



Universidade Estadual de Campinas  
Instituto de Matemática Estatística e Computação Científica  
DEPARTAMENTO DE MATEMÁTICA APLICADA



---

# Otimização da Menor Soma de Valores Ordenados

**Flavio Sakakisbara Yano** <sup>†</sup>

Doutorado em Matemática Aplicada - Imecc - Unicamp

**Orientador: Prof. Dr. Roberto Andreani**

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

<sup>†</sup>Este trabalho teve apoio financeiro da FAPESP - Proc. 02/14203-6.

# Otimização da Menor Soma de Valores Ordenados

Este exemplar corresponde à redação final da tese devidamente corrigida e defendida por **Flavio Sakakisbara Yano** e aprovada pela comissão julgadora.

Campinas, 22 de Junho de 2006.



---

Prof. Dr. **Roberto Andreani.**  
Orientador



---

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

**Banca examinadora:**

Prof. Dr. Roberto Andreani (IMECC/UNICAMP)  
Prof. Dr. Ernesto Julián Goldberg Birgin (IME/USP)  
Prof. Dr. Geraldo Nunes Silva (IBILCE/UNESP)  
Prof. Dr. Lúcio Tunes dos Santos (IMECC/UNICAMP)  
Prof. Dr. Ana Friedlander (IMECC/UNICAMP)

Tese apresentada ao Instituto de Matemática Estatística e Computação Científica, UNICAMP como requisito parcial para obtenção do título de **Doutor em Matemática Aplicada.**

**FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DO IMECC DA UNICAMP**

Bibliotecária: Maria Júlia Milani Rodrigues – CRB8a / 2116

Yano, Flavio Sakakisbara

Y16o Otimização da menor soma de valores ordenados / Flavio Sakakisbara Yano -- Campinas, [S.P. :s.n.], 2006.

Orientadores : Roberto Andreani; José Mario Martínez

Tese (doutorado) - Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1. Otimização. 2. Programação não-linear. 3. Estimativa de parâmetro. I. Andreani, Roberto. II. Martínez, José Mario. III. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. IV. Título.

Título em inglês: Low-sum order value optimization.

Palavras-chave em inglês (Keywords): 1. Optimization. 2. Nonlinear programming. 3. Parameter estimation.

Área de concentração: Otimização

Titulação: Doutor em Matemática Aplicada

Banca examinadora: Prof. Dr. Roberto Andreani (IMECC/UNICAMP)  
Prof. Dr. Ernesto Julián Goldberg Birgin (IME/USP)  
Prof. Dr. Geraldo Nunes Silva (IBILCE/UNESP)  
Prof. Dr. Lúcio Tunes dos Santos (IMECC/UNICAMP)  
Prof. Dr. Ana Friedlander de Martínez Perez (IMECC/UNICAMP)

Data da defesa: 22/06/2006

Programa de Pós-Graduação: Doutorado em Matemática Aplicada

Tese de Doutorado defendida em 22 de junho de 2006 e aprovada

Pela Banca Examinadora composta pelos Profs. Drs.



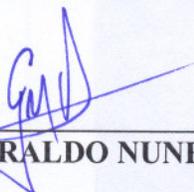
---

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



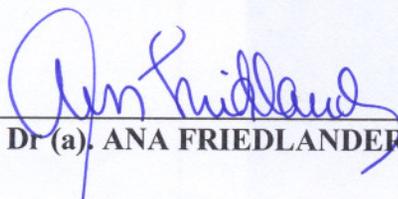
---

Prof. (a). Dr (a). LÚCIO TUNES DOS SANTOS



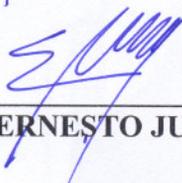
---

Prof. (a). Dr (a). GERALDO NUNES SILVA



---

Prof. (a). Dr (a). ANA FRIEDLANDER DE MARTINEZ PEREZ



---

Prof. (a) Dr. (a) ERNESTO JULIÁN GOLDBERG BIRGIN

Aos meus pais, Alúcio e Rosa.

À minha esposa, Luciani

À minha irmã, Fernanda.

*Dedico*

# Agradecimentos

Primeiramente, aos meus pais, por terem batalhado tanto para me oferecer as melhores oportunidades e as melhores condições para crescer na vida. Hoje, com esta tese de doutorado, espero estar lhes dando um pouco alegria, retribuindo o esforço por eles realizado. Aos meus pais, obrigado por tudo!

Um agradecimento muito especial aos meus orientadores. Ao Prof. Roberto Andreani, agradeço pelo companheirismo, pela paciência e atenção dada desde a iniciação científica, mesmo diante das dificuldades que a vida lhe impôs e que, certamente, é um dos grandes responsáveis pela minha formação. Ao Prof. José Mario Martínez, agradeço pela oportunidade, foi uma grande honra para mim ter trabalhado com um dos mais respeitados e competentes pesquisadores da área. Nossas reuniões ficarão eternamente guardadas em minha lembrança. Nino, Martínez, muito obrigado!

Gostaria de agradecer também à Profa. Ana Friedlander, à Profa. Sandra A. Santos, à Profa. Maria Aparecida D. Ehrhardt, pela ajuda e atenção sem medida, em especial, ao Prof. Lúcio T. dos Santos, que foi sempre muito atencioso em todos os momentos em que Salvatierra e eu necessitamos de apoio para desenvolver o nosso trabalho.

À Fapesp, pelo auxílio financeiro durante esse período.

À minha esposa Luciani, pelo incentivo a cada dia e, principalmente, pelo entendimento e compreensão nos vários momentos em que não pude lhe dar toda a atenção merecida para me dedicar inteiramente a este trabalho.

Aos meus amigos pós-graduandos Chela, Laura, Valéria, Juliano, Feodor, Luziane e Serginho pela ajuda e atenção sem medida. Mário Salvatierra e Valeriano merecem um agradecimento especial. Ao Valeriano agradeço pela ajuda e companheirismo nos momentos mais difíceis que vivemos nas disciplinas obrigatórias e exame de qualificação. Ao Mário agradeço pela ajuda, parceria e companheirismo nos vários trabalhos que realizamos juntos.

Enfim, agradeço a todas as pessoas que contribuíram para a realização deste trabalho.

# Resumo

Dadas  $r$  funções reais  $F_1, \dots, F_r$  definidas em  $\Omega \subset \mathbb{R}^n$  e um inteiro  $p$  entre 1 e  $r$ , o problema de otimização da menor soma de valores ordenados (LOVO) consiste em minimizar a soma das funções que tomam os  $p$  menores valores. Se  $(y_1, \dots, y_r)$  é um vetor de dados e  $T(x, t_i)$  é o valor previsto da observação  $i$  ao adotar um parâmetro  $x \in \Omega$ , é natural definir  $F_i(x) = (T(x, t_i) - y_i)^2$  (erro quadrático da observação  $i$  quando  $x$  é o parâmetro adotado). Neste caso, quando  $p = r$ , temos o clássico problema de quadrados mínimos não-linear. Entretanto, a situação é mais interessante quando  $p$  é menor que  $r$ . Neste caso, é possível desprezar um número pré-determinado de observações ruins. Assim, o problema LOVO aponta como uma ferramenta interessante para fazer estimação robusta de parâmetros. Quando  $p \ll r$  o LOVO pode ser usado para encontrar padrões ocultos em um determinado conjunto de dados. Neste trabalho discutiremos condições de otimalidade, algoritmos para resolver o LOVO serão introduzidos e teoremas de convergência serão provados. Finalmente, experimentos numéricos serão apresentados.

**Palavras-chave:** Otimização de valores ordenados, algoritmos, convergência, estimação robusta de parâmetros, padrões ocultos.

# Abstract

Given  $r$  real functions  $F_1(x), \dots, F_r(x)$  defined in  $\Omega \subset \mathbb{R}^n$  and an integer  $p$  between 1 and  $r$ , the Low Order-Value Optimization problem (LOVO) consists of minimizing the sum of the functions that take the  $p$  smallest values. If  $(y_1, \dots, y_r)$  is a vector of data and  $T(x, t_i)$  is the predicted value of the  $i$ -th observation with the parameters  $x \in \Omega$ , it is natural to define  $F_i(x) = (T(x, t_i) - y_i)^2$  (the quadratic error at observation  $i$  under the parameters  $x$ ). When  $p = r$  this LOVO problem coincides with the classical nonlinear least-squares problem. However, the interesting situation is when  $p$  is smaller than  $r$ . In that case, the solution of LOVO allows one to discard the influence of an estimated number of outliers. Thus, the LOVO problem is an interesting tool for robust estimation of parameters of nonlinear models. When  $p \ll r$  the LOVO problem may be used to find hidden structures in data sets. In this work optimality conditions are discussed, algorithms for solving the LOVO problem are introduced and convergence theorems are proved. Finally, numerical experiments are presented.

**Key words:** Order-Value Optimization, algorithms, convergence, robust estimation of parameters, hidden patterns.

# Sumário

<b>Introdução</b>	<b>1</b>
<b>1 Otimização de Valores Ordenados</b>	<b>7</b>
1.1 Otimalidade . . . . .	7
1.2 Algoritmo Cauchy-Primal para OVO . . . . .	7
1.3 Algoritmo Quase-Newton para OVO . . . . .	9
1.4 Reformulação Diferenciável do OVO . . . . .	13
<b>2 Encontrando Padrões Ocultos com OVO</b>	<b>17</b>
2.1 Metodologia . . . . .	18
2.2 Experimentos Numéricos . . . . .	19
2.2.1 Encontrando Polinômios Ocultos . . . . .	20
2.2.2 Encontrando Circunferências Ocultas . . . . .	21
2.2.3 Encontrando “Bananas” Ocultas . . . . .	24
2.2.4 Encontrando Elipses Ocultas . . . . .	26
2.3 Análise dos Resultados . . . . .	26
<b>3 Otimizando o Valor em Risco VaR com OVO</b>	<b>29</b>
3.1 Valor em Risco VaR . . . . .	29
3.2 Aplicação . . . . .	32
3.3 Experimentos Numéricos . . . . .	36
3.3.1 Aproximações da Hessiana e parâmetros do algoritmo . . . . .	36
3.3.2 Escolha do ponto inicial . . . . .	37
3.3.3 Resultados Numéricos . . . . .	38

---

<b>4</b>	<b>Otimização da Menor Soma de Valores Ordenados</b>	<b>41</b>
4.1	Otimidade . . . . .	41
4.2	Algoritmos para LOVO Irrestrito . . . . .	44
4.2.1	Convergência para pontos críticos fracos . . . . .	44
4.2.2	Convergência para pontos críticos fortes . . . . .	55
4.3	Problema LOVO com Restrições . . . . .	67
4.3.1	Lagrangiano Aumentado Diferenciável . . . . .	67
4.3.2	Método Lagrangiano Aumentado para o LOVO . . . . .	71
<b>5</b>	<b>Quadrados Mínimos LOVO</b>	<b>76</b>
5.1	Ajustando Modelos com “Outliers” . . . . .	77
5.1.1	Implementação . . . . .	77
5.1.2	Caso Irrestrito . . . . .	78
5.1.3	Caso com Restrições . . . . .	85
5.2	Padrão Oculto - Proteína . . . . .	88
	<b>Conclusão</b>	<b>93</b>
	<b>Referências Bibliográficas</b>	<b>95</b>

# Introdução

Dadas  $m$  funções reais  $f_1, \dots, f_m$  definidas em um domínio  $\Omega \subset \mathbb{R}$  e um inteiro  $p \in \{1, \dots, m\}$ , definimos a função de valores ordenados (OVO),  $f : \Omega \rightarrow \mathbb{R}$ , de ordem  $p$  por:

$$f(x) = f_{i_p(x)}(x),$$

para todo  $x \in \Omega$ , onde  $\{i_1(x), \dots, i_m(x)\} = \{1, \dots, m\}$  e

$$f_{i_1(x)}(x) \leq f_{i_2(x)}(x) \leq \dots \leq f_{i_p(x)}(x) \leq \dots \leq f_{i_m(x)}(x).$$

Para fixar idéias, considere as funções reais  $f_1, f_2$  e  $f_3$ , ilustradas na Figura 1. Fixando  $x = 0.4$ , podemos observar que  $f_3(x) \leq f_1(x) \leq f_2(x)$ . Logo, para o caso  $p = 2$ , temos  $f(0.4) = f_{i_2(0.4)}(0.4) = f_1(0.4)$ . Aplicando este raciocínio para todo  $x \in [-1, 1]$  obtemos a função OVO de ordem 2 no domínio  $\Omega = [-1, 1]$ , cuja representação está destacada na figura abaixo.

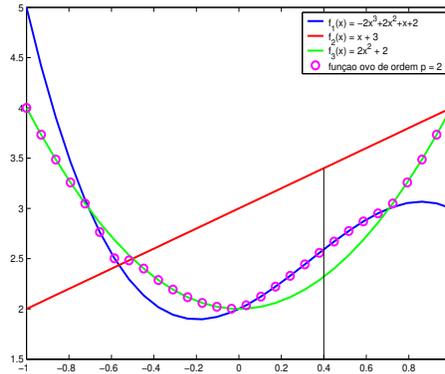


Figura 1: Ilustração da função OVO

Se  $p = 1$ ,  $f(x) = \min\{f_1(x), \dots, f_m(x)\}$ , ao passo que, para  $p = m$  temos  $f(x) = \max\{f_1(x), \dots, f_m(x)\}$ . A função OVO é contínua [4]. Entretanto, mesmo que as funções  $f_i$  sejam diferenciáveis a função OVO é, na maioria dos casos, uma função não-diferenciável. Este fato pode ser visualizado na Figura 1.

O problema de otimização de valores ordenados (OVO - Order Value Optimization) consiste em:

$$\text{Minimizar } f(x) \text{ s.a. } x \in \Omega. \quad (1)$$

O problema (1) é uma generalização do problema Minimax. E, seguindo esta visão descreveremos, a seguir, algumas aplicações para o problema OVO:

1. Suponha que  $\Omega$  é um espaço de decisões e que, para cada  $x \in \Omega$ ,  $f_i(x)$  represente o custo da decisão  $x$  sob o cenário  $i$ . A decisão Minimax corresponde à escolha de  $x$  de forma que o máximo custo seja minimizado. Esta é uma alternativa de decisão muito pessimista e os tomadores de decisão poderiam optar por descartar as piores possibilidades com o intuito de proceder de uma forma mais realista. Quando  $f_i(x)$  representa a perda esperada de um vetor das posições de um portfólio  $x$  sob o cenário  $i$ , a função OVO de ordem  $p$  ( $f_{i_p}(x)$ ) representa o VaR (Value-at-Risk) da decisão  $x$  com nível de confiança  $p/m$  [30, 40, 41].
2. Suponha que temos um problema de estimação de parâmetros onde o espaço dos parâmetros é  $\Omega$  e que  $f_i(x)$  corresponde ao erro quadrático da observação  $i$  quando o parâmetro  $x$  é o escolhido. A estimativa Minimax corresponde àquela que minimiza o máximo erro. Não é novidade alguma que esta estimativa é muito sensível a presença de “outliers” [29]. Por vezes, queremos eliminar (digamos) os 15% maiores erros porque estes podem estar representando observações incorretas. Isto significa minimizar a função de valores ordenados com  $p \approx 0.85 \times m$ , onde  $m$  representa a quantidade de observações.
3. Suponha que queremos identificar padrões ocultos na presença de dados com muito ruído. Nesta aplicação, um padrão é um subconjunto  $S(x) \subset \mathbb{R}^n$ , determinado em função de um parâmetro  $x$ . Por exemplo, podemos definir  $S(x)$  como sendo o conjunto dos pontos em  $\mathbb{R}^2$  que pertencem ao gráfico de uma parábola com coeficientes  $x$ , ou seja,  $S(x) = \{(t, y) \in \mathbb{R}^2 \mid y = x_1 t^2 + x_2 t + x_3\}$ . Um conjunto de pontos  $\{P_1, \dots, P_m\} \subset \mathbb{R}^n$  é dado. Para cada  $i = 1, \dots, m$ ,  $f_i(x)$  é uma “pseudo-distância” entre  $P_i$  e  $S(x)$ . No exemplo da parábola, para cada  $P_i = (t_i, y_i)$ , podemos definir  $f_i(x) = (x_1 t_i^2 + x_2 t_i + x_3 - y_i)^2$ . Somente uma pequena quantidade de pontos pertencem ao padrão oculto correto  $S(x_*)$ , portanto, os parâmetros  $x_*$  são obtidos resolvendo o problema OVO para pequenos valores de  $p$ . Ao contrário

dos exemplos anteriores, os problemas OVO associados a pequenos valores para  $p$  são os mais interessantes.

O problema OVO é difícil porque a função objetivo é não-diferenciável (mesmo que as funções  $f_i$  sejam) e porque este, geralmente, possui muitos minimizadores locais. Em [4], um método tipo Cauchy foi proposto para resolver (1). Já, em [9], propusemos um método Quase-Newton que generaliza o método de Cauchy. Em [5], foi definida uma reformulação que transforma o problema OVO em um problema de programação não-linear diferenciável.

Um outro problema a ser tratado neste trabalho, é o problema de otimização da menor soma ordenada (LOVO - Low Order Value Optimization). Dadas  $r$  funções reais  $F_1, \dots, F_r$  definidas em um domínio  $\Omega \subset \mathbb{R}$  e um inteiro  $p \in \{1, \dots, r\}$ , definimos a função da menor soma de valores ordenados (LOVO),  $S_p : \Omega \rightarrow \mathbb{R}$ , de ordem  $p$  por:

$$S_p(x) = \sum_{j=1}^p F_{i_j(x)}(x)$$

para todo  $x \in \Omega$ , onde  $\{i_1(x), \dots, i_r(x)\} = \{1, \dots, r\}$  e

$$F_{i_1(x)}(x) \leq F_{i_2(x)}(x) \leq \dots \leq F_{i_p(x)}(x) \leq \dots \leq F_{i_r(x)}(x).$$

Se as funções  $F_i$  são contínuas então a função  $S_p$  também é contínua, porque esta é uma soma de funções OVO que, por sua vez, são contínuas [4, 5]. Entretanto, mesmo que as funções  $F_i$  sejam diferenciáveis a função LOVO é, na maioria dos casos, uma função não-diferenciável. Definimos o problema LOVO da seguinte forma:

$$\text{Minimizar } S_p(x) \text{ s.a. } x \in \Omega. \quad (2)$$

Defina  $m = r!/[p!(r-p)!]$ . Claramente, o conjunto  $\{1, \dots, r\}$  contém exatamente  $m$  subconjuntos distintos  $\mathcal{C}_1, \dots, \mathcal{C}_m$  de cardinalidade  $p$ . Para todo  $i = 1, \dots, m$  e  $x \in \Omega$ , definimos:

$$f_i(x) = \sum_{j \in \mathcal{C}_i} F_j(x) \quad (3)$$

e

$$f_{\min}(x) = \min\{f_1(x), \dots, f_m(x)\}.$$

Indutivamente, é fácil verificar que  $f_{\min}(x) = S_p(x)$ , para todo  $x \in \Omega$  e, portanto, o problema LOVO consiste em:

$$\text{Minimizar } f_{\min}(x) \text{ s.a. } x \in \Omega. \quad (4)$$

Isto significa que o problema LOVO é um problema Minimin (caso particular do OVO,  $p = 1$ ). A caracterização (4) do problema LOVO será utilizada para fins teóricos. Entretanto, devemos ter em mente que, na prática, estamos interessados na formulação (2) e que, evidentemente, não precisamos calcular  $f_1, \dots, f_m$  para avaliar  $S_p$ .

Vimos anteriormente, que o problema OVO (minimizar  $F_{i_p(x)}(x)$ ) pode ser aplicado à estimação robusta de parâmetros porque este é uma generalização da clássica regressão Minimax. Entretanto, o LOVO é mais adequado para estimação robusta, se definirmos  $F_i(x)$  de forma apropriada. Se  $y_1, \dots, y_r \in \mathbb{R}$  são observações de um determinado fenômeno que, teoricamente, corresponde a uma lei  $y = T(x, t)$ , devemos definir  $F_i(x)$  como sendo o erro quadrático da  $i$ -ésima observação, ou seja,  $F_i(x) = (T(x, t_i) - y_i)^2$ . A estimação de quadrados mínimos dos parâmetros  $x$  é obtida resolvendo:

$$\text{Minimizar } \sum_{i=1}^r F_i(x) \text{ s.a. } x \in \Omega.$$

Se soubermos a priori que aproximadamente  $r - p$  observações são incorretas, é natural estimarmos os parâmetros  $x$  resolvendo o seguinte problema LOVO:

$$\text{Minimizar } S_p(x) \text{ s.a. } x \in \Omega.$$

Portanto, o problema LOVO acima é uma generalização do problema de quadrados mínimos não-linear, capaz de eliminar a influência de “outliers”.

Quando  $F_i(x)$  representa a perda esperada da decisão  $x$  sob o cenário  $i$ , a função OVO ( $F_{i_p(x)}(x)$ ), corresponde ao VaR da decisão  $x$  com nível de confiança  $p/r$ . Diferentemente do OVO, o problema LOVO não é aplicável para medição de risco. A razão é que, se as funções  $F_i(x)$  representam a perda esperada da decisão  $x$  sob o cenário  $i$ , a função LOVO descarta as maiores perdas (como OVO e VaR), mas, não descarta as menores. Logo, as decisões obtidas com LOVO serão sempre otimistas e arriscadas.

Por outro lado, o problema LOVO com  $p \ll r$  pode ser uma ferramenta útil para encontrar padrões ocultos em situações em que uma pequena quantidade de dados corretos estiver misturada com uma grande quantidade de observações ruins [7, 8, 10].

É importante ressaltar, que a principal contribuição desta tese é a introdução do problema LOVO. Como vimos anteriormente, o problema LOVO está fortemente relacionado com o problema OVO. Desta forma, destinamos os primeiros capítulos para fazer uma revisão bibliográfica sobre o problema OVO. No Capítulo 1, resumimos o material existente na literatura sobre o OVO [4, 5, 9, 10]. Dentre os tópicos discutidos,

destacam-se: condições de otimalidade, algoritmos e seus resultados de convergência e a reformulação diferenciável. No Capítulo 2, apresentamos os resultados numéricos obtidos ao aplicarmos a estratégia OVO para encontrar padrões ocultos [10]. No Capítulo 3, discutimos brevemente sobre a relação do OVO com VaR e apresentamos os resultados numéricos obtidos ao aplicarmos a estratégia OVO para minimizar o VaR do retorno (lucro) de uma instituição financeira, baseando-nos no modelo do sistema bancário ideal de Stiglitz (Prêmio Nobel de Economia em 2001) [9]. No Capítulo 4, definimos dois tipos de condições de otimalidade para o problema LOVO. Para o problema LOVO irrestrito, definimos um algoritmo que converge para pontos críticos fracos e um outro em que temos garantia de convergência para pontos críticos fortes. Em ambos os casos, provamos resultados de convergência local e global. Introduzimos também um algoritmo para o problema LOVO com restrições e provamos seus resultados de convergência. Finalmente, no Capítulo 5, experimentos numéricos relativos à aplicação da estratégia LOVO para estimação robusta de parâmetros e, para encontrar o padrão oculto de uma proteína, são apresentados [7, 8].

### Notação:

$\|\cdot\|$  é a norma-2 de vetores e matrizes;

$\mathcal{B}(z, \delta) = \{x \in \mathbb{R}^n \mid \|x - z\| < \delta\}$ ;

$\mathcal{N}(A)$  denota o núcleo da matriz  $A$ ;

$\mathcal{R}(Z)$  denota a imagem da matriz  $Z$ ;

$\mathbb{R}_+ = \{t \in \mathbb{R} \mid t \geq 0\}$  e  $\mathbb{R}_{++} = \{t \in \mathbb{R} \mid t > 0\}$ .

Dado  $K = \{k_0, k_1, k_2, \dots\}$  tal que  $k_j < k_{j+1}$  e  $k_j \in \mathbb{N}$  para todo  $j \in \mathbb{N}$ , denotamos

$$\lim_{k \in K} z_k = \lim_{j \rightarrow \infty} z_{k_j}.$$

Se  $B \in \mathbb{R}^{n \times n}$ ,  $B > 0$  significa que  $B$  é definida positiva.

$[v]_i$  denota a  $i$ -ésima componente do vetor  $v$ . Se não houver espaço para confusão, denotamos também  $v_i = [v]_i$ .

Se  $v \in \mathbb{R}^n$ , denotamos  $v_+ = (\max\{0, v_1\}, \dots, \max\{0, v_n\})^T$ .

# Capítulo 1

## Otimização de Valores Ordenados

Neste capítulo, resumiremos todo o material existente na literatura sobre o problema OVO [4, 5, 9, 10]. Dentre os tópicos, destacam-se as condições de otimalidade, algoritmos e seus resultados de convergência e a reformulação não-linear diferenciável do OVO.

### 1.1 Otimalidade

Considerando os casos em que o conjunto  $\Omega \subset \mathbb{R}^n$  do problema (1) é convexo e fechado, os autores provaram em [4] a seguinte condição de otimalidade:

**Definição.** Dado  $\varepsilon \geq 0$ ,  $x \in \Omega$  definimos:

$$I_\varepsilon(x) = \{j \mid f(x) - \varepsilon \leq f_j(x) \leq f(x) + \varepsilon\}.$$

**Definição.** Dizemos que  $x$  é  $\varepsilon$ -ótimo se

$$\mathcal{D}(x) \equiv \{d \in \mathbb{R}^n \mid x + d \in \Omega \text{ e } \nabla f_j(x)^T d < 0, \forall j \in I_\varepsilon(x)\} = \emptyset.$$

**Teorema 1.1.** *Se  $x_*$  é um minimizador local de (1) então  $x_*$  é  $\varepsilon$ -ótimo para todo  $\varepsilon \geq 0$ .*

### 1.2 Algoritmo Cauchy-Primal para OVO

Nesta seção vamos descrever o algoritmo Cauchy-Primal proposto em [4] e seu teorema de convergência. Diz-se que este método é “primal” porque nenhuma variável auxiliar ou dual é envolvida. O algoritmo gera uma sequência monótona (os valores

funcionais são decrescentes) usando direções de buscas obtidas da resolução (inexata) de um problema convexo.

**Algoritmo 1.1.** Seja  $x_0 \in \Omega$  um ponto inicial arbitrário. Sejam

$$\alpha \in (0, 1), \Delta > 0, \varepsilon > 0, \eta \in (0, 1], 0 < \sigma_{min} < \sigma_{max} < 1$$

parâmetros do algoritmo. Dado  $x_k \in \Omega$ , os passos para obter a nova iteração  $x_{k+1}$  são:

**Passo 1** (Resolução do Subproblema)

Defina, para todo  $d \in \mathbb{R}^n$ ,

$$M_k(d) \equiv \max_{j \in I_\varepsilon(x)} \nabla f_j(x_k)^T d.$$

Considere o subproblema:

$$\text{Minimizar } M_k(d) \text{ s.a. } x_k + d \in \Omega, \|d\|_\infty \leq \Delta. \quad (1.1)$$

Seja  $\bar{d}_k$  a solução de (1.1). Seja  $d_k$  tal que  $x_k + d_k \in \Omega$ ,  $\|d_k\|_\infty \leq \Delta$  e

$$M_k(d_k) \leq \eta M_k(\bar{d}_k).$$

**Critério de Parada:** Se  $M_k(d_k) = 0$ , parar.

**Passo 2** (Busca Linear)

Defina  $t \leftarrow 1$ . Se

$$f(x_k + td_k) \leq f(x_k) + \alpha t M_k(d_k) \quad (1.2)$$

defina  $t_k = t$ ,  $x_{k+1} = x_k + t_k d_k$  e finalize a iteração. Caso contrário, escolha  $t_{novo} \in [0.1t, 0.9t]$ , defina  $t \leftarrow t_{novo}$  e repita o teste (1.2).

Observe que, quando  $\Omega$  é convexo, o subproblema (1.1) é equivalente ao seguinte problema convexo:

$$\begin{aligned} &\text{Minimizar } w \\ &\nabla f_j(x_k)^T d \leq w, \quad \forall j \in I_\varepsilon(x) \\ &x_k + d \in \Omega, \quad \|d\|_\infty \leq \Delta. \end{aligned}$$

Com a hipótese **A0**, o seguinte teorema foi provado em [4].

**Hipótese A0.** O conjunto  $\Omega$  é fechado e limitado. Além disso, existe  $c > 0$ ,  $L > 0$  tal que, para todo  $x, y \in \Omega$ ,  $j = 1, \dots, m$ ,

$$\|\nabla f_j(x)\|_\infty \leq c, \quad \|\nabla f_j(y) - \nabla f_j(x)\|_\infty \leq L\|y - x\|_\infty.$$

**Teorema 1.2.** Suponha que a Hipótese A0 é satisfeita. Seja  $x_k \in \Omega$  a  $k$ -ésima iteração do Algoritmo 1.1. Então:

1. O algoritmo pára em  $x_k$  se, e somente se,  $x_k$  é  $\varepsilon$ -ótimo. Se o algoritmo não pára em  $x_k$ , então a  $k$ -ésima iteração está bem definida e termina no Passo 2 com o cálculo de  $x_{k+1}$ .
2. Suponha que  $x_* \in \Omega$  é um ponto limite de uma sequência gerada pelo Algoritmo 1.1. Então  $x_*$  é  $\varepsilon$ -ótimo.

## 1.3 Algoritmo Quase-Newton para OVO

O algoritmo Quase-Newton para o problema OVO, que apresentaremos nesta seção, é uma generalização do método Cauchy-Primal. Ao invés de fazer uma aproximação linear das funções  $f_i$ , tentamos uma aproximação quadrática tomando matrizes hessianas semidefinidas positivas do tipo quase-Newton. Em [9], obtivemos os mesmos resultados de convergência global do algoritmo Cauchy-Primal [4] e, sob algumas hipóteses mais restritivas, obtivemos resultados de convergência superlinear e quadrática. A descrição do algoritmo e uma síntese destes resultados serão apresentadas a seguir.

**Algoritmo 1.2.** Seja  $x_0 \in \Omega$  um ponto inicial arbitrário. Sejam

$$\alpha \in (0, 1), \quad \Delta > 0, \quad \varepsilon > 0, \quad 0 < \sigma_{min} < \sigma_{max} < 1, \quad \eta \in (0, 1], \quad c_B > 0$$

parâmetros do algoritmo. Suponha que  $x_k \in \Omega$  e que  $\{B_{k,1}, \dots, B_{k,m}\}$  são matrizes simétricas semidefinidas positivas tais que  $\|B_{k,j}\| \leq c_B$  para todo  $j = 1, \dots, m$ . Os passos para obter a nova iteração  $x_{k+1}$  são:

**Passo 1** (Resolução do Subproblema)

Defina, para todo  $d \in \mathbb{R}^n$ ,

$$M_k(d) \equiv \max_{j \in I_\varepsilon(x_k)} \left\{ \nabla f_j(x_k)^T d + \frac{1}{2} d^T B_{k,j} d \right\}.$$

Considere o subproblema

$$\text{Minimizar } M_k(d) \text{ s.a. } x_k + d \in \Omega, \|d\|_\infty \leq \Delta. \quad (1.3)$$

Seja  $\bar{d}_k$  a solução de (1.3). Seja  $d_k$  tal que  $x_k + d_k \in \Omega$ ,  $\|d_k\|_\infty \leq \Delta$  e

$$M_k(d_k) \leq \eta M_k(\bar{d}_k).$$

**Cr terio de Parada:** Se  $M_k(d_k) = 0$ , parar.

**Passo 2** (Busca Linear)

Defina  $t \leftarrow 1$ . Se

$$f(x_k + td_k) \leq f(x_k) + \alpha t M_k(d_k) \quad (1.4)$$

defina  $t_k = t$ ,  $x_{k+1} = x_k + t_k d_k$  e finalize a itera  o. Caso contr rio, escolha  $t_{\text{nov}} \in [0.1t, 0.9t]$ , defina  $t \leftarrow t_{\text{nov}}$  e repita o teste (1.4).

Observe que, quando  $\Omega$    convexo, o subproblema (1.3)   equivalente ao seguinte problema convexo:

$$\begin{aligned} &\text{Minimizar } w \\ &\nabla f_j(x_k)^T d + \frac{1}{2} d^T B_{k,j} d \leq w, \quad \forall j \in I_\varepsilon(x_k), \\ &x_k + d \in \Omega, \quad \|d\|_\infty \leq \Delta. \end{aligned}$$

Como estamos supondo que as matrizes  $B_{k,j}$  s o semidefinidas positivas (n o necessariamente definidas positivas), o controle do passo  $\|d\|_\infty \leq \Delta$    necess rio para garantir que o subproblema tenha solu  o. Claramente, se  $\Omega$    limitado esta restri  o n o   necess ria, j  que a restri  o  $x_k + d \in \Omega$  garante  $\|d\|_\infty \leq \Delta$ , para  $\Delta$  suficientemente grande.

Abaixo segue o teorema de converg ncia global para o Algoritmo 1.2, provado em [9], enunciando resultados an logos aos apresentados na se  o anterior.

