_0/2$	238.54	213	62334	1094	293	5.1
		$\rho_0/4$	240.14	105	25586	446	244	4.2
		$\rho_0/8$	242.36	67	15350	270	229	4.0
		$\rho_0/10$	242.67	55	16781	293	305	5.3
		$\rho_0/12$	243.84	53	11663	208	220	3.9
		$\rho_0/16$	245.39	45	8918	161	198	3.6
4	160	$\rho_0$	210.63	149	31033	536	208	3.6
		$\rho_0/2$	210.94	133	28648	512	215	3.8
		$\rho_0/4$	211.98	115	23356	408	203	3.5
		$\rho_0/8$	212.15	81	19817	362	245	4.5
		$\rho_0/10$	214.49	70	17726	318	253	4.5
		$\rho_0/12$	215.18	55	15663	284	285	5.2
		$\rho_0/16$	217.25	58	18091	322	312	5.6
5	200	$\rho_0$	198.05	248	64516	1112	260	4.5
		$\rho_0/2$	198.52	166	35895	649	216	3.9
		$\rho_0/4$	198.76	135	28296	502	210	3.7
		$\rho_0/8$	199.57	100	22520	396	225	4.0
		$\rho_0/10$	199.98	93	21918	385	236	4.1
		$\rho_0/12$	200.60	66	16945	297	257	4.5
		$\rho_0/16$	204.67	62	16943	298	273	4.8

Tabela 6.14: Resultados de alguns testes com  $np = 40$  pontos para analisar a influência da variação do parâmetro de barreira inicial  $\rho_0$ .

$nl$	$Rest.$	$\rho_0$	$f$	$it_{total}$	$MVP_{total}$	t(s)	$MVP/it$	$t/it$
3	180	$\rho_0$	574.98	362	126556	4809	350	13.3
		$\rho_0/2$	576.44	172	48130	1852	280	10.8
		$\rho_0/4$	576.66	259	114267	4378	441	16.9
		$\rho_0/8$	578.01	157	53910	2073	343	13.2
		$\rho_0/10$	579.30	100	29991	1154	300	11.5
		$\rho_0/12$	580.86	91	25559	991	281	10.9
		$\rho_0/16$	583.03	69	19569	760	284	11.0
4	240	$\rho_0$	505.47	734	235119	8957	320	12.2
		$\rho_0/2$	505.88	471	148727	5839	316	12.4
		$\rho_0/4$	507.34	210	54603	2091	260	10.0
		$\rho_0/8$	508.98	153	33746	1306	221	8.5
		$\rho_0/10$	509.29	183	48890	1870	267	10.2
		$\rho_0/12$	510.45	102	22884	886	224	8.7
		$\rho_0/16$	512.54	92	23085	892	251	9.7
5	300	$\rho_0$	475.36	590	152214	5826	258	9.9
		$\rho_0/2$	476.13	239	50225	1950	210	8.2
		$\rho_0/4$	476.84	278	58305	2235	210	8.0
		$\rho_0/8$	478.29	135	26530	1028	197	7.6
		$\rho_0/10$	479.03	162	33026	1267	204	7.8
		$\rho_0/12$	480.58	136	28891	1132	212	8.3
		$\rho_0/16$	481.28	89	18400	720	207	8.1

Tabela 6.15: Resultados de alguns testes com  $np = 60$  pontos para analisar a influência da variação do parâmetro de barreira inicial  $\rho_0$ .

$nl$	$Rest.$	$\rho_0$	$f$	$it_{total}$	$MVP_{total}$	t(s)	$MVP/it$	$t/it$
3	240	$\rho_0$	1067.0	500	184683	12542	369	25.1
		$\rho_0/2$	1068.1	439	183705	12514	418	28.5
		$\rho_0/4$	1067.4	563	261171	17796	464	31.6
		$\rho_0/8$	1069.0	172	60489	4164	352	24.2
		$\rho_0/10$	1070.5	250	109532	7441	438	29.8
		$\rho_0/12$	1071.1	189	88560	5992	469	31.7
		$\rho_0/16$	1071.6	328	117426	8056	358	24.6
4	320	$\rho_0$	938.68	592	184108	12512	311	21.1
		$\rho_0/2$	937.04	865	279198	18974	323	21.9
		$\rho_0/4$	935.95	422	122215	8280	290	19.6
		$\rho_0/8$	937.95	342	86697	5906	254	17.3
		$\rho_0/10$	938.98	366	103365	6970	282	19.0
		$\rho_0/12$	940.57	764	212283	14340	278	18.8
		$\rho_0/16$	942.14	164	42720	2928	260	17.9
5	400	$\rho_0$	878.69	1013	351917	23953	347	23.6
		$\rho_0/2$	881.22	837	250626	17055	299	20.4
		$\rho_0/4$	879.77	553	149140	10193	270	18.4
		$\rho_0/8$	882.10	721	193360	13162	268	18.3
		$\rho_0/10$	882.70	260	68052	4662	262	17.9
		$\rho_0/12$	882.78	242	55295	3776	228	15.6
		$\rho_0/16$	884.82	256	59031	4032	231	15.8

Tabela 6.16: Resultados de alguns testes com  $np = 80$  pontos para analisar a influência da variação do parâmetro de barreira inicial  $\rho_0$ .

### 6.3.4 Outra estratégia para a escolha do ponto inicial

Para os testes preliminares, optamos por distribuir aleatoriamente os pontos iniciais em um polígono interior ao polígono original. Porém, ainda assim podemos ter uma concentração maior de pontos em uma determinada região do polígono, o que ocasiona uma lentidão mais acentuada nas primeiras iterações da resolução do problema, até que ocorra uma maior separação destes pontos.

Com a finalidade de obter uma configuração inicial mais bem distribuída, adotamos uma nova técnica para a obtenção dos pontos iniciais, como descrito na sequência.

Inicialmente construímos uma área interna delimitada pelos polígonos com vértices dados por  $0.9V$  e  $0.7V$ , onde  $V$  representa os vértices do polígono original. E então distribuímos os pontos nos quadriláteros internos formados pelos vértices destes polígonos menores, como ilustra a Figura 6.10.

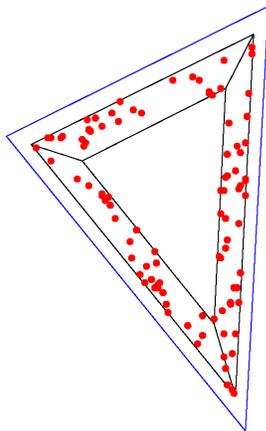


Figura 6.10: Exemplo da nova distribuição dos pontos iniciais.

A distribuição dos pontos em cada quadrilátero interno está relacionada ao tamanho do lado do polígono original correspondente, quanto maior o lado, mais pontos serão distribuídos no respectivo quadrilátero.

Para fazer isso, seja o  $i$ -ésimo lado do polígono  $L_i$  determinado pela reta que une os vértices  $V_i$  e  $V_{i+1}$ , para  $i = 1, \dots, nl - 1$  (ou  $V_{nl}$  e  $V_1$  para  $i = nl$ ), e seja  $L_{\hat{i}}$  o maior lado do polígono.

A idéia é distribuir os  $np$  pontos de tal forma que em cada quadrilátero interno tenhamos  $n_i$  pontos, com  $n_i > n_j$  para todo  $i$  e  $j$  tais que  $L_i > L_j$ .

Então, dado  $\bar{L} = \sum L_i$ , tomamos:

$$n_i = \lfloor \left(\frac{L_i}{\bar{L}}\right) \rfloor np, \quad \text{para } i \neq \hat{i}$$

$$\text{e } n_{\hat{i}} = np - \sum_{i \neq \hat{i}} n_i,$$

onde  $\lfloor (x) \rfloor$  indica a parte inteira de  $x$  e  $\hat{i}$  é o índice correspondente ao maior lado do polígono.

Após determinados quantos pontos estarão em cada quadrilátero interno, distribuimos estes  $n_i$  pontos como uma combinação convexa dos vértices  $0.9V_i$ ,  $0.7V_i$ ,  $0.9V_{i+1}$  e  $0.7V_{i+1}$ . Aqui vale ressaltar que também cuidamos para que a ordem das contas para a realização desta combinação convexa não influencie muito para a concentração dos pontos em apenas um vértice do quadrilátero interno.

Com esta alteração na escolha dos pontos iniciais, apresentamos os novos resultados para o método de restrições ativas nas Tabelas 6.17, 6.18 e 6.19.

Comparando estes resultados com os das Tabelas 6.8, 6.10 e 6.12, notamos um melhor desempenho do algoritmo para uma inicialização mais bem distribuída do ponto inicial.

Isso fica evidente quando analisamos o tempo de execução e o número de produtos matriz-vetor para o mesmo problema com os pontos iniciais diferentes. Os valores apresentados na Tabelas 6.17, 6.18 e 6.19 em geral são menores, mantendo-se a qualidade obtida no valor da função objetivo.

Para todos os testes realizados até esta etapa, comparando os valores obtidos para a função objetivo, fixado o número de pontos do problema, à medida que o número de lados do polígono aumenta, o valor ótimo encontrado diminui. Percebemos que isso acontece por que a área dos polígonos gerados diminui com o aumento do número de lados, com a escolha fixa de  $r_{min}$  e  $r_{max}$  para todos os problemas. Visando corrigir essa distorção, na próxima seção redefinimos tais escolhas.

<i>Prob.</i>	<i>np</i>	<i>Dim.</i>	<i>Rest.</i>	<i>it<sub>RC</sub></i>	<i>cont<sub>H</sub></i>	<i>itint</i>	<i>ativas</i>	<i>f</i>	$\ Z^T g\ $	<i>MVP<sub>tot</sub></i>	<i>t(s)</i>
1	20	40	60	12	11	30	21	50.61	0.0014	3658	18.1
				21	15	40	21	50.55	0.0436	4259	21.4
2	20	40	60	17	15	36	21	50.32	0.0034	4051	20.2
				19	12	42	20	50.62	0.0087	4768	23.6
3	40	80	120	29	19	71	35	238.43	0.1886	8938	160.7
				26	19	62	33	239.40	0.0376	8147	145.5
4	40	80	120	21	16	55	34	238.08	0.0301	7477	134.4
				21	16	53	33	238.62	0.0793	7861	140.7
5	60	120	180	31	20	72	44	577.93★	0.2267	10815	423.7
				29	20	82	46	576.88★	0.3678	14878	583.50
6	60	120	180	32	21	75	44	575.70★	0.3068	13633	533.1
				31	21	79	44	575.60	0.0578	11426	451.6

Tabela 6.17: Resultados dos testes alterando o ponto inicial para polígonos de  $nl = 3$  lados utilizando o método de restrições ativas do capítulo 3.

<i>Prob.</i>	<i>np</i>	<i>Dim.</i>	<i>Rest.</i>	<i>it<sub>RC</sub></i>	<i>cont<sub>H</sub></i>	<i>itint</i>	<i>pface</i>	<i>f</i>	$\ Z^T g\ $	<i>MVP<sub>tot</sub></i>	<i>t(s)</i>
7	20	40	80	13	12	37	22	44.57	0.0024	5179	25.3
				15	14	36	21	44.56	0.0062	4336	21.6
8	20	40	80	16	14	39	22	46.99	0.0029	4775	23.6
				15	13	36	21	47.13	0.0065	3660	18.4
9	40	80	160	28	19	57	32	210.60	0.0688	5382	97.9
				27	20	78	33	210.10	0.0333	12069	213.7
10	40	80	160	26	17	66	33	221.75★	0.3469	11770	207.7
				27	21	59	34	221.38	0.0188	8736	157.2
11	60	120	240	28	21	76	44	505.64	0.0302	16749	655.1
				31	22	79	42	507.00★	0.1004	10233	405.5
12	60	120	240	31	22	76	43	534.56	0.0562	11864	470.3
				23	19	65	44	534.30	0.0503	10855	428.5

Tabela 6.18: Resultados dos testes alterando o ponto inicial para polígonos de  $nl = 4$  lados utilizando o método de restrições ativas do capítulo 3.

<i>Prob.</i>	<i>np</i>	<i>Dim.</i>	<i>Rest.</i>	<i>it<sub>RC</sub></i>	<i>cont<sub>H</sub></i>	<i>itint</i>	<i>ativas</i>	<i>f</i>	$\ Z^T g\ $	<i>MVP<sub>tot</sub></i>	<i>t(s)</i>
13	20	40	100	14	13	37	22	40.09	7.2e-4	4521	22.3
				15	13	39	21	40.26	0.0061	3844	19.2
14	20	40	100	17	15	38	22	42.02	0.0023	6254	30.4
				17	15	40	23	42.00	4.4e-4	4953	24.4
15	40	80	200	22	18	62	33	188.00	0.0149	9771	174.2
				27	20	59	33	188.02	0.1028	8737	155.7
16	40	80	200	28	22	62	34	197.11	0.0111	6664	120.1
				25	20	56	33	197.49	0.0127	7869	140.5
17	60	120	300	26	22	67	43	451.99	0.0047	7847	314.0
				27	19	68	42	452.59	0.1245	12359	486.2
18	60	120	300	37	23	84	42	476.10★	0.1355	9588	379.6
				35	26	81	44	474.87	0.0238	14124	554.6

Tabela 6.19: Resultados dos testes alterando o ponto inicial para polígonos de  $nl = 5$  lados utilizando o método de restrições ativas do capítulo 3.

### 6.3.5 Escolhas de $r_{min}$ e $r_{max}$ para a construção dos polígonos

Antes de realizar os testes computacionais para problemas de maior dimensão, vamos redefinir as escolhas de  $r_{min}$  e  $r_{max}$ , discutidos na Secção 6.2 para a construção dos polígonos.

