

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE MATEMÁTICA ESTATÍSTICA E COMPUTAÇÃO CIENTÍFICA
Departamento de Matemática Aplicada e Computacional

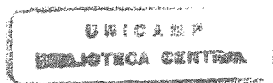
Estudo de Estruturas Especiais Para Aproximação da Matriz Hessiana em Problemas de Minimização em Caixas

Luíz Carlos Neto

Profa. Dra. Maria Aparecida Diniz Ehrhardt
Orientadora

Dissertação apresentada junto ao Departamento de Matemática Aplicada e Computacional do Instituto de Matemática Estatística e Computação Científica da Universidade Estadual de Campinas, para obtenção do título de Mestre em Matemática Aplicada e Computacional.

Campinas
2001



UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE

UNIDADE 3e
N.º CHAMADA: T/UNICAMP
C195e
V. Ex.
TOMBO BC/ 46876
PROC. 16.392/07
C D
PREC. R\$ 11,00
DATA 27/10/07
N.º CPD

CM00161203-2

Estudo de Estruturas Especiais Para Aproximação da Matriz Hessiana em Problemas de Minimização em Caixas

Este exemplar corresponde à redação final da dissertação devidamente corrigida e defendida por Luíz Carlos Neto e aprovada pela comissão julgadora.

Campinas, 20 de Agosto de 2001.



Maria Aparecida Diniz Ehrhardt
Orientadora

Banca Examinadora:

1. Profa. Dra. Maria Aparecida Diniz Ehrhardt – Orientadora – IMECC/UNICAMP
2. Prof. Dr. Jose Mario Martinez – IMECC/UNICAMP
3. Prof. Dr. Ernesto Birgin – IME/USP
4. Profa. Dra. Márcia Aparecida Gomes Ruggiero – Suplente – IMECC/UNICAMP

Dissertação apresentada ao Instituto de Matemática, Estatística e Computação Científica, UNICAMP, como requisito parcial para obtenção do título de MESTRE em MATEMÁTICA APLICADA E COMPUTACIONAL.

00124372

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP

Neto, Luíz Carlos
M858M Estudo de Estruturas Especiais Para Aproximação da Matriz
Hessiana em Problemas de Minimização Irrestrita /Luíz Carlos Neto. –
Campinas, [S.P. :s.n.], 2001.

Orientadora: Maria Aparecida Diniz Ehrhardt.
Dissertação (mestrado) – Universidade Estadual de Campinas, Insti-
tuto de Matemática, Estatística e Computação Científica.

1. Aproximações quase-Newton para as Hessianas 2. Problemas de
minimização em caixas I. Ehrhardt, Maria Aparecida Diniz. II. Uni-
versidade Estadual de Campinas, Instituto de Matemática, Estatística
e Computação Científica. III. Título.

Dissertação de Mestrado defendida em 13 de agosto de 2001 e aprovada

Pela Banca Examinadora composta pelos Profs. Drs.

maria aparecida diniz ehrhardt

Prof (a). Dr (a). MARIA APARECIDA DINIZ EHRHARDT

josé mario martínez pérez

Prof (a). Dr (a). JOSÉ MARIO MARTÍNEZ PÉREZ

ernesto julian goldberg birgin

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

maria aparecida diniz ehrhardt

Resumo

Muitos problemas reais podem ser representados ou aproximados como um problema de programação não-linear, onde a função objetivo e/ou as restrições são não-lineares. Dentre estes podemos citar problemas de controle ótimo de produção e estoque, desenho de estruturas mecânicas, otimização de redes elétricas, modelos de risco de mercado, entre outros (ver [1]).

Destes problemas, considerou-se aqueles onde as variáveis são canalizadas. Para sua resolução, estudou-se dois algoritmos: **BOX-QUACAN**, proposto por Friedlander, Martínez e Santos [13], do tipo região de confiança, e **L-BFGS-B**, de Byrd, Lu, Nocedal e Zhu [3], que trabalha com busca linear.

O enfoque deste estudo está na aproximação da matriz Hessiana, necessária em ambos os códigos.

O trabalho foi feito com o intuito de se obter resultados mais conclusivos em relação à performance de **BOX-QUACAN** com as aproximações secantes de banda para a Hessiana (**BOX-QUACAN Modificado**). Assim, os resultados numéricos de **BOX-QUACAN Modificado** foram comparados com os de **L-BFGS-B** juntamente com o **EASY**, uma versão de **BOX-QUACAN** que trabalha com diferenças finitas para aproximar a Hessiana.

Agradecimentos

São tantas as pessoas para eu lembrar neste espaço que corro o risco de me esquecer de alguém. Espero que isto não ocorra.

Em um trabalho como este, o papel do(a) orientador(a) é fundamental. A começar pelo fato do(a) orientador(a) acreditar no seu trabalho e indo até o momento final do trabalho. Quando este(a) orientador(a) tem uma incrível paciência para mostrar os erros e ainda incentivar seu orientando(a), o desempenho do trabalho melhora muito. Por tudo isto, o mínimo que posso fazer é agradecer, e muito, minha orientadora, Profa. Dra. Maria Aparecida Diniz Ehrhardt, a profa. Cheti.

Não posso deixar de agradecer àqueles que, à distância ou próximos, sempre torceram por mim. Estou me referindo à minha mãe, ao meu pai, às minhas irmãs, Gorete e Lucimara, e ao meu irmão, Thales. Espero que, a partir de agora, possamos viver de maneira mais digna e menos sofrida. Vocês sabem que sempre podem contar comigo.

Agradeço àqueles que foram muito importantes no meu aprendizado, os meus professores, os “mestres”.