**Teorema 1.3.** *Suponha que a Hip tese **A0**   satisfeita. Seja  $x_k \in \Omega$  a  $k$ - sima itera  o do Algoritmo 1.2. Ent o:*

1. *O algoritmo para em  $x_k$  se, e somente se,  $x_k$     $\varepsilon$ - timo. Se o algoritmo n o para em  $x_k$ , ent o a  $k$ - sima itera  o est  bem definida e termina no Passo 2 com o c culo de  $x_{k+1}$ .*

2. *Suponha que  $x_* \in \Omega$  é um ponto limite de uma sequência gerada pelo Algoritmo 1.2. Então  $x_*$  é  $\varepsilon$ -ótimo.*

Discutiremos, a seguir, os resultados de convergência local do Algoritmo 1.2. Sob algumas hipóteses apropriadas e restringindo-nos aos casos em que  $\Omega$  é descrito por um conjunto de restrições lineares, provamos em [9], que a sequência gerada pelo Algoritmo 1.2 converge superlinearmente (ou quadraticamente) para um ponto limite  $x_*$ . A técnica utilizada em [9] para obter os resultados de convergência local consiste em reduzir, ao máximo, o problema OVO à minimização de uma única função diferenciável e imitar os resultados provados em [3], para um método de métrica variável com restrições lineares.

A hipótese **A1** diz que  $\{x_k\}$  é uma sequência infinita gerada pelo Algoritmo 1.2,  $x_*$  é um ponto limite isolado, em dois sentidos. Por um lado, é um único ponto estacionário em uma bola  $\mathcal{B}(x_*, \delta)$ . Por outro lado, para todos os pontos desta bola, o conjunto  $I_\varepsilon(x)$  contém um único índice.

**Hipótese A1.** *Existe um ponto  $x_* \in \Omega$ , uma sequência  $\{x_k\}$  gerada pelo Algoritmo 1.2 e um conjunto infinito de índices  $K \subset \mathbb{N}$  tal que*

$$\lim_{k \in K} x_k = x_*.$$

*Além disso, existem  $\varepsilon > 0$ ,  $\delta > 0$ ,  $j_0 \in \{1, \dots, m\}$  tais que  $x_*$  é o único ponto  $\varepsilon$ -ótimo do problema (1) que pertence a  $\mathcal{B}(x_*, \delta)$  e, para todo  $x \in \Omega \cap \mathcal{B}(x_*, \delta)$ ,*

$$I_\varepsilon(x) = \{j_0\}.$$

A segunda hipótese diz que as matrizes  $B_{k,j}$  são definidas positivas (não apenas semidefinidas positivas como foi suposto anteriormente) e que as normas de  $\|B_{k,j}\|$  e  $\|B_{k,j}^{-1}\|$  são limitadas.

**Hipótese A2.** *Para todo  $k = 0, 1, 2, \dots$ ,  $j = 1, \dots, m$ , a matriz  $B_{k,j}$  é definida positiva,  $\|B_{k,j}\| \leq c_B$  e  $\|B_{k,j}^{-1}\| \leq c_B$ .*

**Hipótese A3.** *O conjunto viável  $\Omega$  é dado por:*

$$\Omega = \{x \in \mathbb{R}^n \mid Ax \leq b\}.$$

A matriz  $A$  tem  $q$  linhas e  $n$  colunas. Sem perda de generalidade, vamos supor que as restrições ativas em  $x_*$  são aquelas correspondentes às primeiras  $s$  linhas de  $A$ .

De agora em diante, definimos:

$$A = (a_1, \dots, a_q)^T, \quad b = (b_1, \dots, b_q)^T,$$

$$\bar{A} = (a_1, \dots, a_s)^T, \quad \bar{b} = (b_1, \dots, b_s)^T.$$

As próximas duas hipóteses dizem, respectivamente, que a condição de complementaridade estrita é satisfeita e que os gradientes das restrições ativas são linearmente independentes em  $x_*$ .

**Hipótese A4.** Os multiplicadores na solução são tais que

$$[\bar{\lambda}_*]_i > 0, \quad \forall i = 1, \dots, s.$$

**Hipótese A5.** Os vetores  $a_1, \dots, a_s$  são linearmente independentes. (Em particular,  $s \leq n$ .)

A hipótese seguinte refere-se à implementação do algoritmo. Esta diz que escolhemos  $d_k = \bar{d}_k$  como sendo a solução do subproblema, pelo menos para  $k$  suficientemente grande.

**Hipótese A6.** Para todo  $k$  suficientemente grande,  $d_k = \bar{d}_k$ .

A hipótese **A7** impõe condições adicionais sobre a diferenciabilidade da função  $f_{j_0}$ . Vamos supor que  $f_{j_0}$  possui derivadas de segunda ordem e que estas derivadas são Lipschitz-contínuas em uma vizinhança de  $x_*$ .

**Hipótese A7.** A Hessiana  $\nabla^2 f_{j_0}(x)$  existe e é contínua para todo  $x$  em uma vizinhança de  $x_*$ . Além disso, existe  $L_2 > 0$  tal que, para todo  $x, y$  nesta vizinhança,

$$\|\nabla^2 f_{j_0}(y) - \nabla^2 f_{j_0}(x)\| \leq L_2 \|y - x\|.$$

A hipótese **A8** é uma condição Dennis-Moré [19]. Esta hipótese exprime os requerimentos mínimos para provar que a convergência é superlinear.

**Hipótese A8.** *Existe  $Z \in \mathbb{R}^{n \times (n-s)}$  tal que  $\mathcal{R}(Z) = \mathcal{N}(\bar{A})$  e*

$$\lim_{k \rightarrow \infty} \frac{\|Z^T [B_{k,j_0} - \nabla^2 f_{j_0}(x_k)] \bar{d}_k\|}{\|\bar{d}_k\|} = 0.$$

A hipótese **A9** especifica uma condição suficiente para que  $x_*$  seja um minimizador local de  $f_{j_0}$  sujeito às restrições ativas.

**Hipótese A9.** *A Hessiana  $\nabla^2 f_{j_0}(x_*)$  é definida positiva no  $\mathcal{N}(\bar{A})$ . Em outras palavras,  $Z^T \nabla^2 f_{j_0}(x_*) Z$  é definida positiva se  $Z$  é tal como em **A8**.*

O teorema final desta seção diz que, se as hipóteses **A0–A9** são satisfeitas, a convergência é superlinear. A convergência é quadrática se as matrizes  $B_{k,j_0}$  são as verdadeiras hessianas.

**Teorema 1.4.** *Suponha que as hipóteses **A0 – A9** são satisfeitas. Então, para  $k$  suficientemente grande, temos  $t_k = 1$  e  $\bar{A}x_k = \bar{b}$ . Além disso,  $\{x_k\}$  converge superlinearmente para  $x_*$ . Finalmente, se, para  $k$  suficientemente grande,*

$$B_{k,j_0} = \nabla^2 f_{j_0}(x_k),$$

*a convergência é quadrática.*

## 1.4 Reformulação Diferenciável do OVO

O problema OVO é um problema de otimização não-convexa e não-diferenciável. Em [5], os autores transformaram o problema OVO em um problema de programação não-linear diferenciável cujas restrições são conhecidas como restrições de equilíbrio. Ver [6].

O teorema a seguir, provado em [5], mostra que resolver (1) é equivalente a resolver um problema de programação não-linear diferenciável.

**Teorema 1.5.** *O ponto  $x \in \Omega$  é uma solução do problema OVO se, e somente se, existem  $r', u', w' \in \mathbb{R}^m$  e  $z' \in \mathbb{R}$  tais que  $(x, r', u', w', z')$  é uma solução de:*

$$\begin{aligned} & \text{Minimizar } z \\ & \text{s. a. } \begin{cases} \sum_{i=1}^m r_i w_i = 0 \\ \sum_{i=1}^m (1 - r_i) u_i = 0 \\ \sum_{i=1}^m r_i = p \\ u_i - z + f_i(x) - w_i = 0, i = 1, \dots, m \\ u \geq 0, 0 \leq r \leq 1, w \geq 0, x \in \Omega. \end{cases} \end{aligned} \quad (1.5)$$

É fácil verificar que, no caso  $p = m$ , que corresponde ao problema Minimax, a reformulação (1.5) reduz-se à clássica reformulação (de programação não-linear) do problema Minimax:

$$\text{Minimizar } z \text{ s.a. } z \geq f_i(x), i = 1, \dots, m, x \in \Omega.$$

Se  $(x, r, u, w, z)$  é um ponto viável da reformulação então  $z \geq f(x)$ . A possibilidade de  $z > f(x)$  para pontos viáveis de (1.5) não está descartada. Entretanto, para minimizadores locais de (1.5), a identidade  $z = f(x)$  necessariamente se verifica. As demonstrações destas propriedades podem ser encontradas em [5].

O seguinte teorema, demonstrado em [5], mostra a relação entre os minimizadores locais do problema OVO e os de sua reformulação. Essencialmente, um minimizador local de (1) gera um minimizador local de (1.5). A recíproca é verdadeira, mas, necessita de uma indispensável hipótese adicional.

**Teorema 1.6.** *Suponha que  $x \in \Omega$  é um minimizador local de (1) e que  $(x, r, u, w, f(x))$  é um ponto viável de (1.5). Então,  $(x, r, u, w, f(x))$  é um minimizador local de (1.5). Reciprocamente, se, para algum  $z \in \mathbb{R}$ , temos que  $(x, r, u, w, z)$  é um minimizador local de (1.5) sempre que  $(x, r, u, w, z)$  seja um ponto viável, então  $z = f(x)$  e  $x$  é um minimizador local de (1).*

A reformulação (1.5) é um problema de minimização diferenciável com restrições de equilíbrio. Isto inclui as restrições de complementaridade:

$$r_i w_i = 0, r_i \geq 0, w_i \geq 0$$

e

$$(1 - r_i) u_i = 0, r_i \leq 1, u_i \geq 0.$$

Restrições de complementaridade são responsáveis pelo fato de nenhum ponto viável satisfazer à condição de qualificação de Mangassarian-Fromovitz [12, 32]. Portanto, minimizadores do problema podem não satisfazer as condições de KKT e isto pode representar uma dificuldade para os algoritmos de programação não-linear. Um resultado muito interessante, provado em [5], é que, pelo menos quando  $\Omega = \mathbb{R}^n$ , este possível obstáculo não existe. Mais precisamente, minimizadores locais de (1) geram pontos KKT de (1.5) mesmo com a falta de regularidade dos pontos.

Uma hipótese essencial para mostrar este resultado é que  $x_*$  deve ser um ponto estacionário de primeira ordem do OVO. Esta é uma condição de otimalidade, apresentada em [5], mais forte que a descrita na Seção 1.1. Antes de enunciá-la, para cada  $x \in \Omega$ , defina:

$$L(x) = \{i \in \{1, \dots, m\} \mid f_i(x) < f(x)\},$$

$$E(x) = \{i \in \{1, \dots, m\} \mid f_i(x) = f(x)\},$$

e

$$G(x) = \{i \in \{1, \dots, m\} \mid f_i(x) > f(x)\}.$$

Claramente, os conjuntos  $L(x)$ ,  $E(x)$  e  $G(x)$ , tais como a função OVO  $f(x)$ , dependem da escolha de  $p$ . Entretanto, não vamos deixar esta dependência explícita com o intuito de simplificar a notação. O teorema a seguir define a condição de otimalidade relacionada com as direções tangentes.

**Teorema 1.7.** *Suponha que  $x$  é um minimizador local de (1) e que  $f_i$  possui primeira derivada contínua em uma vizinhança de  $x$  para todo  $i \in E(x)$ . Então, para todas as direções tangentes  $d$  unitárias,*

$$\#\{i \in E(x) \mid \nabla f_i(x)^T d < 0\} < p - \#L(x).$$

O Teorema 1.7 justifica a definição dos pontos estacionários de primeira ordem.

### Pontos estacionários de primeira-ordem

*Suponha que todas as funções  $f_i$  que definem o problema OVO possuem primeiras derivadas contínuas em um conjunto aberto que contém  $\Omega$ . Dizemos que  $x \in \Omega$  é um ponto estacionário de primeira ordem do problema OVO se, para todas as direções tangentes  $d$  unitárias,*

$$\#\{i \in E(x) \mid \nabla f_i(x)^T d < 0\} < p - \#L(x).$$

Finalmente, o teorema a seguir formaliza o resultado.

**Teorema 1.8.** *Suponha que  $\Omega = \mathbb{R}^n$  e seja  $x_* \in \Omega$  um ponto estacionário de primeira ordem do problema OVO. Sejam  $r, u, w \in \mathbb{R}^m$  tais que  $(x_*, r, u, w, z_*)$  seja um ponto viável de (1.5) com  $z_* = f(x_*)$ . Então,  $(x_*, r, u, w, z_*)$  é um ponto KKT de (1.5).*

A robustez e a eficiência de muitos algoritmos de programação não-linear estão fortemente relacionadas com o cumprimento das condições de KKT na solução. Portanto, o fato dos minimizadores serem pontos KKT pode ser considerado como um forte argumento para tentar resolver o problema OVO com métodos de otimização tradicionais. Porém, esta abordagem não foi muito explorada neste trabalho. Focamos nossa atenção nos métodos primais que têm a vantagem de trabalhar com uma quantidade bem menor de variáveis.

## Capítulo 2

# Encontrando Padrões Ocultos com OVO

A procura de padrões ocultos é uma das mais desafiantes pesquisas existentes em modernos “data mining”. Vários artigos descrevem problemas de encontrar padrões ocultos nas mais diferentes áreas, como Ecologia [23, 33], Administração da Saúde Pública [31], Arte [47], Psiquiatria [25], História Social [46], Demografia [15, 16] e muitos outros.

Considere a nuvem de pontos dada na Figura 2.1. À primeira vista, as duas nuvens de pontos são qualitativamente similares. Entretanto, na Figura 2.1-A existem exatamente cinco pontos que pertencem a uma mesma parábola ao passo que, na Figura 2.1-B, tal conjunto de pontos não existe.

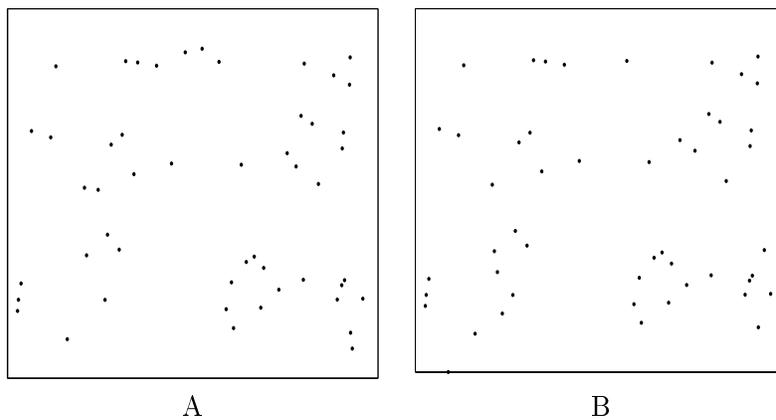


Figura 2.1: Padrões Ocultos

Na Figura 2.2 mostramos, novamente, os pontos da Figura 2.1-A juntamente com a parábola que passa por estes pontos especiais. Em situações como esta, dizemos que a Figura 2.1-A esconde o padrão de uma parábola (ou, simplesmente, esconde uma parábola).

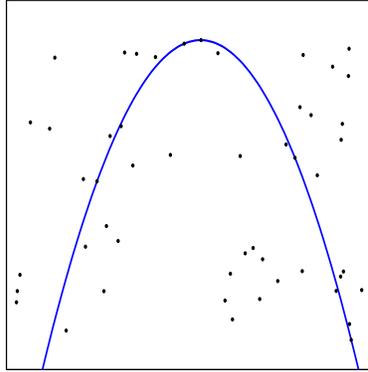


Figura 2.2: Parábola Oculta

Na próxima seção mostraremos como o problema OVO pode ser utilizado para encontrar padrões ocultos deste tipo. Muitos dos exemplos a serem apresentados nas próximas seções são suficientemente simples e pequenos para serem resolvidos de forma exaustiva por métodos enumerativos. Entretanto, se a quantidade de dados ou o número de parâmetros desconhecidos é incrementada, a utilização de procedimentos combinatórios torna-se inviável. Esta é razão pela qual não consideraremos métodos enumerativos em nossas análises.

Devemos mencionar que em muitos artigos de “data-mining” e aplicações, padrões são ocultos no sentido de que estes são difíceis de serem encontrados, ao menos, utilizando procedimentos de ajuste padrões. Em nosso caso, padrões estão ocultos porque estes são revelados tomando em conta somente uma pequena quantidade de dados, devido ao fato de muitas das informações avaliadas estarem severamente corrompidas.

## 2.1 Metodologia

Suponha que  $\{(t_1, y_1), \dots, (t_m, y_m)\} \subset \mathbb{R}^2$  é um conjunto de dados e que sabemos que estes dados contêm muito ruído. Nesta aplicação, um padrão é definido como sendo um subconjunto  $S(x) \subset \mathbb{R}^2$ , determinado em função de um parâmetro  $x$ . Para cada  $i =$

$1, \dots, m$ , defina  $f_i(x)$  como sendo uma “pseudo-distância” entre  $(t_i, y_i)$  e  $S(x)$ . Suponha que esta “pseudo-distância”, para cada par  $(t_i, y_i)$ , possa ser descrita da seguinte forma:

$$f_i(x) = (T(x, t_i) - y_i)^2. \quad (2.1)$$

Se, por exemplo,  $T(x, t) = x_1 t^2 + x_2 t + x_3$  representa uma parábola com coeficientes  $x = (x_1, x_2, x_3)^T$ , então, o padrão  $S(x)$  é formado pelo conjunto dos pontos que passam por esta parábola e (2.1) representa a função de erro quadrático da observação  $i$  se  $x$  for o parâmetro adotado. Nestes casos, na presença de dados com muito ruído, o ajuste de quadrados mínimos da forma  $y_i \approx x_1 t^2 + x_2 t + x_3$  conduz a resultados desastrosos devido à influência opressiva de “outliers”.

Dado  $p \in \{1, \dots, m\}$ , o conjunto de funções (2.1) define um problema OVO para o qual os Algoritmos 1.1 e 1.2 podem ser aplicados. A idéia é resolver este problema para diferentes valores de  $p$ . Se  $p$  é um valor próximo de  $m$  a expectativa é que o valor da função OVO na solução encontrada seja alto, mostrando que existe uma grande quantidade de dados incorretos, fato este que pode ser reforçado ao observar também os valores de  $f_{i_1}, \dots, f_{i_p}$  que representam a “pseudo-distância” dos  $p$  pontos mais próximos ao padrão  $S(x)$ . Somente um pequeno número de pontos pertencem ao padrão oculto correto  $S(x_*)$ , portanto, os parâmetros  $x_*$  são obtidos resolvendo o problema OVO para pequenos valores de  $p$ . Então, quando diminuimos o valor de  $p$ , a função OVO na solução tende a decrescer também. A expectativa é que quando tomamos o valor “correto de  $p$ ”, a função OVO decresça abruptamente, tomando um valor próximo de zero.

## 2.2 Experimentos Numéricos

Os métodos propostos no capítulo anterior, para resolver o problema OVO, são métodos locais, isto é, temos apenas a garantia de convergência para pontos que são  $\varepsilon$ -ótimos. Há, portanto, a necessidade de complementar o método local com uma estratégia de minimização global de forma a obtermos uma maior eficiência. Em [10], inserimos o método local, proposto na Seção 1.2, em uma heurística global onde minimizações locais podem ser melhoradas por uma estratégia “tunneling” baseada no problema de valor inicial que define um oscilador harmônico. Além disso, está incorporada a esta heurística um procedimento “multistart” para gerar diferentes pontos iniciais seguindo alguns critérios para descartar pontos iniciais “pobres”. O mais importante aqui é ter em mente que esta

estratégia global realiza inúmeras minimizações locais com o Algoritmo 1.1. Como o objetivo deste trabalho é mostrar como o OVO pode ser utilizado para encontrar padrões ocultos, não nos aprofundaremos neste assunto e vamos focar mais na implementação do Algoritmo 1.1. Para maiores detalhes da estratégia global, ver [10, 42].

Algumas características relativas à implementação do Algoritmo 1.1 estão descritas abaixo:

- Como mencionado anteriormente, o problema (1.1) é um problema de otimização convexa. Além disso, em nossas aplicações as restrições que definem  $\Omega$  são lineares, portanto (1.1) é um problema de programação linear. Para resolvê-lo usamos a rotina DDLPRS da biblioteca IMSL.
- Os subproblemas são resolvidos exatamente. Isto significa que usamos  $\eta = 0$ .
- Os parâmetros algorítmicos utilizados foram:

$$\theta = 0.5, \Delta = 1, \varepsilon = 10^{-3}, \sigma_{\min} = 0.1, \sigma_{\max} = 0.9.$$

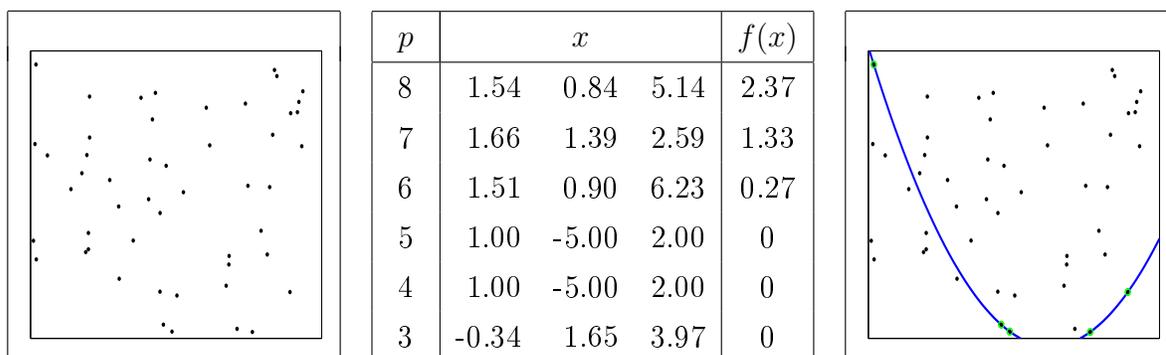
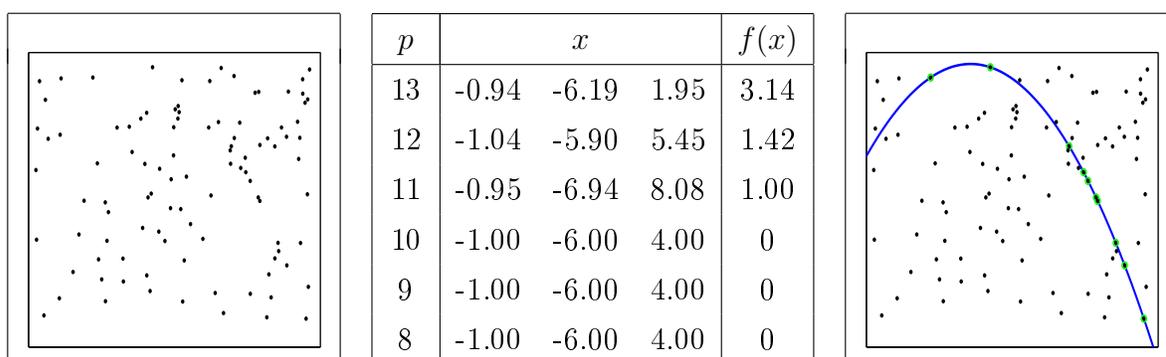
- No procedimento “backtracking” (1.2) tomamos  $t_{new} = 0.5t$ .
- Todos os experimentos deste capítulo foram realizados em um Pentium 4, 2.4 Ghz, 1Gb RAM em FORTRAN de dupla precisão.

### 2.2.1 Encontrando Polinômios Ocultos

Nos experimentos a seguir, o padrão oculto é um polinômio de grau  $n - 1$ . Em todos os casos geramos  $m$  dados de forma aleatória. Dez por cento dos dados são “corretos” no sentido de que eles se ajustam perfeitamente a um polinômio com coeficiente  $x_*$  previamente escolhido. Definimos  $\Omega = [-10, 10]^n$ . Para cada  $i = 1, \dots, m$ , definimos a função de erro da seguinte forma:

$$f_i(x) = \left( \sum_{j=1}^n x_j t_i^{n-j} - y_i \right)^2.$$

Os resultados obtidos ao aplicar o Algoritmo 1.1 (inserido na heurística global) para encontrar polinômios ocultos estão reportados nas Tabelas 2.1-2.6. As figuras à esquerda representam os dados; a primeira coluna da tabela reporta os diferentes valores de  $p$  testados, a partir da segunda coluna reportamos a melhor solução encontrada pelo

Tabela 2.1:  $m = 50$ , “ $p$  correto” = 5,  $x_* = (1, -5, 2)$ Tabela 2.2:  $m = 100$ , “ $p$  correto” = 10,  $x_* = (-1, -6, 4)$ 

algoritmo e o valor da função OVO neste ponto; a figura à direita revela o polinômio oculto e os pontos que fazem parte deste padrão (pontos circulosados em verde).

### 2.2.2 Encontrando Circunferências Ocultas

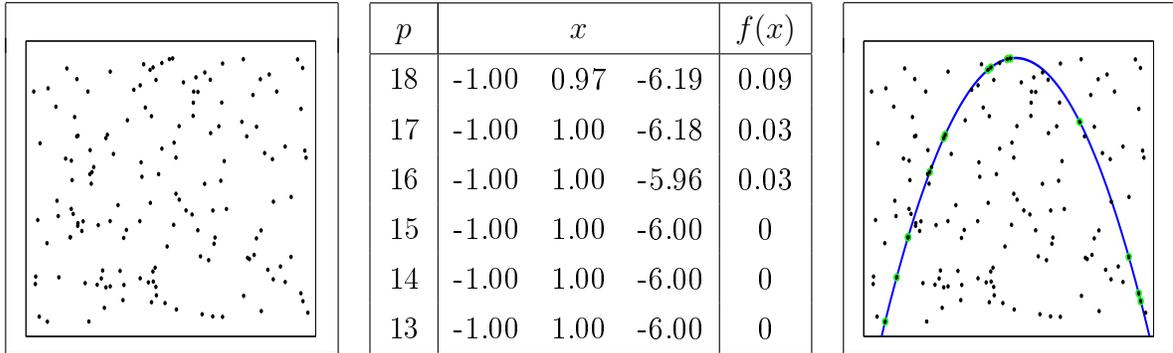
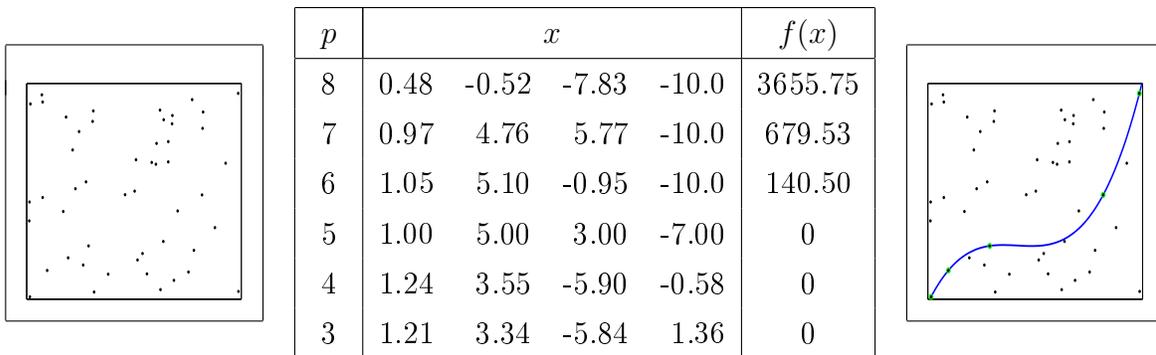
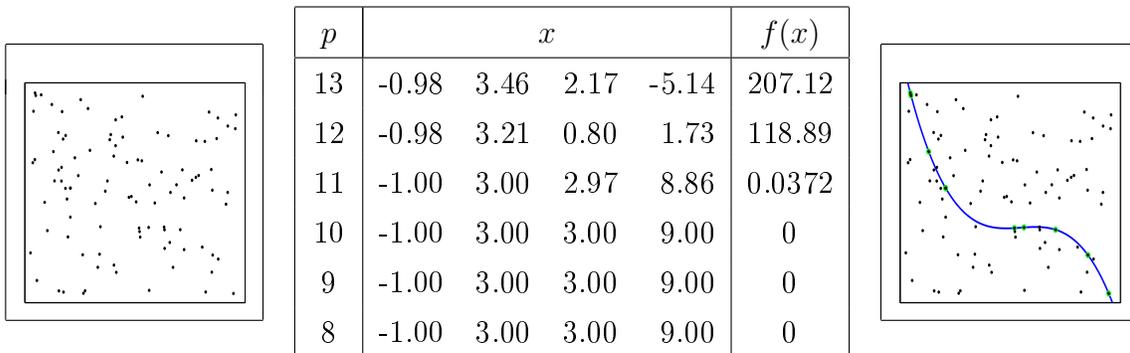
Nos experimentos desta seção, o padrão oculto é uma circunferência. Neste caso, precisamos estimar três parâmetros  $x_1, x_2, x_3 \in \Omega$  onde

$$\Omega = \{x \in \mathbb{R}^3 \mid -10 \leq x_1, x_2 \leq 10, 0 \leq x_3 \leq 10\}.$$

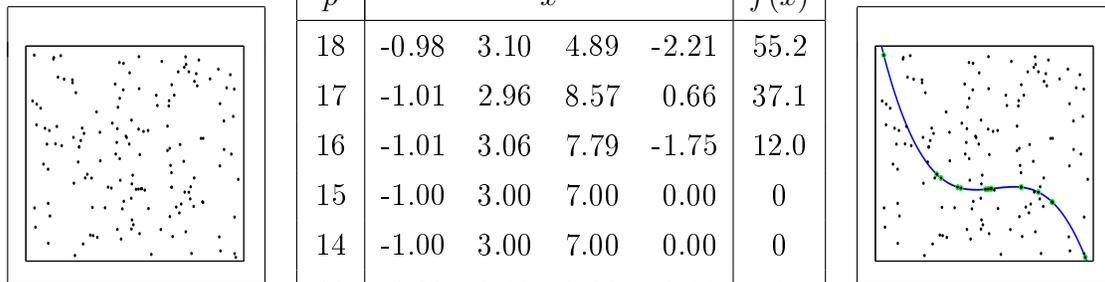
O centro da circunferência oculta é  $(x_1, x_2)$  e o raio é  $x_3$ . As funções  $f_i$  são:

$$f_i(x) = [(t_i - x_1)^2 + (y_i - x_2)^2 - x_3^2]^2.$$

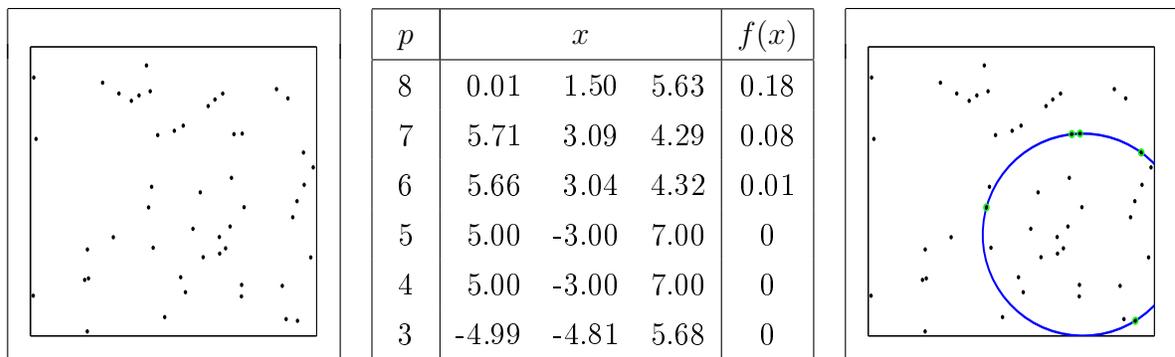
Novamente, geramos  $m$  dados de forma aleatória. Dez por cento dos dados são “corretos” no sentido de que eles pertencem a uma circunferência de centro  $(x_1^*, x_2^*)$  e raio  $x_3^*$  previamente escolhidos. Os resultados, seguindo as mesmas convenções anteriores, estão reportados nas Tabelas 2.7-2.9.