Ao realizar os testes iniciais, notamos que em alguns casos, as áreas dos polígonos gerados de 3, 4 e 5 lados não eram proporcionais, o que dificulta a comparação dos resultados. Em geral, os triângulos gerados na mesma coroa circular delimitada pelos raios  $r_{min}$  e  $r_{max}$  apresentam área muito inferior aos pentágonos gerados na mesma coroa, dificultando a distribuição de um número elevado de pontos nesta região.

Para tentar amenizar este problema, alteramos  $r_{min}$  e  $r_{max}$  dependendo do número de lados do polígono.

Fixamos os valores  $r_{min} = 5$  e  $r_{max} = 6$  para  $nl = 5$  e calculamos a área  $A_5$  do pentágono regular em uma circunferência de raio 5.5.

Então para  $nl = 3$  e  $nl = 4$  calculamos  $r_{medio}^3$  e  $r_{medio}^4$  tais que as áreas  $A_3$  e  $A_4$  do triângulo equilátero em uma circunferência de raio  $r_{medio}^3$  e do quadrado em uma circunferência de raio  $r_{medio}^4$ , respectivamente, fossem iguais a  $A_5$ . Desta forma obtivemos  $r_{medio}^3 \approx 7.5$  e  $r_{medio}^4 \approx 6$ .

Assim adotamos:

$$\begin{aligned} r_{min} &= 7 \quad \text{e} \quad r_{max} = 8, & \text{se } nl = 3, \\ r_{min} &= 5.5 \quad \text{e} \quad r_{max} = 6.5, & \text{se } nl = 4 \text{ e} \\ r_{min} &= 5 \quad \text{e} \quad r_{max} = 6, & \text{se } nl = 5. \end{aligned}$$

Na próxima seção apresentamos os testes finais e a análise dos resultados com estas modificações na geração dos problemas-teste.

## 6.4 Testes com as novas escolhas

Para as novas escolhas de ponto inicial e novas dimensões dos polígonos, realizamos testes para problemas com  $np = 20, 40, 60, 80, 100$  pontos. Os resultados para problemas com polígonos de 3, 4 e 5 lados estão apresentados nas Tabelas 6.20 – 6.25, onde estão mostrados o número de pontos ( $np$ ), o número de restrições ( $Rest.$ ), o total de iterações externas ( $it_{RC}$ , para o método de restrições ativas.  $it_{ext}$  para o método de barreira), o total de iterações internas ( $it_{int}$ ), o valor da função objetivo no ponto final ( $f$ ), a norma do gradiente projetado no ponto final ( $\|Z^T g\|$ ), o total de produtos matriz-vetor ( $MVP_{total}$ ), o tempo total em segundos ( $t$ ), as médias do total de produtos matriz-vetor por iteração interna ( $MVP/it_{int}$ ) e o tempo por produto matriz-vetor ( $t/MVP$ ).

Apresentamos também as médias do número de iterações internas, do tempo total e do total de produtos matriz-vetor para cada conjunto de testes de mesma dimensão.

Como nas tabelas anteriores, quando o algoritmo convergiu para um ponto cujas derivadas de segunda ordem não geram uma matriz positiva definida, representamos com o valor da função objetivo seguido de \* para o método de restrições ativas. Para o método de barreira, além dos critérios de parada discutidos anteriormente, acrescentamos a parada caso o tempo de execução de uma iteração interna ultrapasse 18000s (5 horas), representado nas tabelas pelo tempo total seguido de  $\star$ .

Vale ressaltar também que os problemas 1b–18b são equivalentes aos problemas 1–18 das tabelas anteriores, com o redimensionamento dos polígonos, alterando  $r_{min}$  e  $r_{max}$ .

Analisando as tabelas percebemos que o comportamento observado nos testes iniciais se mantém, com um melhor desempenho de forma geral para o método de restrições ativas. Este comportamento fica mais evidenciado quando comparamos o tempo de execução e o número de produtos matriz-vetor para cada método.

Outra característica que merece destaque é o fato da média de produtos matriz-vetor por iteração interna sofrer pouca alteração, tanto com o aumento da dimensão do problema, com o aumento do número de lados do polígono e consequente aumento do número de restrições do problema. Podemos observar este fato acompanhando  $MVP/it_{int}$  nas tabelas, que se mantém entre 100 e 200 na maior parte dos testes para o método de restrições ativas e entre 200 e 300 em grande parte dos testes para o método de barreira.

Apesar do número de produtos matriz-vetor não sofrer grandes alterações com o aumento de dimensão, vemos que o tempo para a realização de cada produto aumenta conforme a dimensão do problema, como mostrado em  $t/MVP$ . Porém, este aumento

<i>Prob.</i>	<i>np</i>	<i>Rest.</i>	<i>it<sub>RC</sub></i>	<i>it<sub>int</sub></i>	<i>f</i>	$\ Z^T g\ $	<i>MVP<sub>tot</sub></i>	<i>t(s)</i>	$\frac{MVP}{it_{int}}$	$\frac{t}{MVP}$
1b	20	60	15	34	37.43	4.0e-4	4881	24	143.56	0.005
			19	37	37.56	0.0024	4358	22	117.78	0.005
2b	20	60	17	38	37.31	0.0011	4295	22	113.03	0.005
			17	36	37.31	7.4e-4	4201	21	116.69	0.005
Média				36.25			4433	22.25		
3b	40	120	24	62	176.35	0.1378	7605	136	122.66	0.02
			27	58	177.06	0.0336	7086	128	122.17	0.02
4b	40	120	19	59	176.12	0.0143	12242	218	207.49	0.02
			22	61	176.65	0.0400	8133	146	133.33	0.02
Média				60			8766	157.00		
5b	60	180	26	75	427.32	0.1002	11781	465	157.08	0.04
			29	78	426.61	0.1174	12146	483	155.72	0.04
6b	60	180	33	77	426.27	0.2561	10047	405	130.48	0.04
			27	77	426.11 *	0.0696	10706	422	139.04	0.04
Média				76.75			11170	443.75		
19	80	240	32	89	790.89 *	0.3305	17443	1217	195.99	0.07
			35	89	790.90	0.0205	19013	1333	213.63	0.07
20	80	240	29	106	790.28 *	0.2112	17894	1246	168.81	0.07
			34	87	789.28	0.0036	14970	1055	172.07	0.07
Média				92.75			17330	1212.75		
21	100	300	45	129	1272.9 *	0.4086	15262	1664	118.31	0.11
			45	107	1273.2	0.0400	21499	2325	200.93	0.11
22	100	300	33	100	1269.1 *	0.2856	15961	1743	159.61	0.11
			28	90	1269.0 *	0.0642	20324	2202	225.82	0.11
Média				106.5			18261	1983.50		

Tabela 6.20: Resultados dos testes com as novas escolhas para polígonos de  $nl = 3$  lados utilizando o método de restrições ativas do capítulo 3.

segue o mesmo padrão tanto para o método de barreira quanto para o método de restrições ativas e não é influenciado pelo aumento do número de lados do polígono.

<i>Prob.</i>	<i>np</i>	<i>Rest.</i>	<i>it<sub>RC</sub></i>	<i>it<sub>int</sub></i>	<i>f</i>	$\ Z^T g\ $	<i>MVP<sub>tot</sub></i>	<i>t(s)</i>	$\frac{MVP}{it_{int}}$	$\frac{t}{MVP}$
7b	20	80	16	40	40.91	1.2e-4	7057	34	176.43	0.005
			16	39	40.89	9.4e-4	3984	20	102.15	0.005
8b	20	80	17	39	43.23	0.0023	4593	23	117.77	0.005
			18	39	43.23	1.9e-4	3982	20	102.10	0.005
Média				39.25			4904	24.25		
9b	40	160	27	59	192.75	0.1908	10457	186	177.24	0.02
			26	70	193.24	0.0462	10432	186	149.03	0.02
10b	40	160	27	58	203.44 *	0.1342	8816	159	152.00	0.02
			24	56	203.32 *	0.0330	8141	146	145.38	0.02
Média				60.75			9461	169.25		
11b	60	240	31	85	465.02	0.0037	16012	629	188.38	0.04
			42	102	464.39	0.0666	16548	649	162.24	0.04
12b	60	240	27	74	490.14	0.1064	12304	486	166.27	0.04
			30	77	490.09 *	0.0146	14056	552	182.55	0.04
Média				84.5			14730	579.00		
23	80	320	39	90	860.88	0.0449	13811	966	153.46	0.07
			35	87	860.17 *	0.0417	15516	1084	178.34	0.07
24	80	320	41	122	907.19	0.4218	15935	1112	130.61	0.07
			36	101	905.30	0.0967	16480	1149	163.17	0.07
Média				100			15435	1077.75		
25	100	400	42	104	1380.1 *	0.2611	16377	1774	157.47	0.11
			38	100	1378.5 *	0.1434	14959	1674	149.59	0.11
26	100	400	54	120	1455.7	0.1323	15436	1685	128.63	0.11
			48	111	1455.2	0.0681	20011	2191	180.28	0.11
Média				108.8			16695	1831.00		

Tabela 6.21: Resultados dos testes com as novas escolhas para polígonos de  $nl = 4$  lados utilizando o método de restrições ativas do capítulo 3.

<i>Prob.</i>	<i>np</i>	<i>Rest.</i>	<i>it<sub>RC</sub></i>	<i>it<sub>int</sub></i>	<i>f</i>	$\ Z^T g\ $	<i>MVP<sub>tot</sub></i>	<i>t(s)</i>	$\frac{MVP}{it_{int}}$	$\frac{t}{MVP}$
13b	20	100	14	37	40.09	7.2e-4	4521	22	122.19	0.005
			15	39	40.26	0.0061	3844	19	98.56	0.005
14b	20	100	17	38	42.02	0.0023	6254	30	164.58	0.005
			17	40	41.99	4.4e-4	4953	25	123.83	0.005
Média				38.5			4893	24.00		
15b	40	200	22	62	188.00	0.0149	9771	175	157.60	0.02
			27	59	188.02	0.1028	8737	157	148.08	0.02
16b	40	200	28	62	197.10	0.0111	6664	120	107.48	0.02
			25	56	197.48	0.0127	7869	141	140.52	0.02
Média				59.75			8260	148.25		
17b	60	300	26	67	451.98	0.0047	7847	314	117.12	0.04
			27	68	452.59	0.1245	12359	489	181.75	0.04
18b	60	300	37	84	476.10 *	0.1355	9588	380	114.14	0.04
			35	81	474.86	0.0238	14124	552	174.37	0.04
Média				75			10979	433.75		
27	80	400	37	91	835.95 *	0.3650	13136	918	144.35	0.07
			33	92	835.48 *	0.4816	11763	823	127.86	0.07
28	80	400	36	93	879.80 *	0.4680	14795	1024	159.09	0.07
			42	92	878.87	0.1033	12984	916	141.13	0.07
Média				92			13169	920.25		
29	100	500	33	101	1342.1 *	0.0935	16544	1797	163.80	0.11
			37	101	1342.1 *	0.2653	16295	1778	161.34	0.11
30	100	500	33	113	1410.2	0.0600	21878	2372	193.61	0.11
			37	103	1410.0	0.1377	14047	1555	136.38	0.11
Média				104.5			17191	1875.50		

Tabela 6.22: Resultados dos testes com as novas escolhas para polígonos de  $nl = 5$  lados utilizando o método de restrições ativas do capítulo 3.

<i>Prob.</i>	<i>np</i>	<i>Rest.</i>	<i>itext</i>	<i>itint</i>	<i>f</i>	$\ Z^T g\ $	<i>MVP<sub>tot</sub></i>	<i>t(s)</i>	$\frac{MVP}{itint}$	$\frac{t}{MVP}$
1b	20	60	3	66	39.42	0.0026	12917	62	195.71	0.005
			2	33	40.99	0.0023	8219	39	249.06	0.005
2b	20	60	3	27	39.20	2.3e-4	6031	29	223.37	0.005
			2	24	41.01	0.0036	5344	26	222.67	0.005
Média				37.5			8127	39.00		
3b	40	120	2	65	180.60	0.0573	14893	262	229.12	0.02
			2	83	180.02	0.0179	16439	289	198.06	0.02
4b	40	120	2	48	179.88	0.0062	13471	238	280.65	0.02
			2	33	180.24	0.0023	7446	132	225.64	0.02
Média				57.25			13062	230.25		
5b	60	180	2	257	429.52	0.0152	59007	2275	229.60	0.04
			2	279	429.74	0.0662	64739	2490	232.04	0.04
6b	60	180	2	96	428.78	0.0939	19236	746	200.38	0.04
			2	110	429.04	0.0265	34060	1312	309.64	0.04
Média				185.50			44260	1705.75		
19	80	240	2	171	793.95	0.1814	38567	2629	225.54	0.07
			2	665	792.89	0.0109	177011	12070	266.18	0.07
20	80	240	2	122	791.78	0.1701	37400	2538	306.56	0.07
			2	322	791.55	0.0155	120784	8269	375.11	0.07
Média				320.00			93440	6376.50		
21	100	300	1	212	1279.2	0.3372	64785	6875	305.59	0.11
			2	620	1272.9	0.2227	224350	23694*	361.85	0.11
22	100	300	1	139	1274.2	0.1800	34311	3649	246.84	0.11
			2	455	1274.4	22.8159	215774	22762*	474.23	0.11
Média				356.5			134805	14245.00		

Tabela 6.23: Resultados dos testes com as novas escolhas para polígonos de  $nl = 3$  lados utilizando o método com barreira do capítulo 4.