Amigos. Como dizia Milton Nascimento “Amigo é coisa para se guardar, do lado esquerdo do peito”. Da época da G5, grandes amigos das grandes gargalhadas, Carlos Eduardo “Baiano”, Leandro “Bandeirão”, Leduino “Lôlo”. Na E2A, grandes amigos, um pouco mais sério, Anderson “Paulista”, Anderson “Carioca”, Vitor e Cláudio, “O Grande”. Não posso também de esquecer “pessoas ilustres” como Marcelo “Jack”, Max “Noel”, Caubim “Pobre Ousado”, Flávia, Soninha, Wally, Amarélisio, Nilo, Morcegão, Batata, Renatão, Baka. Também, Mariana “Princesinha do

IMECC”, Edmilson “Ed da Mulherada”, Italo “Bugrino”, Marcelo “Dandan”, Daniela, Flávia & Luck, Sérgio Ventura, Júlia Victoria, Lucy, Rosana, Benaia, Roberto “Picareta” Carlos, Fábio “Marido” e Leonardo. Da galera de Passos, grandes amigos, Cláudio, Alex e cia. Deixo aqui o meu abraço.

Agradeço também a todos os brasileiros e brasileiras, os quais financiaram todo o meu estudo. Em especial, aos excluídos, classe que já fiz parte, que vivem neste país sem ter o que comer nem onde morar, onde um animal é mais bem tratado que um ser humano. Deixo aqui o meu grande abraço.

Enfim, não posso de deixar de agradecer a Deus. Se algum dia haver justiça, esta será feita por Ele. Que Ele continue guiando meus passos.

Sumário

Resumo	v
Agradecimentos	vi
1 Introdução	1
2 Um Algoritmo de Região de Confiança para Minimização em Caixas	4
2.1 Introdução	4
2.2 Métodos com estr. glob. para problemas com rest. de canalização . .	4
2.3 O algoritmo BOX-QUACAN	6
3 Atualizações Quase-Newton com Estrutura de Banda	10
3.1 Introdução	10
3.2 Atualizações secantes de variação mínima	11
3.3 Métodos secantes esparsos	12
3.4 Aproximações de banda para as Hessianas	16
3.5 Um breve comentário sobre convergência	20
4 Aproximações BFGS com Memória Limitada	22
4.1 Introdução	22
4.2 O algoritmo	22
4.3 Aproximações BFGS com memória limitada	25
4.4 Implementação do MML	27

4.5	O Ponto de Cauchy generalizado	28
4.6	Um método direto para minimização no subespaço das var. livres . . .	31
5	Experimentos Numéricos	34
5.1	Apresentação dos problemas	35
5.2	Definição de parâmetros e critérios de parada	36
5.3	Resultados numéricos	39
5.3.1	Tabelas gerais	39
5.3.2	Problemas irrestritos	41
5.3.3	Problemas canalizados	42
5.4	Um exemplo de aplicação	45
5.5	Comentários e conclusões	46
	Bibliografia	51

Capítulo 1

Introdução

Muitos problemas reais podem ser representados ou aproximados como um problema de programação não-linear, onde a função objetivo e/ou as restrições são não-lineares. Dentre estes podemos citar problemas de controle ótimo de produção e estoque, desenho de estruturas mecânicas, otimização de redes elétricas, modelos de risco de mercado, entre outros (ver [1]).

Destes problemas, considerou-se aqueles onde as variáveis são canalizadas, ou seja, que podem ser modelados por

$$\begin{aligned} &\text{Minimizar } f(x) \\ &\text{sujeita a } l \leq x \leq u, \end{aligned} \tag{1.0.1}$$

onde $f : \mathbb{R}^n \rightarrow \mathbb{R}$ é uma função não-linear, seu gradiente g está disponível, os vetores l e u representam os limites inferior e superior das variáveis (podem ser $-\infty$ e/ou $+\infty$, respectivamente), e o número n de variáveis é possivelmente grande (problemas de grande porte).

Estudou-se dois algoritmos para a resolução do problema (1.0.1): BOX-QUACAN, proposto por Friedlander, Martínez e Santos [13], do tipo região de confiança, e L-BFGS-B, de Byrd, Lu, Nocedal e Zhu [3], que trabalha com busca linear.

O enfoque deste estudo está na aproximação da matriz Hessiana, necessária em ambos os algoritmos.

BOX-QUACAN permite que o usuário defina a forma desejada para aproximar esta matriz. Optou-se, então, por trabalhar com aproximações secantes de variação mínima que mantêm uma estrutura de banda, com o objetivo de economizar memória, tornando o método adequado para problemas de grande porte. Esta forma de aproximação da Hessiana foi introduzida em [16] onde, além de resultados de convergência, alguns experimentos numéricos foram apresentados. No entanto, uma análise mais aprofundada do desempenho computacional da versão proposta não foi feita.

Neste trabalho, com o intuito de se obter resultados mais conclusivos em relação à performance de BOX-QUACAN com as aproximações secantes de banda para a Hessiana, experimentos numéricos foram realizados e comparados com o *software* L-BFGS-B, que aproxima a Hessiana pela fórmula BFGS [9] usando técnicas de memória limitada. Assim, selecionou-se um conjunto de problemas, os quais foram resolvidos por métodos que têm em comum, fundamentalmente, o uso de aproximações Quase-Newton para a Hessiana, que permitem, de formas diferentes, economia de memória computacional.

Além disso, considerou-se importante avaliar o desempenho destes algoritmos, que utilizam atualizações secantes para a Hessiana, com uma versão de BOX-QUACAN que trabalha com diferenças finitas para aproximar as derivadas parciais de segunda ordem, implementada em EASY. De fato, em EASY, aproxima-se o produto Bu (onde $B \approx \nabla^2 f(x)$ e u é um vetor direção) pelo quociente $\frac{\nabla f(x+tu) - \nabla f(x)}{t}$, onde $t = \frac{\max(10^{-20}, 10^{-8} \|x\|_\infty)}{\|u\|_\infty}$, dado que $u \neq 0$.