Tabela 2.3:  $m = 150$ , “ $p$  correto” = 15,  $x_* = (-1, 1, -6)$ Tabela 2.4:  $m = 50$ , “ $p$  correto” = 5,  $x_* = (1, 5, 3, -7)$ Tabela 2.5:  $m = 100$ , “ $p$  correto” = 10,  $x_* = (-1, 3, 3, 9)$

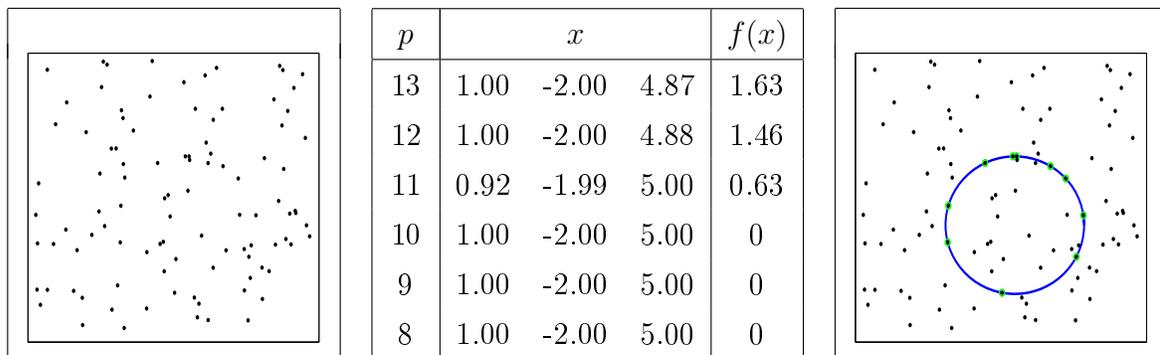
$p$	$x$				$f(x)$
18	-0.98	3.10	4.89	-2.21	55.2
17	-1.01	2.96	8.57	0.66	37.1
16	-1.01	3.06	7.79	-1.75	12.0
15	-1.00	3.00	7.00	0.00	0
14	-1.00	3.00	7.00	0.00	0
13	-1.00	3.00	7.00	0.00	0

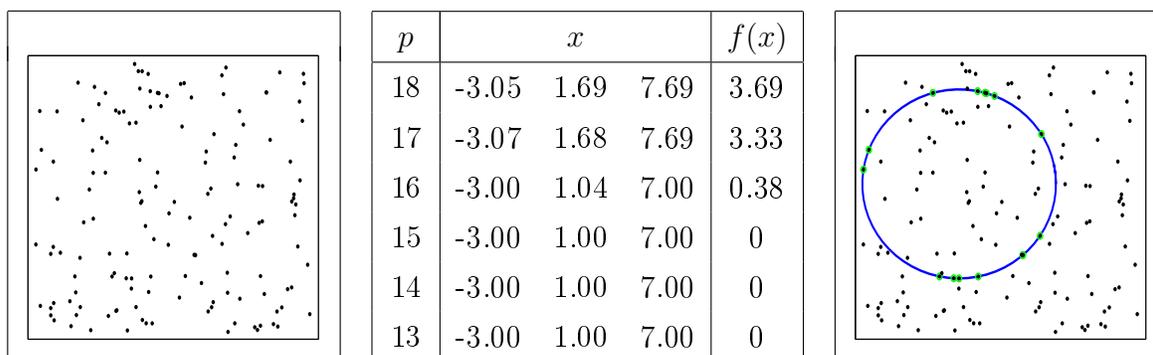
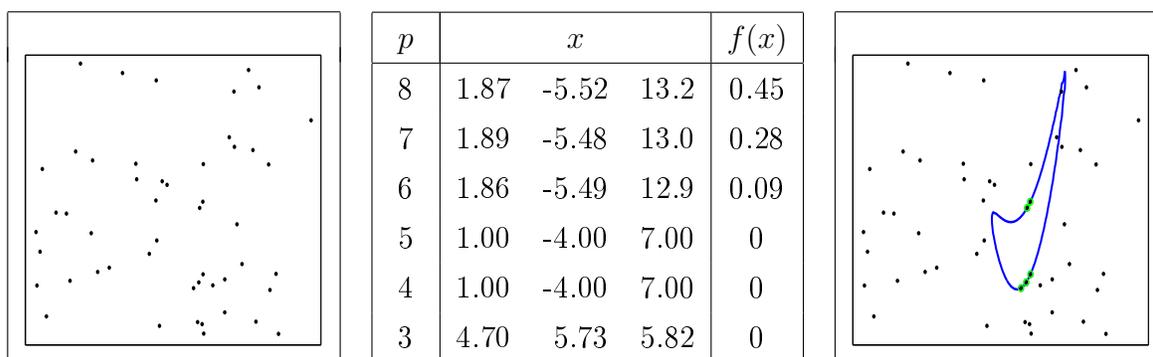

Tabela 2.6:  $m = 150$ , “ $p$  correto” = 15,  $x_* = (-1, 3, 7, 0)$ 

$p$	$x$			$f(x)$
8	0.01	1.50	5.63	0.18
7	5.71	3.09	4.29	0.08
6	5.66	3.04	4.32	0.01
5	5.00	-3.00	7.00	0
4	5.00	-3.00	7.00	0
3	-4.99	-4.81	5.68	0


Tabela 2.7:  $m = 50$ , “ $p$  correto” = 5,  $x^* = (5, -3, 7)$ 

$p$	$x$			$f(x)$
13	1.00	-2.00	4.87	1.63
12	1.00	-2.00	4.88	1.46
11	0.92	-1.99	5.00	0.63
10	1.00	-2.00	5.00	0
9	1.00	-2.00	5.00	0
8	1.00	-2.00	5.00	0


Tabela 2.8:  $m = 100$ , “ $p$  correto” = 10,  $x^* = (1, -2, 5)$

Tabela 2.9:  $m = 150$ , “ $p$  correto” = 15,  $x^* = (-3, 1, 7)$ Tabela 2.10:  $m = 50$ , “ $p$  correto” = 5,  $x^* = (1, -4, 7)$ 

### 2.2.3 Encontrando “Bananas” Ocultas

Nos experimentos desta seção, o padrão oculto é uma curva no plano- $ty$ , da forma:

$$(y - x_2 - (t - x_1)^2)^2 + (1 - (t - x_1))^2 = x_3,$$

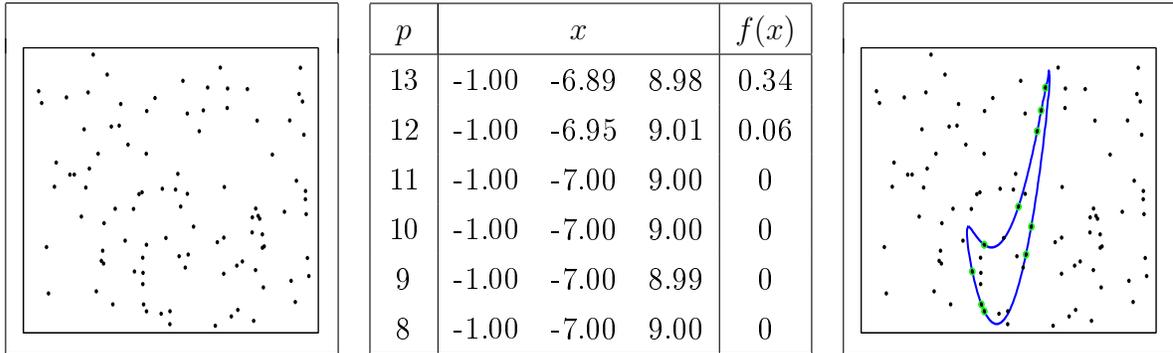
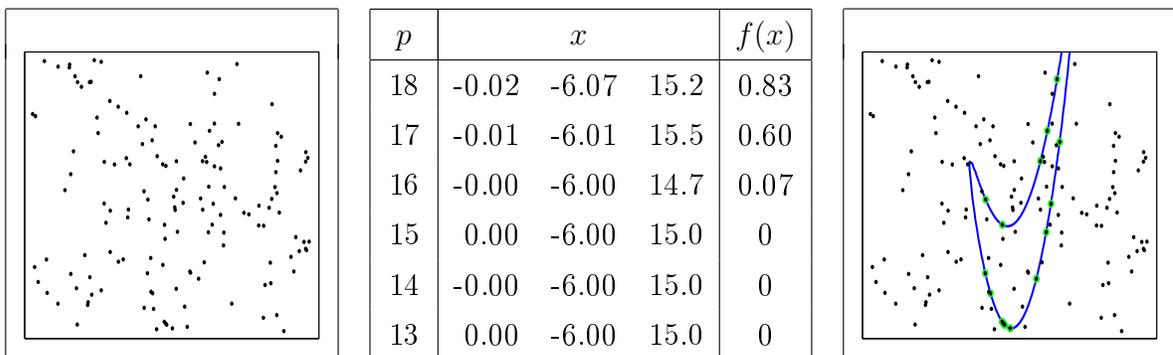
onde  $x_1, x_2$  e  $x_3$  são os parâmetros que precisamos estimar. Definimos:

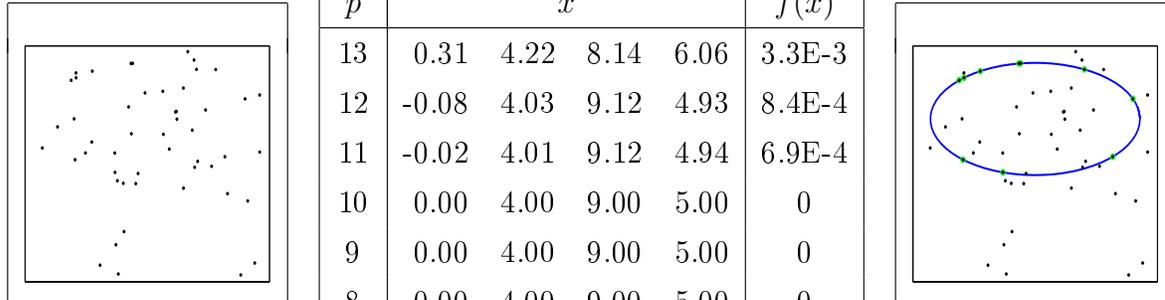
$$\Omega = \{x \in \mathbb{R}^3 \mid -10 \leq x_1, x_2 \leq 10, 0 \leq x_3 \leq 20\}.$$

E, para cada  $i = 1, \dots, m$  a função de erro é dada por:

$$f_i(x) = [(y_i - x_2 - (t_i - x_1)^2)^2 + (1 - (t_i - x_1))^2 - x_3]^2.$$

Novamente, geramos  $m$  dados de forma aleatória. Dez por cento dos dados são “corretos” no sentido de que eles pertencem a uma curva pré-definida pelos parâmetros  $x_*$ . Os resultados, seguindo as mesmas convenções anteriores, estão reportados nas Tabelas 2.10-2.12.

Tabela 2.11:  $m = 100$ , “ $p$  correto” = 10,  $x^* = (-1, -7, 9)$ Tabela 2.12:  $m = 150$ , “ $p$  correto” = 15,  $x^* = (0, -6, 15)$



$p$	$x$				$f(x)$
13	0.31	4.22	8.14	6.06	3.3E-3
12	-0.08	4.03	9.12	4.93	8.4E-4
11	-0.02	4.01	9.12	4.94	6.9E-4
10	0.00	4.00	9.00	5.00	0
9	0.00	4.00	9.00	5.00	0
8	0.00	4.00	9.00	5.00	0

Tabela 2.13:  $m = 50$ , “ $p$  correto” = 10,  $x^* = (0, 4, 9, 5)$ 

### 2.2.4 Encontrando Elipses Ocultas

Nos experimentos desta seção, o padrão oculto é uma elipse:

$$\frac{(t_i - x_1)^2}{x_3^2} + \frac{(y_i - x_2)^2}{x_4^2} = 1.$$

Neste caso, precisamos estimar quatro parâmetros  $x_1, x_2, x_3$  e  $x_4 \in \Omega$  onde

$$\Omega = \{x \in \mathbb{R}^4 \mid -10 \leq x_1, x_2 \leq 10, 0 \leq x_3, x_4 \leq 10\}.$$

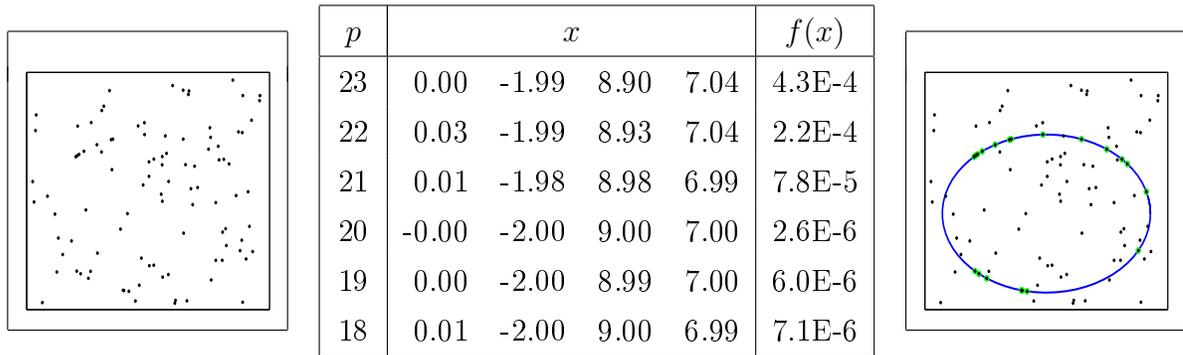
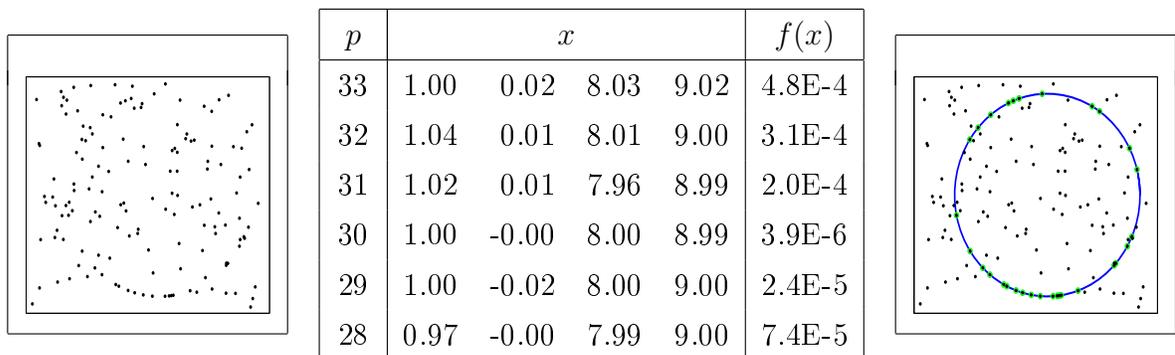
Portanto, as funções de erros são:

$$f_i(x) = \left[ \frac{(t_i - x_1)^2}{x_3^2} + \frac{(y_i - x_2)^2}{x_4^2} - 1 \right]^2.$$

Novamente, geramos  $m$  dados de forma aleatória. Vinte por cento dos dados são “corretos” no sentido de que eles pertencem a uma elipse pré-definida pelos parâmetros  $x_*$ . Os resultados, seguindo as mesmas convenções anteriores, estão reportados nas Tabelas 2.13-2.15.

## 2.3 Análise dos Resultados

Analisando as tabelas e ilustrações correspondentes, apresentadas na seção anterior, podemos concluir que os resultados foram coerentes às expectativas descritas na Seção 2.1. Para os maiores valores de  $p$  testados, obtivemos um valor não-nulo significativo da função OVO. À medida que diminuimos o valor de  $p$ , o valor da função OVO também diminuiu, porém, quando tomamos o valor “correto de  $p$ ”, o valor da função OVO decresceu abruptamente, atingindo o valor zero. Quando este fato não ocorrer,

Tabela 2.14:  $m = 100$ , “ $p$  correto” = 20,  $x^* = (0, -2, 9, 7)$ Tabela 2.15:  $m = 150$ , “ $p$  correto” = 30,  $x_* = (1, 0, 8, 9)$

como foi no caso da elipse, podemos concluir que, apesar de termos valor “correto de  $p$ ” pontos que se ajustam perfeitamente ao padrão (da elipse) existem outros que, apesar de não se ajustarem perfeitamente, estão muito próximos deste padrão, lembrando que esta proximidade é medida pelas funções de “pseudo-distância”  $f_i$ . Mas, mesmo nestes casos, pelo fato dos dados estarem severamente corrompidos, o “salto” da função OVO certamente seria observado, porém, para valores de  $p$  um pouco maiores que o valor “correto de  $p$ ”.

Apesar de termos apresentado duas opções (Algoritmos 1.1 e 1.2) para resolver o problema OVO, utilizamos o Algoritmo 1.1 por ser a única opção no momento em produzimos [10]. Como o intuito deste capítulo é mostrar como o problema OVO pode ser utilizado para encontrar padrões ocultos, não reproduzimos os experimentos com o Algoritmo 1.2. Uma comparação entre os métodos será feita no próximo capítulo.

## Capítulo 3

# Otimizando o Valor em Risco VaR com OVO

Neste capítulo, faremos uma breve introdução do Valor em Risco VaR e sua relação com o problema OVO. Em seguida, apresentaremos os resultados numéricos obtidos ao aplicar a estratégia OVO para maximizar o VaR do retorno (lucro) de uma instituição financeira baseado em uma versão, levemente modificada, do modelo do sistema bancário ideal de Stiglitz [45].

### 3.1 Valor em Risco VaR

O Valor em Risco VaR e o Valor em Risco Condicional CVaR são duas importantes medidas de risco que, nos últimos anos, têm sido amplamente utilizadas para seleção de portfólios e análise de risco. O VaR é de longe a medida de risco mais popular e aceita entre as instituições financeiras. Dado um certo nível de confiança, o VaR é uma estimativa da máxima perda que um tomador de decisões deve esperar durante um período padrão (por exemplo: dia, semana ou ano). Em outras palavras, com certa probabilidade, as perdas não excederão VaR. Ao considerar cenários, o VaR é uma função, com respeito ao vetor das posições, não-diferenciável, não-convexa e com vários minimizadores locais, dificultando seu controle e otimização. Descrições de várias metodologias para modelar o VaR podem ser encontradas no site [www.gloriamundi.org](http://www.gloriamundi.org). O CVaR quantifica as perdas que excedem o VaR e age como um limitante superior para o VaR. Consequentemente, o CVaR é extremamente sensível à presença de cenários

catastróficos. A vantagem é que, em determinadas condições, o CVaR é uma função convexa com respeito ao vetor das posições e pode ser minimizada utilizando técnicas de programação linear [40, 41].

Por definição, com respeito a um especificado nível de confiança  $\beta$ , o  $\beta$ -VaR de um portfólio é a menor quantidade  $\alpha$  tal que, com probabilidade  $\beta$ , a perda não exceda  $\alpha$ , ao passo que, o  $\beta$ -CVaR é a expectância condicional das perdas acima da quantidade  $\alpha$ . Três valores de  $\beta$  são comumente considerados: 0.90, 0.95 e 0.99. Por definição, o  $\beta$ -VaR nunca será menor que o  $\beta$ -CVaR.

Seja  $f(x, y)$  a função de perda associada a um vetor de posições  $x$ , a ser escolhida dentro de um determinado subconjunto  $\Omega$  de  $\mathbb{R}^n$ . O vetor  $x$  pode ser interpretado como sendo a representação de um portfólio e  $\Omega$  o conjunto de portfólios disponíveis (sujeito a várias restrições), mas outras interpretações podem ser feitas. O vetor  $y$  corresponde às incertezas que podem afetar as perdas. Claramente, a perda pode ser negativa e portanto, com efeito, constitui um ganho.

Para cada  $x$ , a perda  $f(x, y)$  é uma variável aleatória que tem uma distribuição em  $\mathbb{R}$  induzida pela distribuição da variável aleatória  $y$ . Na prática, não é necessário ter uma expressão analítica da função densidade de probabilidade de  $y$ . Em geral, assume-se que as perdas são discretamente distribuídas. Esta distribuição discreta da perda pode ser obtida amostrando  $m$  cenários  $y_i$ ,  $i = 1, \dots, m$  de alguma distribuição contínua subjacente. Os pontos amostrais  $y_i$ ,  $i = 1, \dots, m$  podem coincidir, portanto, a probabilidade de pontos discretos são múltiplos de  $1/m$ . Dada uma amostra de tamanho  $m$ , defina as perdas  $f_i(x) = f(x, y_i)$ ,  $i = 1, \dots, m$  para cada vetor de posições  $x$ . Assim, para cada  $x$  podemos ordenar as perdas de forma que:

$$f_{i_1(x)}(x) \leq f_{i_2(x)}(x) \leq \dots \leq f_{i_m(x)}(x).$$

Em função de  $\alpha$  defina para cada  $x$  fixo,  $\psi(x, \alpha)$  como sendo a probabilidade da perda não exceder  $\alpha$ . Apenas para efeito de exemplificação suponha que:

$$f_{i_1(x)}(x) \leq \dots \leq f_{i_p(x)}(x) < f_{i_{p+1}(x)}(x) \leq \dots \leq f_{i_m(x)}(x).$$

Então,  $\psi(x, f_{i_p})$  é igual a  $p/m$  que, naturalmente, coincide com a probabilidade de ocorrência de um cenário pertencente ao subconjunto  $\{y_{i_1}, \dots, y_{i_p}\}$  dentre o total de  $m$  cenários. E, mesmo que tivermos empate na função de ordem  $p$  podemos observar que a probabilidade  $\psi(x, f_{i_p})$  da perda não exceder  $f_{i_p}$  é maior ou igual a  $p/m$ .

Neste contexto, dado um nível de confiança  $\beta \in (0, 1)$  definimos o  $\beta$ -VaR para a perda associada a uma decisão  $x$  por:

$$\beta\text{-VaR}(x) = \min\{\alpha \in \mathbb{R} \mid \psi(x, \alpha) \geq \beta\}. \quad (3.1)$$

No caso em que  $\beta = p/m$  temos, para cada  $x$  fixo, que:

$$\{\alpha \in \{f_{i_1}, \dots, f_{i_m}\} \mid \psi(x, \alpha) \geq \beta\} = \{f_{i_p}, f_{i_{p+1}}, \dots, f_{i_m}\},$$

já que,  $\psi(x, f_{i_p}) \geq p/m$ ,  $\psi(x, f_{i_{p+1}}) \geq (p+1)/m > p/m$  e  $\psi(x, f_{i_m}) = 1 > p/m$ .

Portanto, segue da definição (3.1), que o  $p/m$ -VaR de um vetor de posições  $x$  é  $f_{i_p(x)}(x)$ , ou seja, a função  $p/m$ -VaR corresponde à função OVO de ordem  $p$ . Então, dados  $m$  cenários  $y_i$ ,  $i = 1, \dots, m$ , definindo as funções de perda  $f_i(x) = f(x, y_i)$ ,  $i = 1, \dots, m$ , o problema de encontrar o portfólio  $x \in \Omega$  que minimiza o  $p/m$ -VaR equivale a resolver o problema OVO de ordem  $p$ .

Para finalizar, vamos dar a “visão OVO” do  $p/m$ -CVaR. Em [40, 41], os autores definiram a função auxiliar

$$\tilde{F}_\beta(x, \alpha) = \alpha + \frac{1}{m-p} \sum_{i=1}^m [f_i(x) - \alpha]_+$$

e provaram que o par  $(x_*, \alpha_*)$  que soluciona o problema

$$\min_{(x, \alpha) \in \Omega \times \mathbb{R}} \tilde{F}_\beta(x, \alpha)$$

é tal que  $x_*$  é o portfólio que minimiza o  $p/m$ -CVaR e  $\alpha_*$  é o  $p/m$ -VaR se tivermos

$$\arg \min_{\alpha \in \mathbb{R}} \tilde{F}_\beta(x_*, \alpha) = \{\alpha_*\}.$$

Caso contrário,  $\alpha_*$  é apenas um candidato, já que,  $p/m$ -VaR é o menor valor de  $\alpha$  pertencente ao conjunto  $\arg \min_{\alpha \in \mathbb{R}} \tilde{F}_\beta(x_*, \alpha)$ , ver [40, 41]. Reciprocamente, sabendo que o  $p/m$ -VaR de  $x$  é  $f_{i_p(x)}(x)$  e, restringindo-nos ao caso em que o único elemento de  $\Omega$  é  $x$ , chegamos à seguinte conclusão:

$$\begin{aligned} p/m\text{-CVaR}(x) &= f_{i_p(x)}(x) + \frac{1}{m-p} \sum_{i=1}^m [f_i(x) - f_{i_p(x)}(x)]_+ \\ &= f_{i_p(x)}(x) + \frac{1}{m-p} \sum_{j=p+1}^m [f_{i_j(x)}(x) - f_{i_p(x)}(x)] \\ &= \frac{1}{m-p} \sum_{j=p+1}^m f_{i_j(x)}(x). \end{aligned}$$

Isto significa que  $p/m$ -CVaR é a média das funções acima (ao considerar os índices) da função OVO de ordem  $p$ . Em outras palavras, o  $p/m$ -CVaR corresponde à função da soma dos  $m - p$  maiores valores ordenados que, fazendo uma analogia ao LOVO, definiria um problema HOVO (High Order Value Optimization). Mais ainda, definiria um problema Minimax.

## 3.2 Aplicação

O objetivo deste capítulo é aplicar o OVO para minimizar o VaR da perda (maximizar o VaR do retorno). Apesar da teoria do VaR estar baseada na perda, a interpretação do VaR do retorno é análoga. Dado um nível de confiança  $\beta$ , o  $\beta$ -VaR do retorno representa o mínimo lucro esperado com este nível de confiança.

Em nossa aplicação, o objetivo é maximizar o VaR do retorno de uma instituição financeira. Todo processo de modelagem está baseado no modelo do sistema bancário ideal (SBI) de [45] (o autor Joseph Stiglitz recebeu o Prêmio Nobel de Economia em 2001 por seu trabalho sobre a análise de mercados com informações assimétricas), então, vamos descrever nesta seção as principais características do modelo.

O modelo do SBI discutido aqui se refere às tomadas de decisões de um banco de depósito e investimento. Ou seja, nosso banco toma depósitos dos clientes, concede crédito e aplica em títulos do governo. O banco deve decidir quanto emprestar, quanto cobrar de juros pelos empréstimos concedidos, quanto investir em títulos do governo e, quanto investir em avaliação dos candidatos a empréstimo.

No (levemente modificado) modelo do SBI de um período, as variáveis de decisão são:

- $M$ : valor a ser investido em títulos do governo;
- $N$ : valor a ser concedido em crédito;
- $r$ : taxa de juros anual a ser cobrada pelo crédito concedido;
- $e$ : valor a ser investido em avaliação dos candidatos ao crédito;
- $D$ : soma dos depósitos realizados efetivamente no banco dentro do período considerado;

- $s$ : taxa de juros a ser paga aos depositantes.

Os dados disponíveis são:

- $a$ : patrimônio líquido inicial;
- $\tau$ : compulsório (porcentagem de depósitos que o governo exige que sejam depositados sem juros no Banco Central);
- $\rho$ : taxa de juros paga pelos títulos públicos.

Existem outros dados que, na realidade, são estimativas:

- $N^d$ : dada cada taxa  $r$ ,  $N^d(r)$  estima a demanda por crédito para esta taxa. Se o banco oferece uma taxa muito baixa a demanda por crédito será maior e vice-versa.
- $D^o$ : dada cada taxa  $s$ ,  $D^o(s)$  estima a quantidade de depósitos que o banco receberia para esta taxa. Naturalmente, se  $s$  é grande,  $D^o(s)$  tende a ser grande também e vice-versa.
- $f$ : dado o valor dos gastos com avaliação  $e$ ,  $f(e)$  é uma função que mede a incerteza na previsão de que o devedor foi bem avaliado. Se  $e$  é grande,  $f(e)$  se aproximará de zero, pois a grandes gastos de avaliação corresponde uma baixa incerteza na avaliação. Ao contrário, se  $e$  é pequeno, a incerteza será grande.

No modelo do SBI definido em [45],  $s = \rho$ , mas esta restrição não vai ser imposta aqui. As notações são as mesmas de [45], exceto para a constante  $k$ , que aqui denotamos por  $\tau$ .

As variáveis estão sujeitas à restrição de equilíbrio orçamentário:

$$a + (1 - \tau)D = M + N + e. \quad (3.2)$$

Uma restrição crucial do modelo é que  $N$  não pode exceder a demanda por crédito  $N^d$ , que depende da taxa de juros  $r$ :

$$N \leq N^d(r). \quad (3.3)$$

Uma outra restrição similar do modelo é que  $D$  não pode exceder a demanda por depósitos  $D^o$ , que depende da taxa de juros  $s$ :

$$D \leq D^o(s). \quad (3.4)$$

Por fim, temos as restrições de caixa:

$$\ell_D \leq D \leq u_D, \quad \ell_M \leq M \leq u_M, \quad \ell_N \leq N \leq u_N, \quad (3.5)$$

$$\ell_s \leq s \leq u_s, \quad \ell_e \leq e \leq u_e, \quad \ell_r \leq r \leq u_r. \quad (3.6)$$

A grosso modo, o objetivo do banco é maximizar o patrimônio líquido no final do período que é dado por:

$$\bar{a} = M(1 + \rho) - sD - (1 - \tau)D + V, \quad (3.7)$$

onde  $V$  representa o retorno obtido pelo crédito concedido. Entretanto,  $V$  não é uma quantia fixa, na verdade,  $V$  é uma variável aleatória com  $m$  possíveis valores  $V_1, \dots, V_m$ , limitados superiormente por  $(1 + r)N$ . De fato, o limitante superior é menor, já que, a chance de inadimplência existe mesmo que a taxa de juros seja muito baixa. Aqui, utilizaremos o limitante

$$V_i \leq 0.99995(1 + r)N, \quad \forall i = 1, \dots, m.$$

O máximo retorno obtido pelo crédito concedido é ainda menor. O limitante acima deve ser multiplicado por um coeficiente entre 0 e 1 que decresce com  $r$ . Isto porque a chance de obter devolução total (com juros) dos empréstimos decresce com  $r$  e tende a zero se  $r$  tende a  $\infty$ . Logo, definimos:

$$\text{GRL}_{max} = 0.99995(1 + r)N \times \exp(-\kappa r^2)$$

para um dado  $\kappa > 0$  e postulamos que:

$$V_i \leq \text{GRL}_{max}, \quad \forall i = 1, \dots, m.$$

O verdadeiro retorno obtido pelo crédito concedido sob o cenário  $i$  é então dado por:

$$V_i = \text{GRL}_{max} - z_i \Delta(e, N) \text{GRL}_{max}, \quad (3.8)$$

onde  $\Delta(e, N) \in [0, 1]$  decresce quando  $e$  cresce e, para todo  $i = 1, \dots, m$ , o número  $z_i \in [0, 1]$  define o cenário  $i$ .

Definimos

$$\Delta(e, N) = \frac{\sqrt{\tilde{\gamma}_1^2 (e - \tilde{\gamma}_2)^2 + 10^{-4}} - \tilde{\gamma}(e - \tilde{\gamma}_2)}{2} + \Delta_{\min} \quad (3.9)$$

onde  $\tilde{\gamma}_1 = \frac{5}{N}$ ,  $\tilde{\gamma}_2 = \frac{N}{5}(1 - \Delta_{min})$ . A expressão (3.9) é uma aproximação diferenciável de  $\max\{1 - 5e/N, \Delta_{min}\}$ .

As definições acima são suficientes para definir o problema OVO. Defina  $n = 6$ ,  $x_1 = M$ ,  $x_2 = N$ ,  $x_3 = e$ ,  $x_4 = r$ ,  $x_5 = s$ ,  $x_6 = D$ . Então, as funções  $f_i$  são dadas por:

$$f_i(x) = -\left[M(1 + \rho) - sD - (1 - \tau)D + V_i\right],$$

onde  $V_i$  é definido por (3.8), para todo  $i = 1, \dots, m$ .

Dado  $p \in \{1, \dots, m\}$ , o problema OVO associado a esta aplicação, que maximiza o VaR do retorno com nível de confiança  $p/m$ , é dado por:

$$\text{Minimizar } f_{i_p}(x) \text{ sujeito a (3.2)-(3.6)}. \quad (3.10)$$

Portanto, (3.10) é um problema OVO com restrições. Com definições apropriadas para  $N^d$  e  $D^o$ , as restrições serão lineares e os Algoritmos 1.1 e 1.2 serão perfeitamente aplicáveis.

Em nossa aplicação prática, usamos os seguintes dados:

- $a = 54.6$  bilhões de reais (moeda brasileira atual),  $\tau = 0.5$ ,  $\rho = 0.03$ ,  $\kappa = 0.6$ ,  $\Delta_{min} = 0.1$ .
- As restrições (3.5) e (3.6) que definem os limitantes inferiores e superiores das variáveis são:

$$0 \leq M \leq 5a, \quad (3.11)$$

$$0.1a \leq N \leq 5a, \quad (3.12)$$

$$0 \leq e \leq 5a, \quad (3.13)$$

$$0 \leq r \leq 0.8, \quad (3.14)$$

$$0.02 \leq s \leq 0.1, \quad (3.15)$$

$$a \leq D \leq 5a. \quad (3.16)$$

- As funções  $N^d$  e  $D^o$  são:

$$N^d(r) = 5a - \frac{4.9a}{0.8}r, \quad (3.17)$$

$$D^o(s) = 50as. \quad (3.18)$$

- Em aplicações totalmente realísticas o vetor  $z$  é estimado usando a classificação dos clientes. Então, a distribuição analítica de  $z_i$  é desconhecida. Nestes casos, não é possível obter uma representação analítica fechada da função objetivo e a única alternativa que nos resta é fazer simulações. Entretanto, com o intuito de simplificar nossa apresentação, adotamos aqui a seguinte definição para  $z_i$ ,  $i = 1, \dots, m$ :

$$z_i = \varphi^{-1}(w_i), \quad i = 1, \dots, m,$$

onde

$$\varphi(t) = 3t^2 - 2t^3, \quad \forall t \in [0, 1]$$

e  $w_1, \dots, w_m$  são números randômicos uniformemente distribuídos em  $[0, 1]$ .

### 3.3 Experimentos Numéricos

Para realizar os experimentos numéricos desta seção, utilizamos o Algoritmo 1.2. Estes foram os primeiros testes realizados com este algoritmo, proposto recentemente em [9]. Nesta seção, descrevemos algumas características da implementação do Algoritmo 1.2, o processo de escolha do ponto inicial e os resultados numéricos obtidos.