<i>Prob.</i>	<i>np</i>	<i>Rest.</i>	<i>itext</i>	<i>itint</i>	<i>f</i>	$\ Z^T g\ $	<i>MVP<sub>tot</sub></i>	<i>t(s)</i>	$\frac{MVP}{itint}$	$\frac{t}{MVP}$
7b	20	80	3	39	42.89	6.4e-4	10253	48	262.90	0.005
			3	32	42.89	2.7e-4	10246	48	320.19	0.005
8b	20	80	3	35	45.26	4.1e-4	9962	47	284.63	0.005
			2	25	47.03	0.0015	7236	34	289.44	0.005
Média				32.75			9424	44.25		
9b	40	160	2	43	197.48	0.0669	12888	226	299.72	0.02
			3	61	194.85	0.0125	16808	298	275.54	0.02
10b	40	160	2	66	208.84	0.0808	16163	283	244.89	0.02
			2	52	206.73	0.0230	12711	224	244.44	0.02
Média				55.5			14642	257.75		
11b	60	240	2	124	467.17	0.0017	29980	1169	241.77	0.04
			2	94	468.54	0.0127	21785	848	231.76	0.04
12b	60	240	2	646	492.58	0.0228	145391	5574	225.06	0.04
			2	555	493.05	0.0064	117660	4529	212.00	0.04
Média				354.75			78704	3030.00		
23	80	320	3	753	859.59	0.0139	255288	17287	339.03	0.07
			2	492	861.46	0.0188	130810	8959	265.87	0.07
24	80	320	3	329	908.70	0.0597	76734	5226	233.23	0.07
			2	304	910.13	0.0828	71927	4875	236.60	0.07
Média				469.5			133689	9086.75		
25	100	400	2	219	1382.7	0.0666	62101	6569	283.57	0.11
			2	712	1380.6	0.2427	214542	22669*	301.32	0.11
26	100	400	2	752	1457.4	1.8179	254019	26835*	337.79	0.11
			2	814	1459.5	1.5043	259269	27507*	318.51	0.11
Média				624.25			197482	20895.00		

Tabela 6.24: Resultados dos testes com as novas escolhas para polígonos de  $nl = 4$  lados utilizando o método com barreira do capítulo 4.

<i>Prob.</i>	<i>np</i>	<i>Rest.</i>	<i>itext</i>	<i>itint</i>	<i>f</i>	$\ Z^T g\ $	<i>MVP<sub>tot</sub></i>	<i>t(s)</i>	$\frac{MVP}{itint}$	$\frac{t}{MVP}$
13b	20	100	3	83	42.30	0.0019	18570	87	223.73	0.005
			3	37	42.79	0.0028	8524	41	230.38	0.005
14b	20	100	3	32	44.32	7.3e-4	8543	40	266.97	0.005
			3	33	44.32	7.1e-4	7721	37	233.97	0.005
Média				46.25			10839	51.25		
15b	40	200	3	458	190.01	0.0016	82250	1432	179.59	0.02
			2	67	194.67	0.0537	16719	293	249.54	0.02
16b	40	200	2	66	201.91	0.0174	15952	282	241.70	0.02
			2	38	201.45	0.0104	11926	209	313.84	0.02
Média				157.25			31711	554.00		
17b	60	300	3	655	454.45	0.0084	141476	5431	215.99	0.04
			2	153	462.99	0.1035	32106	1238	209.84	0.04
18b	60	300	2	116	478.97	0.0397	23927	928	206.27	0.04
			2	187	478.70	0.0248	48325	1874	258.42	0.04
Média				277.75			61458	2367.75		
27	80	400	2	397	840.90	0.1387	83000	5609	209.07	0.07
			2	315	847.84	0.1979	63079	4280	200.25	0.07
28	80	400	1	116	887.18	0.1657	23575	1620	203.23	0.07
			2	258	881.53	0.0198	65110	4436	252.36	0.07
Média				271.5			58691	3986.25		
29	100	500	2	975	1345.5	0.3749	255869	27018★	262.43	0.11
			2	1038	1346.0	2.6748	276477	29205★	266.36	0.11
30	100	500	2	722	1412.5	0.6975	211607	22314★	293.08	0.11
			2	201	1413.6	0.1177	49406	5254	245.80	0.11
Média				734.00			198339	20947.75		

Tabela 6.25: Resultados dos testes com as novas escolhas para polígonos de  $nl = 5$  lados utilizando o método com barreira do capítulo 4.

A seguir apresentamos os gráficos comparativos com as médias do número de iterações internas, do tempo total e do total de produtos matriz-vetor para cada dimensão. Para cada gráfico temos representados em azul, vermelho e preto os dados correspondentes aos problemas com  $nl = 3, 4$  e  $5$ , respectivamente.

Na Figura 6.11 apresentamos a média do tempo total de execução para o método de restrições ativas e de barreira, respectivamente. Os pontos representam as médias apresentadas nas Tabelas 6.20–6.25 e as linhas contínuas representam as aproximações realizadas por polinômios do tipo  $y = a_1x + a_2x^2 + a_3x^3$ , onde  $a_1, a_2$  e  $a_3$  são constantes obtidas pelo ajuste por quadrados mínimos.

Analisando estas curvas podemos notar uma suavidade melhor para o ajuste no método de restrições ativas, com previsões para o tempo de execução para dimensões maiores melhores quando comparado com o método de barreira.

Na Tabela 6.26 estão apresentados os valores obtidos para as constantes  $a_1, a_2$  e  $a_3$  em cada caso e as previsões para o tempo de execução (em horas) para dimensões maiores ( $np = 250$  e  $np = 500$ ). Estas previsões para problemas de maior dimensão nos permitem concluir que o método de barreira deve ser melhorado para se mostrar mais competitivo.

Ressaltamos aqui que para o caso da resolução do método com barreira, o ajuste através de quadrados mínimos foi feito desconsiderando-se os testes para  $np = 100$  pontos, desde que para alguns destes testes não houve convergência para uma solução. Notamos também que o ajuste não foi satisfatório para o caso de polígonos com  $nl = 5$  lados para o método de barreira, onde – na Tabela 6.26 indica que o valor da estimativa foi negativo.

Contudo, analisando a Tabela 6.26 observamos um crescimento rápido do tempo de execução com o aumento da dimensão do problema, o que torna impraticável a resolução de problemas de grande porte, em especial para o método de barreira. Porém, há modificações a serem feitas neste método que podem torná-lo mais rápido, o que não foi realizado neste trabalho, mas que pode ser feito futuramente, por exemplo usando-se a estratégia de barreira estabilizada (cf. [70, §16.4]).

Para os problemas de polígonos com  $nl = 3$  lados resolvidos com o método de restrições ativas, realizamos um teste com  $np = 250$  pontos e obtivemos a convergência para uma solução com o tempo total de execução de 7.64 horas, acima da média estimada de 5.51 horas para o tempo de execução.

Porém, realizamos também o ajuste de quadrados mínimos para os tempos mínimo

$nl$	método	constantes			tempo médio previsto (hs)	
		$a_1$	$a_2$	$a_3$	$T(250)$	$T(500)$
3	Rest. Ativas	-4.68	0.19	$5.9e - 4$	5.51	32.96
	Barreira	29.62	-1.90	$3.15e - 2$	106.10	968.56
4	Rest. Ativas	-3.39	0.19	$1.75e - 4$	3.98	19.44
	Barreira	10.88	-1.33	$3.27e - 2$	119.62	-
5	Rest. Ativas	2.42	-0.06	$2.23e - 3$	8.80	73.62
	Barreira	-32.69	1.55	$-6.38e - 3$	1044.72	-

Tabela 6.26: Constantes dos ajustes por quadrados mínimos dos modelos  $T(n) = a_1n + a_2n^2 + a_3n^3$  da Figura 6.11, com as previsões dos tempos médios de execução para problemas com 250 e 500 pontos.

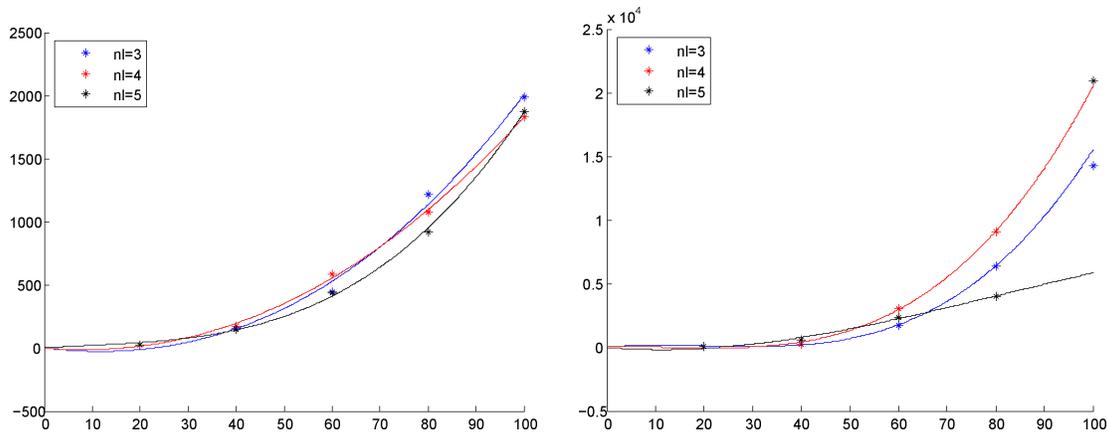


Figura 6.11: Gráficos da média do tempo total de execução pelo número de pontos do problema ( $np = 20, 40, 60, 80, 100$ ) para problemas com  $nl = 3, 4, 5$  resolvidos pelo método de restrições ativas e de barreira, respectivamente. O ajuste da curva foi feito via quadrados mínimos e os valores das constantes dos modelos estão na Tabela 6.26.

e máximo obtidos para os testes com número de pontos até  $np = 100$ , como é ilustrado na Figura 6.12. Os valores obtidos para as constantes  $a_1$ ,  $a_2$  e  $a_3$  estão na Tabela 6.27, bem como as estimativas para o tempo mínimo e máximo para  $np = 250$  e  $np = 500$  pontos.

Assim observamos que o tempo de execução obtido para o problema com  $np = 250$  está dentro da faixa esperada pela estimativa via quadrados mínimos, entre a estimativa mínima ( $T = 3.64$  horas) e a estimativa máxima ( $T = 8.54$  horas).

$nl$	tempo	constantes			tempo médio previsto (hs)	
		$a_1$	$a_2$	$a_3$	$T(250)$	$T(500)$
3	mínimo	-5.33	0.22	$5.95e - 5$	3.64	16.31
	médio	-4.68	0.19	$5.91e - 4$	5.52	32.96
	máximo	-1.45	$8.29e - 3$	$1.66e - 3$	8.54	63.17

Tabela 6.27: Constantes dos ajustes por quadrados mínimos dos modelos  $T(n) = a_1n + a_2n^2 + a_3n^3$  da Figura 6.12, com as previsões dos tempos médios de execução para problemas com 250 e 500 pontos.

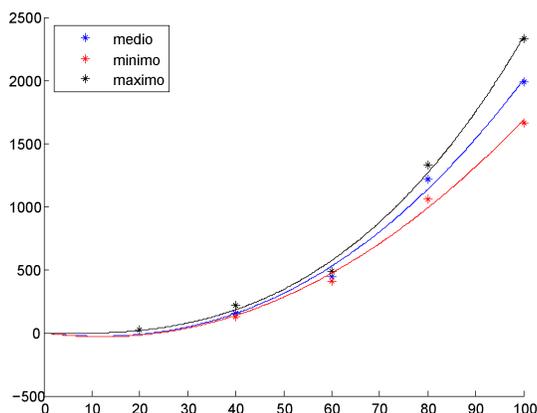


Figura 6.12: Gráficos da média, do mínimo e do máximo do tempo total de execução pelo número de pontos do problema ( $np = 20, 40, 60, 80, 100$ ) para problemas com  $nl = 3$  resolvidos pelo método de restrições ativas. O ajuste das curvas foi feito via quadrados mínimos e os valores das constantes dos modelos estão na Tabela 6.27.

Na Figura 6.13 são mostradas as configurações inicial e final para o problema com  $np = 250$  pontos testado para o método de restrições ativas.

A dimensão maior desse problema oferece um resultado visualmente mais interessante. A configuração ótima é bastante peculiar, com 135 pontos no interior do polígono, que correspondem a 54% dos pontos do problema.

Após analisar o tempo de execução dos algoritmos, na Figura 6.14 podemos avaliar o comportamento da média do número de iterações internas com o crescimento da dimensão do problema. Podemos notar um comportamento quase linear para os dois métodos, com um padrão monótono mais acentuado para o método de restrições ativas.