Segue então, no capítulo 2, a descrição do algoritmo BOX-QUACAN como um algoritmo de região de confiança para minimização em caixas.

No capítulo 3, faz-se uma discussão sobre as matrizes de banda obtidas através de atualizações secantes de variação mínima, que são utilizadas para aproximar a Hessiana no algoritmo descrito no capítulo 2.

O algoritmo L-BFGS-B é apresentado no capítulo 4, onde são definidas as aproximações BFGS com memória limitada.

Finalmente, o capítulo 5 descreve os experimentos numéricos. São apresentados os problemas e os critérios de parada usados nos algoritmos. Através de tabelas contendo os resultados obtidos para cada um dos três algoritmos, para cada um dos problemas, fêz-se uma análise geral do desempenho dos algoritmos abordados.

Capítulo 2

Um Algoritmo de Região de Confiança para Minimização em Caixas

2.1 Introdução

Descreve-se a seguir, uma idéia geral de duas estratégias globalizadoras para problemas com restrições de canalização: a busca unidimensional e a região de confiança. É apresentado também o algoritmo BOX-QUACAN do tipo região de confiança, introduzido por Friedlander, Martínez e Santos [13] para resolver o problema (1.0.1).

2.2 Métodos com estratégias globalizadoras para problemas com restrições de canalização

Definem-se métodos locais como aqueles que convergem se a aproximação inicial estiver suficientemente próxima de um ponto estacionário da função f . No entanto, apesar destes métodos serem muito eficientes, eles falham para determinados problemas. Para contornar esta situação, são utilizadas estratégias para induzir o método

a convergir independentemente da aproximação inicial. Esse tipo de método é chamado de globalmente convergente. Entre as estratégias aplicadas para obtenção de convergência global, estão aquelas que utilizam técnicas de busca unidimensional e aquelas baseadas em regiões de confiança. Nos dois procedimentos que serão descritos, o método é induzido a convergir pela escolha da direção e do tamanho do passo que produzam um efetivo decréscimo da função objetivo f em cada iteração.

Quando se utiliza estratégia de busca unidimensional, determina-se inicialmente uma direção d a partir de x_k , tal que $\nabla f(x_k)^\top d < 0$ (direção de descida) e, então, calcula-se um tamanho de passo $\lambda_k > 0$ de modo que $f(x_k + \lambda_k d)$ seja “suficientemente menor” que $f(x_k)$. Se a matriz Hessiana B_k , ou uma aproximação dela, é definida positiva (ou seja, $v^\top B_k v > 0$, para todo vetor $v \in \mathbb{R}^n$, v não nulo), então uma direção de descida d é dada pela solução do sistema linear $B_k d = -\nabla f(x_k)$.

Quando as aproximações das Hessianas não são definidas positivas, a técnica de busca unidimensional não é adequada devido à dificuldade em determinar-se direções de descida. No capítulo 4, descreve-se o algoritmo **L-BFGS-B**, que utiliza busca unidimensional, desde que mantém aproximações definidas positivas para a Hessiana.

Nos métodos que trabalham com região de confiança, define-se primeiramente uma aproximação quadrática para a função objetivo e uma região ao redor do iterando atual na qual acredita-se que a aproximação seja adequada. Dessa forma, encontra-se um candidato factível para a próxima iteração, nesta região. Este candidato é tomado tal que haja uma redução suficiente no valor do modelo quadrático da função objetivo. Se o valor da função objetivo, calculado neste ponto, satisfaz a condição de decréscimo suficiente, o novo ponto é aceito como o próximo iterando, podendo a região ser aumentada ou não. Se a condição de decréscimo suficiente não é satisfeita,

o ponto é rejeitado e o tamanho da região de confiança é reduzido.

Na k -ésima iteração, constrói-se o modelo quadrático para o decréscimo da função objetivo na vizinhança do ponto x_k definido por

$$f(x_k + s) - f(x_k) \approx q_k(s) \equiv \frac{1}{2}s^\top B_k s + g_k^\top s, \quad (2.2.1)$$

onde $g_k = \nabla f(x_k)$ e B_k é uma aproximação simétrica para a matriz Hessiana $\nabla^2 f(x_k)$.

Denotando-se $\mathcal{B} = \{x \in \mathbb{R}^n \mid l \leq x \leq u\}$ como o conjunto factível, define-se a região de confiança como

$$\mathcal{B}_\Delta = \{s \in \mathbb{R}^n \mid x_k + s \in \mathcal{B}, \ \|Ds\| \leq \Delta\}, \quad (2.2.2)$$

onde Δ é um escalar positivo representando o raio da região de confiança, $\|\cdot\|$ é uma norma arbitrária em \mathbb{R}^n e D é uma matriz de escalamento relacionada com os limites da região factível.

O subproblema então a ser resolvido é

$$\min_{s \in \mathcal{B}_\Delta} q_k(s). \quad (2.2.3)$$

Não é necessário que a solução de (2.2.3) seja obtida de forma exata para que se tenha convergência global do algoritmo principal, como será visto ao longo deste capítulo.

Descreve-se abaixo o algoritmo BOX-QUACAN.

2.3 O algoritmo BOX-QUACAN

Descreve-se nesta seção, em linhas gerais, o algoritmo BOX-QUACAN para a resolução do problema (1.0.1), o qual é um método de região de confiança, como descrito na seção 2.2. Deve-se ressaltar que a matriz B_k , que aproxima $\nabla^2 f(x_k)$, não é necessariamente definida positiva.

Inicialização.

Sejam σ_1 , σ_2 , α , Δ_{min} , tais que $0 < \sigma_1 \leq \sigma_2 < 1$, $\alpha \in (0, 1)$ e $\Delta_{min} > 0$. Considere inicialmente um ponto factível x_0 , uma matriz simétrica B_0 (a aproximação da Hessiana), uma matriz $D_0 \in \mathbb{R}^{n \times n}$ (a matriz de escalamento) e um raio $\Delta^0 \geq \Delta_{min}$. Em uma iteração k , dados x_k factível, B_k simétrica, D_k não-singular e $\Delta^k \geq \Delta_{min}$, obtém-se x_{k+1} realizando-se os passos a seguir.