#### 3.3.1 Aproximações da Hessiana e parâmetros do algoritmo

Se o número de cenários  $m$  é pequeno, é possível calcular ou atualizar uma matriz  $B_{k,j}$  distinta para cada  $j = 1, \dots, m$ . Porém, não temos condições de realizar estes cálculos se  $m$  é grande. Se adotarmos um método Newtoniano ( $B_{k,j} = \nabla^2 f_j(x_k)$ ) podemos calcular  $B_{k,j}$  somente quando  $j \in I_\varepsilon(x_k)$ . Esta é uma alternativa interessante, mas, não temos como garantir que a Hessiana será uma matriz semidefinida positiva. Em nossas implementações, adotamos o método BFGS [24]. Além disso, com o intuito de reduzir o custo computacional, postulamos que  $B_k = B_{k,j}$ , para todo  $j = 1, \dots, m$ ,  $k \in \mathbb{N}$ . As matrizes  $B_k$  foram geradas tomando  $B_0 = I$  e, para todo  $k \in \mathbb{N}$ ,

$$s_k = x_{k+1} - x_k, \quad y_k = \nabla f_{i_p(x_k)}(x_{k+1}) - \nabla f_{i_p(x_k)}(x_k),$$

$$B_{k+1} = B_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k}$$

se  $y_k^T s_k > 0$ , ao passo que,  $B_{k+1} = B_k$  se  $y_k^T s_k \leq 0$ . É de conhecimento de todos, que isto é suficiente para garantir que  $B_k$  seja definida positiva, para todo  $k \in \mathbb{N}$  [24].

Finalmente, se  $\|B_k\|_\infty > 10^3 n$ , substituímos  $B_k$  pela matriz identidade.

Os parâmetros algorítmicos utilizados foram:

$$\Delta = 1.0, \theta = 0.5, \sigma_{\min} = 0.1, \sigma_{\max} = 0.9, \eta = 1, \varepsilon = 10^{-6}.$$

O critério de convergência foi  $M_k(d_k) \geq -10^{-6}$ .

Para resolver o subproblema do Passo 1 do algoritmo utilizamos a rotina DNCONG [43, 44] da biblioteca IMSL.

### 3.3.2 Escolha do ponto inicial

Dado  $p \in \{1, \dots, m\}$ , o problema OVO associado ao modelo do SBI é o definido em (3.10). Antes de iniciar o Algoritmo 1.2 geramos 1000 pontos aleatórios em  $\Omega$  e avaliamos a função OVO de ordem  $p$  em cada um destes pontos. Definimos a aproximação inicial  $x_0$  como sendo o ponto que atingir o menor valor da função OVO de ordem  $p$ . Em nossas implementações, o conjunto  $\Omega$  é definido pelas restrições (3.2), (3.11)-(3.18). Para gerar pontos iniciais aleatórios em  $\Omega$  procedemos da seguinte forma:

1. Sorteamos  $r$  entre 0 and 0.8 aleatoriamente. Desta forma, a restrição (3.14) é satisfeita.
2. Sorteamos  $N$ , de forma aleatória, entre  $0.1a$  e  $N^d(r)$ . Desta forma, as restrições (3.12) e (3.17) são satisfeitas.
3. Sorteamos  $e$  aleatoriamente entre 0 e  $\frac{N}{5}(1 - \Delta_{\min})$ . Desta forma, a restrição (3.13) é satisfeita.
4. (a) Se  $2(N + e - a) < a$ , sorteamos  $D$  entre  $a$  e  $5a$ .  
 (b) Se  $a \leq 2(N + e - a) \leq 5a$ , sorteamos  $D$  entre  $2(N + e - a)$  e  $5a$ .  
 (c) Se  $2(N + e - a) > 5a$ , é impossível satisfazer todas as restrições. Neste caso, retornamos ao sorteio de  $N$  e  $e$ .
5. Calculamos  $s = \frac{D}{50a}$  que é a menor taxa que satisfaz as restrições (3.15) e (3.18).
6. Finalmente, calculamos  $M = a + (1 - \tau)D - N - e$ . Logo, a restrição de balanço (3.2) é satisfeita. Como  $a + (1 - \tau)D \geq N + e$ ,  $M$  também satisfaz a restrição (3.11).

$p$	$M$	$N$	$e$	$r$	$s$	$D$	n.iter	n.feval	$-f(x_0)$	$-f(x_*)$
10	0.	110.	0.	0.488	0.040	110.	30	31	70.1	71.6
50	0.	90.8	0.	0.541	0.027	72.5	22	25	61.6	62.3
100	0.	69.5	12.4	0.541	0.020	54.6	19	29	58.6	59.4
500	0.	69.4	12.5	0.541	0.020	54.6	45	55	56.2	56.7
700	75.5	5.46	0.989	0.541	0.020	54.6	22	33	55.4	55.9
900	75.5	5.46	0.991	0.541	0.020	54.6	19	29	55.2	55.8
950	75.4	5.46	0.991	0.541	0.020	54.6	12	22	54.9	55.8
990	75.4	5.46	0.992	0.541	0.020	54.6	25	37	55.0	55.7
995	75.4	5.46	0.993	0.541	0.020	54.6	22	29	54.7	55.7
999	75.4	5.46	0.993	0.541	0.020	54.6	20	29	55.0	55.7

Tabela 3.1: Resultados obtidos pelo Algoritmo 1.2

### 3.3.3 Resultados Numéricos

Os experimentos foram realizados em um Pentium 4, 2.4 Ghz, 1Gb RAM em FORTRAN de dupla precisão.

Tomando  $m = 1000$ , as soluções  $x^* = (M, N, e, r, s, D)$  obtidas pelo Algoritmo 1.2 para diferentes valores de  $p$  estão reportadas na Tabela 3.1. Nesta tabela, n.iter representa o número de iterações realizadas pelo Algoritmo 1.2 e n.feval representa o número de avaliações da função OVO.

Observe que a solução é, essencialmente, a mesma para todos valores de  $p \geq 700$ . A aversão ao risco, juntamente com as restrições deste problema, geram pequena oferta de crédito a taxas de juros muito altas ( $N$  pequeno,  $r$  grande). Este resultado era esperado, mas, a surpresa é que o grau de aversão ao risco (medido pelo nível de confiança  $p/m$  do VaR) nem precisou ser muito próximo de 1 para que esta solução fosse obtida. É interessante reportar, para cada valor de  $p$ , a média e o desvio padrão do patrimônio líquido no final do período (3.7), seu valor máximo e seu valor mínimo. Estas quantidades são definidas abaixo e reportadas na Tabela 3.2.

$$\text{Média do PL} = \mu = \frac{1}{m} \sum_{i=1}^m -f_i(x^*),$$

$p$	Máximo PL	Mínimo PL	Média PL	Desvio PL
10	79.7	-57.7	-2.05	24.4
50	77.4	-36.6	9.60	20.2
100	61.3	51.7	55.6	1.70
500	61.1	52.1	55.8	1.60
700	56.4	55.7	56.0	0.125
900	56.4	55.7	55.9	0.125
950	56.4	55.7	55.9	0.125
990	56.4	55.7	55.9	0.124
995	56.4	55.7	55.9	0.124
999	56.4	55.7	55.9	0.124

Tabela 3.2: Estatísticas do Patrimônio Líquido Final

$$\begin{aligned} \text{Desvio do PL} &= \sqrt{\frac{1}{m-1} \sum_{i=1}^m [-f_i(x^*) - \mu]^2}, \\ \text{Máximo PL} &= \max\{-f_1(x_*), \dots, -f_m(x_*)\}, \\ \text{Mínimo PL} &= \min\{-f_1(x_*), \dots, -f_m(x_*)\}. \end{aligned}$$

Na Tabela 3.2, podemos observar que, quando a aversão ao risco aumenta ( $p$  grande) o máximo PL decresce. Entretanto, este decréscimo não existe para  $p \geq 700$ . Reciprocamente, o mínimo PL aumenta quando  $p$  cresce. Além disso, para  $p \geq 700$  a aversão ao risco produz um desvio padrão da distribuição final muito pequeno. Conseqüentemente, a diferença entre o máximo e o mínimo PL é pequena. É interessante observar que a média do PL cresce para  $p = 10$  até  $p = 700$  e decresce para  $p = 700$  em diante.

Com o intuito de alcançar uma melhor compreensão do comportamento do Algoritmo 1.2, investigamos as iterações individuais do método para o caso  $p = 900$ . Observamos que:

- Para todas as iterações foi verificado que o conjunto  $I_\varepsilon(x_k)$  tem apenas um elemento (sempre o mesmo). Nas últimas 15 iterações tivemos  $t_k = 1$  (a primeira aproximação foi aceita). Observe, na Tabela 3.1, que pelo fato da diferença entre

$n.iter$  e  $n.feval$  ter sido pequena, a primeira aproximação foi aceita na maioria das iterações para todos os problemas (todos os valores de  $p$ ).

- A restrição (3.4) foi ativa na solução encontrada. As variáveis  $N$ ,  $s$  e  $D$  atingiram seus limitantes inferiores. Estas quatro restrições foram ativas nas últimas 5 iterações.
- Os últimos 10 quocientes  $\|x_{k+1} - x_*\|/\|x_k - x_*\|$  foram:

$$0.75, 0.71, 0.83, 0.94, 0.56, 0.04, 0.59, 0.06, 0.53, 0.005.$$

Estas observações são compatíveis com os resultados teóricos de convergência super-linear do algoritmo.

Novamente, para o caso  $p = 900$ , tentamos resolver o problema (3.10) com o Algoritmo 1.1 definindo os mesmos parâmetros, ponto inicial e critério de parada utilizados no Algoritmo 1.2. Após 50000 iterações o critério de parada  $M_k(d_k) > -10^{-6}$  não foi atingido. Além disso, observamos que:

- Como no Algoritmo 1.2, para todas as iterações do Algoritmo 1.1, foi verificado que o conjunto  $I_\varepsilon(x_k)$  tem apenas um elemento (sempre o mesmo).
- A distância euclidiana entre a solução encontrada pelo Algoritmo 1.2 e os pontos obtidos pelo Algoritmo 1.1 após 10000, 20000, 30000, 40000 e 50000 iterações foram:

$$1.0346, 0.58938, 0.46413, 0.39863, \text{ e } 0.35249, \text{ respectivamente.}$$

- A diferença entre o menor valor obtido pelo método quase-Newton e o valor da função OVO nos pontos obtidos pelo Algoritmo 1.1 após 10000, 20000, 30000, 40000 e 50000 iterações foram:

$$0.01865, 0.00601, 0.00372, 0.00274, \text{ e } 0.00214, \text{ respectivamente.}$$

- O número de avaliações da função OVO após 50000 iterações do Algoritmo 1.1 foi 741663.
- Para todas as iterações do Algoritmo 1.1,  $0 < t_k \leq 0.0625 < 1$ . Logo, a primeira aproximação foi sempre rejeitada.

Estas observações mostram que, embora o Algoritmo 1.1 convirja para a solução, esta convergência, quando comparada com o Algoritmo 1.2, é muito lenta.

# Capítulo 4

## Otimização da Menor Soma de Valores Ordenados

Neste capítulo, discutiremos as condições de otimalidade e algoritmos para o problema LOVO utilizando, como já alertamos anteriormente, a formulação (4).

### 4.1 Otimalidade

Para todo  $x \in \Omega$  definimos:

$$I(x) = \{i \in \{1, \dots, m\} \mid f_i(x) = f_{\min}(x)\}.$$

No Lema 4.1, provamos que um minimizador global  $x_*$  de (4) é, necessariamente, um minimizador global de  $f_i(x)$ , para todo  $i \in I(x_*)$ . Como consequência, mostramos no Teorema 4.2 que a mesma propriedade vale para minimizadores locais.

**Lema 4.1.** *Seja  $A \subset \Omega$ ,  $x_* \in A$ . Se  $x_*$  é um minimizador global de  $f_{\min}(x)$  sujeito a  $x \in A$ , então  $x_*$  é um minimizador global de  $f_i(x)$  sujeito a  $x \in A$ , para todo  $i \in I(x_*)$ . Em particular (tomando  $A = \Omega$ ), se  $x_*$  é um minimizador global de (4) então  $x_*$  é um minimizador global de  $f_i(x)$ , para todo  $i \in I(x_*)$*

*Prova.* Suponha que, para algum  $i \in I(x_*)$ ,  $x_*$  não é um minimizador global de  $f_i(x)$  em  $A$ . Então, existe  $y \in A$  tal que  $f_i(y) < f_i(x_*)$ . Logo, segue das definições de  $f_{\min}$  e  $I(x_*)$  que:

$$f_{\min}(y) \leq f_i(y) < f_i(x_*) = f_{\min}(x_*).$$

Mas, isto significa, que  $x_*$  não é um minimizador global de  $f_{min}(x)$  em  $A$ , contradizendo a hipótese.  $\square$

**Teorema 4.2.** *Se  $x_* \in \Omega$  é um minimizador local de (4) então, para todo  $i \in I(x_*)$ ,  $x_*$  é um minimizador local de  $f_i(x)$  sujeito a  $x \in \Omega$ .*

*Prova.* Defina  $\epsilon > 0$  de forma que  $x_*$  seja um minimizador global de  $f_{min}(x)$  em  $A$ , onde

$$A = \{x \in \Omega \mid \|x - x_*\| \leq \epsilon\}.$$

Segue, do Lema 4.1, que  $x_*$  é um minimizador global de  $f_i(x)$  em  $A$ , para todo  $i \in I(x_*)$ . Portanto,  $x_*$  é um minimizador local de  $f_i(x)$  em  $\Omega$ , para todo  $i \in I(x_*)$ .  $\square$

**Observação.** A recíproca do Lema 4.1 não é verdadeira, mesmo que as funções  $f_i$  sejam contínuas. Considere  $A = \Omega = \mathbb{R}$ ,  $f_1(x) = (x - 1)^2$ ,  $f_2(x) = x$ . Embora  $x_* = 1$  seja um minimizador global de  $f_i(x)$  para todo  $i \in I(x_*) = \{1\}$ , este ponto não é um minimizador global de  $f_{min}$ . Entretanto, como veremos abaixo, a recíproca do Teorema 4.2 é verdadeira se as funções  $f_i$  são contínuas.

**Proposição 4.3.** *Suponha que  $x_*$  é um minimizador local de  $f_i$ , para todo  $i \in I(x_*)$ , e que  $f_i$  é contínua em  $x_*$ , para todo  $i \notin I(x_*)$ . Então,  $x_*$  é um minimizador local de (4).*

*Prova.* Defina  $\epsilon > 0$  de forma que:

$$f_i(x_*) > f_{min}(x_*) + \epsilon, \quad \text{para todo } i \notin I(x_*).$$

Como  $f_i$  é contínua, para todo  $i \notin I(x_*)$ , existe  $\delta_1 > 0$  tal que

$$f_i(x) > f_{min}(x_*), \quad \text{para todo } i \notin I(x_*), \quad (4.1)$$

sempre que  $\|x - x_*\| \leq \delta_1$ .

Por hipótese, existe  $\delta_2 > 0$  tal que, para todo  $i \in I(x_*)$ ,

$$f_i(x) \geq f_i(x_*) = f_{min}(x_*), \quad (4.2)$$

sempre que  $\|x - x_*\| \leq \delta_2$ .

Defina  $\delta = \min\{\delta_1, \delta_2\}$ . Por (4.1) e (4.2) temos, para todo  $x \in \Omega$  tal que  $\|x - x_*\| \leq \delta$ , e para todo  $i = 1, \dots, m$ , que:

$$f_i(x) \geq f_{min}(x_*).$$

Portanto,

$$f_{min}(x) \geq f_{min}(x_*),$$

para todo  $x \in \Omega$  tal que  $\|x - x_*\| \leq \delta$ .  $\square$

Seja  $\Phi$  uma função diferenciável em um aberto que contém  $\Omega$  e considere o problema de programação não-linear (PNL)

$$\text{Minimizar } \Phi(x) \text{ sujeito a } x \in \Omega. \quad (4.3)$$

Condições Necessárias de Otimalidade (CNO) são condições que devem ser satisfeitas pelos minimizadores locais de (4.3). Por exemplo, se  $\Omega = \mathbb{R}^n$ , a condição “ $\nabla\Phi(x) = 0$ ” é uma CNO. Nos problemas de otimização com restrições as CNO, geralmente, são descritas da seguinte forma: *se  $x_*$  satisfaz uma condição de qualificação, então as condições de KKT são satisfeitas*. Veja, por exemplo [13]. Condições de qualificação envolvem somente propriedades do conjunto  $\Omega$ , enquanto que, as condições de KKT envolvem o gradiente da  $f$  e os gradientes das restrições.

O Teorema 4.2, permite-nos provar o seguinte Corolário.

**Corolário 4.4.** *Seja  $x_* \in \Omega$  um minimizador local do problema (4) e suponha que todas as funções  $f_i$  são diferenciáveis em um aberto que contém  $\Omega$ . Então, para todo  $i \in I(x_*)$ ,  $x_*$  satisfaz as condições necessárias de otimalidade associadas ao problema*

$$\text{Minimizar } f_i(x) \text{ sujeito a } x \in \Omega. \quad (4.4)$$

*Prova.* Pelo Teorema 4.2,  $x_*$  é um minimizador local de  $f_i$ , para todo  $i \in I(x_*)$ . Portanto,  $x_*$  satisfaz as condições necessárias de otimalidade associadas a estes problemas.

$\square$

Dada uma CNO para um PNL, o Corolário 4.4 motiva as seguintes definições:

**Definição.** Dizemos que  $x_* \in \Omega$  é um *ponto crítico forte* se, para todo  $i \in I(x_*)$ ,  $x_*$  satisfaz a CNO associada ao problema (4.4) e dizemos que  $x_* \in \Omega$  é um *ponto crítico fraco* se existe  $i \in I(x_*)$  tal que  $x_*$  satisfaz a CNO associada ao problema (4.4).

## 4.2 Algoritmos para LOVO Irrestrito

Baseados na última definição, vamos propor, nesta seção, dois algoritmos para o problema LOVO irrestrito: um que converge para pontos críticos fracos e outro que converge para pontos críticos fortes.

### 4.2.1 Convergência para pontos críticos fracos

Os algoritmos de otimização para resolver problemas de programação não-linear do tipo (4.3), são iterativos. Geralmente, a cada iteração, é necessário avaliar o valor funcional, os gradientes e, talvez, a segunda derivada da função objetivo e das restrições. Códigos que implementam algoritmos de programação não-linear devem conter subrotinas que calculam estas quantias.

Na presença dos problemas (2) ou (4) a idéia de usar um método qualquer de otimização diferenciável que seja computacionalmente bem estabelecido e teoricamente bem fundamentado é atrativa. A cada vez que o (talvez não existente)  $\nabla f_{min}(x)$  é exigido pelo algoritmo, podemos escolher  $i \in I(x)$  e “definir”

$$\nabla f_{min}(x) \leftarrow \nabla f_i(x). \quad (4.5)$$

(Podemos proceder de forma similar se o algoritmo também exigir a Hessiana).

Para muitos problemas não diferenciáveis, aplicar tal estratégia pode ser catastrófico. Entretanto, vamos mostrar aqui, que nos casos (2) e (4), as consequências são menos severas. Essencialmente, vamos mostrar que a convergência para pontos críticos fracos necessariamente ocorre.

O Algoritmo 4.1, definido abaixo, é aplicável à minimização irrestrita ( $\Omega = \mathbb{R}^n$ ) de  $f_{min}(x)$ . Vamos supor que as funções  $f_i$  são continuamente diferenciáveis, para todo  $x \in \mathbb{R}^n$ . Este algoritmo deve ser interpretado como a aplicação direta de um método diferenciável de minimização irrestrita ao problema LOVO irrestrito com a “avaliação incorreta” (4.5).

**Algoritmo 4.1.** Seja  $x_0 \in \mathbb{R}^n$  um ponto inicial arbitrário. Sejam

$$\theta \in (0, 1), \alpha \in (0, 1), M > 1, \beta > 0, t_{one} > 0$$

parâmetros do algoritmo. Dado  $x_k \in \mathbb{R}^n$ , os passos para obter a nova iteração  $x_{k+1}$  são:

**Passo 1** Escolha  $\nu(k) \in I(x_k)$ . Se  $\|\nabla f_{\nu(k)}(x_k)\| = 0$ , parar.

**Passo 2** Calcule  $d_k \in \mathbb{R}^n$  tal que

$$\nabla f_{\nu(k)}(x_k)^T d_k \leq -\theta \|d_k\| \|\nabla f_{\nu(k)}(x_k)\| \quad \text{e} \quad \|d_k\| \geq \beta \|\nabla f_{\nu(k)}(x_k)\|. \quad (4.6)$$

**Passo 3** Calcule  $t_k > 0$ ,  $x_{k+1} \in \mathbb{R}^n$ , tal que

$$f_{\min}(x_{k+1}) \leq f_{\min}(x_k) + \alpha t_k \nabla f_{\nu(k)}(x_k)^T d_k \quad (4.7)$$

e

$$\left[ t_k \geq t_{\text{one}} \text{ ou } f_{\min}(x_k + \bar{t}_k d_k) > f_{\min}(x_k) + \alpha \bar{t}_k \nabla f_{\nu(k)}(x_k)^T d_k \text{ para algum } \bar{t}_k \leq M t_k \right]. \quad (4.8)$$

A estratégia de busca linear (4.7)-(4.8) admite diferentes implementações. A mais direta é o “backtracking”. Neste caso,  $t_k$  é escolhido como sendo o primeiro número da sequência  $\{1, 2^{-1}, 2^{-2}, \dots\}$  que satisfaz (4.7) e  $x_{k+1} = x_k + t_k d_k$ . Neste caso,  $t_{\text{one}} = 1$  e  $M = 2$ . Entretanto, a escolha baseada em (4.7)-(4.8) admite procedimentos de busca mais sofisticados e eficientes. Veja, por exemplo, [14].

Lembremos que, no problema LOVO irrestrito, um ponto crítico fraco é um ponto tal que  $\nabla f_i(x) = 0$ , para algum  $i \in I(x)$ . Nos seguintes teoremas, provamos que o algoritmo pára em  $x_k$  somente se  $x_k$  é um ponto crítico fraco e que pontos limite de sequências geradas pelo Algoritmo 4.1 são pontos críticos fracos.

**Teorema 4.5.** *O Algoritmo 4.1 está bem definido e termina em  $x_k$  somente se  $x_k$  é um ponto crítico fraco.*

*Prova.* Suponha que  $x_k$  não é um ponto crítico fraco e defina  $i = \nu(k)$ . Então,  $\nabla f_i(x_k) \neq 0$ . Segue de (4.6) e da diferenciabilidade de  $f_i$  que:

$$\lim_{t \rightarrow 0} \frac{f_i(x_k + t d_k) - f_i(x_k)}{t} = \nabla f_i(x_k)^T d_k < 0.$$

Desta forma,

$$\lim_{t \rightarrow 0} \frac{f_i(x_k + t d_k) - f_i(x_k)}{t \nabla f_i(x_k)^T d_k} = 1.$$

Como  $\alpha < 1$ , para  $t$  suficientemente pequeno temos

$$\frac{f_i(x_k + t d_k) - f_i(x_k)}{t \nabla f_i(x_k)^T d_k} \geq \alpha.$$

Como  $\nabla f_i(x_k)^T d_k < 0$ , deduzimos que

$$f_i(x_k + td_k) \leq f_i(x_k) + \alpha t \nabla f_i(x_k)^T d_k.$$

Mas,  $f_{min}(x_k + td_k) \leq f_i(x_k + td_k)$  e  $f_{min}(x_k) = f_i(x_k)$ . Logo,

$$f_{min}(x_k + td_k) \leq f_{min}(x_k) + \alpha t \nabla f_i(x_k)^T d_k, \quad (4.9)$$

para  $t$  suficientemente pequeno.

Portanto, escolhendo  $t_k$  como sendo o primeiro número da sequência

$$\{t_{one}, t_{one}/M, t_{one}/M^2, \dots\}$$

que satisfaz (4.9), as condições (4.7) e (4.8) são satisfeitas.

Isto prova que, sempre que  $x_k$  não é um ponto crítico fraco, um ponto  $x_{k+1}$  satisfazendo (4.7)-(4.8) pode ser encontrado, logo o algoritmo está bem definido.  $\square$

Vale ressaltar que o Teorema 4.5 diz que, se o Algoritmo 4.1 termina em  $x_k$ , então  $x_k$  é um ponto crítico fraco, mas a recíproca não é verdadeira. Por exemplo, defina  $n = 1$ ,  $m = 2$ ,  $f_1(x) = x$ ,  $f_2(x) = x^2$ . Claramente, 0 é um ponto crítico fraco já que  $\nabla f_2(0) = 0$ . Entretanto, se  $x_k = 0$  e escolhermos  $\nu(k) = 1$  o algoritmo não pára em  $x_k = 0$  e, certamente, encontrará um ponto (melhor)  $x_{k+1}$  tal que  $f_{min}(x_{k+1}) < f_{min}(0)$ .

**Teorema 4.6.** *Se  $x_*$  é um ponto limite de uma sequência gerada pelo Algoritmo 4.1, então,  $x_*$  é um ponto crítico fraco. Além disso, se  $\lim_{k \in K} x_k = x_*$  e um mesmo  $i = \nu(k) \in I(x_k)$  é escolhido no Passo 1 do algoritmo para infinitos índices  $k \in K$ , então  $i \in I(x_*)$  e  $\nabla f_i(x_*) = 0$ . Finalmente,*

$$\lim_{k \in K} \|\nabla f_{\nu(k)}(x_k)\| = 0. \quad (4.10)$$

*Prova.* Seja  $x_* \in \mathbb{R}^n$  um ponto limite da sequência gerada pelo Algoritmo 4.1. Seja  $K = \{k_0, k_1, k_2, k_3, \dots\}$  uma sequência infinita de índices tal que:

1. Existe  $i \in \{1, \dots, m\}$  tal que  $i = \nu(k)$  para todo  $k \in K$ ;
2.  $\lim_{k \in K} x_k = x_*$ .

Como  $\{1, \dots, m\}$  é finito, a sequência  $K$  e o índice  $i$  necessariamente existem.

Pela continuidade de  $f_i$ ,

$$\lim_{k \in K} f_i(x_k) = f_i(x_*). \quad (4.11)$$

Claramente, como  $i = \nu(k)$ , temos

$$f_i(x_k) \leq f_\ell(x_k) \text{ para todo } \ell \in \{1, \dots, m\},$$

para todo  $k \in K$ .

Aplicando o limite em ambos os membros da desigualdade, obtemos  $f_i(x_*) \leq f_\ell(x_*)$  para todo  $\ell \in \{1, \dots, m\}$ . Portanto,

$$i \in I(x_*). \quad (4.12)$$

Como  $k_{j+1} \geq k_j + 1$  segue, pela própria definição do Algoritmo 4.1, que

$$\begin{aligned} f_i(x_{k_{j+1}}) &= f_{\min}(x_{k_{j+1}}) \leq f_{\min}(x_{k_j+1}) \\ &\leq f_{\min}(x_{k_j}) + \alpha t_{k_j} \nabla f_i(x_{k_j})^T d_{k_j} < f_{\min}(x_{k_j}) = f_i(x_{k_j}), \end{aligned} \quad (4.13)$$

para todo  $j \in \mathbb{N}$ .

De (4.7), (4.11) e (4.13), obtemos:

$$\lim_{j \rightarrow \infty} t_{k_j} \nabla f_i(x_{k_j})^T d_{k_j} = 0.$$

Logo, por (4.6),

$$\lim_{j \rightarrow \infty} t_{k_j} \|\nabla f_i(x_{k_j})\| \|d_{k_j}\| = 0. \quad (4.14)$$

Se, para alguma subsequência  $K_1 \subset K$ ,  $\lim_{k \in K_1} \|\nabla f_i(x_k)\| = 0$ , deduzimos que  $\nabla f_i(x_*) = 0$  e a tese está provada. Portanto, precisamos apenas analisar a possibilidade de que  $\|\nabla f_i(x_k)\| > \epsilon$ , para todo  $k \in K$  e algum  $\epsilon > 0$ . Neste caso, por (4.14),

$$\lim_{k \in K} t_k \|d_k\| = 0. \quad (4.15)$$

Se, para alguma subsequência,  $\|d_k\| \rightarrow 0$ , a condição (4.6) também implica que  $\nabla f_i(x_k) \rightarrow 0$  e  $\nabla f_i(x_*) = 0$ . Então, precisamos apenas considerar o caso em que  $\lim_{k \in K} t_k = 0$ . Sem perda de generalidade, podemos supor que  $t_k < t_{one}$ , para todo  $k \in K$ . Então, por (4.8), para todo  $k \in K$  existe  $\bar{t}_k > 0$  tal que

$$f_i(x_k + \bar{t}_k d_k) \geq f_{\min}(x_k + \bar{t}_k d_k) > f_{\min}(x_k) + \alpha \bar{t}_k \nabla f_i(x_k)^T d_k = f_i(x_k) + \alpha \bar{t}_k \nabla f_i(x_k)^T d_k. \quad (4.16)$$

Além disso, por (4.8) e (4.15),

$$\lim_{k \in K} \bar{t}_k \|d_k\| = 0. \quad (4.17)$$

Defina  $s_k = \bar{t}_k d_k$  para todo  $k \in K$ . Então, por (4.17),

$$\lim_{k \in K} \|s_k\| = 0. \quad (4.18)$$

Por (4.16) e pelo Teorema do Valor Médio, para todo  $k \in K$  existe  $\xi_k \in [0, 1]$  tal que

$$\nabla f_i(x_k + \xi_k s_k)^T s_k = f_i(x_k + s_k) - f_i(x_k) > \alpha \nabla f_i(x_k)^T s_k. \quad (4.19)$$

Além disso, por (4.6),

$$\frac{\nabla f_i(x_k)^T s_k}{\|s_k\|} \leq -\theta \|\nabla f_i(x_k)\|, \quad (4.20)$$

para todo  $k \in K$ .

Defina  $K_1 \subset K$ ,  $s \in \mathbb{R}^n$  de forma que  $\lim_{k \in K_1} s_k / \|s_k\| = s$ .

Por (4.18), dividindo ambos os membros da inequação (4.19) por  $\|s_k\|$  e, aplicando o limite para  $k \in K_1$ , obtemos:

$$\nabla f_i(x_*)^T s \geq \alpha \nabla f_i(x_*)^T s.$$

Como  $\alpha < 1$  e  $\nabla f_i(x_k)^T d_k < 0$ , para todo  $k$ , isto implica que  $\nabla f_i(x_*)^T s = 0$ . Então, aplicando o limite em (4.20), concluímos que  $\nabla f_i(x_*) = 0$ . Portanto, por (4.12),  $x_*$  é um ponto crítico fraco.

Finalmente, vamos provar (4.10). Se (4.10) não é satisfeita, existe  $j$  e um conjunto infinito de índices  $k \in K$  tal que  $j = \nu(k)$  e  $\|\nabla f_j(x_k)\| > \epsilon$ , para algum  $\epsilon > 0$ . Isto implica que  $j \in I(x_*)$  e  $\|\nabla f_j(x_*)\| \neq 0$ , contradizendo a primeira parte da prova.  $\square$

O resto desta seção vai tratar dos resultados de convergência local do Algoritmo 4.1. A escolha de  $x_{k+1}$  neste algoritmo impõe que  $f_{\min}(x_{k+1}) \leq f_{\min}(x_k) + \alpha t_k \nabla f_{\nu(k)}(x_k)^T d_k$ . Esta propriedade é claramente satisfeita se  $x_{k+1} = x_k + t_k d_k$ , mas, para aumentar a probabilidade de convergência a minimizadores globais, outras definições para  $x_{k+1}$ , que aceleram o decrescimento da função objetivo a cada iteração são possíveis e, até, desejáveis. Para convergência local, entretanto, a distância entre  $x_{k+1}$  e  $x_k$  deve ser pequena se  $x_k$  está próximo de “se tornar” um ponto crítico. Este requerimento está descrito na

seguinte Hipótese **B1**.

**Hipótese B1** Suponha que o Algoritmo 4.1 está implementado de forma que exista  $b > 0$  tal que

$$\|x_{k+1} - x_k\| \leq b \|\nabla f_{\nu(k)}(x_k)\|, \quad (4.21)$$

para todo  $k \in \mathbb{N}$ .

A Hipótese **B1** é compatível com buscas lineares baseadas em (4.8). Para escolhas de  $d_k$  do tipo direção do gradiente, direção de Newton ou quase-Newton, geralmente, temos  $\|d_k\| = O(\|\nabla f_{\nu(k)}(x_k)\|)$ . Obviamente, “backtracking” preserva esta propriedade com  $t_k d_k$  substituindo  $d_k$ . Então, é possível obter um ponto  $x_{k+1}$  da forma  $x_k + t_k d_k$  satisfazendo (4.21).

Nossa estratégia para provar convergência local superlinear tem três partes. No Teorema 4.7 mostramos que, sob uma hipótese de isolamento, se  $x_*$  é um ponto limite do algoritmo, a sequência toda converge para  $x_*$ . No Teorema 4.8 provamos que, se o algoritmo é iniciado perto de um minimizador local estrito, a sequência toda converge para este minimizador. Nem o Teorema 4.8 pode ser reduzido ao Teorema 4.7, tampouco, o Teorema 4.7 é uma consequência do Teorema 4.8 (a hipótese sobre  $x_*$  no Teorema 4.7 é mais fraca). Entretanto, ambos teoremas mostram que, em muitos casos, a convergência da sequência toda para um ponto  $x_*$  deve ser esperada. Sob esta hipótese e supondo que as direções de busca são soluções inexatas de sistemas lineares do tipo quase-Newton, com uma condição de Dennis-Moré compatível, mostraremos que a convergência é superlinear.