Já na Figura 6.15 podemos ver o comportamento da média do número de produtos matriz-vetor com o crescimento da dimensão do problema. O comportamento da curva é também quase linear. Novamente, ao analisar os gráficos, observamos um comporta-

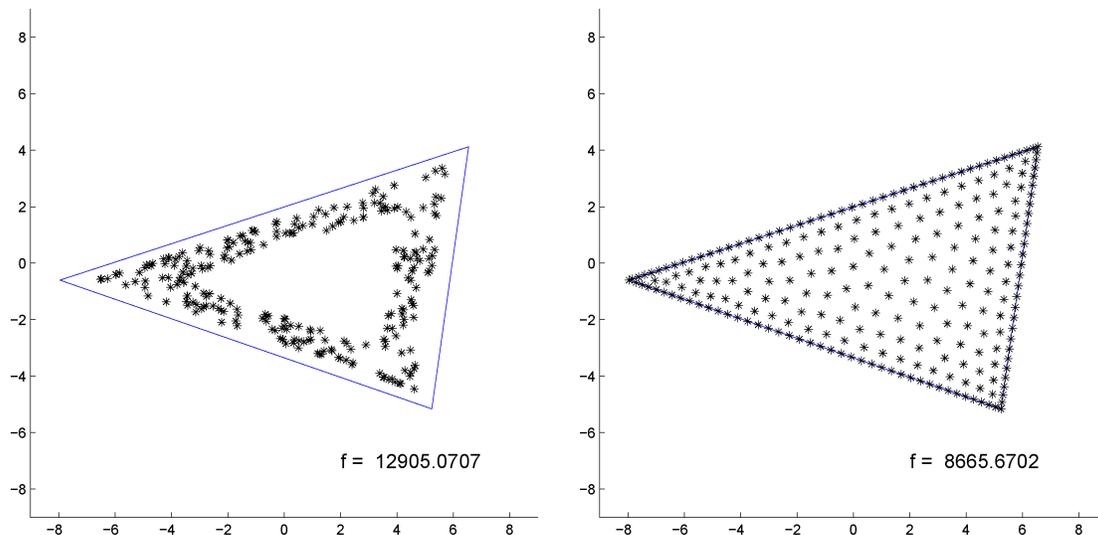


Figura 6.13: Resultados para um problema com  $np = 250$  pontos de dimensão  $n = 500$ , obtidos com o método de restrições ativas em  $it_{RC} = 62$  iterações de região de confiança, com  $cont_H = 41$  modelos quadráticos construídos e  $MVP_{total} = 39455$  produtos matriz por vetor, calculados em um tempo de 27514 segundos. Os gráficos representam as configurações inicial e final dos pontos, respectivamente.

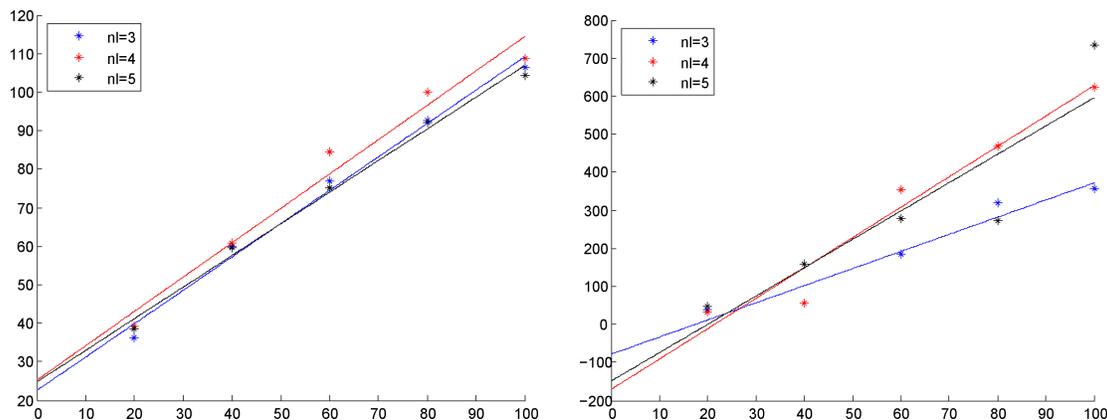


Figura 6.14: Gráficos da média do número total de iterações internas pelo número de pontos do problema ( $np = 20, 40, 60, 80, 100$ ) para problemas com  $nl = 3, 4, 5$  resolvidos pelo método de restrições ativas e de barreira, respectivamente.

mento mais instável para o método de barreira.

A seguir apresentamos a visualização gráfica dos resultados de alguns problemas das Tabelas 6.20 – 6.25.

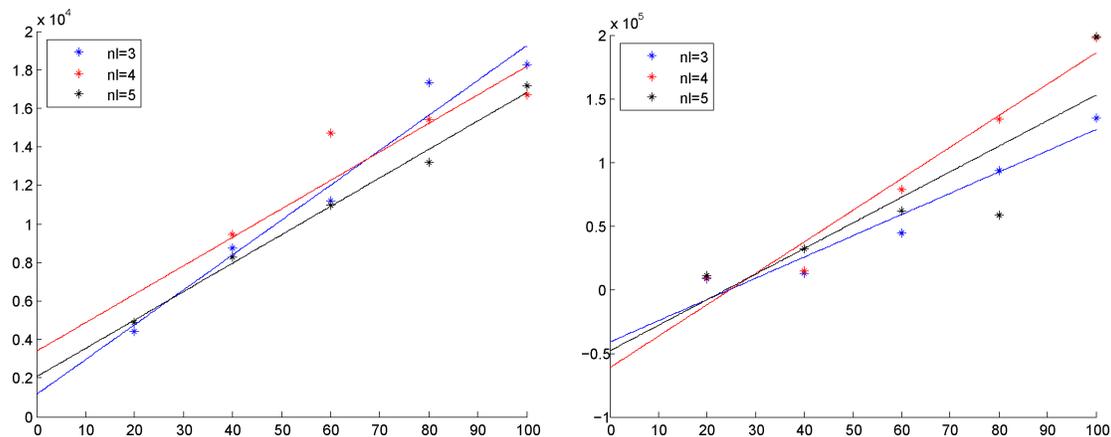


Figura 6.15: Gráficos da média do número total de produtos matriz-vetor pelo número de pontos do problema ( $np = 20, 40, 60, 80, 100$ ) para problemas com  $nl = 3, 4, 5$  resolvidos pelo método de restrições ativas e de barreira, respectivamente.

Nas Figuras 6.16, 6.17 e 6.18 apresentamos os problemas com 40 pontos em polígonos de 3, 4 e 5 lados respectivamente. Vemos em cada conjunto de gráficos, as configurações iniciais dos problemas, a solução para o método de restrições ativas e a solução para o método com barreira, na primeira, segunda e terceira linha de cada conjunto de gráficos, respectivamente.

Notamos que, apesar de iniciarmos os algoritmos com os mesmos pontos iniciais, em geral temos a convergência para soluções distintas. Isso se deve ao fato do problema apresentar vários minimizadores locais e globais, o que pode também contribuir para uma maior dificuldade de resolução do problema.

Outra evidência a favor do algoritmo proposto com restrições ativas neste conjunto de testes é o fato de, na maioria dos problemas, a solução apresentar um valor de função objetivo menor do que a solução obtida com o método com barreira, apesar deste último trabalhar um pouco mais. Os valores da função objetivo em cada ponto estão destacados nas figuras.

Além da convergência para soluções distintas para cada método, temos também casos em que o mesmo algoritmo converge para pontos diferentes alterando o ponto inicial de um mesmo problema. Nas Figuras 6.19 e 6.20 apresentamos os resultados para os problemas 5b e 15b, respectivamente, com dois pontos iniciais diferentes para cada problema.

Notamos na Figura 6.19 que as soluções obtidas pelo método de barreira (última linha) são ligeiramente diferentes. Contudo, a diferença de solução para o método de

restrições ativas é mais evidente, com 17 pontos no interior, 40 nas faces e 3 nos vértices do polígono (46 restrições ativas) na primeira solução e 19 pontos no interior, 38 nas faces e 3 nos vértices do polígono (44 restrições ativas) na segunda solução.

Na Figura 6.20 ilustramos um outro caso de convergência para soluções distintas para os dois métodos.

Outro fato importante na resolução dos problemas é uma característica que pode ser verificada para o método de barreira com a introdução da sequência forçante descrita anteriormente em (6.7). A primeira iteração externa do método realiza a maior separação dos pontos, sendo as demais responsáveis apenas por um ajuste mais fino para atingir a precisão exigida.

Podemos ver esta característica exemplificada na Figura 6.21, onde ilustramos a configuração inicial e as 3 iterações externas do problema 17b resolvido com o método de barreira. Após a primeira iteração já podemos notar uma definição dos pontos que ficarão na face e no interior do polígono na solução.

Isso confirma a vantagem ao introduzir a sequência forçante na escolha da precisão para o critério de parada.

Outro ponto a ser destacado é o comportamento não monótono da sequência  $\{\|Z^T g\|\}$  utilizada para detectar otimalidade do problema, em função das iterações do algoritmo de região de confiança. Um exemplo ilustrativo é mostrado na Figura 6.22, cujos resultados correspondem ao Problema 7b com dois pontos iniciais distintos. Nos gráficos da esquerda ilustramos os resultados para o método com restrições ativas, com a variação de  $\|Z^T g\|$  pelo número de iterações de região de confiança sem a diminuição do raio. Os gráficos da direita apresentam os resultados para o método de barreira, onde são mostradas as variações de  $\|Z^T g_{exp}\|$  pelo número de iterações bem sucedidas de região de confiança, isto é, sem a diminuição do raio. As linhas pontilhadas verticais representam as transições entre as iterações externas do algoritmo.

Pelas figuras percebemos um decrescimento não monótono de  $\|Z^T g\|$  e de  $\|Z^T g_{exp}\|$ , com esta característica mais acentuada para o método de barreira. Isso pode estar relacionado ao fato de que os subproblemas dos método de barreira são mais mal condicionados devido à introdução do parâmetro de barreira.

Para o algoritmo com restrições ativas também analisamos um problema com maior dimensão ( $n = 500$ ) utilizando  $np = 250$  pontos distribuídos em um polígono com  $nl = 3$  lados. Este é o mesmo problema apresentado na Figura 6.13, onde vemos

as configurações iniciais e finais dos pontos. Na Figura 6.23 observamos a função que descreve o decréscimo da norma do gradiente projetado por iteração. Podemos observar um comportamento parecido para os problemas com  $np = 20$  pontos ilustrados na Figura 6.22, com um destaque para o comportamento não monótono da sequência da norma do gradiente projetado.

Este comportamento não monótono da norma do gradiente projetado se acentua em problemas de dimensão maior, como podemos verificar nos gráficos ilustrados na Figura 6.22 e na Figura 6.23. Observamos também que este comportamento não monótono é mais acentuado na resolução com o algoritmo de barreira, o que nos sugere uma possível modificação futura, como por exemplo, a introdução de preconditionadores para acelerar a resolução dos subproblemas quadráticos à medida que o condicionamento piora com o aumento do parâmetro da função barreira.

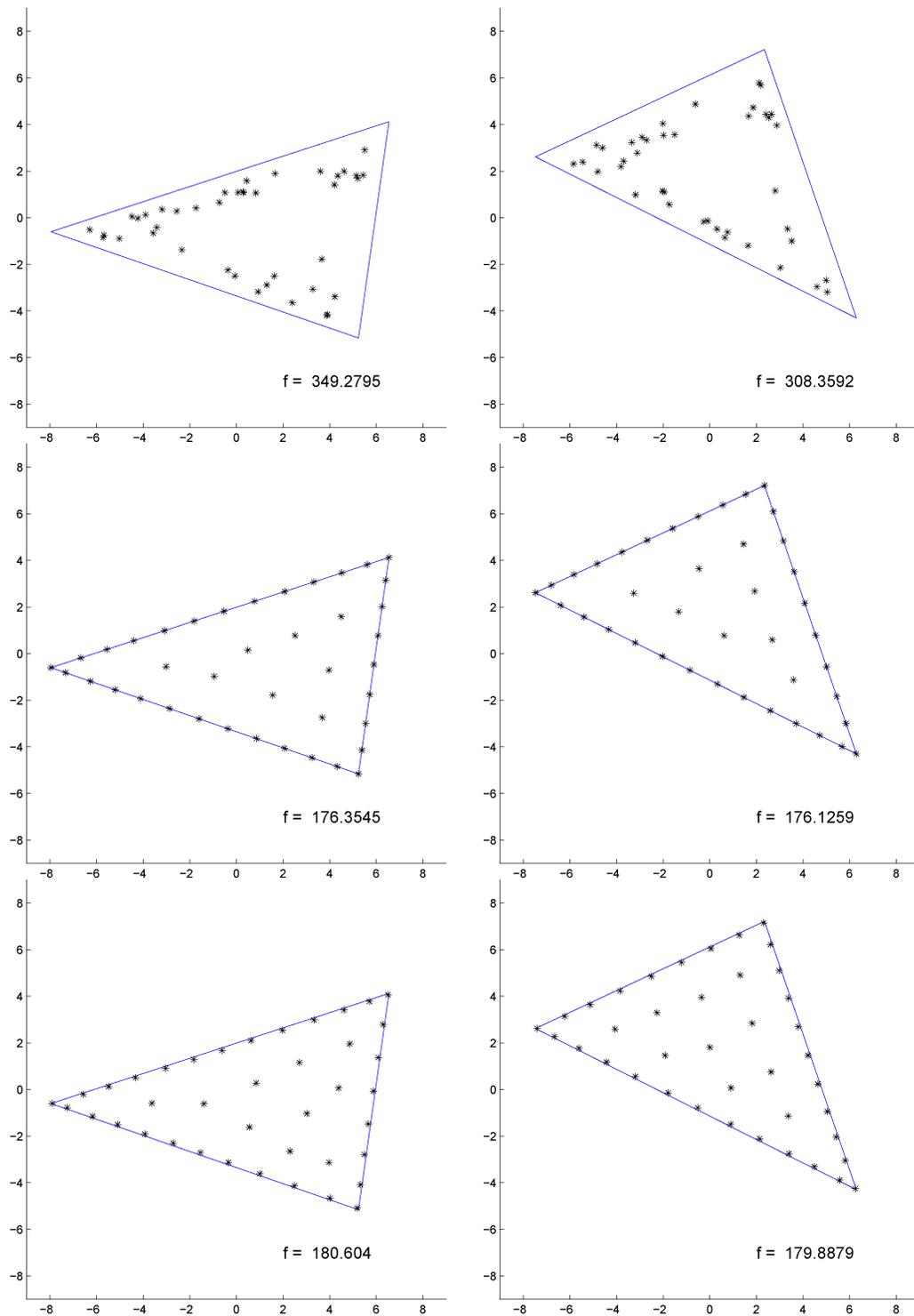


Figura 6.16: Resultados dos problemas 3b e 4b ( $nl = 3$  e  $np = 40$ ) para cada método. Na primeira linha temos a configuração inicial dos pontos para cada problema. Na segunda linha, a solução com o algoritmo de restrições ativas e na terceira linha, a solução com o algoritmo de barreira.