Passo 1 (Define o raio inicial da região de confiança).

Faça $\Delta \leftarrow \Delta^k$.

Passo 2 (Calcula o passo preliminar).

Calcule $\bar{s}_k(\Delta)$ resolvendo aproximadamente o subproblema

$$\begin{aligned} \min \quad & q_k(s) \\ \text{s. a} \quad & l \leq x_k + s \leq u \\ & \|D_k s\|_\infty \leq \Delta, \end{aligned} \tag{2.3.1}$$

com $q_k(s)$ definida como em (2.2.1).

Passo 3 (Testa o decréscimo suficiente).

Se

$$f(x_k + \bar{s}_k(\Delta)) \leq f(x_k) + \alpha q_k(\bar{s}_k(\Delta)) \tag{2.3.2}$$

então defina $s_k = \bar{s}_k(\Delta)$, $x_{k+1} = x_k + s_k$, $\Delta_k = \Delta$ e pare. Caso contrário, $\Delta \leftarrow \Delta_{new} \in [\sigma_1 \|D_k \bar{s}_k(\Delta)\|_\infty, \sigma_2 \Delta]$ e repita o passo 2.

Para se obter a solução aproximada de (2.3.1), utiliza-se um algoritmo que minimiza quadráticas em uma caixa, denominado QUACAN (ver [2] e [13]). Este combina iterações de gradientes conjugados e gradientes projetados nas diferentes faces de seu

domínio. Como o algoritmo está voltado para problemas de grande porte, ele utiliza somente produtos de matriz-vetor; nenhuma fatoração de matrizes é necessária.

Dois parâmetros usados em QUACAN têm papel importante no desempenho de BOX e merecem ser destacados. São os parâmetros $\tau \in (0, 1)$, usado no critério de parada, e $\eta \in (0, 1)$, usado no critério para abandonar uma face da caixa.

O critério de parada utilizado em QUACAN é dado por

$$\|P[\nabla q_k(s), \mathcal{B}_{\Delta_k}]\|_2 \leq \tau \|P[\nabla q_k(0), \mathcal{B}_{\Delta_k}]\|_2, \quad (2.3.3)$$

onde $P[x, \mathcal{S}]$ denota a projeção de x no conjunto \mathcal{S} e τ é a precisão desejada. No capítulo 5, onde se colocam os experimentos numéricos realizados, é feito um estudo sobre o valor usado para τ neste trabalho.

Com relação ao critério de abandono de uma face da caixa, isto ocorrerá, em uma dada iteração j ($j = 0, 1, \dots$) do “solver” quadrático, se a desigualdade

$$\|g_c(x_j)\| > \eta \|g_p(x_j)\|,$$

for satisfeita, onde g_c , denominado “gradiente chopado”, é a componente de g_p que fornece a direção de busca para deixar a face, como pode ser visto em [13] e [2]. Se $\eta \approx 1$, as faces são mais exploradas antes de seu abandono. Se $\eta \approx 0$, as faces são mais frequentemente abandonadas. Seguindo sugestões de trabalhos anteriores, como [2] e [12], usou-se, neste trabalho, $\eta = 0.9$.

Como foi aqui descrito, o *software* BOX-QUACAN implementa um método de região de confiança para minimização em caixas. Seus resultados gerais de convergência podem ser encontrados em [13]. Resumidamente, sobre a convergência global do algoritmo, pode-se colocar que, sob certas hipóteses, como continuidade das derivadas parciais de primeira ordem e supondo que as aproximações para as Hessianas, em cada

iteração k , são limitadas, todo ponto limite da seqüência gerada pelo método é um ponto estacionário (Karush-Kuhn-Tucker) de (1.0.1).

Capítulo 3

Atualizações Quase-Newton com Estrutura de Banda

3.1 Introdução

Na busca de uma solução para o problema

$$\begin{aligned} \min \quad & f(x) \\ \text{s. a} \quad & x \in \Omega \subseteq \mathbb{R}^n \end{aligned} \tag{3.1.1}$$

onde $f : \mathbb{R}^n \rightarrow \mathbb{R}$, é desejável que as matrizes Hessianas sejam aproximadas, pois seu cálculo exato pode não ser possível ou ser custoso computacionalmente. Uma das estratégias utilizadas é o uso dos métodos secantes, como descreve-se abaixo.

Considere a solução exata x_* para o problema (3.1.1). A uma dada iteração k , seja x_k uma estimativa para x_* e B_k uma aproximação para $\nabla^2 f(x_k)$. A iteração seguinte determina uma melhor estimativa x_{k+1} para x_* e a matriz B_k é atualizada, obtendo-se uma aproximação B_{k+1} de $\nabla^2 f(x_{k+1})$. Estas matrizes devem satisfazer a equação secante

$$B_{k+1}s_k = y_k, \tag{3.1.2}$$

onde $s_k, y_k \in \mathbb{R}^n$, $s_k \neq 0$, são definidos por

$$s_k = x_{k+1} - x_k, \quad y_k = \nabla f(x_{k+1}) - \nabla f(x_k). \quad (3.1.3)$$

Do ponto de vista computacional, é importante que se tire proveito da estrutura do problema para a redução do gasto de memória com armazenamento e do número de operações, tornando o algoritmo mais eficiente.

Em problemas de grande porte, as matrizes Hessianas são, em geral, esparsas. Assim, é desejável que as aproximações para as Hessianas sejam esparsas e simétricas, e que satisfaçam a equação secante (3.1.2). Uma técnica para obter estas aproximações, utilizando atualização secante de variação mínima, será descrita na próxima seção.