**Definição.** Dizemos que  $x_*$  é *fortemente isolado* se existe  $\epsilon > 0$  tal que para todo  $x \in B(x_*, \epsilon) - \{x_*\}$  e para todo  $i \in I(x)$ , temos  $\nabla f_i(x) \neq 0$ . Em outras palavras, existe uma pequena vizinhança de  $x_*$  que não contém pontos críticos fracos.

**Teorema 4.7.** *Suponha que  $x_*$  é fortemente isolado, a sequência  $\{x_k\}$  é gerada pelo Algoritmo 4.1, a Hipótese **B1** é satisfeita e  $\lim_{k \in K} x_k = x_*$ , para algum conjunto infinito de índices  $K \subset \mathbb{N}$ . Então,  $x_*$  é um ponto crítico fraco e*

$$\lim_{k \rightarrow \infty} x_k = x_*.$$

*Prova.* O fato de  $x_*$  ser um ponto crítico fraco é consequência do Teorema 4.6.

Por (4.10) e (4.21), temos:

$$\lim_{k \in K} \|x_{k+1} - x_k\| = 0. \quad (4.22)$$

Como  $x_*$  é fortemente isolado, existe  $\epsilon > 0$  tal que  $\nabla f_i(x) \neq 0$ , para todo  $i \in I(x)$ , sempre que  $x \in B(x_*, \epsilon) - \{x_*\}$ .

Segue de (4.22) e da hipótese do teorema que existe  $k_1 \in K$  tal que

$$\|x_{k+1} - x_k\| < \epsilon/2 \text{ e } \|x_k - x_*\| < \epsilon/2,$$

para todo  $k \in K, k \geq k_1$ .

Defina:

$$C = \{x \in \mathbb{R}^n \mid \epsilon/2 \leq \|x - x_*\| \leq \epsilon\}.$$

Claramente,  $C$  é compacto e não contém pontos críticos fracos. Então, pelo Teorema 4.6,  $C$  não pode conter uma quantidade infinita de iterandos. Portanto, temos duas possibilidades:

1. Existe  $k_2 \in \mathbb{N}$  tal que  $\|x_k - x_*\| \leq \epsilon/2$ , para todo  $k \geq k_2$ .
2. Existem infinitas iterações  $k \geq k_1$ , tais que  $\|x_k - x_*\| \leq \epsilon/2$  e  $\|x_{k+1} - x_k\| > \epsilon/2$ .

No primeiro caso, como  $x_*$  é o único candidato a ponto limite na bola com raio  $\epsilon/2$  temos que a sequência  $\{x_k\}$  converge para  $x_*$ .

Analisemos o segundo caso. Defina  $K_1 \subset \mathbb{N}$  de forma que  $\|x_k - x_*\| \leq \epsilon/2$  e  $\|x_{k+1} - x_k\| > \epsilon/2$ , para todo  $k \in K_1$ . Como todas as iterações pertencem à bola de raio  $\epsilon/2$  e  $x_*$  é o único candidato a ponto limite nesta bola, segue que:

$$\lim_{k \in K_1} x_k = x_*.$$

Portanto, por (4.10),

$$\lim_{k \in K_1} \|\nabla f_{\nu(k)}(x_k)\| = 0.$$

De (4.21), isto implica que:

$$\lim_{k \in K_1} \|x_{k+1} - x_k\| = 0,$$

contradizendo a hipótese  $\|x_{k+1} - x_k\| \geq \epsilon/2, \forall k \in K_1$ . Isto significa que o segundo caso mencionado acima é impossível. Logo, a prova está completa.  $\square$

**Teorema 4.8.** *Suponha que  $x_*$  é um minimizador local estrito fortemente isolado. Seja  $\{x_k\}$  uma sequência gerada por um Algoritmo do tipo 4.1 que satisfaça a Hipótese **B1**. Então, existe  $\delta_1 > 0$  tal que  $\|x_0 - x_*\| \leq \delta_1$  implica que:*

$$\lim_{k \rightarrow \infty} x_k = x_*.$$

*Prova.* Defina  $\epsilon > 0$  de forma que  $x_*$  seja um minimizador global estrito de  $f_{min}$  na bola  $B(x_*, \epsilon)$  e que esta bola não contenha pontos críticos fracos além de  $x_*$ . Vamos provar que existe  $\delta \in (0, \epsilon/2)$  tal que

$$\|x_k - x_*\| \leq \delta \Rightarrow \|x_{k+1} - x_k\| \leq \epsilon/2. \quad (4.23)$$

Suponha, por contradição, que não existe  $\delta$  satisfazendo (4.23). Dado  $x \in \mathbb{R}^n$  denote por  $x_+$  o possível *sucessor* de  $x$  por uma iteração do Algoritmo 4.1. Sob a hipótese de que (4.23) não é satisfeita, temos que existe uma sequência  $\{z_\ell\}$  tal que  $\lim_{\ell \rightarrow \infty} z_\ell = x_*$  e

$$\|z_\ell - (z_\ell)_+\| > \epsilon/2, \quad \text{para todo } \ell = 0, 1, 2, \dots$$

De (4.21), segue que, para todo  $\ell \in \mathbb{N}$ , existe  $j_\ell$  tal que  $f_{j_\ell}(z_\ell) = f_{min}(z_\ell)$  e  $\|\nabla f_{j_\ell}(z_\ell)\| > \zeta$ , para algum  $\zeta > 0$ . Tome  $j$  tal que  $j_\ell = j$  infinitas vezes. Então,  $j \in I(z_\ell)$ , para todo  $\ell$  e  $\|\nabla f_j(z_\ell)\| > \zeta$ . Isto implica que  $j \in I(x_*)$  e  $\|\nabla f_j(x_*)\| \neq 0$ . Isto é contraditório, já que,  $x_*$  é um minimizador local e, conseqüentemente,  $x_*$  é um ponto crítico forte. Portanto, (4.23) é verdadeira.

Seja  $c$  o mínimo de  $f_{min}(x)$  no conjunto definido por  $\delta \leq \|x - x_*\| \leq \epsilon$ . Defina  $\delta_1 \in (0, \delta)$  de forma que:

$$\|x - x_*\| \leq \delta_1 \Rightarrow f_{min}(x) < c.$$

Vamos provar por indução que, tomando  $\|x_0 - x_*\| \leq \delta_1$ , temos  $\|x_k - x_*\| \leq \epsilon/2$  e  $f(x_k) < c$  para todo  $k$ . Pela definição de  $\delta_1$ , a afirmação é válida para  $k = 0$ . Para o passo indutivo, observe que, por (4.23), temos  $\|x_{k+1} - x_*\| \leq \epsilon$ . Mas, como  $f(x_{k+1}) < f(x_k) < c$  segue da definição de  $c$  que  $\|x_{k+1} - x_*\| \leq \epsilon/2$ .

Portanto, a sequência toda está contida em  $B(x_*, \epsilon/2)$ . Como  $x_*$  é o único ponto crítico fraco nesta bola, segue do Teorema 4.6 que a sequência toda converge para  $x_*$ , como queríamos demonstrar.  $\square$

**Hipótese B2.** Na implementação do Algoritmo 4.1 temos:

- $\alpha \in \left(0, \frac{1}{2}\right)$ .
- A direção  $d_k$  é uma solução de

$$B_k d = -\nabla f_{\nu(k)}(x_k) + r_k, \quad (4.24)$$

onde  $B_k \in \mathbb{R}^{n \times n}$  é simétrica definida positiva e

$$\|r_k\| \leq \eta_k \|\nabla f_{\nu(k)}(x_k)\|. \quad (4.25)$$

- Se

$$f_{min}(x_k + d_k) \leq f_{min}(x_k) + \alpha \nabla f_{\nu(k)}(x_k)^T d_k,$$

escolhemos  $t_k = 1$  e  $x_{k+1} = x_k + d_k$ .

- O conjunto  $\{\|B_k^{-1}\|, k \in \mathbb{N}\}$  é limitado.

A seguir, faremos alguns comentários sobre as características do Algoritmo 4.1, implementado de forma a satisfazer a Hipótese **B2**.

1. Restringimos o coeficiente  $\alpha$  ao intervalo  $(0, 1/2)$  porque isto favorece a aceitação do passo  $t_k = 1$ , como poderá ser observado na demonstração do teorema.
2. A direção  $d_k$  provém da resolução inexata de uma equação quase-Newton. As matrizes  $B_k$  são aproximações definidas positivas da Hessiana.
3. Quando  $x_k + d_k$  satisfaz a condição de decréscimo suficiente aceitamos o passo  $t_k = 1$  e o ponto  $x_{k+1}$  é definido por  $x_k + d_k$  aumentando, assim, a probabilidade de realizarmos um passo do tipo Newton.
4. Se  $\|B_k\| < \frac{1}{\beta}$ , a condição  $\|d_k\| \geq \beta \|\nabla f_{\nu(k)}(x_k)\|$  é satisfeita. Além disso, se o número de condição  $\|B_k\| \|B_k^{-1}\|$  é menor ou igual a  $\frac{1}{\theta}$ , a condição do ângulo (4.6) é satisfeita. Claramente, é sempre possível escolher uma matriz  $B_k$  que satisfaça ambos os requerimentos.

O Teorema 4.9 completa a teoria de convergência do Algoritmo 4.1. Supondo que as Hipóteses **B1** e **B2** são satisfeitas, mostraremos que se a sequência  $\{x_k\}$  converge para um minimizador local tal que todas as Hessianas relevantes são definidas positivas e as matrizes  $B_k$  satisfazem uma condição de Dennis-Moré, a convergência é superlinear e, eventualmente,  $t_k = 1$ .

**Teorema 4.9.** *Suponha que:*

1. A sequência  $\{x_k\}$  é gerada pelo Algoritmo 4.1 e que as Hipóteses **B1** and **B2** são satisfeitas;
2.  $x_*$  é um minimizador local;
3.  $f_i$  admite segunda derivada contínua em uma vizinhança de  $x_*$ , para todo  $i \in I(x_*)$ ;
4.  $\nabla^2 f_i(x_*) > 0$ , para todo  $i \in I(x_*)$ ;
5.  $\lim_{k \rightarrow \infty} x_k = x_*$ ;
6. A condição Dennis-Moré

$$\lim_{k \rightarrow \infty} \frac{\|[B_k - \nabla^2 f_{\nu(k)}(x_k)]d_k\|}{\|d_k\|} = 0 \quad (4.26)$$

e a condição de Newton-Inexata

$$\lim_{k \rightarrow \infty} \eta_k = 0 \quad (4.27)$$

são satisfeitas.

Então,

- Existe  $k_0 \in \mathbb{N}$  tal que  $t_k = 1$ , para todo  $k \geq k_0$ .
- A sequência  $\{x_k\}$  converge superlinearmente para  $x_*$ .

*Prova.* Pela continuidade das funções  $f_i$ , existe  $k_1 \in \mathbb{N}$  tal que, para todo  $k \geq k_1$ ,

$$I(x_k) \subset I(x_*).$$

Pela fórmula de Taylor, para todo  $k \geq k_1$ , temos que:

$$\begin{aligned} f_{\nu(k)}(x_k + d_k) - f_{\nu(k)}(x_k) - \alpha d_k^T \nabla f_{\nu(k)}(x_k) &= (1 - \alpha) d_k^T \nabla f_{\nu(k)}(x_k) + \frac{1}{2} d_k^T \nabla^2 f_{\nu(k)}(x_k) d_k + o(\|d_k\|^2) \\ &= (1 - \alpha) d_k^T [\nabla f_{\nu(k)}(x_k) + \nabla^2 f_{\nu(k)}(x_k) d_k] + \left(\alpha - \frac{1}{2}\right) d_k^T \nabla^2 f_{\nu(k)}(x_k) d_k + o(\|d_k\|^2). \end{aligned}$$

Mas,  $B_k d_k + \nabla f_{\nu(k)}(x_k) = r_k$  e, por (4.6), (4.25) e (4.27),  $\|r_k\| = o(\|\nabla f_{\nu(k)}(x_k)\|) = o(\|d_k\|)$ . Portanto,

$$f_{\nu(k)}(x_k + d_k) - f_{\nu(k)}(x_k) - \alpha d_k^T \nabla f_{\nu(k)}(x_k)$$

$$\begin{aligned}
&= (1 - \alpha)(d_k)^T r_k + (1 - \alpha)d_k^T [\nabla^2 f_{\nu(k)}(x_k) - B_k]d_k + \left(\alpha - \frac{1}{2}\right)d_k^T \nabla^2 f_{\nu(k)}(x_k)d_k + o(\|d_k\|^2) \\
&= (1 - \alpha)d_k^T [\nabla^2 f_{\nu(k)}(x_k) - B_k]d_k + \left(\alpha - \frac{1}{2}\right)d_k^T \nabla^2 f_{\nu(k)}(x_k)d_k + o(\|d_k\|^2).
\end{aligned}$$

Mas, por (4.26),

$$(1 - \alpha)d_k^T [\nabla^2 f_{\nu(k)}(x_k) - B_k]d_k = o(\|d_k\|^2),$$

e portanto,

$$f_{\nu(k)}(x_k + d_k) - f_{\nu(k)}(x_k) - \alpha d_k^T \nabla f_{\nu(k)}(x_k) = \left(\alpha - \frac{1}{2}\right)d_k^T \nabla^2 f_{\nu(k)}(x_k)d_k + o(\|d_k\|^2). \quad (4.28)$$

Seja  $\mu > 0$  uma cota inferior para os autovalores de  $\nabla^2 f_i(x_*)$ ,  $i \in I(x_*)$ . Então, existe  $k_2 > k_1$  tal que  $\mu/2$  é uma cota inferior para os autovalores de  $\nabla^2 f_{\nu(k)}(x_k)$ , para todo  $k \geq k_2$ . Logo, para todo  $k \geq k_2$ , temos:

$$\frac{d_k^T \nabla^2 f_{\nu(k)}(x_k)d_k}{\|d_k\|^2} \geq \mu/2.$$

Como  $\alpha < 1/2$ , por (4.28), temos:

$$\frac{f_{\nu(k)}(x_k + d_k) - f_{\nu(k)}(x_k) - \alpha d_k^T \nabla f_{\nu(k)}(x_k)}{\|d_k\|^2} \leq \left(\alpha - \frac{1}{2}\right)\frac{\mu}{2} + \frac{o(\|d_k\|^2)}{\|d_k\|^2}, \quad (4.29)$$

para  $k \geq k_2$ . Mas, como  $\{\|B_k^{-1}\|, k \in \mathbb{N}\}$  é limitado e  $\nabla f_{\nu(k)}(x_k) \rightarrow 0$ , por (4.24) e (4.25) temos que  $\|d_k\| \rightarrow 0$ . Logo, aplicando o limite em (4.29) para  $k \rightarrow \infty$ , obtemos

$$f_{\nu(k)}(x_k + d_k) - f_{\nu(k)}(x_k) - \alpha d_k^T \nabla f_{\nu(k)}(x_k) \leq 0, \quad (4.30)$$

para  $k$  suficientemente grande. Logo, pela definição do algoritmo, existe  $k_0 \in \mathbb{N}$  tal que  $t_k = 1$  para todo  $k \geq k_0$ . Portanto, a primeira parte da tese está provada.

Pela primeira parte da tese e pela Hipótese **B2** temos que:

$$x_{k+1} - x_k = d_k \text{ para todo } k \geq k_0.$$

Então, pela fórmula de Taylor

$$\begin{aligned}
\nabla f_{\nu(k)}(x_{k+1}) &= \nabla f_{\nu(k)}(x_k) + \nabla^2 f_{\nu(k)}(x_k)d_k + o(\|d_k\|) \\
&= B_k d_k + \nabla f_{\nu(k)}(x_k) + [\nabla^2 f_{\nu(k)}(x_k) - B_k]d_k + o(\|d_k\|) \\
&= r_k + [\nabla^2 f_{\nu(k)}(x_k) - B_k]d_k + o(\|d_k\|).
\end{aligned}$$

Assim como na primeira parte da demonstração, temos que  $\|r_k\| = o(\|d_k\|)$ . Portanto,

$$\nabla f_{\nu(k)}(x_{k+1}) = [\nabla^2 f_{\nu(k)}(x_k) - B_k]d_k + o(\|d_k\|).$$

Logo, por (4.26),

$$\lim_{k \rightarrow \infty} \frac{\|\nabla f_{\nu(k)}(x_{k+1})\|}{\|x_{k+1} - x_k\|} = 0.$$

Pela continuidade e não-singularidade da Hessiana em  $x_*$ , deduzimos que:

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x_*\|}{\|x_{k+1} - x_k\|} = 0.$$

Claramente, isto implica que:

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x_*\|}{\|x_{k+1} - x_*\| + \|x_k - x_*\|} = 0.$$

Portanto, após breve manipulação algébrica, obtemos a convergência superlinear da sequência  $\{x_k\}$ .  $\square$

### 4.2.2 Convergência para pontos críticos fortes

Na seção anterior introduzimos o Algoritmo 4.1 que, resumindo, converge para pontos críticos fracos. O Algoritmo 4.1 pode convergir para pontos que não são pontos críticos fortes e que, claro, estão longe de ser minimizadores do problema LOVO irrestrito. Por exemplo, considere o problema definido por  $f_1(x) = x$ ,  $f_2(x) = x^2$ ,  $m = 2$ . Para todo  $x \in (0, 1)$  temos  $f_{min}(x) = x^2$ . Portanto, é fácil definir uma sequência  $x_k \in (0, 1)$  gerada pelo Algoritmo 4.1 que converge para 0. Claramente, 0 é um ponto crítico fraco, mas não é um ponto crítico forte. O objetivo desta seção é introduzir e analisar um algoritmo para o LOVO irrestrito que converge para pontos críticos fortes.

**Algoritmo 4.2.** Seja  $x_0 \in \mathbb{R}^n$  um ponto inicial arbitrário. Sejam

$$\theta \in (0, 1), \alpha \in (0, 1), M > 1, \beta > 0, t_{one} > 0, \varepsilon > 0, \delta > 0$$

parâmetros do algoritmo. Dado  $x_k \in \mathbb{R}^n$ , os passos para obter a nova iteração  $x_{k+1}$  são:

**Passo 1** Se  $\|\nabla f_i(x_k)\| = 0$  para todo  $i \in I(x_k)$ , parar. Se  $\|\nabla f_i(x_k)\| > \delta$  para todo  $i \in I(x_k)$ , escolha  $i \in I(x_k)$  e defina  $J_k = \{i\}$ . Caso contrário, defina:

$$J_k = \{j \in \{1, \dots, m\} \mid f_j(x_k) \leq f_{min}(x_k) + \varepsilon \text{ e } \nabla f_j(x_k) \neq 0\}.$$

**Passo 2** Para todo  $i \in J_k$ , calcule  $d_k^i \in \mathbb{R}^n$  tal que

$$\nabla f_i(x_k)^T d_k^i \leq -\theta \|d_k^i\| \|\nabla f_i(x_k)\| \quad \text{e} \quad \|d_k^i\| \geq \beta \|\nabla f_i(x_k)\|. \quad (4.31)$$

**Passo 3** Para todo  $i \in J_k$ , calcule  $t_k^i > 0$  tal que

$$f_i(x_k + t_k^i d_k^i) \leq f_i(x_k) + \alpha t_k^i \nabla f_i(x_k)^T d_k^i \quad (4.32)$$

e

$$\left[ t_k^i \geq t_{one} \quad \text{ou} \quad f_i(x_k + \bar{t}_k d_k^i) > f_i(x_k) + \alpha \bar{t}_k \nabla f_i(x_k)^T d_k^i \quad \text{para algum} \quad \bar{t}_k \leq M t_k^i \right]. \quad (4.33)$$

**Passo 4** Calcule  $x_{k+1} \in \mathbb{R}^n$  tal que

$$f_{min}(x_{k+1}) \leq \min_{i \in J_k} \{f_i(x_k + t_k^i d_k^i)\}. \quad (4.34)$$

No Algoritmo 4.2, se  $\|\nabla f_i(x^k)\| > \delta$ , para todo  $i \in I(x_k)$ , realizamos uma iteração do tipo Algoritmo 4.1. Se, para algum  $i \in I(x_k)$  a norma do gradiente é menor do que  $\delta$  calculamos direções de descida para todas as funções  $f_i$  tais que  $f_i(x_k) \approx f_{min}(x_k)$  (com precisão  $\varepsilon$ ). Então, realizamos buscas lineares ao longo de todas estas direções e terminamos definindo  $x_{k+1}$  de forma que seja no mínimo tão bom quanto o melhor dos pontos obtidos nas buscas lineares. A seguir, mostraremos que o Algoritmo 4.2 está bem definido e pára apenas em pontos críticos forte.

**Teorema 4.10.** *O Algoritmo 4.2 está bem definido e pára em  $x_k$  se, e somente se,  $x_k$  é um ponto crítico forte. Além disso, se o algoritmo não pára em  $x_k$ ,*

$$f_{min}(x_{k+1}) < f_{min}(x_k), \quad (4.35)$$

para todo  $k = 0, 1, 2, \dots$

*Prova.* Se  $x_k$  é um ponto crítico forte, o Passo 1 garante que o algoritmo pára em  $x_k$ . Agora, temos que mostrar que, se  $x_k$  não é um ponto crítico forte, a iteração que define o Algoritmo 4.2 pode ser completada em tempo finito e que  $x_{k+1}$  satisfaz (4.35).

Se  $x_k$  não é um ponto crítico forte, existe  $i \in I(x_k)$  tal que  $\|\nabla f_i(x_k)\| \neq 0$ . Logo, o conjunto  $J_k$  é não-vazio e, por construção, para todo  $i \in J_k$ ,  $\nabla f_i(x_k) \neq 0$ . Portanto,

como na prova do Teorema 4.5, para todo  $i \in J_k$  e  $t$  suficientemente pequeno, a condição de decréscimo suficiente

$$f_i(x_k + td_k^i) \leq f_i(x_k) + \alpha t \nabla f_i(x_k)^T d_k^i$$

é satisfeita. Assim, escolhendo  $t_k^i$  como sendo o primeiro número da sequência

$$\{t_{one}, t_{one}/M, t_{one}/M^2, \dots\}$$

que satisfaz (4.32), as condições (4.32) e (4.33) são satisfeitas. Logo, o algoritmo está bem definido. Agora, considere  $i \in I(x_k)$  tal que  $\nabla f_i(x_k) \neq 0$ . Como  $i \in J_k$  temos:

$$f_i(x_k + t_k^i d_k^i) \leq f_i(x_k) + \alpha t_k^i \nabla f_i(x_k)^T d_k^i = f_{min}(x_k) + \alpha t_k^i \nabla f_i(x_k)^T d_k^i < f_{min}(x_k).$$

Portanto, (4.35) segue de (4.34).  $\square$

No Lema 4.11 provamos que uma subsequência convergente gerada pelo Algoritmo 4.2 tem, no máximo, um número finito de iterações do tipo Algoritmo 4.1.

**Lema 4.11.** *Suponha que  $\{x_k\}$  é uma sequência gerada pelo Algoritmo 4.2 e que  $K$  é um conjunto infinito de índices tal que  $\lim_{k \in K} x_k = x_*$ . Então, para todo  $k \in K$  suficientemente grande,*

$$\min_{i \in I(x_k)} \{\|\nabla f_i(x_k)\|\} \leq \delta.$$

*Prova.* Suponha que a tese não é verdadeira. Então, existe  $K_1$ , uma subsequência infinita de índices de  $K$ , tal que

$$\|\nabla f_i(x_k)\| > \delta, \text{ para todo } i \in I(x_k), k \in K_1. \quad (4.36)$$

Defina  $K_1 = \{k_0, k_1, k_2, k_3, \dots\}, k_j < k_{j+1}$ , para todo  $j$ , e

$$y_j = x_{k_j}, \text{ para todo } j = 0, 1, 2, \dots$$

Por (4.36) e pela definição de  $J_k$ , nestes casos, temos que  $\{y_j\}$  é uma sequência gerada por iterações do tipo Algoritmo 4.1. Portanto, existe  $i \in \{1, \dots, m\}$  tal que  $J_{k_j} = \{i\} \subset I(x_{k_j})$  infinitas vezes. Pelo Teorema 4.6,  $i \in I(x_*)$  e  $\nabla f_i(x_*) = 0$ . Portanto, pela continuidade de  $\nabla f_i$ ,  $\lim_{j \rightarrow \infty} \|\nabla f_i(x_{k_j})\| = 0$ . Isto implica que a desigualdade (4.36) é falsa.  $\square$

No Teorema 4.12, provamos que o Algoritmo 4.2 necessariamente converge para pontos críticos fortes.

**Teorema 4.12.** *Se  $x_*$  é um ponto limite da sequência gerada pelo Algoritmo 4.2, então  $x_*$  é um ponto crítico forte. Além disso, dado  $\epsilon > 0$ , existe  $k \in \mathbb{N}$  tal que*

$$\|\nabla f_i(x_k)\| \leq \epsilon, \quad \text{para todo } i \in I(x_k).$$

*Prova.* Seja  $K = \{k_0, k_1, k_2, \dots\}$  um conjunto infinito de índices tal que

$$\lim_{k \in K} x_k = x_*.$$

Pelo Lema 4.11 e pela definição do Algoritmo 4.2, podemos supor, sem perda de generalidade, que

$$J_k = \{j \in \{1, \dots, m\} \mid f_j(x_k) \leq f_{\min}(x_k) + \epsilon \text{ e } \nabla f_j(x_k) \neq 0\},$$

para todo  $k \in K$ .

Suponha que  $i \in I(x_*)$ . Nossa meta é provar que  $\nabla f_i(x_*) = 0$ . Claramente,  $f_i(x_*) = f_{\min}(x_*)$ . Logo, pela continuidade das funções  $f_i$  e  $f_{\min}$ ,

$$f_i(x_k) \leq f_{\min}(x_k) + \epsilon, \quad (4.37)$$

para  $k \in K$  suficientemente grande. Por continuidade, se  $\nabla f_i(x_k)$  se anula infinitas vezes para  $k \in K$ , nada temos a provar. Caso contrário, podemos supor, sem perda de generalidade, que  $\nabla f_i(x_k) \neq 0$  para todo  $k \in K$ . Portanto, por (4.37),  $i \in J_k$ , para todo  $k \in K$ . Além disso,

$$\lim_{k \in K} f_i(x_k) - f_{\min}(x_k) = f_i(x_*) - f_{\min}(x_*) = 0. \quad (4.38)$$

Pela definição do algoritmo, para  $j$  suficientemente grande temos:

$$\begin{aligned} f_{\min}(x_{k_{j+1}}) &< f_{\min}(x_{k_j+1}) \leq f_i(x_{k_j} + t_{k_j}^i d_{k_j}^i) \leq f_i(x_{k_j}) + \alpha t_{k_j}^i \nabla f_i(x_{k_j})^T d_{k_j}^i \\ &= f_{\min}(x_{k_j}) + [f_i(x_{k_j}) - f_{\min}(x_{k_j})] + \alpha t_{k_j}^i \nabla f_i(x_{k_j})^T d_{k_j}^i. \end{aligned} \quad (4.39)$$

Por (4.31),  $\alpha t_{k_j}^i \nabla f_i(x_{k_j})^T d_{k_j}^i < 0$ . Suponha, por um momento, que existe  $c > 0$ ,  $j_0 \in \mathbb{N}$ , tal que

$$\alpha t_{k_j}^i \nabla f_i(x_{k_j})^T d_{k_j}^i < -c, \quad (4.40)$$

para todo  $j \geq j_0$ . Mas, por (4.38), existe  $j_1 \geq j_0$  tal que

$$f_i(x_{k_j}) - f_{\min}(x_{k_j}) < c/2, \quad (4.41)$$

para todo  $j \geq j_1$ . Logo, por (4.39), (4.40) e (4.41), temos que

$$f_{\min}(x_{k_{j+1}}) \leq f_{\min}(x_{k_j}) - c/2,$$

para todo  $j \geq j_1$ . Isto implica que  $\lim_{j \rightarrow \infty} f_{\min}(x_{k_j}) = -\infty$  e contradiz o fato de que, por continuidade,  $f_{\min}(x_{k_j}) \rightarrow f_{\min}(x_*)$ . Portanto, a existência de  $c$  e  $j_0$  satisfazendo (4.40) é impossível. Isto implica que existe  $K_1$ , uma subsequência infinita de índices de  $K$ , tal que

$$\lim_{k \in K_1} \alpha t_k^i \nabla f_i(x_k)^T d_k^i = 0.$$

Portanto, por (4.31),

$$\lim_{k \in K_1} t_k^i \|\nabla f_i(x_k)\| \|d_k^i\| = 0.$$

O resto da demonstração é similar à prova do Teorema 4.6. Se, para alguma subsequência  $K_2 \subset K_1$ ,  $\lim_{k \in K_2} \nabla f_i(x_k) = 0$ , nada temos a provar. Logo, vamos supor que  $\|\nabla f_i(x_k)\| > \zeta$ , para todo  $k \in K_1$  e algum  $\zeta > 0$ . Neste caso,

$$\lim_{k \in K} t_k^i \|d_k^i\| = 0. \quad (4.42)$$

Se, para alguma subsequência  $K_3 \subset K_1$ ,  $\lim_{k \in K_3} \|d_k^i\| = 0$ , então, por (4.31),  $\lim_{k \in K_3} \|\nabla f_i(x_k)\| = 0$  e, portanto,  $\nabla f_i(x_*) = 0$ . Logo, precisamos apenas considerar o caso em que  $\|d_k^i\| > \zeta$ , para todo  $k \in K_1$  e algum  $\zeta > 0$ . Neste caso, por (4.42),

$$\lim_{k \in K_1} t_k^i = 0.$$

Portanto, sem perda de generalidade, podemos supor que  $t_k^i < t_{one}$ , para todo  $k \in K_1$ . Então, por (4.33), existe  $\bar{t}_k^i \leq M t_k^i$ ,  $s_k = \bar{t}_k^i d_k^i$  tal que

$$f_i(x_k + s_k) > f_i(x_k) + \alpha \nabla f_i(x_k)^T s_k, \quad \text{para todo } k \in K_1 \quad (4.43)$$

e, por (4.42),

$$\lim_{k \in K_1} \|s_k\| = 0. \quad (4.44)$$

Logo, por (4.43) e pelo Teorema do Valor Médio, existe  $\xi_k \in [0, 1]$  tal que

$$\nabla f_i(x_k + \xi_k s_k)^T s_k = f_i(x_k + s_k) - f_i(x_k) > \alpha \nabla f_i(x_k)^T s_k, \quad (4.45)$$

para todo  $k \in K_1$ . Além disso, por (4.31),

$$\frac{\nabla f_i(x_k)^T s_k}{\|s_k\|} \leq -\theta \|\nabla f_i(x_k)\|, \quad (4.46)$$

para todo  $k \in K_1$ . Seja  $K_4$  uma subsequência de  $K_1$  tal que

$$\lim_{k \in K_4} \frac{s_k}{\|s_k\|} = s.$$

Por (4.44), dividindo ambos os membros de (4.45) por  $\|s_k\|$  e tomando limites para  $k \in K_4$ , obtemos:

$$\nabla f_i(x_*)^T s \geq \alpha \nabla f_i(x_*)^T s.$$

Como  $\alpha < 1$  e  $\nabla f_i(x_k)^T d_k < 0$ , para todo  $k$ , isto implica que  $\nabla f_i(x_*)^T s = 0$ . Aplicando o limite em ambos os membros de (4.46) obtemos  $\|\nabla f_i(x_*)\| = 0$ .

Agora, vamos provar a segunda parte da teorema. Suponhamos por absurdo que a tese não é verdadeira, então existe  $K_5$ , um subconjunto infinito de  $K$ , e  $\epsilon > 0$  tal que para todo  $k \in K_5$  existe  $i \in I(x_k)$  tal que  $\|\nabla f_i(x_k)\| > \epsilon$ . Claramente, o mesmo índice  $i$  deve se repetir infinitas vezes, e, tomando limites, obtemos que  $i \in I(x_*)$  e  $\|\nabla f_i(x_*)\| \geq \epsilon$ . Isto contradiz a primeira parte do teorema.  $\square$

Para provar convergência local procedemos de forma similar aos passos seguidos no Algoritmo 4.1. A Hipótese **B3** diz que a distância entre duas iterações consecutivas é menor ou igual que a máxima norma dos gradientes em  $J_k$ . Isto é sempre possível se escolhermos  $d_k^i$  como sendo direções do tipo gradiente, Newton ou quase-Newton.

**Hipótese B3** Suponha que o Algoritmo 4.2 é implementado de forma que exista  $b > 0$  tal que

$$\|x_{k+1} - x_k\| \leq b \max\{\|\nabla f_i(x_k)\|, i \in J_k\}, \quad (4.47)$$

para todo  $k \in \mathbb{N}$ .

**Definição.** Dizemos que  $x_*$  é *fortemente isolado* se existe  $\epsilon > 0$  tal que, para todo  $x \in B(x_*, \epsilon) - \{x_*\}$ , existe  $i \in I(x)$  tal que  $\nabla f_i(x) \neq 0$ . Em outras palavras, existe uma pequena vizinhança de  $x_*$  que não contém pontos críticos fortes.