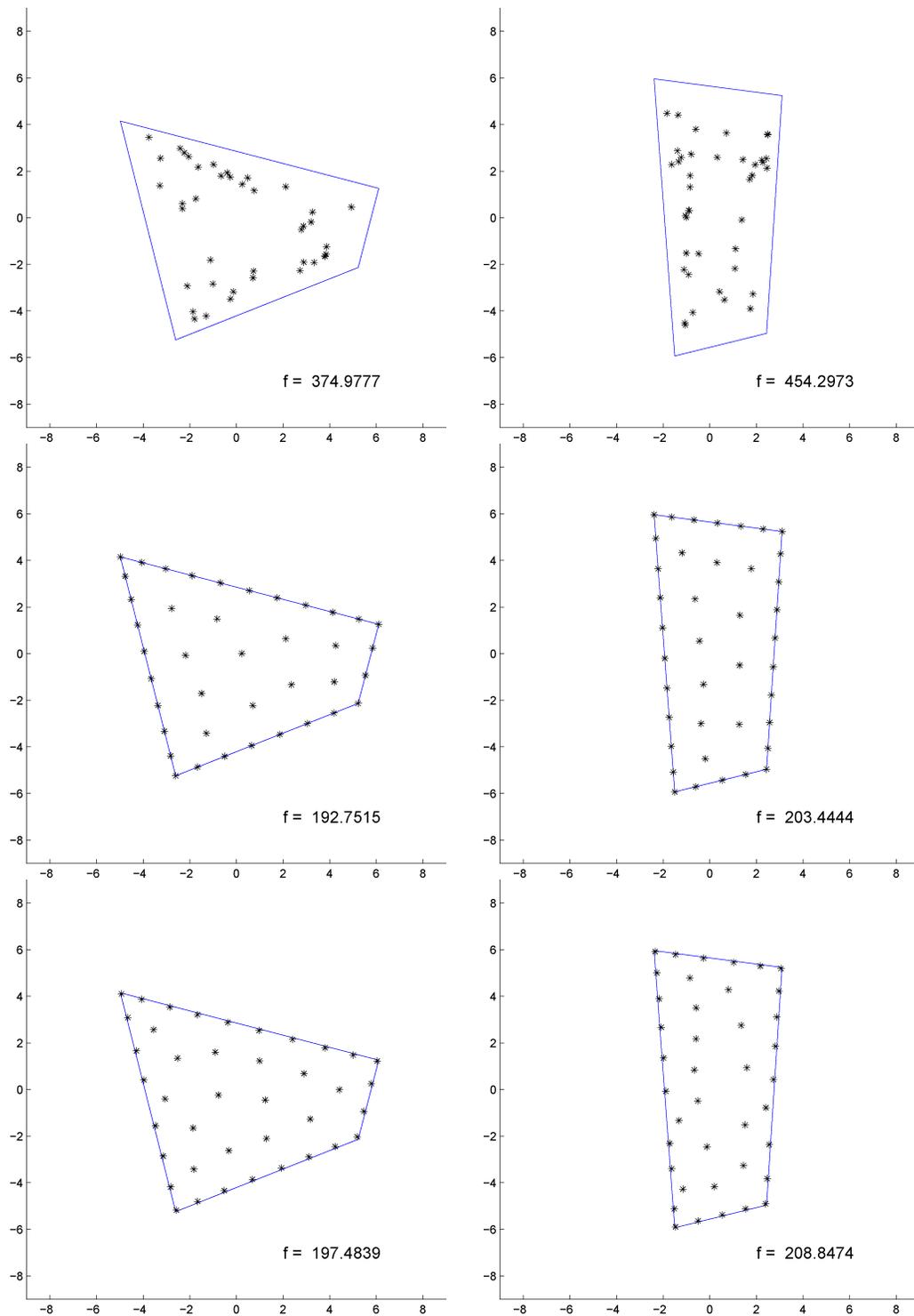


Figura 6.17: Resultados dos problemas 9b e 10b ( $nl = 4$  e  $np = 40$ ) para cada método. Na primeira linha temos a configuração inicial dos pontos para cada problema. Na segunda linha, a solução com o algoritmo de restrições ativas e na terceira linha, a solução com o algoritmo de barreira.

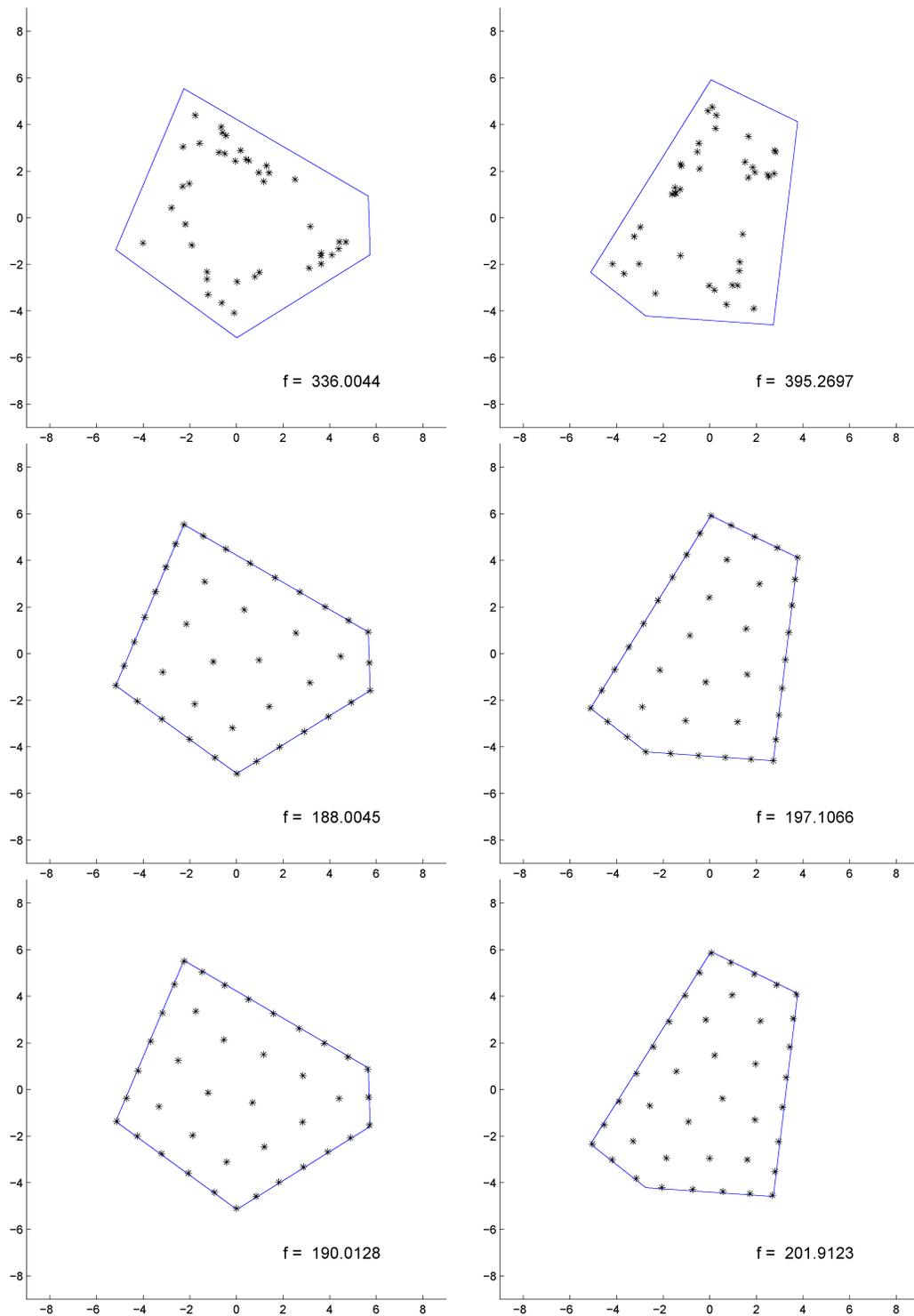


Figura 6.18: Resultados dos problemas 15b e 16b ( $nl = 5$  e  $np = 40$ ) para cada método. Na primeira linha temos a configuração inicial dos pontos para cada problema. Na segunda linha, a solução com o algoritmo de restrições ativas e na terceira linha, a solução com o algoritmo de barreira.

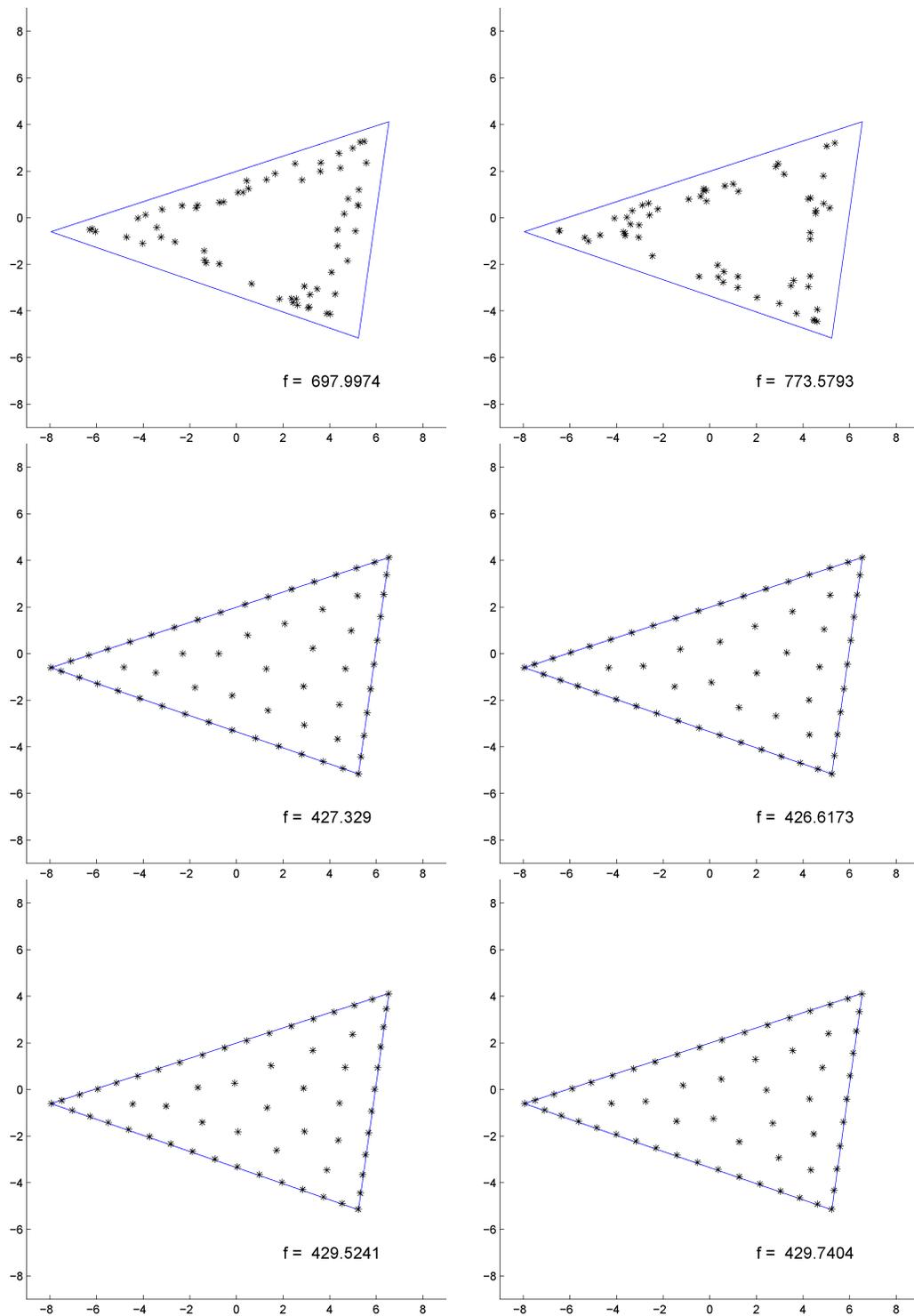


Figura 6.19: Resultados do problema 5b ( $nl = 3$  e  $np = 60$ ) para dois pontos iniciais diferentes. Na primeira linha temos as configurações iniciais dos pontos. Na segunda linha, a solução com o algoritmo de restrições ativas e na terceira linha, a solução com o algoritmo de barreira.