3.2 Atualizações secantes de variação mínima

Descreve-se nesta seção a idéia de atualização das matrizes Hessianas, na qual procura-se minimizar a diferença entre a aproximação atual e a anterior, satisfazendo a equação secante (3.1.2).

A equação secante pode ser motivada a partir das propriedades satisfeitas por uma função quadrática. Seja $q(x) = \frac{1}{2}x^\top Bx + b^\top x + c$, $B \in \mathbb{R}^{n \times n}$ simétrica, $x, b \in \mathbb{R}^n$, $c \in \mathbb{R}$. Então, $\nabla q(x) = Bx + b$ e $\nabla q(x+d) = B(x+d) + b$, $\forall d \in \mathbb{R}^n$. Portanto

$$\nabla q(x+d) - \nabla q(x) = B(x+d) - Bx = Bd. \quad (3.2.1)$$

Pode-se impor, então, que a matriz B_{k+1} satisfaça a igualdade (3.2.1) da seguinte forma:

$$\nabla f(x_{k+1}) - \nabla f(x_k) = B_{k+1}(x_{k+1} - x_k), \quad (3.2.2)$$

gerando a equação secante

$$B_{k+1}s_k = y_k.$$

Considere agora o conjunto de matrizes dado por

$$Q(y, s) = \{M \in \mathbb{R}^{n \times n} \mid Ms = y\}. \quad (3.2.3)$$

Deseja-se resolver o seguinte problema

$$\begin{aligned} \min \quad & \|B - B_k\|_F \\ \text{s. a} \quad & B \in Q(y_k, s_k), \end{aligned} \quad (3.2.4)$$

onde $\|\cdot\|_F$ é definida como a norma de Frobenius e é dada por $\|B\|_F = (\sum_{i=1}^n \sum_{j=1}^n b_{ij}^2)^{1/2}$.

Para obter a solução de (3.2.4), enuncia-se o lema e o teorema a seguir, cujas demonstrações podem ser vistas em [9].

Lemma 3.2.1. *Sejam $\alpha \in \mathbb{R}$, $v \in \mathbb{R}^n$, $v \neq 0$. Então a única solução para*

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \|x\|_2 \\ \text{s. a} \quad & v^\top x = \alpha \end{aligned}$$

é

$$x = \alpha v / \langle v, v \rangle.$$

Teorema 3.2.2. *Sejam $B_k \in \mathbb{R}^{n \times n}$, $s_k, y_k \in \mathbb{R}^n$, $s \neq 0$, e $Q(y, s)$ como em (3.2.3). Então a única solução para*

$$\min_{B \in Q(y_k, s_k)} \|B - B_k\|_F$$

é

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k) s_k^\top}{\langle s_k, s_k \rangle}. \quad (3.2.5)$$

A equação (3.2.5) é conhecida como atualização secante de Broyden ou secante de variação mínima.

3.3 Métodos secantes esparsos

Desenvolve-se nesta seção uma breve teoria sobre métodos secantes esparsos, que têm como objetivo manter uma dada estrutura de esparsidade nas atualizações secantes.

Seja $Z \in \mathbb{R}^{n \times n}$ uma matriz cujos elementos são 0 ou 1.

Seja $SP(Z)$ o subespaço das matrizes $n \times n$ com o padrão de zeros de Z , definido por

$$SP(Z) = \{M \in \mathbb{R}^{n \times n} : M_{ij} = 0 \text{ se } Z_{ij} = 0, 1 \leq i, j \leq n\}. \quad (3.3.1)$$

Supondo que $B_k \in SP(Z)$, formula-se o seguinte problema:

$$\begin{aligned} \min_{B \in \mathbb{R}^{n \times n}} \quad & \|B - B_k\|_F \\ \text{s. a} \quad & Bs = y, B \in SP(Z), \end{aligned} \quad (3.3.2)$$

onde, para simplificar a notação, usou-se s e y para representar s_k e y_k , respectivamente.

Aqui, deseja-se que B satisfaça a equação secante e tenha o mesmo padrão de esparsidade de Z . Ou seja, temos uma restrição a mais no problema definido em (3.2.4).

Seja $N \in \mathbb{R}^{n \times n}$ e seja a matriz do operador de projeção $P_Z : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}$ dada por

$$(P_Z(N))_{ij} = \begin{cases} 0, & \text{se } Z_{ij} = 0, \\ N_{ij}, & \text{se } Z_{ij} = 1. \end{cases} \quad (3.3.3)$$

Para $v \in \mathbb{R}^n$, defina $v_i \in \mathbb{R}^n$ por

$$(v_i)_j = \begin{cases} 0, & \text{se } Z_{ij} = 0, \\ v_j, & \text{se } Z_{ij} = 1. \end{cases} \quad (3.3.4)$$

Assim, enuncia-se o seguinte teorema:

Teorema 3.3.1. *Sejam $Z \in \mathbb{R}^{n \times n}$ uma matriz 0–1, $SP(Z)$ definido por (3.3.1) e P_Z por (3.3.3). Considere $B_k \in SP(Z)$, $s, y \in \mathbb{R}^n$, e defina s_i , $i = 1, \dots, n$, por (3.3.4). Se $s_i = 0$ somente quando $y_i = 0$, então a solução para (3.3.2) é a atualização secante esparsa*

$$B_{k+1} = B_k + P_Z(D^+(y - B_k s)s^\top), \quad (3.3.5)$$

onde $D^+ \in \mathbb{R}^{n \times n}$ é definida por

$$D^+ = \begin{cases} d_{ii}^+ = \begin{cases} \frac{1}{s_i^\top s_i}, & \text{se } s_i \neq 0, \\ 0, & \text{se } s_i = 0, \end{cases} \\ d_{ij}^+ = 0, & \text{se } i \neq j, \end{cases} \quad (3.3.6)$$