**Definição.** Dado  $a > 0$ , dizemos que  $x_*$  é *a-verticalmente isolado* se

$$f_i(x_*) > f_{\min}(x_*) + a,$$

para todo  $i \notin I(x_*)$ .

O Teorema 4.13 é similar ao Teorema 4.7 da seção anterior. Provamos que, se as hipóteses de isolamento são satisfeitas, um ponto limite da sequência gerada pelo Algoritmo 4.2 é necessariamente um ponto limite da sequência toda. Além disso, no Teorema 4.14 provamos que convergência para minimizadores locais estritos ocorre se o ponto inicial está suficientemente próximo de tal solução.

**Teorema 4.13.** *Suponha que  $x_*$  é fortemente isolado e  $a$ -verticalmente isolado com  $a > \varepsilon$ . Suponha que a sequência  $\{x_k\}$  é gerada pelo Algoritmo 4.2, a Hipótese **B3** é satisfeita e  $\lim_{k \in K} x_k = x_*$ , para algum conjunto infinito de índices  $K \subset \mathbb{N}$ . Então,  $x_*$  é um ponto crítico forte e*

$$\lim_{k \rightarrow \infty} x_k = x_*.$$

*Prova.* O fato de  $x_*$  ser um ponto crítico forte é uma consequência do Teorema 4.12. Pela hipótese de isolamento vertical, para  $k \in K$  suficientemente grande, temos que  $J_k \subset I(x_*)$ . Como  $\nabla f_i(x_*) = 0$ , para todo  $i \in I(x_*)$ , temos por (4.47) que:

$$\lim_{k \in K} \|x_{k+1} - x_k\| = 0. \quad (4.48)$$

Como  $x_*$  é fortemente isolado, existe  $\epsilon > 0$  tal que para todo  $x \in B(x_*, \epsilon) - \{x_*\}$  existe  $i \in I(x)$  tal que  $\nabla f_i(x) \neq 0$ .

Por (4.48) e pela hipótese do teorema, existe  $k_1 \in K$  tal que

$$\|x_{k+1} - x_k\| < \epsilon/2 \text{ e } \|x_k - x_*\| < \epsilon/2,$$

para todo  $k \in K, k \geq k_1$ .

Assim como no Teorema 4.7, defina:

$$C = \{x \in \mathbb{R}^n \mid \epsilon/2 \leq \|x - x_*\| \leq \epsilon\}.$$

Claramente,  $C$  é compacto e não contém pontos críticos fortes. Então, pelo Teorema 4.12,  $C$  não pode conter uma quantidade infinita de iterandos. Portanto, temos duas possibilidades:

1. Existe  $k_2 \in \mathbb{N}$  tal que  $\|x_k - x_*\| \leq \epsilon/2$ , para todo  $k \geq k_2$ .
2. Existem infinitas iterações  $k \geq k_1$ , tais que  $\|x_k - x_*\| \leq \epsilon/2$  e  $\|x_{k+1} - x_k\| > \epsilon/2$ .

No primeiro caso, como  $x_*$  é o único candidato a ponto limite na bola com raio  $\epsilon/2$  temos que a sequência  $\{x_k\}$  converge para  $x_*$ .

Analisemos o segundo caso. Defina  $K_1 \subset \mathcal{N}$  de forma que  $\|x_k - x_*\| \leq \epsilon/2$  e  $\|x_{k+1} - x_k\| > \epsilon/2$ , para todo  $k \in K_1$ . Como todas as iterações pertencem à bola de raio  $\epsilon/2$  e  $x_*$  é o único candidato a ponto limite nesta bola, segue que:

$$\lim_{k \in K_1} x_k = x_*.$$

Pela hipótese de isolamento vertical, temos que  $J_k \subset I(x_*)$  para  $k \in K_1$  suficientemente grande. Além disso,  $\nabla f_i(x_*) = 0$ , para todo  $i \in I(x_*)$ . Então, por (4.47),

$$\lim_{k \in K_1} \|x_{k+1} - x_k\| = 0,$$

contradizendo a hipótese  $\|x_{k+1} - x_k\| \geq \epsilon/2$ ,  $\forall k \in K_1$ . Isto significa que o segundo caso mencionado acima é impossível. Logo, a prova está completa.  $\square$

**Teorema 4.14.** *Suponha que  $x_*$  é um minimizador local estrito fortemente isolado e  $a$ -verticalmente isolado com  $a > \epsilon$ . Seja  $\{x_k\}$  uma sequência gerada por um Algoritmo do tipo 4.2 que satisfaça a Hipótese **B3**. Então, existe  $\epsilon > 0$  tal que  $\|x_0 - x_*\| \leq \epsilon$  implica que:*

$$\lim_{k \rightarrow \infty} x_k = x_*.$$

*Prova.* Defina  $\epsilon > 0$  de forma que  $x_*$  seja um minimizador global estrito de  $f_{min}$  na bola  $B(x_*, \epsilon)$  e que esta bola não contenha pontos críticos fortes além de  $x_*$ . Vamos provar que existe  $\delta \in (0, \epsilon/2)$  tal que

$$\|x_k - x_*\| \leq \delta \Rightarrow \|x_{k+1} - x_k\| \leq \epsilon/2. \quad (4.49)$$

De fato, como  $x_*$  é um ponto crítico forte,  $\nabla f_i(x_*) = 0$ , para todo  $i \in I(x_*)$ . Mas, a hipótese de isolamento vertical com  $a > \epsilon$  implica que existe uma vizinhança de  $x_*$  tal que  $J_k \subset I(x_*)$ . Então, pela continuidade dos gradientes e pela hipótese (4.47), obtemos (4.49).

O restante da demonstração é idêntico à prova do Teorema 4.14. Seja  $c$  o mínimo de  $f_{min}(x)$  no conjunto definido por  $\delta \leq \|x - x_*\| \leq \epsilon$ . Defina  $\delta_1 \in (0, \delta)$  de forma que:

$$\|x - x_*\| \leq \delta_1 \Rightarrow f_{min}(x) < c.$$

Vamos provar por indução que, tomando  $\|x_0 - x_*\| \leq \delta_1$ , temos  $\|x_k - x_*\| \leq \epsilon/2$  e  $f(x_k) < c$ , para todo  $k$ . Pela definição de  $\delta_1$  a afirmação é válida para  $k = 0$ . Para o passo indutivo, observe que, por (4.49), temos  $\|x_{k+1} - x_*\| \leq \epsilon$ . Mas, como  $f(x_{k+1}) < f(x_k) < c$  segue da definição de  $c$  que  $\|x_{k+1} - x_*\| \leq \epsilon/2$ .

Portanto, a sequência toda está contida em  $B(x_*, \epsilon/2)$ . Como  $x_*$  é o único ponto crítico forte nesta bola, o Teorema 4.12 implica que a sequência toda converge para  $x_*$ , como queríamos demonstrar.  $\square$

A hipótese de isolamento vertical é essencial para provar os Teoremas 4.13 e 4.14. De fato, considere o problema definido por  $f_1(x) = x^2$ ,  $f_2(x) = x + \epsilon/2$ , onde o isolamento vertical não se verifica. O ponto  $x_* = 0$  é o único minimizador local forte deste problema. Entretanto, para  $x_0$  suficientemente próximo de  $x_*$ ,  $J_0 = \{1, 2\}$ . Tomando  $d_0^2 = -1$  teremos  $f_{min}(x^1) < 0$  e, a convergência para 0 seria impossível. Então, a tese do Teorema 4.14 não vale neste caso.

A Hipótese **B4** especifica a implementação do Algoritmo 4.2 para que se tenha convergência local superlinear. Assim como no Algoritmo 4.1, vamos supor que as direções  $d_k^i$  são soluções inexatas de sistemas lineares de equações do tipo Newton. Para aumentar a probabilidade de se ter iterações do tipo Newton puro faz-se a escolha (4.53) abaixo. Isto será suficiente para provar, no Teorema 4.15 que, em condições similares àquelas do Teorema 4.9, a convergência é superlinear.

**Hipótese B4.** Na implementação do Algoritmo 4.2 temos:

- $\alpha \in \left(0, \frac{1}{2}\right)$ .
- Para todo  $i \in J_k$ , a direção  $d_k^i$  é uma solução de

$$B_k^i d = -\nabla f_i(x_k) + r_k, \quad (4.50)$$

onde  $B_k^i \in \mathbb{R}^{n \times n}$  é simétrica definida positiva e

$$\|r_k\| \leq \eta_k \|\nabla f_i(x_k)\|. \quad (4.51)$$

- Se

$$f_i(x_k + d_k^i) \leq f_i(x_k) + \alpha \nabla f_i(x_k)^T d_k^i,$$

então escolhemos  $t_k^i = 1$ .

- Se existe  $j \in J_k$  tal que  $t_k^j = 1$ , escolhemos

$$x_{k+1} = x_k + t_k^i d_k^i, \quad (4.52)$$

onde

$$f_i(x_k + t_k^i d_k^i) = \min\{f_j(x_k + t_k^j d_k^j), j \in J_k\}. \quad (4.53)$$

- Existe  $C > 0$  tal que, para todo  $k \in \mathbb{N}$ ,  $i \in J_k$ ,  $\|(B_k^i)^{-1}\| \leq C$  e a Hipótese **B3** é satisfeita.

Se  $\eta_k$  é suficientemente pequeno e  $\|B_k^i\| < \frac{1}{\beta}$ , a condição  $d_k^i \geq \beta \|\nabla f_i(x_k)\|$  é satisfeita. Além disso, se o número de condição  $\|B_k^i\| \|(B_k^i)^{-1}\|$  é menor ou igual a  $\frac{1}{\theta}$ , a condição do ângulo (4.31) é satisfeita. Claramente, é sempre possível escolher uma matriz  $B_k^i$  que satisfaça ambos os requerimentos.

**Teorema 4.15.** *Suponha que:*

1. A sequência  $\{x_k\}$  é gerada pelo Algoritmo 4.2 e a Hipótese **B4** é satisfeita;
2.  $x_*$  é um minimizador local;
3. Para todo  $i \in I(x_*)$ , a função  $f_i$  admite segunda derivada contínua em uma vizinhança de  $x_*$ ;
4.  $\nabla^2 f_i(x_*) > 0$ , para todo  $i \in I(x_*)$ ;
5.  $\lim_{k \rightarrow \infty} x_k = x_*$ ;
6. Para todo  $i \in I(x_*)$ , a condição de Dennis-Moré

$$\lim_{k \rightarrow \infty} \frac{\|[B_k^i - \nabla^2 f_i(x_k)]d_k^i\|}{\|d_k^i\|} = 0 \quad (4.54)$$

e a condição de Newton-Inexata

$$\lim_{k \rightarrow \infty} \eta_k = 0 \quad (4.55)$$

são satisfeitas.

Então,

- Existe  $k_0 \in \mathbb{N}$  tal que, para todo  $k \geq k_0$  e  $i \in I(x_*)$ , temos que  $i \in J_k$  e  $t_k^i = 1$ .

- Se  $i \in J_k$  é tal que

$$f_i(x_k + t_k^i d_k^i) = \min\{f_j(x_k + t_k^j d_k^j), j \in J_k\}, \quad (4.56)$$

para infinitos índices  $k$ , então  $i \in I(x_*)$ .

- Existe  $k_1 \in \mathbb{N}$  tal que, para todo  $k \geq k_1$ , existe  $\iota(k) \in J_k \cap I(x_*)$  tal que  $t_k^{\iota(k)} = 1$  e

$$x_{k+1} = x_k + d_k^{\iota(k)}. \quad (4.57)$$

- A sequência  $\{x_k\}$  converge superlinearmente para  $x_*$ .

*Prova.* Seja  $i \in I(x_*)$ . Como  $x_*$  deve ser um ponto crítico forte, temos que  $\nabla f_i(x_*) = 0$ . Entretanto, como  $\nabla^2 f_i(x_*)$  é definida positiva,  $\nabla f_i(x_k) \neq 0$  para  $k$  suficientemente grande. Como  $f_i(x_*) = f_{\min}(x_*)$ , pela continuidade de  $f_i$  e  $f_{\min}$  temos que  $f_i(x_k) \leq f_{\min}(x_k) + \varepsilon$ , para  $k$  suficientemente grande. Então, existe  $k'_0 \in \mathbb{N}$  tal que  $i \in J_k$ , para todo  $k \geq k'_0$ .

Pela fórmula de Taylor, para todo  $k \geq k'_0$ , temos que:

$$\begin{aligned} f_i(x_k + d_k^i) - f_i(x_k) - \alpha (d_k^i)^T \nabla f_i(x_k) &= (1 - \alpha) (d_k^i)^T \nabla f_i(x_k) + \frac{1}{2} (d_k^i)^T \nabla^2 f_i(x_k) (d_k^i) + o(\|d_k^i\|^2) \\ &= (1 - \alpha) (d_k^i)^T [\nabla f_i(x_k) + \nabla^2 f_i(x_k) d_k^i] + (\alpha - \frac{1}{2}) (d_k^i)^T \nabla^2 f_i(x_k) d_k^i + o(\|d_k^i\|^2). \end{aligned}$$

Por (4.31), (4.51) e (4.55) temos que  $\|r_k\| = o(\|d_k^i\|)$ . Portanto,

$$\begin{aligned} f_i(x_k + d_k^i) - f_i(x_k) - \alpha (d_k^i)^T \nabla f_i(x_k) \\ = (1 - \alpha) (d_k^i)^T [\nabla^2 f_i(x_k) - B_k^i] d_k^i + (\alpha - \frac{1}{2}) (d_k^i)^T \nabla^2 f_i(x_k) d_k^i + o(\|d_k^i\|^2). \end{aligned}$$

Mas, por (4.54),

$$(1 - \alpha) (d_k^i)^T [\nabla^2 f_i(x_k) - B_k^i] d_k^i = o(\|d_k^i\|^2),$$

portanto,

$$f_i(x_k + d_k^i) - f_i(x_k) - \alpha (d_k^i)^T \nabla f_i(x_k) = \left(\alpha - \frac{1}{2}\right) (d_k^i)^T \nabla^2 f_i(x_k) d_k^i + o(\|d_k^i\|^2). \quad (4.58)$$

Seja  $\mu > 0$  uma cota inferior para os autovalores de  $\nabla^2 f_i(x_*)$ . Então, existe  $k_2 > k'_0$  tal que  $\mu/2$  é uma cota inferior para os autovalores de  $\nabla^2 f_i(x_k)$ , para todo  $k \geq k_2$ . Então, para todo  $k \geq k_2$ , temos:

$$\frac{(d_k^i)^T \nabla^2 f_i(x_k) d_k^i}{\|d_k^i\|^2} \geq \mu/2.$$

Como  $\alpha < 1/2$ , por (4.58), temos que

$$\frac{f_i(x_k + d_k^i) - f_i(x_k) - \alpha(d_k^i)^T \nabla f_i(x_k)}{\|d_k^i\|^2} \leq \left(\alpha - \frac{1}{2}\right) \frac{\mu}{2} + \frac{o(\|d_k^i\|^2)}{\|d_k^i\|^2}, \quad (4.59)$$

para  $k \geq k_2$ . Mas, como  $\|(B_k^i)^{-1}\| \leq C$ , para todo  $k$  e  $\nabla f_i(x_*) = 0$ , temos que  $\|d_k^i\| \rightarrow 0$ . Então, aplicando o limite em (4.59) para  $k \rightarrow \infty$ , obtemos

$$f_i(x_k + d_k^i) - f_i(x_k) - \alpha(d_k^i)^T \nabla f_i(x_k) \leq 0, \quad (4.60)$$

para  $k$  suficientemente grande. Então, pela Hipótese **B4**, existe  $k_0 \geq k_2$  tal que  $t_k^i = 1$ , para todo  $k \geq k_0$ . Portanto, a primeira parte da prova está completa.

Vamos provar a segunda parte da tese. Pela primeira parte da demonstração, para  $k$  suficientemente grande, escolhemos  $x_{k+1}$  usando (4.52) e (4.53). Suponha que (4.56) vale para infinitos índices  $k \in K$ . Então, para todo  $k \in K$ ,

$$f_i(x_{k+1}) = f_i(x_k + t_k^i d_k^i) \leq f_{\min}(x_k).$$

Tomando o limite para  $k \in K$ , obtemos  $f_i(x_*) = f_{\min}(x_*)$ . Logo,  $i \in I(x_*)$ .

A terceira parte da demonstração é uma consequência das duas primeiras. Para  $k$  suficientemente grande,  $J_k \cap I(x_*) \neq \emptyset$ . Portanto, pela primeira parte da tese, para  $k$  suficientemente grande, existe  $i$  tal que  $t_k^i = 1$ . Então, por (4.52) e pela segunda parte da tese,  $x_{k+1} = x_k + t_k^{\iota(k)} d_k^{\iota(k)}$  e  $\iota(k) \in I(x_*)$ , para todo  $k$  suficientemente grande. Então, novamente, pela primeira parte da demonstração, obtemos (4.57).

Agora, estamos aptos a provar a última parte do teorema. Como na primeira parte da prova, por (4.31), (4.51) e (4.55) temos que  $\|r_k\| = o(\|\nabla f_{\iota(k)}(x_k)\|) = o(\|d_k^{\iota(k)}\|)$ . Então, pela fórmula de Taylor

$$\begin{aligned} \nabla f_{\iota(k)}(x_{k+1}) &= \nabla f_{\iota(k)}(x_k) + \nabla^2 f_{\iota(k)}(x_k) d_k^{\iota(k)} + o(\|d_k^{\iota(k)}\|) \\ &= B_k^{\iota(k)} d_k^{\iota(k)} + \nabla f_{\iota(k)}(x_k) + [\nabla^2 f_{\iota(k)}(x_k) - B_k^{\iota(k)}] d_k^{\iota(k)} + o(\|d_k^{\iota(k)}\|). \end{aligned}$$

Então, por (4.50), (4.51) e (4.55),

$$\nabla f_{\iota(k)}(x_{k+1}) = [\nabla^2 f_{\iota(k)}(x_k) - B_k^{\iota(k)}] d_k^{\iota(k)} + o(\|d_k^{\iota(k)}\|).$$

Logo, por (4.54),

$$\lim_{k \rightarrow \infty} \frac{\|\nabla f_{\iota(k)}(x_{k+1})\|}{\|x_{k+1} - x_k\|} = 0.$$

Mas, pela continuidade e não-singularidade de  $\nabla^2 f_i(x)$  em  $x_*$ , isto implica que:

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x_*\|}{\|x_{k+1} - x_k\|} = 0.$$

Daí, segue que:

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x_*\|}{\|x_{k+1} - x_*\| + \|x_k - x_*\|} = 0.$$

Portanto, após breve manipulação algébrica, obtemos a convergência superlinear.  $\square$

## 4.3 Problema LOVO com Restrições

Nesta seção, o conjunto viável  $\Omega$  não é mais todo o espaço  $\mathbb{R}^n$ . Vamos supor que o conjunto  $\Omega$  é descrito por um conjunto de equações e inequações e definiremos um algoritmo Lagrangiano Aumentado globalmente convergente para resolver o problema LOVO com restrições. Para este propósito, precisamos primeiramente, recordar um método Lagrangiano Aumentado apropriado para resolver problemas de otimização diferenciáveis com restrições.

### 4.3.1 Lagrangiano Aumentado Diferenciável

Considere o problema:

$$\text{Minimizar } f(x) \text{ s.a. } h(x) = 0, g(x) \leq 0, \quad (4.61)$$

onde  $f : \mathbb{R}^n \rightarrow \mathbb{R}, h : \mathbb{R}^n \rightarrow \mathbb{R}^{n_h}, g : \mathbb{R}^n \rightarrow \mathbb{R}^{n_g}$ . Vamos supor que  $f, h, g$  são continuamente diferenciáveis.

Para todo  $x \in \mathbb{R}^n, \rho \in \mathbb{R}_{++}, \lambda \in \mathbb{R}^{n_h}, \mu \in \mathbb{R}_+^{n_g}$  definimos o Lagrangiano Aumentado [27, 35, 37, 38]:

$$L(x, \lambda, \mu, \rho) = f(x) + \rho \left[ \left\| h(x) + \frac{\lambda}{\rho} \right\|^2 + \left\| \left( g(x) + \frac{\mu}{\rho} \right)_+ \right\|^2 \right]. \quad (4.62)$$

O Algoritmo 4.3 é um método Lagrangiano Aumentado para resolver o problema diferenciável (4.61). Essencialmente, este é um caso particular do algoritmo de Lagrangiano Aumentado, com restrições do nível inferior arbitrárias, descrito em [2] e implementado na página (web) **Tango**.

**Algoritmo 4.3.** Seja  $x_0 \in \mathbb{R}^n$  um ponto inicial arbitrário. Os parâmetros para a execução do algoritmo são:

$$\begin{aligned} \tau &\in [0, 1), \gamma > 1, \\ -\infty &< \bar{\lambda}_{\min} < \bar{\lambda}_{\max} < \infty, \\ 0 &\leq \bar{\mu}_{\max} < \infty, \\ \rho_1 &\in \mathbb{R}_{++}, \\ [\bar{\lambda}_1]_j &\in [\bar{\lambda}_{\min}, \bar{\lambda}_{\max}], \forall j = 1, \dots, n_h, \\ [\bar{\mu}_1]_j &\in [0, \bar{\mu}_{\max}], \forall j = 1, \dots, n_g. \\ \varepsilon_1 &> 0. \end{aligned}$$

**Passo 1** (Inicialização)

Defina  $k \leftarrow 1$ . Para  $j = 1, \dots, n_g$ , calcule

$$[\sigma_0]_j = \max\{g_j(x_0), 0\}.$$

**Passo 2** (Resolução do subproblema)

Calcule  $x_k \in \Omega$  tal que

$$\|\nabla L(x_k, \lambda_k, \mu_k, \rho_k)\|_\infty \leq \varepsilon_k.$$

**Passo 3** (Estimativa dos multiplicadores)

Para todo  $j = 1, \dots, n_h$ , calcule

$$[\lambda_{k+1}]_j = [\bar{\lambda}_k]_j + \rho_k h_j(x_k)$$

e

$$[\bar{\lambda}_{k+1}]_j \in [\bar{\lambda}_{\min}, \bar{\lambda}_{\max}].$$

Para todo  $j = 1, \dots, n_g$ , calcule

$$[\mu_{k+1}]_j = \max\{0, [\bar{\mu}_k]_j + \rho_k g_j(x_k)\},$$

$$[\sigma_k]_j = \max\left\{g_j(x_k), -\frac{[\bar{\mu}_k]_j}{\rho_k}\right\},$$

e

$$[\bar{\mu}_{k+1}]_j \in [0, \bar{\mu}_{\max}].$$

**Passo 4** (Atualização dos parâmetros de penalização)

Se

$$\max\{\|h(x_k)\|_\infty, \|\sigma_k\|_\infty\} \leq \tau \max\{\|h(x_{k-1})\|_\infty, \|\sigma_{k-1}\|_\infty\}, \quad (4.63)$$

defina

$$\rho_{k+1} \geq \rho_k. \quad (4.64)$$

Caso contrário, defina

$$\rho_{k+1} \geq \gamma\rho_k. \quad (4.65)$$

**Passo 5** (Começar uma nova iteração externa)Calcule  $\varepsilon_{k+1} > 0$ . Defina  $k \leftarrow k + 1$  e volte para o Passo 2.

A única diferença entre o Algoritmo 4.3 e o algoritmo introduzido em [2] (no caso em que restrições do nível inferior não estão presentes) está nas regras de atualização (4.64) e (4.65). Em [2], os autores definem  $\rho_{k+1} = \rho_k$  quando (4.63) é satisfeita e,  $\rho_{k+1} = \gamma\rho_k$ , caso contrário. Esta diferença não afeta as provas dos seguintes teoremas de convergência.

**Teorema 4.16.** *Suponha que  $\{x_k\}$  é uma sequência gerada pelo Algoritmo 4.3 com  $\varepsilon_k \rightarrow 0$  e que  $x_*$  é um ponto limite desta sequência. Então,  $x_*$  é um ponto estacionário do problema*

$$\text{Minimizar } \sum_{j=1}^{n_h} h_j(x)^2 + \sum_{j=1}^{n_g} \max\{0, g_j(x)\}^2.$$

*Prova.* Ver Teorema 3.1 de [2]. □

**Teorema 4.17.** *Suponha que  $\{x_k\}$  é uma sequência gerada pelo Algoritmo 4.3 com  $\varepsilon_k \rightarrow 0$ ,  $x_*$  é um ponto limite e que a condição de dependência linear positiva constante (CPLD) [11, 36] é satisfeita em  $x_*$ . Então,  $x_*$  é um ponto KKT de (4.61).*

*Prova.* Ver Teorema 3.2 de [2]. □

O resultado final, que fala da limitação dos parâmetros de penalização associados ao Algoritmo 4.3, é dado no Teorema 4.18. Uma hipótese crucial é o fato de que a precisão usada para resolver os subproblemas deve tender a zero mais rápido que a medida de viabilidade. Este tipo de exigência é usual em muitos métodos do tipo Lagrangiano

Aumentado e Multiplicadores [12, 13, 17, 18, 20, 21, 22, 26].

**Hipótese C1.** Suponha que

1. A sequência  $\{x_k\}$  é gerada pela aplicação do Algoritmo 4.3 ao problema (4.61) e

$$\lim_{k \rightarrow \infty} x_k = x_*.$$

2. Em (4.64), a regra  $\rho_{k+1} = \rho_k$  é aplicada.
3. O ponto  $x_*$  é viável ( $h(x_*) = 0, g(x_*) \leq 0$ ).
4. Os gradientes

$$\{\nabla h_j(x_*)\}_{j=1}^{n_h}, \{\nabla g_j(x_*)\}_{\{j \mid g_j(x_*)=0\}},$$

são linearmente independentes.

5. Tem-se complementaridade estrita em  $x_*$ . Isto significa que, se  $\mu_* \in \mathbb{R}_+^{n_g}$  é o vetor de multiplicadores de Lagrange correspondentes às restrições  $g(x) \leq 0$ , então:

$$g_j(x_*) = 0 \Rightarrow [\mu_*]_j > 0.$$

6. As funções  $f, h, g$  possuem segundas derivadas contínuas em uma vizinhança de  $x_*$ .
7. Defina o subespaço tangente  $T$  como sendo o conjunto formado por todo  $z \in \mathbb{R}^n$  tal que

$$\nabla h(x_*)^T z = 0,$$

$$\nabla g_j(x_*)^T z = 0,$$

para todo  $j$  tal que  $g_j(x_*) = 0$ . Então, para todo  $z \in T, z \neq 0$ ,

$$z^T [\nabla^2 f(x_*) + \sum_{i=1}^{n_h} [\lambda_*]_i \nabla^2 h_i(x_*) + \sum_{j=1}^{n_g} [\mu_*]_j \nabla^2 g_j(x_*)] z > 0.$$

**Teorema 4.18.** *Suponha que a Hipótese C1 é satisfeita. Adicionalmente, suponha que:*

1. *Existe uma sequência  $\eta_k \rightarrow 0$  tal que*

$$\varepsilon_k \leq \eta_k \max\{\|h(x_k)\|_\infty, \|\sigma_k\|_\infty\}, \forall k \in \mathbb{N}.$$

2.  $[\lambda_*]_j \in (\bar{\lambda}_{\min}, \bar{\lambda}_{\max}) \forall j = 1, \dots, n_h$  e  $[\mu_*]_j \in (\bar{\mu}_{\min}, \bar{\mu}_{\max}), \forall j = 1, \dots, n_g$ .
3.  $[\bar{\lambda}_{k+1}]_j$  é a projeção de  $[\lambda_{k+1}]_j$  em  $[\bar{\lambda}_{\min}, \bar{\lambda}_{\max}]$ ,  $\forall j = 1, \dots, n_h$  e  $[\bar{\mu}_{k+1}]_j$  é a projeção de  $[\mu_{k+1}]_j$  em  $[0, \bar{\mu}_{\max}]$ ,  $\forall j = 1, \dots, n_g$ ,  $k \in \mathbb{N}$ .

Então, a sequência  $\{\rho_k\}$  dos parâmetros de penalização é limitada.

*Prova.* Ver Teorema 4.3 de [2]. □

### 4.3.2 Método Lagrangiano Aumentado para o LOVO

Agora, temos condições de definir extensões naturais do Algoritmo 4.3 para o problema LOVO. Quando for preciso obter soluções de subproblemas de minimização ir-restrita, podemos utilizar os Algoritmos 4.1 ou 4.2.

Considere o problema:

$$\text{Minimizar } f_{\min}(x) \text{ s.a. } h(x) = 0, g(x) \leq 0, \quad (4.66)$$

onde  $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ , para todo  $i = 1, \dots, m$ ,  $h : \mathbb{R}^n \rightarrow \mathbb{R}^{n_h}$ ,  $g : \mathbb{R}^n \rightarrow \mathbb{R}^{n_g}$  e todas estas funções são diferenciáveis.

Assim como em (4.62), para todo  $x \in \mathbb{R}^n$ ,  $\rho \in \mathbb{R}_{++}$ ,  $\lambda \in \mathbb{R}^{n_h}$ ,  $\mu \in \mathbb{R}_+^{n_g}$  definimos o Lagrangiano Aumentado associado a  $f_i$  por:

$$L_i(x, \lambda, \mu, \rho) = f_i(x) + \rho \left[ \left\| h(x) + \frac{\lambda}{\rho} \right\|^2 + \left\| \left( g(x) + \frac{\mu}{\rho} \right)_+ \right\|^2 \right],$$

definimos o Lagrangiano Aumentado associado a  $f_{\min}$  por:

$$L_{\min}(x, \lambda, \mu, \rho) = f_{\min}(x) + \rho \left[ \left\| h(x) + \frac{\lambda}{\rho} \right\|^2 + \left\| \left( g(x) + \frac{\mu}{\rho} \right)_+ \right\|^2 \right],$$

e, ainda, definimos:

$$I_{\min}(x) = \{i \in \{1, \dots, m\} \mid L_i(x, \lambda, \mu, \rho) = L_{\min}(x, \lambda, \mu, \rho)\}.$$

**Algoritmo 4.4.** Seja  $x_0 \in \mathbb{R}^n$  um ponto inicial arbitrário. Os parâmetros para a execução do algoritmo são:

$$\begin{aligned} \tau &\in [0, 1), \gamma > 1, \\ -\infty &< \bar{\lambda}_{\min} < \bar{\lambda}_{\max} < \infty, \end{aligned}$$

$$0 \leq \bar{\mu}_{\max} < \infty,$$

$$\rho_1 \in \mathbb{R}_{++},$$

$$[\bar{\lambda}_1]_j \in [\bar{\lambda}_{\min}, \bar{\lambda}_{\max}], \quad \forall j = 1, \dots, n_h,$$

$$[\bar{\mu}_1]_j \in [0, \bar{\mu}_{\max}], \quad \forall j = 1, \dots, n_g.$$

$$\varepsilon_1 > 0.$$

**Passo 1** (Inicialização)

Defina  $k \leftarrow 1$ . Para  $j = 1, \dots, n_g$ , calcule

$$[\sigma_0]_j = \max\{g_j(x_0), 0\}.$$

**Passo 2** (Resolução do subproblema)

Calcule  $x_k \in \Omega$  tal que

$$\|\nabla L_i(x_k, \bar{\lambda}_k, \bar{\mu}_k, \rho_k)\|_{\infty} \leq \varepsilon_k, \quad (4.67)$$

para algum  $i \in I_{\min}(x_k)$ .

**Passo 3** (Estimativa dos multiplicadores)

Para todo  $j = 1, \dots, n_h$ , calcule:

$$[\lambda_{k+1}]_j = [\bar{\lambda}_k]_j + \rho_k h_j(x_k)$$

e

$$[\bar{\lambda}_{k+1}]_j \in [\bar{\lambda}_{\min}, \bar{\lambda}_{\max}].$$

Para todo  $j = 1, \dots, n_g$ , calcule:

$$[\mu_{k+1}]_j = \max\{0, [\bar{\mu}_k]_j + \rho_k g_j(x_k)\},$$

$$[\sigma_k]_j = \max\left\{g_j(x_k), -\frac{[\bar{\mu}_k]_j}{\rho_k}\right\},$$

e

$$[\bar{\mu}_{k+1}]_j \in [0, \bar{\mu}_{\max}].$$

**Passo 4** (Atualização dos parâmetros de penalização)

Se

$$\max\{\|h(x_k)\|_\infty, \|\sigma_k\|_\infty\} \leq \tau \max\{\|h(x_{k-1})\|_\infty, \|\sigma_{k-1}\|_\infty\},$$

defina

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

Caso contrário, defina

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

**Passo 5** (Começar uma nova iteração externa)Calcule  $\varepsilon_{k+1} > 0$ . Defina  $k \leftarrow k + 1$  e volte para o Passo 2.

Uma maneira óbvia de resolver (4.67) é aplicar o Algoritmo 4.1 ou o Algoritmo 4.2 ao problema

$$\text{Minimizar } L_{\min}(x, \bar{\lambda}_k, \bar{\mu}_k, \rho_k).$$

Ambos algoritmos garantem que um ponto satisfazendo (4.67) pode ser encontrado, contanto que a sequência gerada seja limitada. Por outro lado, a limitação das sequências geradas pelos Algoritmos 4.1 ou 4.2 pode ser garantida se tivermos relações apropriadas entre a função objetivo e as restrições.

No Teorema 4.19, provamos que o Algoritmo 4.4 encontra pontos estacionários da inviabilidade das restrições.