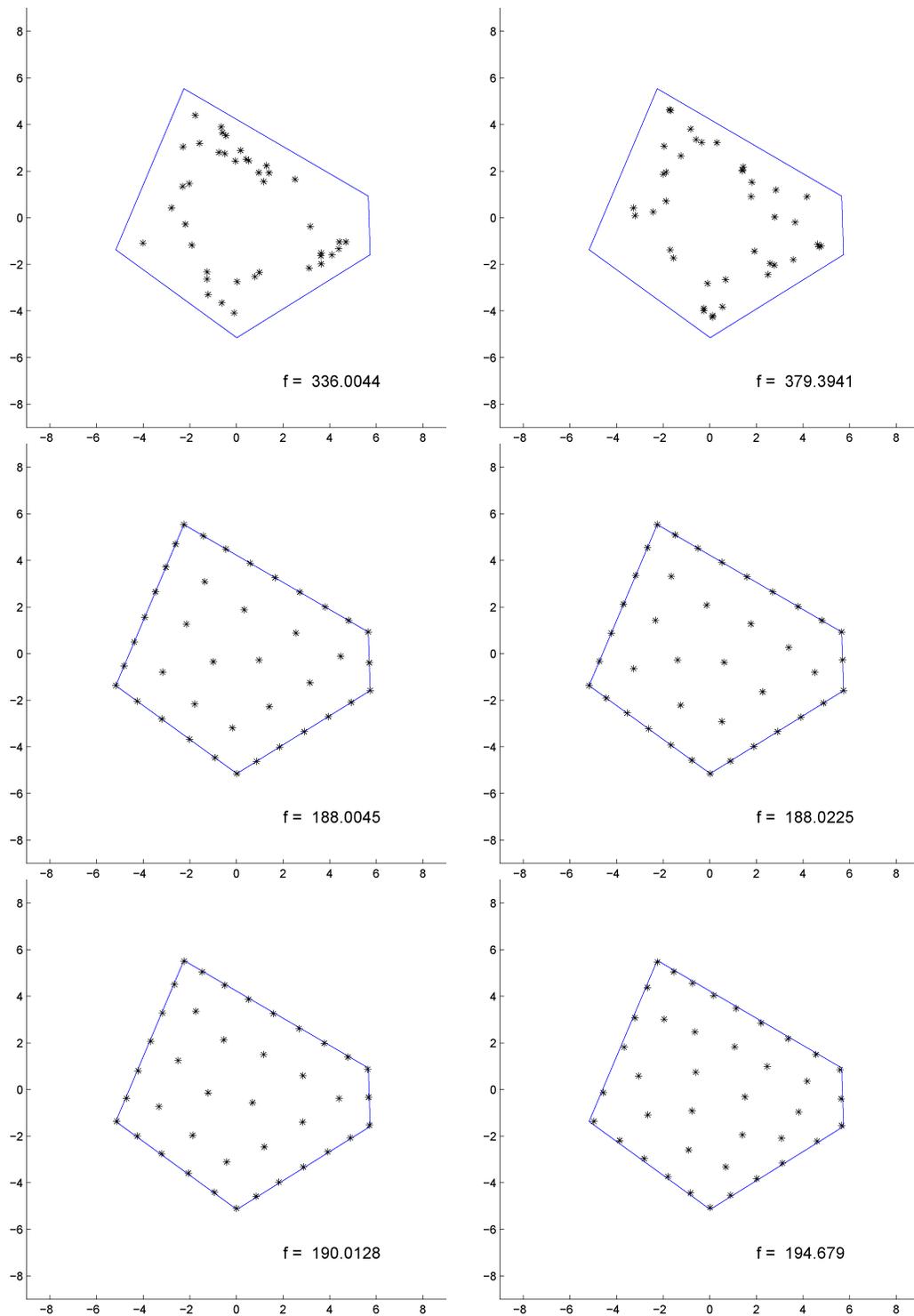


Figura 6.20: Resultados do problema 15b ( $nl = 5$  e  $np = 40$ ) para dois pontos iniciais diferentes. Na primeira linha temos as configurações iniciais dos pontos. Na segunda linha, a solução com o algoritmo de restrições ativas e na terceira linha, a solução com o algoritmo de barreira.

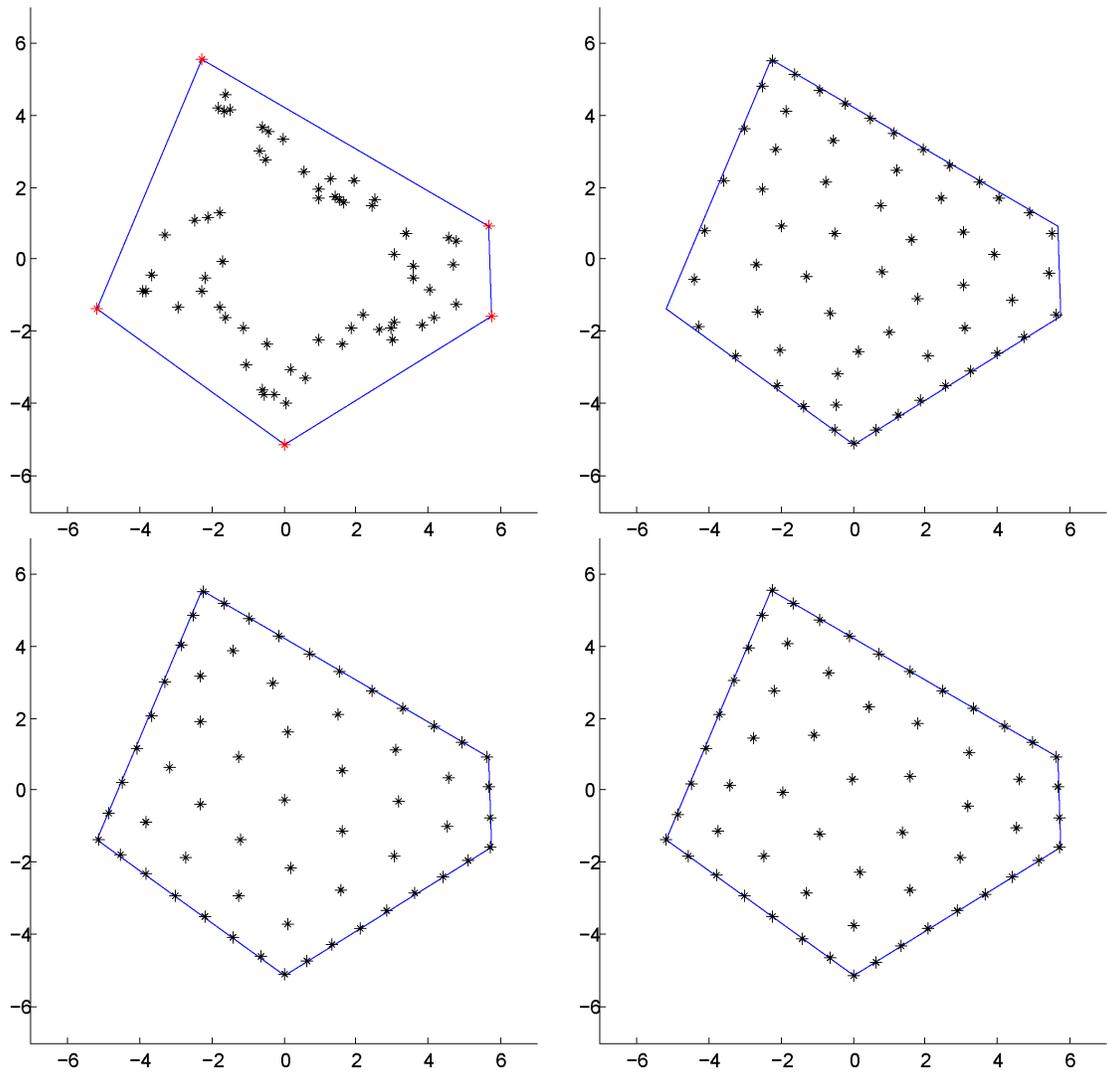


Figura 6.21: Gráficos ilustrativos das iterações externas para a resolução do Problema 17b ( $nl = 5$  e  $np = 60$ ) com o algoritmo de barreira.

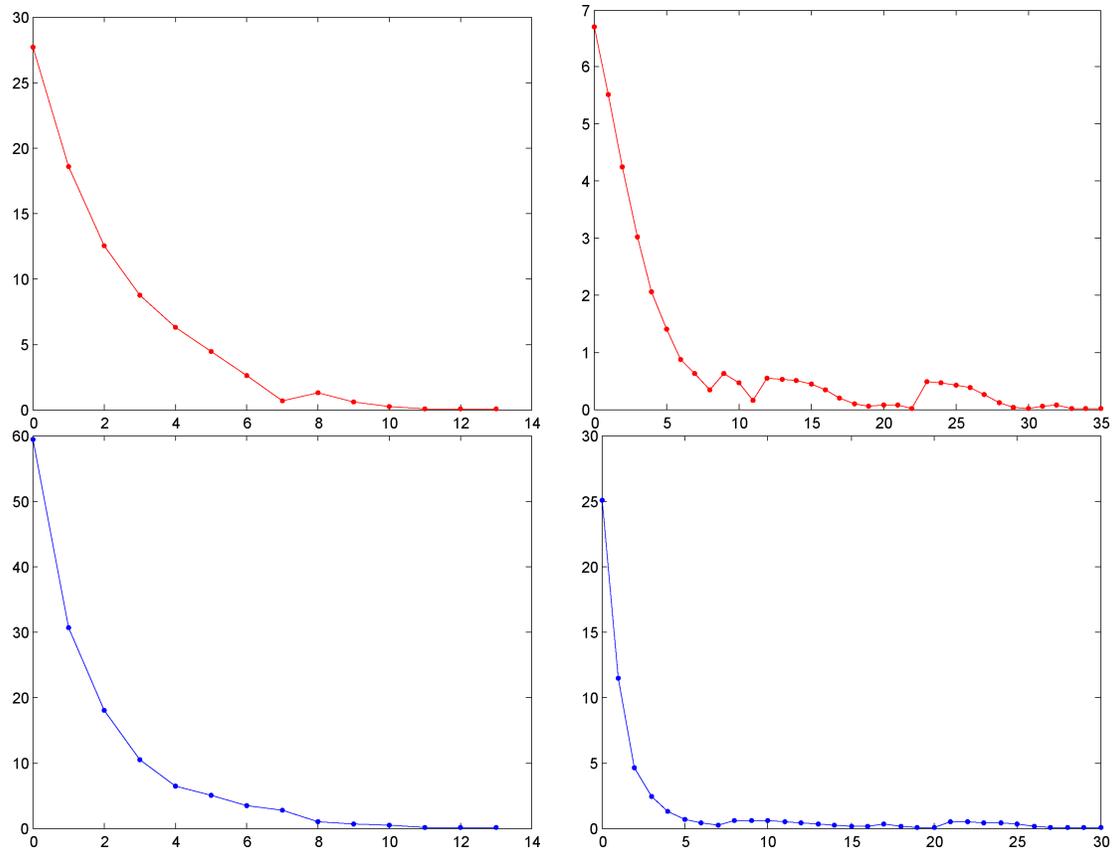


Figura 6.22: Gráficos ilustrativos do comportamento da sequência da norma do gradiente projetado a cada iteração. As figuras correspondem ao Problema 7b ( $nl = 4$  e  $np = 20$ ) com dois pontos iniciais distintos. A resolução com os algoritmos de restrições ativas e de barreira está associada aos gráficos da esquerda e da direita, respectivamente.

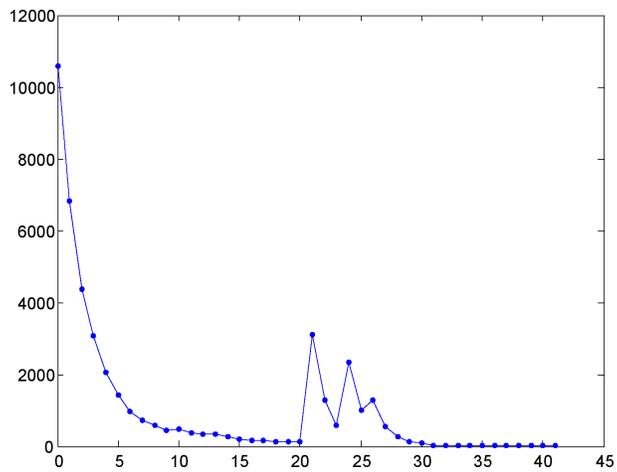


Figura 6.23: Sequência da norma do gradiente projetado a cada iteração para o problema com  $np = 250$  pontos de dimensão  $n = 500$ , obtidos com o método de restrições ativas.

Com este conjunto de testes pudemos validar os dois métodos propostos para a resolução dos problemas considerados. Apesar dos problemas-teste possuírem estrutura favorável para a geração de instâncias potencialmente de grande porte, no contexto livre de fatorações, percebemos que a distribuição de pontos em polígonos é um problema difícil e desafiador, com uma grande quantidade de minimizadores locais. Infelizmente, o tempo computacional dispendido não permitiu que problemas efetivamente grandes fossem considerados nesta tese. De qualquer forma, comprovamos um melhor desempenho para o algoritmo de restrições ativas em comparação à abordagem usando barreira na forma em que ambos algoritmos foram implementados neste trabalho. Naturalmente há pontos que podem ser melhorados nos dois algoritmos em um trabalho futuro, visando diminuir o tempo de execução computacional e aumentar a dimensão dos problemas resolvidos.

# Considerações finais

Neste trabalho propusemos dois algoritmos distintos com regiões de confiança para a resolução de problemas com restrições lineares gerais. O primeiro é baseado na estratégia de restrições ativas proposta por Gay [55] e o segundo utiliza técnicas de barreira para a não negatividade das variáveis de folga. Para ambos algoritmos analisamos as propriedades de boa definição e de convergência global. Para o algoritmo de restrições ativas também incluímos um resultado de convergência local. Como trabalhamos efetivamente com as informações referentes às derivadas segundas da função objetivo, obtivemos convergência para minimizadores que cumprem condições de otimalidade de segunda ordem.

Para testar os algoritmos utilizamos uma família de problemas específicos de distribuição de pontos em polígonos que ainda não tinha sido utilizada neste contexto na literatura. Tais problemas podem ser vistos como uma modificação do problema de distribuição de pontos em caixas, sugerido por Powell [98] e [99]. Para resolver os subproblemas de região de confiança nos dois algoritmos propostos utilizamos o algoritmo de Rojas, Santos e Sorensen [102], desenvolvido de acordo com a filosofia livre de fatorações de matrizes. Foi nessa linha que preparamos a implementação dos algoritmos e dos problemas-teste. Exploramos toda a estrutura dos problemas, de modo que as fatorações que utilizamos correspondem apenas a pequenos blocos de informações dentro do problema como um todo.