Prova: ver [10]

Seja \mathcal{A} um subespaço afim definido por propriedades tais como simetria e esparsidade. Dada uma matriz $B \in \mathcal{A}$, reformula-se o problema (3.3.2) como

$$\begin{aligned} \min_{B \in \mathbb{R}^{n \times n}} \|B - B_k\|_F \\ \text{s. a } B \in \mathcal{A} \cap Q(y, s). \end{aligned} \quad (3.3.7)$$

Neste caso, B deve pertencer a $\mathcal{A} \cap Q(y, s)$. Pode-se ter $\mathcal{A} \cap Q(y, s) \neq \emptyset$ ou $\mathcal{A} \cap Q(y, s) = \emptyset$. Encontra-se então uma matriz B em \mathcal{A} que é a projeção de variação mínima de B_k em \mathcal{A} mais próxima de $Q(y, s)$.

Definindo-se $B_0 \in \mathcal{A}$ e o subespaço paralelo a \mathcal{A} , $\Upsilon = \{B - B_0 : B \in \mathcal{A}\}$, a solução do problema (3.3.7) é dada pelo

Teorema 3.3.2. *Seja $s \neq 0$ e sejam $P_{\mathcal{A}}$ e P_{Υ} projetores ortogonais em \mathcal{A} e Υ , respectivamente. Seja \mathcal{P} uma matriz $n \times n$ cuja j -ésima coluna é*

$$\left[P_{\Upsilon} \left(\frac{e_j s^\top}{s^\top s} \right) \right] s \quad (3.3.8)$$

e seja $B_k \in \mathcal{A}$. Se v é uma solução qualquer para

$$\min_{v \in \mathbb{R}^n} \| \mathcal{P}v - (y - B_k s) \|_2 \quad (3.3.9)$$

ou equivalentemente para

$$\min_{v \in \mathbb{R}^n} \left\| P_{\Upsilon} \left(\frac{v s^\top}{s^\top s} \right) s - (y - B_k s) \right\|_2, \quad (3.3.10)$$

então

$$B_{k+1} = B_k + P_{\Upsilon} \left(\frac{v s^\top}{s^\top s} \right) \quad (3.3.11)$$

é a matriz em \mathcal{A} mais próxima de B_k , dentre as matrizes de \mathcal{A} que estão a uma distância mínima de $Q(y, s)$. Se o mínimo é zero, então $B_{k+1} \in \mathcal{A} \cap Q(y, s)$.

Prova: ver [9].

Deriva-se de (3.3.9) e (3.3.11) a expressão

$$B_{k+1} = B_k + P_{\Upsilon} \left(\frac{\mathcal{P}^+(y - B_k s) s^\top}{s^\top s} \right), \quad (3.3.12)$$

onde \mathcal{P}^+ denota a pseudoinversa de \mathcal{P} (ver em [10]).

Usando-se o teorema 3.3.2, definindo-se $\mathcal{A} = \Upsilon = SP(Z)$ e $P_{\mathcal{A}} = P_{\Upsilon} = P_Z$ dadas por (3.3.3), considerando-se

$$P_{.j} = \left[P_Z \left(\frac{e_j s^\top}{s^\top s} \right) \right] s = \frac{\mathbf{s}_j^\top s}{s^\top s} e_j \quad (3.3.13)$$

e, conseqüentemente, $\mathcal{P} = (\frac{1}{s^\top s}) \text{diag}(\mathbf{s}_j^\top \mathbf{s}_j)$ (onde $\text{diag}(\mathbf{s}_j^\top \mathbf{s}_j)$ representa a matriz diagonal, cujos elementos da diagonal são $\mathbf{s}_j^\top \mathbf{s}_j$, para $j = 1, \dots, n$), reescreve-se (3.3.12) como

$$B_{k+1} = B_k + P_Z (\text{diag}(\mathbf{s}_j^\top \mathbf{s}_j))^+ (y - B_k s) s^\top \quad (3.3.14)$$

Para o caso simétrico, utiliza-se o

Teorema 3.3.3. *Seja $\mathcal{A} = \Upsilon = SP(Z) \cap \{B \in \mathbb{R}^{n \times n} : B = B^\top\}$. Obtém-se, para qualquer matriz $M \in \mathbb{R}^{n \times n}$,*

$$P_{\mathcal{A}}(M) = P_{\Upsilon}(M) = \frac{1}{2} P_Z (M + M^\top), \quad (3.3.15)$$

Prova: ver [9].

Nesta situação, a matriz \mathcal{P} do teorema (3.3.2) torna-se

$$\mathcal{P} = \frac{1}{2s^\top s} [\text{diag}(\mathbf{s}_j^\top \mathbf{s}_j) + P_Z(ss^\top)], \quad (3.3.16)$$

e a equação (3.3.12) pode ser reescrita como

$$B_{k+1} = B_k + P_Z (v s^\top + s v^\top), \quad \text{onde } v = \mathcal{P}^+(y - B_k s). \quad (3.3.17)$$

3.4 Aproximações de banda para as Hessianas

Com base na teoria desenvolvida para matrizes esparsas, é feito um estudo para um caso especial em que as aproximações para a matriz Hessiana são matrizes de banda simétricas, introduzido em [16].

Seja $\mathcal{A} \subset \mathbb{R}^{n \times n}$ o subespaço das matrizes simétricas com $(2d + 1)$ diagonais não-nulas, $d = 0, 1, \dots$, e B_k a aproximação para a matriz Hessiana em uma certa iteração k . Obtido o passo s , calcula-se $x_{k+1} = x_k + s$ e $y = \nabla f(x_{k+1}) - \nabla f(x_k)$. Assim, B_{k+1} será a solução do problema

$$\min \|B - B_k\|_F^2, \quad (3.4.1)$$

onde B é o minimizador de $\|Bs - y\|_2^2$, $B \in \mathcal{A}$.