**Teorema 4.19.** *Suponha que  $\{x_k\}$  é uma sequência infinita gerada pelo Algoritmo 4.4 com  $\varepsilon_k \rightarrow 0$  e que  $x_*$  é um ponto limite desta sequência. Então,  $x_*$  é um ponto estacionário de*

$$\text{Minimizar } \sum_{j=1}^{n_h} h_j(x)^2 + \sum_{j=1}^{n_g} \max\{0, g_j(x)\}^2.$$

Prova. Como  $\{x_k\}$  é uma sequência infinita, existe  $i \in \{1, \dots, m\}$  tal que (4.67) é satisfeita para  $f_i$  infinitas vezes. Tomando a subsequência de  $\{x_k\}$  correspondente, podemos pensar que esta subsequência foi gerada pelo Algoritmo 4.3. Portanto, o resultado segue do Teorema 4.16.  $\square$

**Teorema 4.20.** *Suponha que  $\{x_k\}$  é uma sequência gerada pelo Algoritmo 4.4 com  $\varepsilon_k \rightarrow 0$ ,  $x_*$  é um ponto limite e a condição de qualificação CPLD é satisfeita em  $x_*$ . Então, existe  $i \in I(x_*)$  tal que  $x_*$  é um ponto KKT de*

$$\text{Minimizar } f_i(x) \text{ s.a. } h(x) = 0, g(x) \leq 0.$$

*Prova.* Assim como no Teorema 4.19, considere uma subsequência de  $\{x_k\}$  tal que (4.67) é satisfeita com o mesmo índice  $i$  para todos os termos desta subsequência. Novamente, podemos pensar que esta sequência foi gerada pelo Algoritmo 4.3. Pelo Teorema 4.17, existe  $x_*$  satisfazendo a tese deste teorema. O fato de  $i \in I(x_*)$  segue, trivialmente, do fato que  $L_i(x_k, \bar{\lambda}_k, \bar{\mu}_k, \rho_k) \leq L_j(x_k, \bar{\lambda}_k, \bar{\mu}_k, \rho_k)$ , para todo  $j$ .  $\square$

O resultado final, que fala da limitação dos parâmetros de penalização associados ao Algoritmo 4.4, é dado no Teorema 4.21. Assim como nos teoremas anteriores, a técnica consiste em reduzir o problema LOVO a um problema de programação não-linear diferenciável. Entretanto, neste caso, necessitaremos de uma hipótese adicional: dada uma sequência convergente gerada pelo Algoritmo 4.4, vamos supor que existe um único índice  $i_{min}$  tal que  $f_{min}(x_k) = f_{i_{min}}(x_k)$ , para todo  $k$  suficientemente grande. Desta forma, é possível garantir que o algoritmo, definitivamente, se comporta como o Algoritmo 4.3 para a minimização de  $f_{i_{min}}$ .

**Hipótese C2.** Suponha que:

1. A sequência  $\{x_k\}$  é gerada pela aplicação do Algoritmo 4.4 ao problema (4.61) e

$$\lim_{k \rightarrow \infty} x_k = x_*.$$

2. O ponto  $x_*$  é viável ( $h(x_*) = 0, g(x_*) \leq 0$ ).

3. Existe  $i_{min} \in \{1, \dots, m\}$  tal que

$$f_{i_{min}}(x_k) = f_{min}(x_k) < f_i(x^k),$$

para todo  $k$  suficientemente grande e  $i \neq i_{min}$ .

4. Os gradientes

$$\{\nabla h_j(x_*)\}_{j=1}^{n_h}, \{\nabla g_j(x_*)\}_{\{j \mid g_j(x_*)=0\}}$$

são linearmente independentes.

5. Tem-se complementaridade estrita em  $x_*$ . Isto significa que, se  $\mu_* \in \mathbb{R}_+^{n_g}$  é o vetor dos multiplicadores de Lagrange correspondentes às restrições  $g(x) \leq 0$ , então

$$g_j(x_*) = 0 \Rightarrow [\mu_*]_j > 0.$$

6. As funções  $f_{i_{min}}, h, g$  possuem segunda derivada contínuas em uma vizinhança de  $x_*$ .

7. Defina o subespaço tangente  $T$  como sendo o conjunto formado por todo  $z \in \mathbb{R}^n$  tal que

$$\nabla h(x_*)^T z = 0,$$

$$\nabla [g(x_*)]_j^T z = 0,$$

para todo  $j$  tal que  $g_j(x_*) = 0$ . Então, para todo  $z \in T, z \neq 0$ ,

$$z^T [\nabla^2 f_{i_{min}}(x_*) + \sum_{j=1}^{n_h} [\lambda_*]_j \nabla^2 h_j(x_*) + \sum_{j=1}^{n_g} [\mu_*]_j \nabla^2 g_j(x_*)] z > 0.$$

**Teorema 4.21.** *Suponha que a Hipótese C2 é satisfeita. Adicionalmente, suponha que:*

1. *Existe uma sequência  $\eta_k \rightarrow 0$  tal que*

$$\varepsilon_k \leq \eta_k \max\{\|h(x_k)\|_\infty, \|\sigma_k\|_\infty\}, \forall k \in \mathbb{N}.$$

2.  $[\lambda_*]_j \in (\bar{\lambda}_{\min}, \bar{\lambda}_{\max}) \forall j = 1, \dots, n_h$  e  $[\mu_*]_j \in (\bar{\mu}_{\min}, \bar{\mu}_{\max}), \forall j = 1, \dots, n_g$ .
3.  $[\bar{\lambda}_{k+1}]_j$  é uma projeção de  $[\lambda_{k+1}]_j$  em  $[\bar{\lambda}_{\min}, \bar{\lambda}_{\max}]$ ,  $\forall j = 1, \dots, n_h$ , e  $[\bar{\mu}_{k+1}]_j$  é a projeção de  $[\mu_{k+1}]_j$  em  $[0, \bar{\mu}_{\max}]$ ,  $\forall j = 1, \dots, n_g, k \in \mathbb{N}$ .

*Então, a sequência  $\{\rho_k\}$  dos parâmetros de penalização é limitada.*

*Prova.* Para  $k$  suficientemente grande, a sequência pode ser pensada como sendo uma sequência gerada pelo Algoritmo 4.3 com a Hipótese C1. Logo, o resultado segue do Teorema 4.18.  $\square$

# Capítulo 5

## Quadrados Mínimos LOVO

Suponha que  $\{(t_1, y_1), \dots, (t_m, y_m)\} \subset \mathbb{R}^2$  é um conjunto de dados e que sabemos que alguns deles estão “incorretos”, ou seja, há presença de “outliers”. Suponha que  $T(x, t_i)$  é o valor previsto da observação  $i$  ao adotar um parâmetro  $x \in \Omega$ . O ajuste de quadrados mínimos da forma  $y_i \approx T(x, t_i)$  leva a resultados insatisfatórios devido à influência opressiva de “outliers”.

A proposta LOVO para estimação robusta de parâmetros consiste em definir, para cada  $i = 1, \dots, r$ , a função de erro

$$F_i(x) = (T(x, t_i) - y_i)^2.$$

Dado  $p \in \{1, \dots, r\}$ , este conjunto de funções define um problema LOVO para o qual os Algoritmos 4.1 e 4.2 (para o caso irrestrito) e 4.4 (para problemas com restrições) podem ser aplicados. Quando  $p = r$  este problema LOVO coincide com o clássico problema de quadrados mínimos não-linear. Entretanto, a situação fica muito mais interessante quando  $p$  é menor que  $r$ . Neste caso, é possível desprezar um número pré-determinado de “outliers”. Logo, podemos dizer que o problema LOVO é uma generalização do problema de quadrados mínimos não-linear.

A idéia é resolver o problema LOVO para diferentes valores de  $p$ . Se  $p$  é igual a  $r$ , a expectativa é que o valor da função LOVO na solução encontrada seja grande, ou seja, as funções de erro  $F_{i_1}, \dots, F_{i_r}$  que definem o somatório irão indicar que, para alguns pontos, o ajuste não está bom, principalmente para aqueles cujos índices correspondentes estão próximos de  $i_r$ . Quando este fato ocorre, temos um forte indício de que existem dados incorretos ou “outliers”. Quando  $p$  decresce, a função LOVO tende a decrescer também. Obviamente, uma das razões para este decréscimo é o fato da quantidade de termos na

soma da função LOVO ser cada vez menor, mas, a expectativa é que, quando tomarmos o valor de “ $p$  correto”, a magnitude deste decréscimo seja bem maior.

## 5.1 Ajustando Modelos com “Outliers”

### 5.1.1 Implementação

Uma das principais consequências da teoria introduzida no capítulo anterior é que, mesmo com a não-diferenciabilidade do problema LOVO, se ignorarmos a multiplicidade do gradiente em um dado ponto  $x_k$  e utilizarmos diretamente um método de minimização diferenciável, os prejuízos que poderíamos vir a ter são mais amenos. De fato, o fator mais agravante corresponde a não termos garantia de convergência para minimizadores globais, mas, esta inconveniência é compartilhada pela maioria dos métodos de programação não-linear.

Muitos algoritmos de otimização para problemas diferenciáveis, quando aplicados ao LOVO, podem ser considerados casos particulares dos Algoritmos 4.1 e 4.4. Com esta propriedade em mente, utilizamos, em nossos experimentos, as versões com e sem restrições do ALGENCAN, que é um código de programação não-linear disponível na página do projeto Tango ([www.ime.usp.br/~egbirgin/tango](http://www.ime.usp.br/~egbirgin/tango)) [1, 2, 14], sem modificar as definições “default” dos parâmetros.

Implementamos também uma versão do Algoritmo 4.2 que satisfaz às Hipóteses **B3** e **B4**. Algumas características relativas a esta implementação estão descritas abaixo:

- Definimos  $d_k^i$ , para cada  $i \in J_k$ , como sendo a direção de Gauss-Newton correspondente, ou seja,

$$B_k^i = 2\Phi_i'(x_k)^T \Phi_i'(x_k)$$

onde  $\Phi_i'(x)$  é o jacobiano de  $\Phi_i(x)$  em relação a  $x$ ,  $\Phi_i(x)$  é um vetor de tamanho  $p$  cujas componentes são  $\phi_j(x) = (T(x, t_j) - y_j)$  e os  $p$  índices  $j$  são tais que  $j \in \mathcal{C}_i$ . Como não temos garantia de que  $B_k^i$  será sempre definida positiva, quando necessário, corrigimos  $B_k^i$  somando uma constante positiva  $\mu$  em sua diagonal até que esta seja definida positiva.

- A estratégia de busca linear (4.7)-(4.8) implementada foi o “backtracking”, isto é,  $t_k$  foi escolhido como sendo o primeiro número da sequência  $\{1, 2^{-1}, 2^{-2}, \dots\}$  que satisfaz (4.7) e  $x_{k+1} = x_k + t_k d_k$ . Neste caso,  $t_{one} = 1$  e  $M = 2$ .

- Os parâmetros algorítmicos utilizados foram:

$$\theta = 10^{-6}, \alpha = 0.1, \beta = 10^{-2}, \varepsilon = 10^{-4}, \delta = 10^{-2}.$$

- O critério de parada foi:  $\|\nabla f_i(x_k)\| < 10^{-4}$ , para todo  $i \in I(x_k)$ .

Todos os experimentos desta seção foram realizados em um Pentium 4, 3.2 Ghz, 2Gb RAM em FORTRAN de dupla precisão.

### 5.1.2 Caso Irrestrito

Para ilustrar o comportamento da estratégia LOVO consideramos um problema irrestrito bastante simples, porém, muito ilustrativo onde  $T(x, t_i)$  é definida por:

$$T(x, t_i) = x_1 \exp[-t_i x_5] + x_2 \exp[-(t_i - x_9)^2 x_6] + \\ x_3 \exp[-(t_i - x_{10})^2 x_7] + x_4 \exp[-(t_i - x_{11})^2 x_8].$$

Esta é a função de Osborne 2 (problema (19) de [34]). Em [34],  $r = 65$ . Aqui, introduzimos 13 pontos adicionais, que representam os erros sistemáticos. O critério para determinar esta quantidade foi gerar aproximadamente 20% de “outliers” em função da quantidade de dados “corretos” fornecidos em [34]. Os resultados estão reportados nas Tabelas 5.1 e 5.2. As respectivas tabelas trazem as seguintes informações:

- $p$  - inteiro entre  $\{1, \dots, r\}$  que define o problema LOVO.
- $k_{max}$  - quantidade de minimizações realizadas (número de pontos iniciais gerados aleatoriamente).
- $k$  - esta coluna indica que a melhor solução foi encontrada na  $k$ -ésima minimização. Nas Tabelas 5.1 e 5.2,  $k = 0$  significa que a melhor solução foi obtida definindo  $x_0$  como sendo o ponto inicial sugerido em [34].
- $x$  - melhor solução encontrada após  $k_{max}$  minimizações.
- $S_p(x)$  - valor da função LOVO avaliada em  $x$ .
- $S_{p+1}/S_p$  - quociente (razão) entre os valores reportados na coluna anterior.
- n.feval - número de avaliações da função LOVO na  $k$ -ésima minimização.

$p$	Número de minimizações		Melhor Solução encontrada							
	$k_{max}$	$k$	4 primeiros termos de $x$				$S_p(x)$	$S_{p+1}/S_p$	n.feval	in.iter
78	50	0	1.27	0.38	0.48	0.59	2.74	-	58	36
77	50	0	1.32	0.56	0.61	0.60	2.39	1.15	56	37
76	50	0	1.34	0.51	0.62	0.61	2.02	1.19	33	20
75	50	0	1.33	0.46	0.61	0.62	1.76	1.14	29	20
74	50	0	1.31	0.50	0.61	0.81	1.53	1.15	36	25
73	50	0	1.32	0.39	0.60	0.80	1.32	1.16	30	18
72	50	0	1.32	0.41	0.60	0.70	1.11	1.19	29	18
71	50	0	1.32	0.37	0.61	0.68	0.94	1.19	30	16
70	50	0	1.34	0.40	0.61	0.66	0.75	1.24	24	16
69	50	0	1.34	0.41	0.61	0.61	0.61	1.23	22	15
68	50	0	1.33	0.35	0.60	0.61	0.42	1.45	25	15
67	50	27	1.34	0.60	0.35	0.56	0.28	1.49	115	52
66	50	0	1.29	0.35	0.60	0.57	0.17	1.66	21	13
65	50	0	1.31	0.43	0.63	0.60	0.04	4.26	37	21
64	50	0	1.26	0.39	0.62	0.58	0.03	1.17	26	17
63	100	76	1.26	0.38	0.61	0.58	0.03	1.20	28	17

Tabela 5.1: Resultados do Algencan aplicado à função Osborne 2.

- in.iter - número de iterações internas do algoritmo Algencan na  $k$ -ésima minimização.
- out.iter - número de iterações externas do algoritmo Algencan na  $k$ -ésima minimização.

Para problemas irrestritos, o número de iterações externas do Algencan é sempre igual a 1. Por este motivo suprimimos esta coluna das Tabelas 5.1 e 5.2.

- n.iter - número de iterações do Algoritmo 4.2 na  $k$ -ésima minimização.

**Observação:** O mínimo encontrado para  $p = 65$ , o valor de “ $p$  correto”, coincide com a solução reportada em [34].

		Número de minimizações	Melhor Solução encontrada							
$p$	$k_{max}$	$k$	4 primeiros termos de $x$				$S_p(x)$	$S_{p+1}/S_p$	n.feval	n.iter
78	50	15	1.27	0.48	0.59	0.38	2.74	-	5542	364
77	50	0	1.31	0.56	0.61	0.60	2.39	1.15	39	18
76	50	0	1.34	0.51	0.62	0.61	2.02	1.19	27	13
75	50	0	1.33	0.46	0.61	0.62	1.76	1.14	24	13
74	50	0	1.32	0.38	0.59	0.63	1.56	1.13	21	15
73	50	0	1.31	0.51	0.61	0.71	1.32	1.18	22	13
72	50	0	1.32	0.41	0.60	0.70	1.11	1.19	18	12
71	50	0	1.32	0.37	0.61	0.68	0.94	1.19	20	14
70	50	0	1.34	0.40	0.61	0.66	0.75	1.24	16	11
69	50	0	1.34	0.38	0.61	0.63	0.60	1.26	14	11
68	50	19	1.33	0.35	0.60	0.61	0.42	1.42	98	29
67	50	0	1.34	0.35	0.56	0.60	0.28	1.49	15	12
66	100	0	1.27	0.35	0.60	0.57	0.17	1.66	13	10
65	200	158	1.31	0.63	0.60	0.43	0.04	4.26	381	95
64	100	80	1.31	0.60	0.42	0.63	0.04	1.14	471	61
63	50	47	1.31	0.42	0.61	0.64	0.03	1.09	315	49

Tabela 5.2: Resultados do Algoritmo 4.2 aplicado a função Osborne 2.

Analisando as tabelas, podemos observar que quando  $p$  toma um valor próximo de  $r$  obtemos um alto valor da função LOVO, indicando que há presença de “outliers”. À medida que diminuimos o valor de  $p$  notamos um decréscimo da função LOVO, porém, como já havíamos alertado anteriormente, a magnitude deste decréscimo tem uma proporção muito maior quando atingimos o valor “correto de  $p$ ”.

Ainda fazendo uma comparação entre as tabelas podemos concluir que, para estes experimentos, os pontos críticos fortes parecem estar isolados, ou seja, o conjunto  $I(x_*)$  é composto por um único índice, já que, as soluções encontradas por Algencan e pelo Algoritmo 4.2 são praticamente as mesmas, apesar de termos com Algencan apenas a garantia de convergência para pontos críticos fracos. Como a garantia de convergência para pontos críticos fortes não foi determinante, o fato do Algencan possuir um processo de busca linear mais sofisticado do que a estratégia “backtracking”, adotada na implementação do Algoritmo 4.2, levou a um melhor desempenho do Algencan ao compararmos o número de iterações e a quantidade de minimizações necessárias para obtermos a melhor solução encontrada.

Dado  $p$ , seja  $x_*$  a solução do problema LOVO obtida pelo algoritmo. Em todas as figuras, a curva representa a imagem de  $T(x_*, t)$  como uma função de  $t$ , os pontos representam os dados  $(t_i, y_i)$  e os pontos circulados são aqueles que possuem os  $r - p$  maiores erros ao adotar o parâmetro  $x_*$ , conseqüentemente, estes foram os pontos descartados pelo algoritmo. Na Figura 5.1, ilustramos a melhor solução obtida pelos algoritmos para os problemas LOVO definidos pelos valores  $p = 78, 75, 70$  e  $65$ .

Seguindo a mesma metodologia, realizamos mais alguns experimentos com os problemas testes (8),(9),(15) e (17) de [34]. As melhores soluções obtidas por Algencan e pelo Algoritmo 4.2 foram idênticas, chegamos às mesmas conclusões gerando tabelas semelhantes às Tabelas 5.1 e 5.2, portanto, por ser mais ilustrativo, optamos por apresentar somente a representação gráfica da melhor solução obtida pelos algoritmos para diferentes valores de  $p$ , seguindo as mesmas convenções da Figura 5.1.

O problema teste (8) corresponde à função de Bard e  $T(x, t_i)$  é definida por:

$$T(x, t_i) = \left( x_1 + \frac{u_i}{v_i x_2 + w_i x_3} \right),$$

onde  $u_i = i$ ,  $v_i = 16 - i$ ,  $w_i = \min\{u_i, v_i\}$ . Em [34],  $r = 15$ . Aqui, introduzimos 3 pontos adicionais representando os “outliers”. Os resultados estão representados na Figura 5.2.

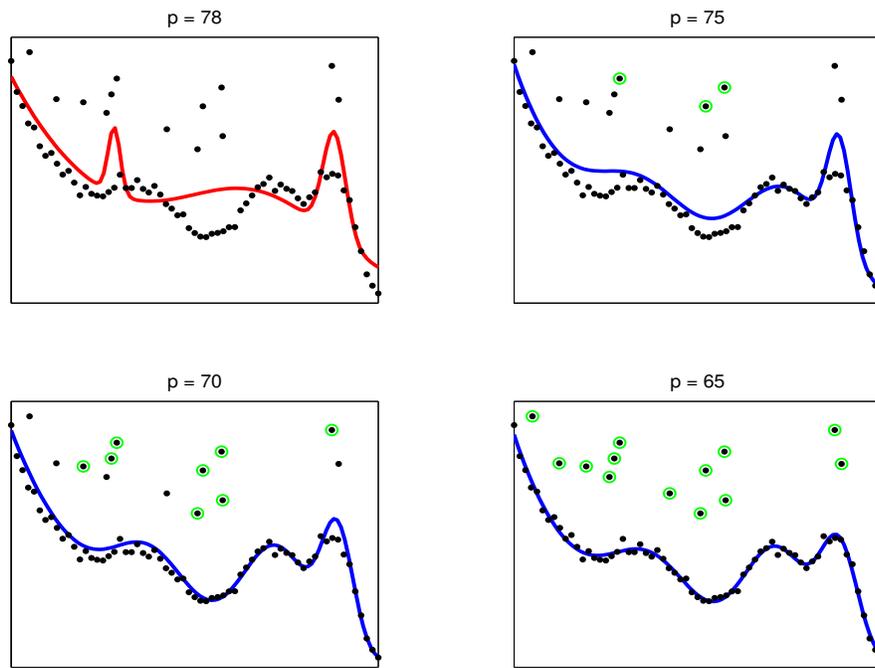


Figura 5.1: Representação gráfica das soluções (Osborne 2).

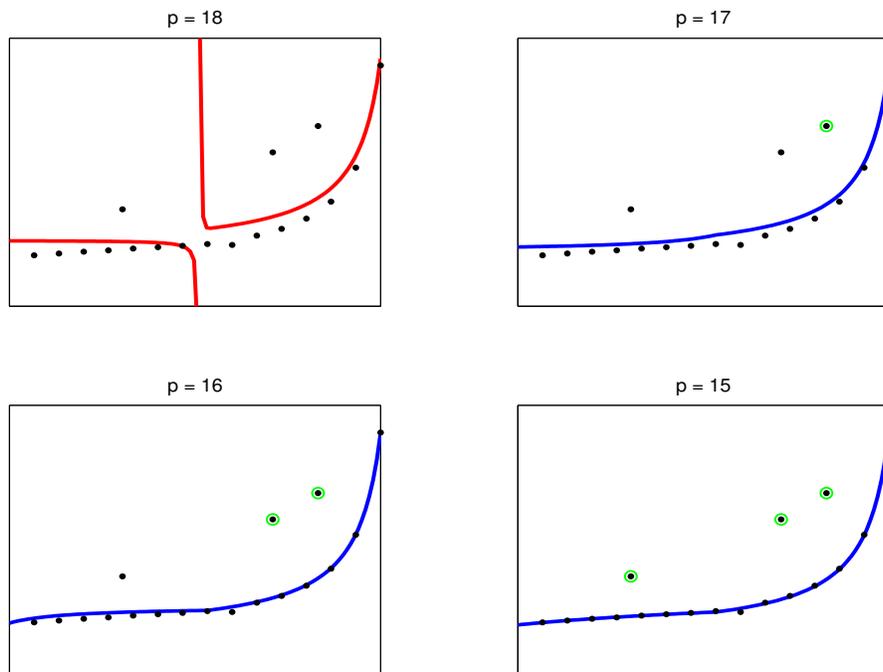


Figura 5.2: Representação gráfica das soluções (Bard).

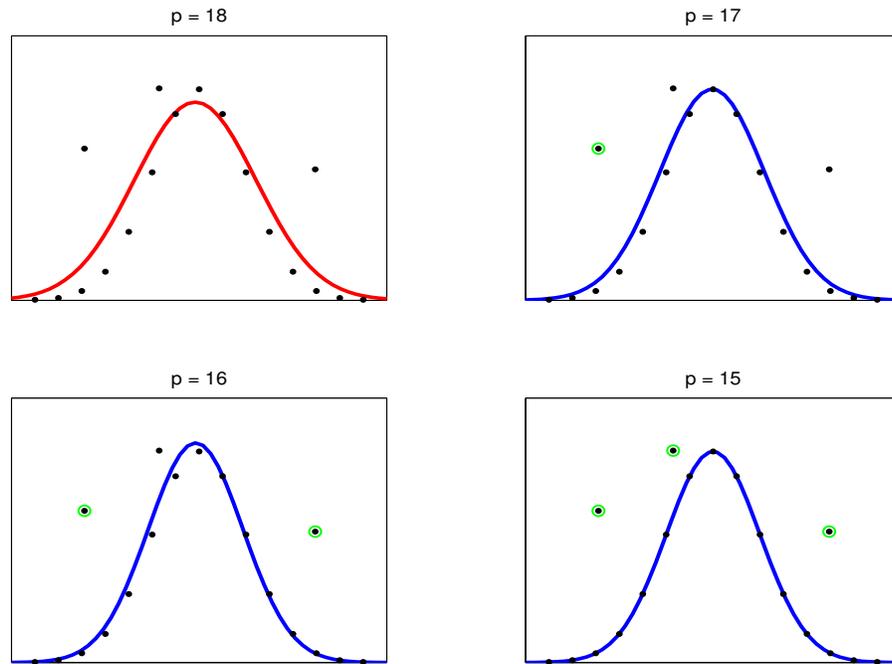


Figura 5.3: Representação gráfica das soluções (Gaussiana).

O problema teste (9) corresponde à função Gaussiana e  $T(x, t_i)$  é definida por:

$$T(x, t_i) = x_1 \exp\left(\frac{-x_2(t_i - x_3)^2}{2}\right).$$

Em [34],  $r = 15$ . Aqui, introduzimos 3 pontos adicionais representando os “outliers”. Os resultados estão representados na Figura 5.3.

O problema teste (15) corresponde à função de Kowalik e Osborne e  $T(x, t_i)$  é definida por:

$$T(x, t_i) = \frac{x_1(t_i^2 + t_i x_2)}{(t_i^2 + t_i x_3 + x_4)}.$$

Em [34],  $r = 11$ . Aqui, introduzimos 2 pontos adicionais representando os “outliers”. Os resultados estão representados na Figura 5.4.

O problema teste (17) corresponde à função de Osborne 1 e  $T(x, t_i)$  é definida por:

$$T(x, t_i) = (x_1 + x_2 \exp[-t_i x_4] + x_3 \exp[-t_i x_5]).$$

Em [34],  $r = 33$ . Aqui, introduzimos 6 pontos adicionais representando os “outliers”. Os resultados estão representados na Figura 5.5.

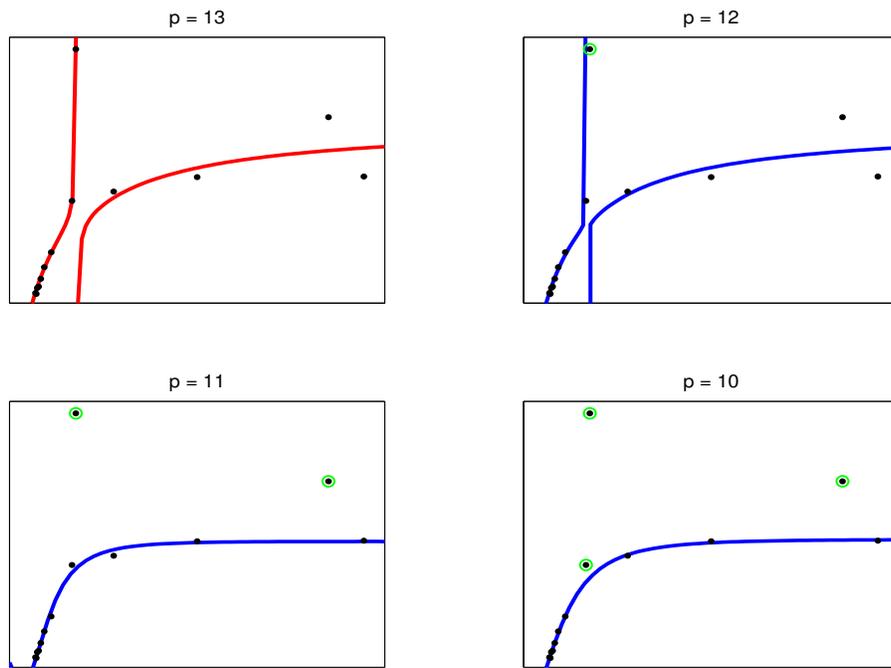


Figura 5.4: Representação gráfica das soluções (Kowalik).

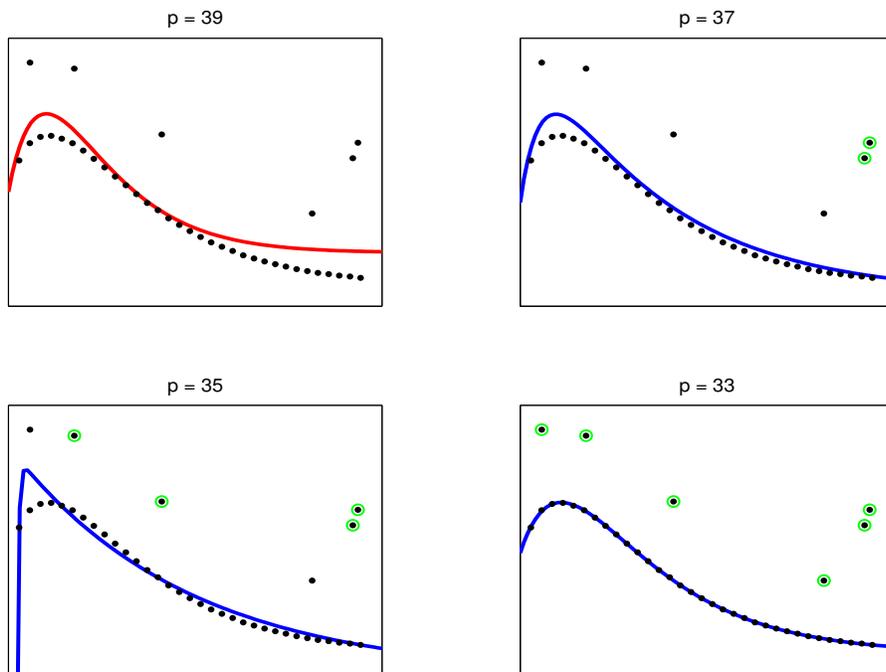
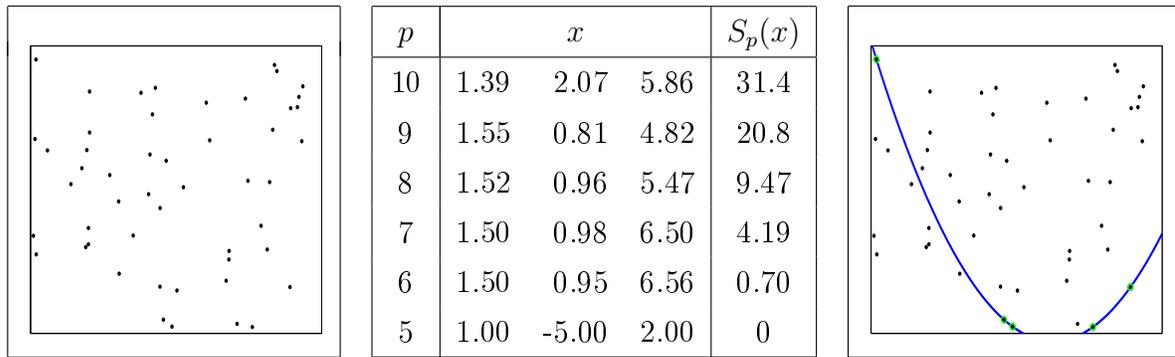


Figura 5.5: Representação gráfica das soluções (Osborne 1).

Tabela 5.3:  $m = 50$ , “ $p$  correto” = 5,  $x_* = (1, -5, 2)$ 

A interpretação destes resultados (figuras) é bastante simples. Quando  $p = r$ , temos a representação gráfica do clássico ajuste de quadrados mínimos (curvas em vermelho) não-linear, a medida que diminuimos o valor de  $p$ , os  $r - p$  pontos que possuem os maiores erros são descartados e a solução LOVO representa o ajuste de quadrados mínimos aos  $p$  pontos restantes. Condizendo com este fato, observamos que o valor da função LOVO na melhor solução encontrada para o valor “correto de  $p$ ”, para cada um dos problemas testes, foi exatamente o mesmo valor reportado em [34]. Ressaltamos que a escolha dos pontos que são descartados é feita iterativamente pelo próprio algoritmo. Esta é, sem dúvida, a principal vantagem da estratégia LOVO para resolver problemas de estimação robusta de parâmetros.

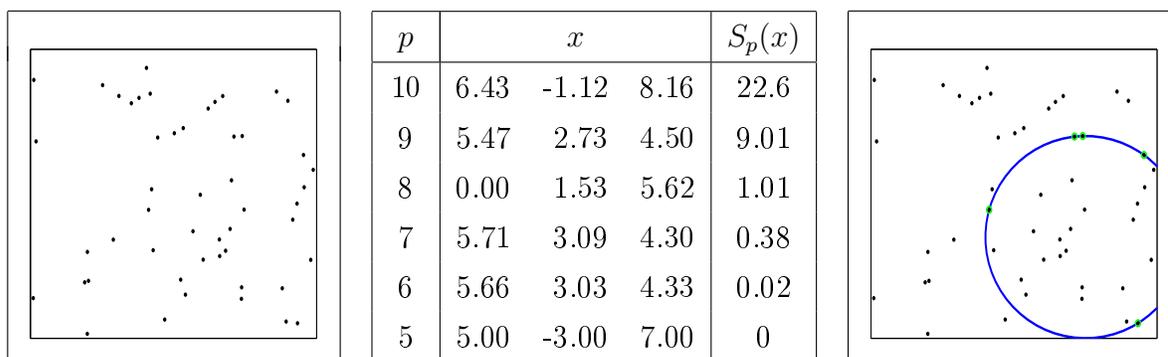
### 5.1.3 Caso com Restrições

Na seção anterior, foi possível observar que, na prática, a convergência para pontos críticos fracos que não são minimizadores locais é um fato que não ocorre com tanta frequência. Por esta razão, optamos por utilizar somente o Algencan para realizar os experimentos desta seção, considerando-o como um caso particular do Algoritmo 4.4.