O problema abordado possui uma estrutura favorável para as fatorações  $QR$  utilizadas. Porém, vale destacar, que o algoritmo pode ser adaptado para outras famílias de problemas. A fatoração  $QR$  foi apenas utilizada para obter informações da matriz  $Z$  e pode ser substituída por outro tipo de fatoração mais conveniente a cada problema com outra estrutura específica.

Para os testes realizados neste trabalho, os dois algoritmos resolveram a maior parte dos problemas gerados, obtendo pontos estacionários de segunda ordem na maioria dos

casos, com uma vantagem em tempo de execução e desempenho para o algoritmo de restrições ativas.

Notamos também que, para o algoritmo com barreira, as iterações mais caras são as últimas, com o aumento do valor do parâmetro  $\rho_\ell$ . Porém, obtivemos na primeira iteração externa um bom espalhamento dos pontos quando comparados com a configuração inicial dos pontos. Isso nos faz pensar para trabalhos futuros em um algoritmo híbrido, utilizando apenas uma iteração do método com barreira e as demais, do método de restrições ativas.

Podemos também pensar em outras melhorias para o algoritmo com elementos de barreira. Em Griva, Nash e Sofer [70, §16.4] é feito um estudo para a incorporação de elementos preconditionadores em algoritmos tradicionais de barreira, podendo ser aproveitadas informações da matriz cujas colunas geram o núcleo das restrições. Acreditamos que as ideias deste estudo, bem como estratégias adaptativas para atualizar o parâmetro de barreira, como as de [93], [117] possam servir de ponto de partida para melhorias em nosso trabalho a fim de melhorar o desempenho do algoritmo de barreira com região de confiança proposto.

# Referências Bibliográficas

- [1] E. D. Andersen. *The homogeneous and self-dual model and algorithm for linear optimization*. MOSEK Technical Report: TR-2009-1. April 14, 2009, (Last revised: June 02, 2009), MOSEK ApS, Fruebjergvej 3 Box 16, 2100 Copenhagen, Denmark. 5 pág.
- [2] E. D. Andersen, B. Jensen, J. Jensen, R. Sandvik & U. Worsøe. *MOSEK version 6*. MOSEK Technical report: TR-2009-3. October 20, 2009, MOSEK ApS, Fruebjergvej 3 Box 16, 2100 Copenhagen, Denmark. 7 pág.
- [3] E. D. Andersen, B. Jensen, R. Sandvik & U. Worsøe. *The improvements in MOSEK version 5*. MOSEK Technical report: TR-2007-1. September 25, 2007, MOSEK ApS, Fruebjergvej 3 Box 16, 2100 Copenhagen, Denmark. 7 pág.
- [4] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen. *LAPACK User's Guide* (<http://www.netlib.org/lapack/lug/>), Third Edition, SIAM, Philadelphia, 1999.
- [5] R. Andreani, E. G. Birgin, J. M. Martínez & M. L. Schuverdt. On Augmented Lagrangian methods with general lower-level constraints, *SIAM Journal on Optimization* 18 (2007) 1286-1309.
- [6] R. Andreani, E. G. Birgin, J. M. Martínez & M. L. Schuverdt. Augmented Lagrangian methods under the Constant Positive Linear Dependence constraint qualification, *Mathematical Programming* 111 (2008) 5-32.
- [7] R. Andreani, E. G. Birgin, J. M. Martínez & J. Yuan. Spectral projected gradient and variable metric methods for optimization with linear inequalities, *IMA Journal of Numerical Analysis* 25 (2005) 221-252.

- [8] M. Andretta, E. G. Birgin & J. M. Martínez. Practical active set Euclidian trust-region method with spectral projected gradients for bound-constrained optimization. *Optimization* 54 (2005) 305-325.
- [9] M. Andretta, E. G. Birgin & J. M. Martínez. Partial Spectral Projected Gradient Method with Active-Set Strategy for Linearly Constrained Optimization, *Numerical Algorithms* 53 (2010) 23-52.
- [10] M. S. Bazaraa, H. D. Sherali & C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*. 2 ed. New York, John Wiley, 1993.
- [11] H. Y. Benson & D. F. Shanno. Interior-Point Methods for Nonconvex Nonlinear Programming: Regularization and Warmstarts. *Computational Optimization and Applications* 40:2 (2008) 143-189.
- [12] H. Benson, D. F. Shanno & R. J. Vanderbei. A Comparative Study of Large Scale Nonlinear Optimization Algorithms. In: *High Performance Algorithms and Software for Nonlinear Optimization* (G. Di Pillo & A. Murli eds.) pp.95-127, Kluwer Academic Publishers, Massachusetts, USA, 2003.
- [13] H. Y. Benson, D. F. Shanno & R. J. Vanderbei. Interior-Point Methods for Nonconvex Nonlinear Programming: Jamming and Comparative Numerical Testing. *Mathematical Programming* 99:1 (2005) 35-48.
- [14] E. G. Birgin, E. V. Castelani, A. L. Martinez & J. M. Martínez. Outer Trust-Region Method for Constrained Optimization, *Journal of Optimization Theory and Applications*, a aparecer, 2011.
- [15] E. G. Birgin, R. Castillo & J. M. Martínez. Numerical comparison of Augmented Lagrangian algorithms for nonconvex problems, *Computational Optimization and Applications* 31 (2005) 31-56.
- [16] E. G. Birgin, C. A. Floudas & J. M. Martínez. Global minimization using an Augmented Lagrangian method with variable lower-level constraints. *Mathematical Programming* 125 (2010) 139-162.
- [17] E. G. Birgin & J. M. Martínez. A spectral conjugate gradient method for unconstrained optimization, *Applied Mathematics and Optimization* 43 (2001) 117-128.

- [18] E. G. Birgin & J. M. Martínez. Large-scale active-set box-constrained optimization method with spectral projected gradients, *Computational Optimization and Applications* 23 (2002) 101-125.
- [19] E. G. Birgin & J. M. Martínez. Structured minimal-memory inexact quasi-Newton method and secant preconditioners for Augmented Lagrangian Optimization, *Computational Optimization and Applications* 39 (2008) 1-16.
- [20] E. G. Birgin & J. M. Martínez. Improving ultimate convergence of an Augmented Lagrangian method, *Optimization Methods and Software* 23 (2008) 177-195.
- [21] E. G. Birgin, J. M. Martínez & M. Raydan. Nonmonotone spectral projected gradient methods on convex sets, *SIAM Journal on Optimization* 10 (2000) 1196-1211.
- [22] E. G. Birgin, J. M. Martínez & M. Raydan. Algorithm 813: SPG - Software for convex-constrained optimization. *ACM Transactions on Mathematical Software* 27 (2001) 340-349 .
- [23] E. G. Birgin, J. M. Martínez & M. Raydan. Inexact Spectral Projected Gradient methods on convex sets. *IMA Journal on Numerical Analysis* 23 (2003) 539-559.
- [24] J. F. Bonnans & M. Bouhtou. The trust region affine interior point algorithm for convex and nonconvex quadratic programming. *RAIRO, Recherche Opérationnelle* 29:2 (1995) 195-217.
- [25] J. F. Bonnans & C. Pola. A trust region interior point algorithm for linearly constrained optimization. *SIAM Journal of Optimization* 7:3 (1997) 717-731.
- [26] J. P. Boyle & R. L. Dykstra. A method for finding projections onto the intersection of convex sets in Hilbert spaces. In: *Advances in Order Restricted Statistical Inference, Lecture Notes in Statistics* 37 pp.28-47, Springer-Verlag, Berlin, 1985.
- [27] R. Byrd, M. E. Hribar & J. Nocedal. An Interior Point Method for Large Scale Nonlinear Programming, *SIAM Journal on Optimization* 9:4 (1999) 877-900.
- [28] R. Byrd, J.C. Gilbert & J. Nocedal, A Trust Region Method Based on Interior Point Techniques for Nonlinear Programming, *Mathematical Programming A* 89 (2000) 149-185.

- [29] R. H. Byrd, N. I. M. Gould, J. Nocedal & R. A. Waltz. An algorithm for nonlinear optimization using linear programming and equality constrained subproblems. *Mathematical Programming B* 100:1 (2004) 27-48.
- [30] R. Byrd, G. Liu & J. Nocedal, On the Local Behavior of an Interior Point Method for Nonlinear Programming, in *Numerical Analysis* (D.F. Griffiths, D.J. Higham & G.A. Watson, eds.) pp.37-56, Addison Wesley Longman, Harlow, Essex, 1997.
- [31] R. H. Byrd, J. Nocedal & R. A. Waltz, KNITRO: An Integrated Package for Nonlinear Optimization in *Large-Scale Nonlinear Optimization*, (G. di Pillo & M. Roma, eds.) pp.35-59, Springer-Verlag, Berlin, Heidelberg, 2006.
- [32] T. F. Coleman & Y. Li, On the Convergence of Reflective Newton Methods for Large-Scale Nonlinear Minimization Subject to Bounds, *Mathematical Programming* 67:2 (1994) 189-224.
- [33] T. F. Coleman & Y. Li, An Interior, Trust Region Approach for Nonlinear Minimization Subject to Bounds, *SIAM Journal on Optimization* 6 (1996) 418-445
- [34] T. F. Coleman & Y. Li, A Reflective Newton Method for Minimizing a Quadratic Function Subject to Bounds on some of the Variables, *SIAM Journal on Optimization* 6 (1996) 1040-1058.
- [35] T. F. Coleman & Y. Li. Combining trust region and affine scaling for linearly constrained nonconvex minimization. In: *Advances in Nonlinear Programming*, (Y. Yuan, ed.) pp.219-250, Kluwer Academic Publishers, Boston, Dordrecht and London, 1998.
- [36] T. F. Coleman & Y. Li. A trust region and affine scaling interior point method for nonconvex minimization with linear inequality constraints. *Mathematical Programming* 88 (2000) 1-31.
- [37] T. Coleman & A. Verma. A Preconditioned conjugate gradient approach to linear equality constrained minimization. *Computational Optimization and Applications* 20 (2001) 61-72.
- [38] A. R. Conn, N. I. M. Gould, Ph. L. Toint, *LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A)*, Springer-Verlag, Berlin, Heidelberg, 1992.

- [39] A. R. Conn, N. I. M. Gould & Ph. L. Toint, *Trust-Region Methods*, SIAM, Philadelphia, 2000.
- [40] F. E. Curtis, O. Schenk & A. Wächter, An interior-point algorithm for large-scale nonlinear optimization with inexact step computations *SIAM Journal on Scientific Computing* 32:6 (2010) 3447-3475.
- [41] F. E. Curtis, J. Huber, O. Schenk & A. Wächter, On the Implementation of an Interior-Point Algorithm for Nonlinear Optimization with Inexact Step Computations, disponível em <http://fgb.informatik.unibas.ch/people/oschenk/publications/references/imips.pdf>, manuscrito, 25 pág. 2011. Último acesso em 18/abril/2011.
- [42] J. E. Dennis Jr. & R. B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. SIAM, Philadelphia, 1996.
- [43] E. D. Dolan, R. Fourer, J. J. Moré & T. S. Munson. *The NEOS Server for Optimization - Version 4 and Beyond*. Preprint ANL/MCS-P947-0202, Argonne National Laboratory, February 2002. Mathematics and Computer Science Division, Argonne, Illinois. 29 pág.
- [44] Z. Dostál. *Optimal Quadratic Programming Algorithms*. New York, Springer-Verlag, 2009.
- [45] A. S. Drud. CONOPT: A GRG code for large sparse dynamic nonlinear optimization problems. *Mathematical Programming* 31:2 (1985) 153-191.
- [46] A. S. Drud. CONOPT - A Large-Scale GRG Code. *ORSA Journal on Computing* 6 (1992) 207-216.
- [47] L. Eldén, P. C. Hansen & M. Rojas. Minimization of Linear Functionals Defined on Solutions of Large-Scale Discrete Ill-Posed Problems. *BIT Numerical Mathematics* 45:2 (2005) 329-340.
- [48] J. B. Erway, P. E. Gill & J. D. Griffin. Iterative methods for finding a trust-region step. *SIAM Journal of Optimization* 20:2 (2009) 1110-1131.
- [49] A. V. Fiacco & G. P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. SIAM, Philadelphia, 1990.

- [50] A. L. Forsgren, P. E. Gill & M. H. Wright. Interior Methods for Nonlinear Optimization. *SIAM Review* 44:4 (2002) 525-597.
- [51] A. L. Forsgren & W. Murray. Newton Methods for Large- Scale Linear Equality Constrained Minimization. *SIAM Journal on Matrix Analysis and Applications* 14 (1993) 560-587.
- [52] A. L. Forsgren & W. Murray. Newton Methods for Large-Scale Linear Inequality Constrained Minimization, *SIAM Journal of Optimization* 7 (1997) 162-176.
- [53] R. Fourer, D. M. Gay & B. W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*, 2nd ed. Brooks/Cole Publishing Company/Cengage Learning, 2002.
- [54] D. M. Gay. *On convergence testing in model/trust-region algorithms for unconstrained optimization*. Computing Science, Technical Report 104, September 16, 1982, Bell Laboratories, Murray Hill, New Jersey, 07974. 10 pág.
- [55] D. M. Gay. A trust-region approach to linearly constrained optimization. In: *Numerical Analysis* (D.F. Griffiths ed.) *Lecture Notes in Mathematics* 1066 pp.72-105, Springer-Verlag, Berlin, 1984.
- [56] P. E. Gill & W. Murray. Newton-type methods for linearly constrained optimization. In: *Numerical Methods for constrained optimization* (P.E. Gill & W. Murray eds.) pp.29-66. Academic Press, New York, 1974.
- [57] P. E. Gill, W. Murray & M. A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Journal on Optimization* 12:4 (2002) 979-1006.
- [58] P. E. Gill, W. Murray, M. A. Saunders. *User's Guide for SNOPT Version 7: Software for Large-Scale Nonlinear Programming*, Systems Optimization Laboratory, Stanford University, June 2008, 116 pág. Disponível em <http://sbsi-sol-optimize.com/manuals/SNOPTManual.pdf>.
- [59] P. E. Gill, W. Murray, M. A. Saunders & M. H. Wright. Procedures for Optimization Problems with a Mixture of Bounds and General Linear Constraints, *ACM Trans. Math. Software* 10 (1984) 282-298.
- [60] P. E. Gill, W. Murray, M. A. Saunders & M. H. Wright. Maintaining LU factors of a general sparse matrix. *Linear Algebra and its Applications* 88/89 (1987) 239-270.