Usando-se os teoremas 3.3.2 e 3.3.3, resolve-se o problema (3.4.1). Utilizando-se (3.3.8), mostra-se a seguir como determinar a matriz \mathcal{P} .

Seja Z redefinida como

$$Z_{ij} = \begin{cases} 1, & \text{se } |i - j| \leq d, \\ 0, & \text{se } |i - j| > d, \end{cases} \quad (3.4.2)$$

s definido por

$$s = (s_1, s_2, \dots, s_n)^\top \quad (3.4.3)$$

e \mathbf{s}_i , $i = 1, \dots, n$, dado por

$$(\mathbf{s}_i)_j = \begin{cases} 0, & \text{se } Z_{ij} = 0, \\ s_j, & \text{se } Z_{ij} = 1. \end{cases} \quad (3.4.4)$$

Nota-se, pela definição acima, que \mathbf{s}_i^\top tem o mesmo modelo de esparsidade da linha i da matriz Z e que

$$\mathbf{s}_i^\top s = \mathbf{s}_i^\top \mathbf{s}_i. \quad (3.4.5)$$

Sejam $SP(Z)$ como definido em (3.3.1), $P_Z(N)_{ij}$ definido como em (3.3.3) e $\mathcal{A} = SP(Z) \cap \{B \in \mathbb{R}^{n \times n} : B = B^\top\}$.

A coluna j da matriz \mathcal{P} definida em (3.3.16) é escrita como

$$\frac{1}{2s^\top s} P_Z(e_j s^\top + s e_j^\top) s. \quad (3.4.6)$$

Observando-se (3.3.3) e (3.4.4), tem-se a j -ésima linha da matriz $P_Z(e_j s^\top + s e_j^\top)$, dada por

$$\mathbf{s}_j^\top + [0 \dots 0 \ s_j \ 0 \dots 0]. \quad (3.4.7)$$

Por simetria, a j -ésima coluna é dada por

$$\mathbf{s}_j + [0 \dots 0 \ s_j \ 0 \dots 0]^\top. \quad (3.4.8)$$

A partir de (3.4.7) e (3.4.8) obtém-se

$$P_Z(e_j s^\top + s e_j^\top) = e_j \mathbf{s}_j^\top + \mathbf{s}_j e_j^\top. \quad (3.4.9)$$

E de (3.4.6), chega-se à expressão abaixo que representa a coluna j de \mathcal{P}

$$\frac{1}{2s^\top s} ([\mathbf{s}_j^\top \ \mathbf{s}_j] e_j + s_j \mathbf{s}_j). \quad (3.4.10)$$

Denotando-se $b = y - B_k s$, reescreve-se o problema (3.3.9) como

$$\min_{v \in \mathbb{R}^n} \| \mathcal{P} v - b \|_2 \quad (3.4.11)$$

A matriz \mathcal{P} é uma matriz simétrica com $2d+1$ diagonais não-nulas, $d = 0, 1, \dots$, ou seja, \mathcal{P} é uma matriz simétrica de banda. Conhecendo-se esta estrutura *a priori*, pode-se armazená-la eficientemente. Utiliza-se a estratégia de armazenamento somente dos valores da diagonal principal e das d diagonais superiores. A matriz \mathcal{P} possui também algumas propriedades especiais, cujas demonstrações podem ser encontradas em [16] e que são mencionadas abaixo:

- P1** Seja $1 \leq j \leq n$. Se $s_j = 0$, então todos os elementos da coluna j de \mathcal{P} são nulos, exceto possivelmente o elemento da diagonal principal. Como \mathcal{P} é simétrica, o mesmo ocorre com a linha j .
- P2** Os elementos da diagonal principal de \mathcal{P} são sempre positivos ou nulos.
- P3** Se o elemento \mathcal{P}_{jj} da diagonal principal de \mathcal{P} é nulo, então todos os elementos da coluna (linha) j são nulos.
- P4** Seja j tal que $1 \leq j \leq n$. Se $\mathcal{P}_{jj} = 0$, então
- a \mathcal{P} tem todas as colunas (linhas) nulas até a coluna (linha) j .
 - b As colunas (linhas) $j+1, \dots, j+d$ são nulas, exceto possivelmente o elemento da diagonal principal.
- P5** Se $s \neq 0$ e $0 \leq d \leq n-2$, então \mathcal{P} é uma matriz semidefinida positiva. Se $s \neq 0$ e $d = n-1$, então \mathcal{P} é uma matriz definida positiva.

Observando-se as propriedades de \mathcal{P} acima, há dois casos a considerar para a resolução do problema (3.4.11):

- i** \mathcal{P} é inversível: neste caso, existe uma solução única obtida pela resolução do sistema $\mathcal{P}v = b$. Aqui, todos os elementos da diagonal principal de \mathcal{P} são não nulos. Pode-se mostrar então que \mathcal{P} é uma matriz definida positiva (ver [16]). A solução do problema (3.4.11) é obtida utilizando-se a fatoração de Cholesky de \mathcal{P} . Através desta fatoração, obtém-se $\mathcal{P} = LL^\top$, onde L é uma matriz triangular inferior com a diagonal principal positiva e d diagonais inferiores não-nulas (ver [14]). Escreve-se então, observando-se o problema (3.4.11), $LL^\top v = b$ e resolve-se os sistemas $Lc = b$ e $L^\top v = c$, obtendo-se a solução v_* desejada para o

problema (3.4.11). Faz-se também um armazenamento da matriz L de forma eficiente. Isto consiste em armazenar-se apenas a diagonal principal e as d diagonas inferiores não-nulas.