A estratégia LOVO também pode ser aplicada para encontrar padrões ocultos que, particularmente, podem ser considerados como problemas de ajuste com restrições de caixa. Definindo

$$F_i(x) = \left( \sum_{j=1}^n x_j t_i^{n-j} - y_i \right)^2,$$

tal como na seção 2.2.1, podemos encontrar polinômios ocultos. Os resultados, reproduzindo os experimentos da Tabela 2.1, estão reportados na Tabela 5.3.



$p$	$x$			$S_p(x)$
10	6.43	-1.12	8.16	22.6
9	5.47	2.73	4.50	9.01
8	0.00	1.53	5.62	1.01
7	5.71	3.09	4.30	0.38
6	5.66	3.03	4.33	0.02
5	5.00	-3.00	7.00	0

Tabela 5.4:  $m = 50$ , “ $p$  correto” = 5,  $x^* = (5, -3, 7)$ 

Similarmente, podemos encontrar circunferências ocultas definindo

$$F_i(x) = [(t_i - x_1)^2 + (y_i - x_2)^2 - x_3^2]^2,$$

tal como na seção 2.2.2. Os resultados, reproduzindo os experimentos da Tabela 2.7, estão reportados na Tabela 5.4.

Agora, vamos descrever uma outra aplicação para o LOVO com restrições. Considere um intervalo  $[a, b]$ . Seja  $t_1, \dots, t_n$  uma partição deste intervalo de forma que pontos consecutivos sejam equidistantes por um incremento  $h$ ,  $t_1 = a$  e  $t_n = b$ . A incógnita deste problema é um vetor  $x = (x_1, \dots, x_n)$  que representa a discretização da solução de uma equação diferencial de segunda ordem, digamos  $\ddot{x} = f(x, t)$ . Isto significa que as restrições

$$x_{i+1} - 2x_i + x_{i-1} - h^2 f(x_i, t_i) = 0, \quad i = 2, \dots, n - 1. \quad (5.1)$$

devem ser satisfeitas por  $x$ . Usualmente, um problema de valor de contorno (PVC) consiste em encontrar  $x$  que satisfaça (5.1) e que, adicionalmente,  $x_1$  e  $x_n$  tenham valores de contorno dados. Mas, esta aplicação busca o seguinte objetivo: a partir de  $n$  observações  $y_1, \dots, y_n$  observadas nos instantes  $t_1, \dots, t_n$ , respectivamente, encontrar os verdadeiros valores de  $x_1, \dots, x_n$  que mais se pareçam com estas observações. Logo, a formulação LOVO para este problema é a seguinte:

$$\text{Minimizar } \sum_{i=1}^n [y_i - x_i]^2 \quad \text{s.a. (5.1)}. \quad (5.2)$$

Suponha que, por algum motivo, as observações próximas à região de contorno sejam difíceis de serem recuperadas. Nestas situações, a formulação LOVO pode ser bastante útil, já que, podemos encontrar o melhor ajuste desprezando algumas observações, ou seja, considerando apenas  $p$  observações.

$p$	Número de minimizações		Melhor solução encontrada								
	$k_{max}$	$k$	$x_1$	$x_n$	$\alpha$	$\beta$	$\gamma$	$S_p(x)$	n.feval	in.iter	out.iter
21	100	100	7.2	8.6	0.2	1.7	0.4	5.27	121	74	6
20	100	97	7.4	8.1	0.3	2.6	1.1	1.71	77	45	6
19	100	95	45.7	8.2	4.1	38.9	31.4	0.13	364	335	2
18	100	45	7.1	0.7	1.0	9.3	7.3	0.04	96	59	3
17	100	88	7.1	-2.1	1.0	9.5	7.1	0.02	88	49	3
16	100	2	3.9	6.1	0.1	0.9	2.0	0.01	74	52	4
15	100	62	3.9	6.0	0.1	1.1	2.3	0.00	70	44	4
14	100	47	3.9	5.9	0.1	1.2	2.2	0.00	120	87	4

Tabela 5.5: Resultados do Algencan aplicado ao problema de contorno.

Um segundo objetivo desta aplicação é estimar alguns parâmetros que definem  $f(x, t)$ . Por exemplo, se  $f(x, t) = \alpha e^x - \beta t(x^2 + 1) - \gamma x \sin(tx)$ , vamos estimar conjuntamente: a discretização da solução do PVC que mais se ajusta aos dados e as constantes  $\alpha, \beta$  e  $\gamma$  que definem a equação diferencial.

Para gerar os dados procedemos da seguinte forma: a partir da solução de um dado PVC, geramos o vetor de dados  $y$  inserindo um pequeno ruído na solução, porém, em n.out pontos, próximos à região de contorno, introduzimos uma perturbação positiva consideravelmente maior para produzir os erros sistemáticos.

Os resultados da estratégia LOVO aplicada ao PVC original:

$$\begin{cases} \ddot{x} = \alpha e^x - \beta t(x^2 + 1) - \gamma x \sin(tx) \\ x(0) = 4, x(2) = 6 \end{cases}$$

onde  $\alpha = 0.1$ ,  $\beta = 1$  e  $\gamma = 2$  com  $h = 0.1$  ( $r = 21$ ) e n.out = 6, estão reportados na Tabela 5.5 e Figura 5.6.

Analisando os resultados, podemos observar que, quando  $p = r$ , temos um alto valor da função LOVO devido à presença de observações erradas próximas à região de contorno. À medida que diminuimos o valor de  $p$ , notamos um decréscimo da função LOVO e, ao atingirmos o valor “correto de  $p$ ”, os parâmetros  $\alpha, \beta, \gamma$  e as condições de contorno do PVC original são recuperados.

Realizamos mais alguns experimentos deste gênero, mas, vamos mostrar os resultados

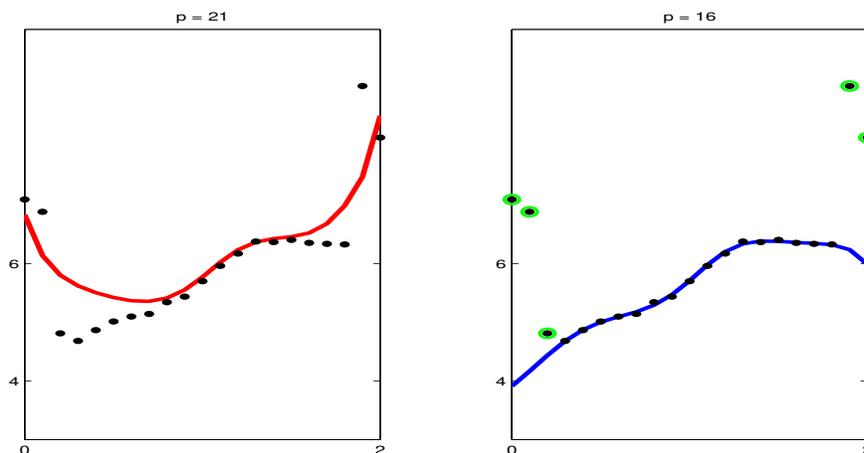


Figura 5.6: Soluções LOVO para diferentes valores de  $p$ .

obtidos apenas graficamente, já que, a dinâmica das tabelas é muito parecida com a Tabela 5.5.

Os resultados da estratégia LOVO aplicada ao PVC original:

$$\begin{cases} \ddot{x} = \alpha \sqrt{e^{2t} - x^2} \\ x(0) = 0, x(2) = 2 \end{cases}$$

onde  $\alpha = 2$  com  $h = 0.1$  ( $r = 21$ ) e  $n.out = 6$ , estão reportados Figura 5.7.

Os resultados da estratégia LOVO aplicada ao PVC original:

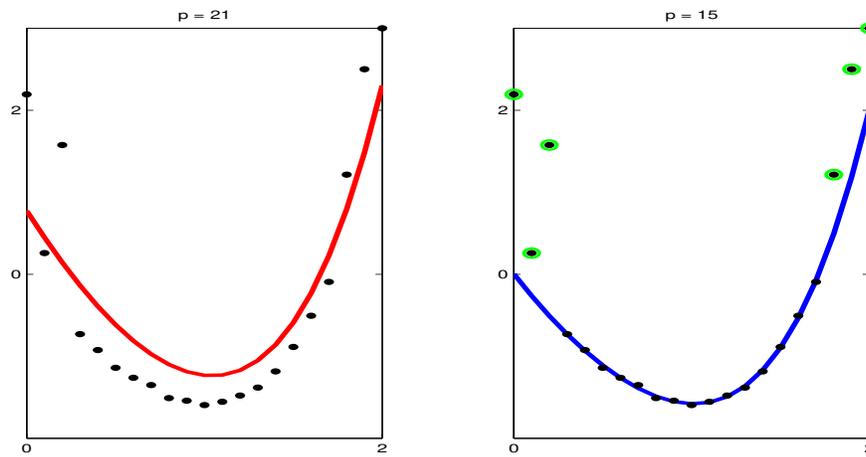
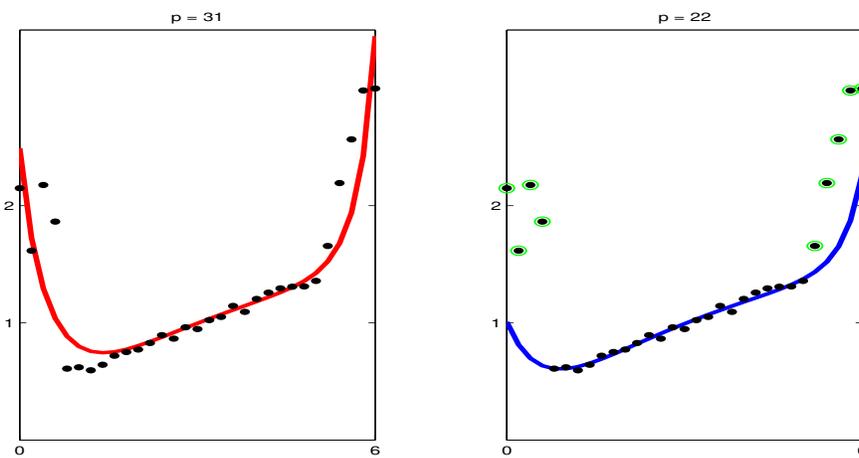
$$\begin{cases} \ddot{x} = \alpha x + \beta x^3 + \gamma t \\ x(0) = 1, x(6) = 2 \end{cases}$$

onde  $\alpha = 2$ ,  $\beta = 1$  e  $\gamma = -1$  com  $h = 0.2$  ( $r = 31$ ) e  $n.out = 9$ , estão reportados Figura 5.8.

## 5.2 Padrão Oculto - Proteína

Sejam  $\mathcal{Q} = \{Q_1, \dots, Q_N\} \subset \mathbb{R}^{dim}$ ,  $\mathcal{P} = \{P_1, \dots, P_M\} \subset \mathbb{R}^{dim}$ ,  $N \leq M$ . O objetivo é encontrar a estrutura definida por  $\mathcal{Q}$  no conjunto  $\mathcal{P}$ . Mais precisamente, objetivamos encontrar um operador de deslocamento,  $D : \mathbb{R}^{dim} \rightarrow \mathbb{R}^{dim}$ , tal que  $\{D(Q_1), \dots, D(Q_N)\}$  se ajuste a algum subconjunto de  $\mathcal{P}$ .

Defina  $\mathcal{N}$  como sendo o conjunto de  $N$ -uplas  $\nu = (\nu(1), \dots, \nu(N))$ , onde  $\nu(i) \in \{1, \dots, M\}$ , para todo  $i = 1, \dots, N$ . (Em outras palavras,  $\mathcal{N} = \{1, \dots, M\}^N$ .)

Figura 5.7: Soluções LOVO para diferentes valores de  $p$ .Figura 5.8: Soluções LOVO para diferentes valores de  $p$ .

Seja  $D$  um deslocamento. Para todo  $\nu \in \mathcal{N}$  defina:

$$f_\nu(D) = \sum_{i=1}^N \|D(Q_i) - P_{\nu(i)}\|^2.$$

Finalmente,

$$f_{min}(D) = \min_{\nu \in \mathcal{N}} f_\nu(D).$$

Se existe um conjunto com  $N$  pontos de  $\mathcal{P}$  que se ajuste exatamente a um deslocamento  $D$  de  $\mathcal{Q}$  temos  $f_{min}(D) = 0$ . O problema de minimizar  $f_{min}$  segue da teoria introduzida no capítulo anterior.

Afortunadamente, para avaliar  $f_{min}$  não precisamos calcular todas as funções  $f_\nu$ . De fato, dado um deslocamento  $D$ , calculamos, para todo  $i = 1, \dots, N$ ,  $P_{c(i)}(D) \in \mathcal{P}$  tal que

$$\|D(Q_i) - P_{c(i)}(D)\| \leq \|D(Q_i) - P\|, \quad \forall P \in \mathcal{P}. \quad (5.3)$$

Então,

$$f_{min}(D) = \sum_{i=1}^N \|D(Q_i) - P_{c(i)}(D)\|^2.$$

As duas situações mais comuns, em aplicações, correspondem a  $dim = 2$  e  $dim = 3$ . No primeiro caso, o deslocamento pode ser representado por três parâmetros: a translação do centro de gravidade de  $\mathcal{Q}$  e o ângulo de rotação. Nos casos tridimensionais, deslocamentos podem ser representados pelo vetor de translação e três rotações, embora outras alternativas sejam possíveis.

Uma generalização deste problema consiste em encontrar uma estrutura comum para os conjuntos  $\mathcal{P}$  e  $\mathcal{Q}$ . Suponha que queremos encontrar um deslocamento  $D$  tal que existam  $R \leq N$  pontos de  $\mathcal{Q}$  (digamos,  $Q_{j_1}, \dots, Q_{j_R}$ ) tais que  $D(Q_{j_1}), \dots, D(Q_{j_R})$  se ajustem a  $R$  pontos de  $\mathcal{P}$ . Neste caso, definimos  $\mathcal{M}$  como sendo o produto cartesiano dos subconjuntos de  $R$  elementos de  $\{1, \dots, N\}$  com as  $R$ -uplas de  $\{1, \dots, M\}$ . Para todo  $\nu = (\{j_1, \dots, j_R\}, \{i_1, \dots, i_R\}) \in \mathcal{M}$ , definimos

$$f_\nu(D) = \sum_{\ell=1}^R \|D(Q_{j_\ell}) - P_{i_\ell}\|^2$$

e o objetivo é minimizar  $f_{min}(D) \equiv \min_{\nu \in \mathcal{M}} f_\nu(D)$ . Novamente, o cálculo de  $f_{min}$  é simples: para todo  $i = 1, \dots, N$ , calcule  $P_{c(i)}(D) \in \mathcal{P}$  como em (5.3). Então,  $f_{min}(D)$  é a soma dos  $R$  menores valores de  $\|D(Q_i) - P_{c(i)}(D)\|^2$ .

Embora a definição mais óbvia de um operador de deslocamento envolva somente translações e rotações, definições mais generalizadas podem ser elaboradas. Por exemplo, a introdução de um parâmetro adicional permite que consideremos variações de escala, de maneira que uma dada forma possa ser reconhecida em uma estrutura, independentemente do seu tamanho. Além disso, se substituirmos a norma Euclidiana da diferença por outra função distância, podemos obter muitas medidas alternativas similares e adequadas para os problemas sob consideração.

Em nossa aplicação, os pontos de  $\mathcal{P}$ , representados na Figura 5.9(a) em cinza claro, são os átomos de carbono da cadeia principal de uma proteína (253 átomos): o receptor do hormônio tireoideano, ligado a um hormônio sintético conhecido como IH5. Os pontos de  $\mathcal{Q}$ , em preto na Figura 5.9(a), são os átomos de um pedaço da mesma proteína, porém, obtidos de uma estrutura ligeiramente diferente: estes átomos correspondem aos carbonos da cadeia principal do receptor do hormônio tireoideano ligado a outro hormônio sintético, conhecido como GC24. Como as duas estruturas são da mesma proteína, mas, associadas a diferentes hormônios, estas são ligeiramente diferentes. Portanto, não existe um subconjunto de  $\mathcal{P}$  que coincide exatamente com o conjunto  $\mathcal{Q}$ , mas, existe uma região em  $\mathcal{P}$  que se assemelha ao conjunto  $\mathcal{Q}$ . O objetivo aqui é identificar qual o conjunto de pontos da proteína que mais se assemelha aos pontos do fragmento. Em outras palavras, queremos saber se existe um padrão estrutural do tipo definido por  $\mathcal{Q}$  na estrutura definida por  $\mathcal{P}$ . Este tipo de problema é conhecido como problema do reconhecimento de enovelamentos de proteínas, que consiste em encontrar os padrões de organização dos átomos de uma proteína em outras.

Utilizamos uma estratégia “multistart”, já que, este tipo de problema possui muitos minimizadores locais. As variáveis do problema são os elementos que definem o deslocamento  $D$ : três variáveis para definição da translação e três variáveis para definição da rotação ao redor dos eixos das coordenadas. Seja  $\mathcal{B} \subset \mathbb{R}^3$  a menor caixa que contém a proteína  $\mathcal{P}$ . A aproximação inicial para o vetor de translação foi  $\xi - O$  onde  $O$  é o centro de gravidade de  $\mathcal{Q}$  e  $\xi$  é um ponto aleatório em  $\mathcal{B}$ . Os ângulos iniciais foram sorteados uniformemente entre 0 e  $2\pi$ .

A melhor solução foi obtida na terceira tentativa, após 0.19 segundos de tempo de CPU, incluindo impressões. A última execução do algoritmo irrestrito realizou 21 iterações. Logo, o Algoritmo 4.1 foi executado três vezes, encontrando pontos críticos nos dois primeiros casos. Na média, a distância entre os pontos deslocados de  $\mathcal{Q}$  e os

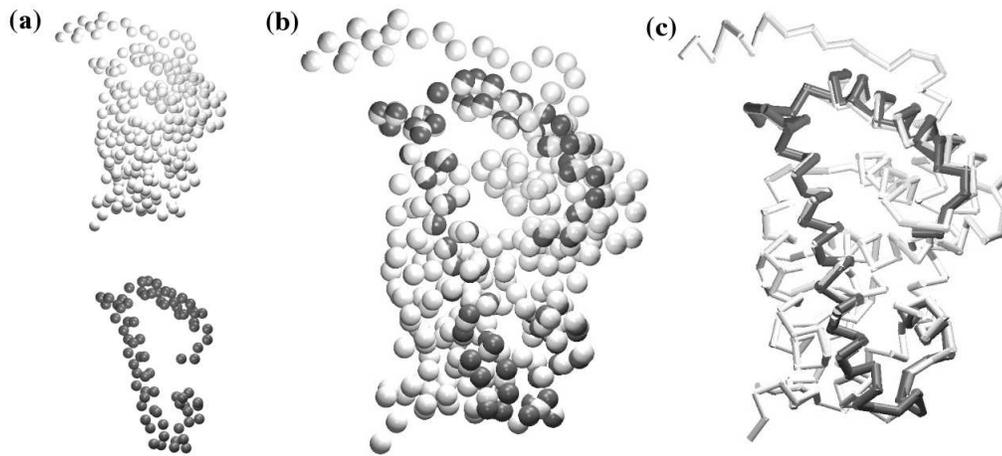


Figura 5.9: Encontrando padrões de uma proteína com LOVO.

pontos de  $\mathcal{P}$  foi de 1.07 angstroms (a melhor solução encontrada é correta do ponto de vista da função da proteína). Na Figura 5.9(b), mostramos a sobreposição dos pontos da melhor solução encontrada. Notamos que, mesmo quando o alinhamento é bom, seu reconhecimento não é óbvio. A Figura 5.9(c) mostra a mesma solução, mas, os átomos de carbono da cadeia principal da proteína estão conectados de acordo com sua sequência, mostrando com maior clareza o alinhamento obtido (o fragmento está em preto e a proteína está em cinza claro).

# Conclusão

Apresentamos o problema de Otimização de Valores Ordenados (OVO), um problema de otimização contínua, não-diferenciável e, geralmente, não-convexo. Este problema é uma generalização do problema Minimax e possui muitas aplicações, muitas delas ainda a serem descobertas. Neste trabalho, enfatizamos a aplicação de encontrar padrões ocultos e a otimização do VaR, especialmente, quando as funções de perda  $f_i$  são não-lineares e as restrições são lineares.

Para resolver o problema OVO apresentamos o algoritmo Cauchy-Primal, cuja implementação confia na resolução de subproblemas de programação linear e que tem funcionado satisfatoriamente. Introduzimos também um algoritmo do tipo quase-Newton para problemas OVO onde o conjunto viável  $\Omega$  é convexo. Ambos possuem resultados de convergência global no sentido de que todo ponto limite satisfaz uma condição de otimalidade. Para o algoritmo quase-Newton provamos convergência local (superlinear e quadrática) se algumas hipóteses adicionais são satisfeitas. O desempenho deste método foi qualitativamente superior ao desempenho do método Cauchy-Primal. Além disso, as hipóteses usadas para provar convergência local, apesar de serem bastante restritivas, parecem estar sendo satisfeitas, pelo menos para um exemplo em particular.

O problema OVO também é um potencial problema de otimização global. Os algoritmos locais para o OVO não são suficientemente eficientes para enfrentar problemas com múltiplos pontos críticos, portanto, definições de estratégias globais são necessárias (ver, por exemplo, [10]). Além disso, este parece ser um desafiante problema de otimização que generaliza a otimização diferenciável. Portanto, a expectativa é de realização de muita pesquisa tanto na introdução quanto na análise de métodos para este problema. Um campo promissor de pesquisa é o de adaptar algoritmos clássicos de otimização ao problema OVO (ver, por exemplo, [24]).

O problema de Otimização da Menor Soma de Valores Ordenados (LOVO) introduzido neste trabalho é, em geral, não-diferenciável e não-convexo. Este problema é uma generalização do problema clássico de quadrados mínimos. Por outro lado, este pode ser visto como um problema Minimin, ou seja, um caso particular do problema

OVO ( $p = 1$ ). Olhando por este ponto de vista, introduzimos algoritmos para resolver o LOVO com e sem restrições. Uma importante consequência da teoria, confirmada pelos experimentos, é que, ao contrário de muitos problemas não-diferenciáveis (mesmo sendo convexos) as consequências de ignorar a não-diferenciabilidade, ou seja, a multiplicidade dos gradientes em um determinado ponto  $x_k$ , não são tão severas. Resumindo, algoritmos de otimização diferenciável quando aplicados a este problema convergem para pontos críticos fracos e algoritmos específicos convergem para pontos críticos fortes. Isto permite que tiremos vantagem da disponibilidade de eficientes “softwares” de otimização diferenciável.

Aplicações em reconhecimento de padrões ocultos e estimação robusta de parâmetros parecem ser promissoras. O LOVO se mostra uma ferramenta muito interessante para atacar estes tipos de problemas que, por sua vez, são muito importantes em muitas áreas da Ciência e Engenharia. Indubitavelmente, na presença de aplicações tecnológicas específicas será necessário desenvolver algoritmos específicos, mas, a possibilidade de utilizar “softwares” gerais com resultados razoáveis é muito encorajadora.

Futuras pesquisas sobre este assunto podem incluir:

- Exploração de reformulações diferenciáveis tais como a proposta em [5] para o problema OVO.
- Adaptação e desenvolvimento de estratégias de otimização global para encontrar pontos iniciais adequados com o intuito de evitar a atratividade de minimizadores locais-não-globais.
- Desenvolvimento de algoritmos para o problema LOVO com restrições com convergência para pontos críticos fortes.
- Extensões da estratégia LOVO para o caso em que  $p$  não é pré-fixado.
- Problema de programação não-linear com restrições LOVO e OVO.
- Algoritmos de Programação Quadrática Sequencial, Pontos Interiores e Restauração para problemas LOVO com restrições não-lineares.

# Referências Bibliográficas

- [1] R. Andreani, E. G. Birgin, J. M. Martínez and M. L. Schuverdt. Augmented Lagrangian methods under the Constant Positive Linear Dependence constraint qualification. To appear in *Mathematical Programming*.
- [2] R. Andreani, E. G. Birgin, J. M. Martínez and M. L. Schuverdt. On Augmented Lagrangian methods with general lower-level constraints. Technical Report MCDO 050303, Department of Applied Mathematics, State University of Campinas, Brazil, 2005.
- [3] R. Andreani, E. G. Birgin, J. M. Martínez and J. Yuan. Spectral projected gradient and variable metric methods for optimization with linear inequalities. *IMA Journal on Numerical Analysis* 25, pp. 221-252 (2005).
- [4] R. Andreani, C. Dunder and J. M. Martínez. Order-Value Optimization: formulation and solution by means of a primal Cauchy method, *Mathematical Methods of Operations Research* 58, pp. 387-399 (2003).
- [5] R. Andreani, C. Dunder and J. M. Martínez. Nonlinear-Programming Reformulation of the Order-Value Optimization Problem, *Mathematical Methods of Operations Research* 61, pp. 365-384 (2005).
- [6] R. Andreani and J. M. Martínez. On the solution of mathematical programming problems with equilibrium constraints. *Mathematical Methods of Operations Research* 54, pp.345-358 (2001).
- [7] R. Andreani, J. M. Martínez, L. Martínez and F. Yano. Continuous Optimization Methods for Structure Alignments. To appear in *Mathematical Programming*.

- [8] R. Andreani, J. M. Martínez, L. Martínez and F. Yano. Low Order-Value Optimization and applications. Technical Report MCDO 051013, Department of Applied Mathematics, State University of Campinas, Brazil, October 2005.
- [9] R. Andreani, J. M. Martínez, M. Salvatierra and F. Yano. Quasi-Newton methods for order-value optimization and value-at-risk calculations. *Pacific Journal of Optimization* 2, pp. 11-33 (2006).
- [10] R. Andreani, J. M. Martínez, M. Salvatierra and F. Yano. Global Order-Value Optimization by means of a multistart harmonic oscillator tunneling strategy. Leo Liberti, Nelson Maculan (eds.), *Global Optimization: From Theory to Implementation*, Vol. 84 of the Nonconvex Optimization and Its Applications series, Springer, 2006.
- [11] R. Andreani, J. M. Martínez and M. L. Schuverdt. On the relation between the Constant Positive Linear Dependence condition and quasinormality constraint qualification, *Journal of Optimization Theory and Applications* 125, pp. 473–485 (2005).
- [12] D. P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, 1982; republished by Athena Scientific, 1996.
- [13] D. P. Bertsekas. *Nonlinear Programming*, 2nd edition, Athena Scientific, Belmont, Massachusetts, 1999.
- [14] E. G. Birgin and J. M. Martínez. Large-scale active-set box-constrained optimization method with spectral projected gradients, *Computational Optimization and Applications* 23, pp. 101-125 (2002).
- [15] E. Cam and J. Y. Monnat. Stratification based on reproductive state reveals contrasting patterns of age-related variation in demographic parameters in the kittiwake. *OIKOS* 90, pp. 560-574 (2000).
- [16] L. L. Cavalli-Sforza. Genes, peoples, and languages. *Proceedings of the National Academy of Sciences of the United States of America* 94, pp. 7719-7724 (1997).
- [17] A. R. Conn, N. I. M. Gould, A. Sartenaer and Ph. L. Toint. *Convergence properties of an Augmented Lagrangian algorithm for optimization with a combination of*

- general equality and linear constraints*, *SIAM Journal on Optimization* 6, pp. 674–703 (1996).
- [18] A. R. Conn, N. I. M. Gould and Ph. L. Toint. A globally convergent Augmented Lagrangian algorithm for optimization with general constraints and simple bounds, *SIAM Journal on Numerical Analysis* 28, pp. 545–572 (1991).
- [19] J. E. Dennis Jr. and J. J. Moré. A characterization of superlinear convergence and its application to quasi-Newton methods. *Mathematics of Computation* 28, pp. 549–560 (1974).
- [20] Z. Dostál. Inexact semi-monotonic Augmented Lagrangians with optimal feasibility convergence for convex bound and equality constrained quadratic programming. To appear in *SIAM Journal on Numerical Analysis*.
- [21] Z. Dostál, A. Friedlander and S. A. Santos. Augmented Lagrangian with adaptive precision control for quadratic programming with simple bounds and equality constraints, *SIAM Journal on Optimization* 13, pp. 1120–1140 (2003).
- [22] Z. Dostál, F. A. M. Gomes and S. A. Santos. Duality based domain decomposition with natural coarse space for variational inequalities, *Journal of Computational and Applied Mathematics* 126, pp. 397–415 (2000).
- [23] X. Espadaler and C. Gómez. The species body-size distribution in Iberian ants is parameter independent. *Vie et Milieu* 52, pp. 103–107 (2002).
- [24] R. Fletcher. *Practical Methods of Optimization*, Academic Press, London, 1987.
- [25] R. Gronner and L. Manheim. Hidden patterns. *Arch. Gen. Psychiat.* 17, pp. 248 (1967).
- [26] W. W. Hager. Analysis and implementation of a dual algorithm for constrained optimization, *Journal of Optimization Theory and Applications* 79, pp. 37–71 (1993).
- [27] M. R. Hestenes. Multiplier and gradient methods, *Journal of Optimization Theory and Applications* 4, pp. 303–320 (1969).
- [28] L. Holm and C. Sander. Mapping the Protein Universe, *Science* 273, pp. 595–602 (1996).

- [29] P. J. Huber. *Robust Statistics*. Wiley, New York, 1981.
- [30] P. Jorion. *Value at risk: the new benchmark for managing financial risk*, 2nd edition. Mc Graw-Hill, New York, 2001.
- [31] H. C. Koh and S. K. Leong. Data mining applications in the context of casemix. *Annals Academy of Medicine Singapore* 30, pp. 41-49 (2001).
- [32] O. L. Mangasarian and S. Fromovitz. The Fritz-John necessary optimality conditions in presence of equality and inequality constraints, *Journal of Mathematical Analysis and Applications* 17, pp. 37-47 (1967)
- [33] M. May and L. Ragia. Spatial subgroup discovery applied to the analysis of vegetation data. *Lecture Notes in Artificial Intelligence* 2569, pp. 49-61 (2002).
- [34] J. J. Moré, B. S. Garbow and K. E. Hillstrom. Testing unconstrained optimization software, *ACM Transactions on Mathematical Software* 7, pp. 17-41 (1981).
- [35] M. J. D. Powell. A method for nonlinear constraints in minimization problems, in *Optimization*, R. Fletcher (ed.), Academic Press, New York, NY, pp. 283-298 (1969).
- [36] L. Qi and Z. Wei. On the constant positive linear dependence condition and its application to SQP methods, *SIAM Journal on Optimization* 10, pp. 963-981 (2000).
- [37] R. T. Rockafellar. The multiplier method of Hestenes and Powell applied to convex programming, *Journal of Optimization Theory and Applications* 12, pp. 555-562 (1973).
- [38] R. T. Rockafellar. Augmented Lagrange multiplier functions and duality in non-convex programming, *SIAM Journal on Control* 12, pp. 268-285 (1974).
- [39] R. T. Rockafellar. Lagrange multipliers and optimality, *SIAM Review* 35, pp. 183-238 (1993).
- [40] R. T. Rockafellar and S. Uryasev. Optimization of conditional value-at-risk. *Journal of Risk* 2, pp. 21-41 (2000).

- 
- [41] R. T. Rockafellar and S. Uryasev. Conditional value-at-risk for general loss distributions, *Journal of Banking and Finance* 26, pp. 1443-1471 (2002).
- [42] L. T. Santos, F. Yano, M. Salvatierra, J. M. Martínez, R. Andreani and M. Tygel. A global optimization algorithm applied to the common reflection surface (CRS) problem. *Journal of Seismic Exploration* 14, pp. 217-233 (2005).
- [43] K. Schittkowski. The non-linear programming method of Wilson, Hand and Powell with an augmented Lagrangian type line search function. 1. Convergence Analysis. *Numerische Mathematik* 38, pp. 83-114 (1981).
- [44] K. Schittkowski. The non-linear programming method of Wilson, Hand and Powell with an augmented Lagrangian type line search function. 2. An efficient implementation with linear least-squares subproblems. *Numerische Mathematik* 38, pp. 115-127 (1981).
- [45] J. E. Stiglitz and B. Greenwald. *Towards a new paradigm in Monetary Economics*. Cambridge University Press, Cambridge, 2003.
- [46] P. Street. The logic and limits of “plant loyalty”: Black workers, white labor, and corporate racial paternalism in Chicago’s stockyards, 1916-1940. *Journal of Social History* 29, pp. 659-& (1996).
- [47] J. W. Weaver. Hidden patterns in Joyce ‘Portrait of the artist as a young man’. *South Atlantic Bulletin* 41, pp. 63-63 (1976).