- [61] P. E. Gill, W. Murray & M. H. Wright. *Practical Optimization*. San Diego, Academic Press, 1981.
- [62] P. E. Gill, W. Murray & M. H. Wright. *Numerical Linear Algebra and Optimization*, Vol. 1, Addison Wesley, 1991.
- [63] D. Goldfarb & A. Idnani. A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical Programming* 27 (1983) 1-33.
- [64] G. H. Golub & C. F. van Loan. *Matrix Computations*. 3ed. Baltimore and London, The Johns Hopkins University Press, 1996.
- [65] N. I. M. Gould, D. Orban & Ph. L. Toint. *Results from a numerical evaluation of LANCELOT B*. Numerical Analysis Group Internal Report 2002-1, Rutherford Appleton Laboratory, Chilton, Oxfordshire, England, 2002.
- [66] N. I. M. Gould, D. Orban & Ph. L. Toint. GALAHAD, a library of thread-safe Fortran 90 packages for large-scale nonlinear optimization. *ACM Transactions on Mathematical Software* 29:4 (2004) 353-372.
- [67] N. I. M. Gould & Ph. L. Toint. An iterative working-set method for large-scale nonconvex quadratic programming. *Applied Numerical Mathematics* 43 (2002) 109-128.
- [68] N. I. M. Gould & Ph. L. Toint. Numerical methods for large-scale non-convex quadratic programming. In: *Trends in Industrial and Applied Mathematics* (A.H. Siddiqi & M. Kočvara, eds.) pp.149-179, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002.
- [69] S. Gratton, Ph. L. Toint, A. Troltsch. *An active-set trust-region method for derivative-free nonlinear bound-constrained optimization*. Technical Report TR/PA/10/70, CERFACS, France, 2010, 20pág.
- [70] I. Griva, S. G. Nash & A. Sofer, *Linear and Nonlinear Optimization*, SIAM, Philadelphia, 2009.
- [71] W. W. Hager, B. A. Mair & H. Zhang. An affine-scaling interior-point CBB method for box-constrained optimization. *Mathematical Programming* 119 (2009) 1-32.

- [72] W. W. Hager & H. Zhang. A new active set algorithm for box constrained optimization. *SIAM Journal on Optimization* 17 (2006) 526-557.
- [73] A. Izmailov & M. Solodov. *Otimização Volume 1: Condições de otimalidade, elementos de análise convexa e de dualidade*, IMPA, Rio de Janeiro, 2005.
- [74] A. Kearsley. Matrix-free algorithm for the large-scale constrained trust-region subproblem. *Optimization Methods and Software* 21 (2006) 233-245.
- [75] J. Lampe, M. Rojas, D. C. Sorensen & H. Voss. *Accelerating the LSTRS algorithm*. IMM Technical Report 2009-09, July 9, 2009, Department of Informatics and Mathematical Modelling. Technical University of Denmark, Lyngby, Denmark. 11 pág. (A aparecer em *SIAM Journal on Scientific Computing*).
- [76] C. L. Lawson & R. J. Hanson. *Solving Least Squares Problems*. SIAM, Philadelphia, 1995.
- [77] R. B. Lehoucq, D. C. Sorensen & C. Yang, *ARPACK Users Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, SIAM, Philadelphia, 1998.
- [78] C. J. Lin & J. J. Moré. Incomplete Cholesky factorizations with limited memory. *SIAM Journal on Scientific Computing* 21 (1999) 24-45.
- [79] B. Llanas & C. Moreno. Finding the projection on a polytope: an iterative method. *Computers & Mathematics with Applications* 32 (1996) 33-39.
- [80] D. G. Luenberger. *Linear and Nonlinear Programming*. 2nd ed. Kluwer, Boston, Dordrecht and London, 1984.
- [81] I. J. Lustig, R. E. Marsten & D. F. Shanno. The Primal-Dual Interior Point Method on the Cray Supercomputer. In: *Large-scale numerical optimization* (T. F. Coleman, Y. Li eds.) pp.70-80, SIAM, Philadelphia, 1990.
- [82] J. M. Martínez, E. A. Pilotta & M. Raydan. Spectral gradient methods for linearly constrained optimization. *Journal of Optimization Theory and Applications* 125 (2005) 629-651.
- [83] J. M. Martínez & S. A. Santos. *Métodos Computacionais de Otimização*, IMPA, SBM, 20o. Colóquio Brasileiro de Matemática, Rio de Janeiro, 1995.

- [84] G. P. McCormick. A second order method for the linearly constrained nonlinear programming problem. In: *Nonlinear Programming* (J.B.Rosen, O.L Mangasarian & K. Ritter eds.) pp.207-243, Academic Press, London and New York, 1970.
- [85] S. Mehrotra. On the implementation of a (primal-dual) interior point method. *SIAM Journal on Optimization* 2 (1992) 575-601.
- [86] R. D. C. Monteiro & Y. Wang. Trust region affine scaling algorithms for linearly constrained convex and concave programs. *Mathematical Programming* 80 (1998) 283-313.
- [87] J. J. Moré & D. C. Sorensen, Computing a trust region step. *SIAM Journal on Scientific Computing* 4 (1983) 553-572.
- [88] J. J. Moré & G. Toraldo, On the solution of large quadratic programming problems with bound constraints, *SIAM Journal on Optimization* 1 (1991) 93-113.
- [89] B. A. Murtagh & M. A. Saunders. Large-scale linearly constrained optimization. *Mathematical Programming* 14 (1978) 41-72.
- [90] B. A. Murtagh & M. A. Saunders. *MINOS 5.5 User's Guide*, Technical Report SOL 83-20R, Stanford University, 1983-1998, 150 pág. Disponível em <http://www.stanford.edu/group/SOL/guides/minos55.pdf>.
- [91] S. G. Nash & A. Sofer. *Why Extrapolation Helps Barrier Methods*, Technical Report (September 11, 1998), George Mason University. Operations Research and Engineering Department. Fairfax, VA 22030. 19 pág.
- [92] J. Nocedal & S. J. Wright. *Numerical Optimization*. New York, Springer-Verlag, 1999.
- [93] J. Nocedal, A. Wächter & R. A. Waltz. Adaptive barrier update strategies for nonlinear interior methods. *SIAM Journal on Optimization* 19:4 (2009) 1674-1693.
- [94] J. M. Ortega & W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*. Academic Press, London and New York, 1970.
- [95] C. Perkins. A convergence analysis of Dykstra's algorithm for polyedral sets. *SIAM Journal on Numerical Analysis* 40:2 (2003) 792-804.

- [96] T. Plantenga & R. A. Waltz. *KNITRO Performance on Large Nonlinear Problems*, Ziena Optimization, Inc. August 2006. 3 pág. Disponível em <http://www.ziena.com/studies.htm>.
- [97] M. J. D. Powell. On the quadratic programming algorithm of Goldfarb and Idnani. *Mathematical Programming Study* 25 (1985) 46-61.
- [98] M. J. D. Powell. A tolerant algorithm for linearly constrained optimization calculations. *Mathematical Programming* 45 (1989) 547-566.
- [99] M. J. D. Powell. Developments of NEWUOA for minimization without derivatives, *IMA Journal of Numerical Analysis* 28 (2008) 649-664.
- [100] M. J. D. Powell. *The BOBYQA algorithm for bound constrained optimization without derivatives*. Technical Report NA2009/06, August, 2009. Department of Applied Mathematics and Theoretical Physics, Cambridge, England. 39 pág.
- [101] M. Rojas & D. C. Sorensen. A trust-region approach to the regularization of large-scale discrete forms of ill-posed problems. *SIAM Journal on Scientific Computing* 23 (2002) 1842-1860.
- [102] M. Rojas, S. A. Santos & D. C. Sorensen. A New Matrix-Free Algorithm for the Large-Scale Trust-Region Subproblem. *SIAM Journal on Optimization* 11:3 (2000) 611-646.
- [103] M. Rojas, S. A. Santos & D. C. Sorensen. Algorithm 873: LSTRS: MATLAB software for large-scale trust-region subproblems and regularization. *ACM Transactions on Mathematical Software* 34(2):11 (2008) 28p.
- [104] J. B. Rosen. The gradient projection method for nonlinear programming: Part I. Linear constraints, *Journal of the SIAM* 8:1 (1960) 181-217.
- [105] K. Schittkowski. *QL: a Fortran code for convex quadratic programming - User's Guide*, Version 2.11, 2005. 7 pág. Disponível em <http://www.ai7.uni-bayreuth.de/QL.pdf> (acessado em 27 de maio de 2011).
- [106] D. F. Shanno & R. J. Vanderbei. Interior-Point Methods for Nonconvex Nonlinear Programming: Orderings and Higher-Order Methods. *Mathematical Programming* 87:2 (2000) 303-316.

- [107] G. A. Shultz, R. B. Schnabel & R. H. Byrd. A Family of Trust-Region-Based Algorithms for Unconstrained Minimization with Strong Global Convergence Properties. *SIAM Journal on Numerical Analysis* 22:1 (1985) 47-67.
- [108] K. Schittkowski. An Active Set Strategy for Solving Optimization Problems with up to 200,000,000 Nonlinear Constraints. *Applied Numerical Mathematics* 59 (2009) 2999-3007.
- [109] D. C. Sorensen. Newton's method with a model trust -region modification, *SIAM Journal on Numerical Analysis* 19 (1982) 409-426.
- [110] T. Steihaug. The conjugate gradient method and trust regions in large scale optimization, *SIAM Journal on Numerical Analysis* 20 (1983) 626-637.
- [111] G. W. Stewart. *Matrix Algorithms, Volume I: Basic Decompositions*. SIAM, Philadelphia, 2001.
- [112] G. W. Stewart. *Matrix Algorithms, Volume II: Eigensystems*. SIAM, Philadelphia, 2001.
- [113] Ph. L. Toint. Towards an efficient sparsity exploiting Newton method for minimization. In: *Sparse Matrices and Their Uses* (I.S. Duff, ed.) pp.57-88, Academic Press, London and New York, 1981.
- [114] P. Tseng. Convergent Infeasible Interior-Point Trust-Region Methods for Constrained Minimization. *SIAM Journal on Optimization* 13:2 (2002) 432-469.
- [115] R. J. Vanderbei. LOQO: An interior point code for quadratic programming. *Optimization Methods and Software* 12 (1999) 451-484.
- [116] R. J. Vanderbei. *LOQO User's Manual - Version 4.05*, Technical Report No. ORFE-99 (September 13, 2006), Princeton University, School of Engineering and Applied Science, Department of Operations Research and Financial Engineering, Princeton, New Jersey, 08544.
- [117] R. J. Vanderbei & D.F. Shanno. An interior-point algorithm for nonconvex non-linear programming. *Computational Optimization and Applications* 13 (1999) 231-252.

- [118] A. Wächter. *An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, January, 2002.
- [119] A. Wächter & L. T. Biegler. Failure of global convergence for a class of interior point methods for nonlinear programming. *Mathematical Programming* 88:3 (2000) 565-587.
- [120] A. Wächter & L. T. Biegler. Line Search Filter Methods for Nonlinear Programming: Motivation and Global Convergence, *SIAM Journal on Optimization* 16 (2005) 1-31.
- [121] A. Wächter & L. T. Biegler. On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming. *Mathematical Programming* 106:1 (2006) 25-57.
- [122] R. A. Waltz & T. D. Plantenga. *KNITRO User's Manual Version 7.0*, Ziena Optimization, Inc. September 2010. 99 pág. Disponível em <http://www.ziena.com/documentation.htm>.
- [123] Y. Wang & D. Zhu. An affine scaling optimal path method with interior backtracking curvilinear technique for linear constrained optimization, *Applied Mathematics and Computation* 207 (2009) 178-196.
- [124] P. Wolfe. Methods of nonlinear programming. In: *Recent Advances in Mathematical Programming*, (R.L. Graves & P.Wolfe eds.) pp.67-86, McGraw Hill, New York, 1963.
- [125] P. Wolfe. Methods for nonlinear constraints. In: *Nonlinear Programming*, (J. Abadie ed.) pp.99-131, North-Holland, Amsterdam, 1967.
- [126] M. H. Wright. Interior methods for constrained optimization. In: *Acta Numerica*, (A. Iserles ed.) pp.341-407, Cambridge University Press, Port Melbourne, 1992.
- [127] Y. Ye, M. J. Todd & S. Mizuno. An  $O(\sqrt{n}L)$ -iteration homogeneous and self-dual linear programming algorithm. *Mathematics of Operations Research* 19 (1994) 53-67.
- [128] Y. Ye & E. Tse. An extension of Karmarkar's projective algorithm for convex quadratic programming, *Mathematical Programming* 44 (1989) 157-179.

- [129] D. Zhu. Nonmonotonic back-tracking trust region interior point algorithm for linear constrained optimization. *Journal of Computational and Applied Mathematics* 155 (2003) 285-305.
- [130] D. Zhu. An affine scaling reduced preconditional conjugate gradient path method for linear constrained optimization. *Applied Mathematics and Computation* 184 (2007) 181-198.