- ii \mathcal{P} é não-inversível: neste caso, o problema (3.4.11) tem infinitas soluções. A solução procurada é a de norma-2 mínima, v_* , onde $\|v_*\|_2 \leq \|v\|_2$, para todo v que satisfaz (3.4.11). Poderia-se utilizar a decomposição em valores singulares (SVD) para obter-se a solução desejada, mas isto é caro computacionalmente (ver [14]). Tirando-se proveito das propriedades P1 a P5, descritas acima, pode-se obter v_* de forma mais eficiente.

Se há elementos nulos na diagonal principal de \mathcal{P} , esta matriz não é definida positiva e, pela propriedade P3, vê-se que é uma matriz singular. Decompõe-se então \mathcal{P} em n_b blocos independentes, com a eliminação da linha e da coluna correspondentes ao elemento nulo da diagonal principal. Repete-se este processo para todos os elementos nulos da diagonal principal. Cada bloco, que será denotado por $\bar{\mathcal{P}}_{j_b}$, para $j_b = 1, \dots, n_b$, é uma matriz de banda com todos os elementos da diagonal principal não nulos e, portanto, definida positiva (ver [16]). Fazendo-se as mudanças correspondentes para b , obtém-se o vetor \bar{b}_{j_b} . Resolve-se, então, os sistemas $\bar{\mathcal{P}}_{j_b} \bar{v}_{j_b} = \bar{b}_{j_b}$. As soluções $(\bar{v}_*)_{j_b}$ podem ser calculadas como no primeiro caso. Estas fornecem as coordenadas $(v_*)_j$ da solução v_* procurada, para todo j tal que $\mathcal{P}_{jj} \neq 0$. Se $\mathcal{P}_{jj} = 0$, faz-se $(v_*)_j = 0$ para obter-se a solução de norma mínima.

Analisando-se o caso em que \mathcal{P} é uma matriz diagonal, a solução v_* de norma mínima é obtida diretamente calculando-se $(v_*)_j = \frac{b_j}{\mathcal{P}_{jj}}$, se $\mathcal{P}_{jj} \neq 0$ e $(v_*)_j = 0$ se $\mathcal{P}_{jj} = 0$.

Com o que foi desenvolvido até aqui, descreve-se agora como fazer o cálculo das aproximações de banda para as Hessianas.

De (3.3.17) tem-se que

$$B_{k+1} = B_k + \frac{1}{2s^\top s} P_Z(v_* s^\top + s v_*^\top). \quad (3.4.12)$$

Denotando-se $\bar{W} = v_* s^\top + s v_*^\top$ e $\bar{\bar{W}} = \frac{1}{2s^\top s} P_Z(\bar{W})$, obtém-se a atualização da matriz Hessiana

$$B_{k+1} = B_k + \bar{\bar{W}} \quad (3.4.13)$$

onde

$$\bar{\bar{W}}_{ij} = \begin{cases} \bar{W}_{ij}, & \text{se } |i - j| \leq d, \\ 0, & \text{se } |i - j| > d. \end{cases}$$

3.5 Um breve comentário sobre convergência

Na resolução de problemas de minimização com restrições, uma característica relevante para qualquer algoritmo é que as restrições ativas na solução possam ser identificadas em um número finito de iterações. Isto reduzirá o algoritmo a um método para minimização irrestrita e teoremas de convergência local para o problema irrestrito podem ser aplicados (ver [13]).

Na solução do problema (3.1.1), onde $\Omega \equiv \mathbb{R}^n$, a cada iteração k , tem-se

$$x_{k+1} = x_k - B_k^{-1} \nabla f(x_k), \quad (3.5.1)$$

com a atualização secante de variação mínima B_{k+1} obtida com a solução do problema (3.4.1). Pode-se mostrar que, sob certas hipóteses, a seqüência gerada por (3.5.1) tem convergência local superlinear (ver [16]).

O algoritmo BOX-QUACAN identifica as restrições ativas em um número finito de passos (ver em [13]). Desta forma, o método global para minimização com restrições canalizadas, dado pelo algoritmo descrito na seção 2.3, é reduzido a um método global para minimização irrestrita e, nas vizinhanças da solução x_* , seu comportamento é semelhante ao do método local dado por (3.5.1), uma vez que utiliza-se a atualização (3.4.1) para a aproximação da Hessiana necessária em BOX-QUACAN.

Ao algoritmo BOX-QUACAN com as atualizações (3.4.1) para B_k , dá-se o nome de BOX-QUACAN Modificado. Os experimentos numéricos feitos com este algoritmo serão descritos e analisados no capítulo 5.

Capítulo 4

Aproximações BFGS com Memória Limitada

4.1 Introdução

Descreve-se, neste capítulo, um algoritmo para resolver o problema (1.0.1), proposto por Byrd, Lu, Nocedal e Zhu, em 1995 [3]. Este algoritmo é baseado no método do gradiente projetado e usa busca linear, diferentemente do algoritmo BOX-QUACAN descrito no capítulo 2, que trabalha com região de confiança. São utilizadas atualizações quase-Newton com memória limitada para a aproximação da Hessiana da função objetivo. O armazenamento destas aproximações é linear em n (ver [4]).

4.2 O algoritmo

Seja x_k a aproximação para a solução do problema (1.0.1) gerada na iteração k do método ($k = 0, 1, \dots$), a partir de x_0 factível dado. Denota-se f_k o valor da função f em x_k , g_k , o gradiente de f em x_k , e B_k , matriz simétrica e definida positiva, a aproximação para $\nabla^2 f(x_k)$. O modelo quadrático de f em x_k é dado por

$$m_k(x) = f_k + g_k^\top (x - x_k) + \frac{1}{2} (x - x_k)^\top B_k (x - x_k). \quad (4.2.1)$$

Definindo-se

$$x(t) = P(x_k - tg_k, l, u), \quad (4.2.2)$$

onde

$$P(x, l, u)_i = \begin{cases} l_i & \text{se } x_i < l_i, \\ x_i & \text{se } x_i \in [l_i, u_i], \\ u_i & \text{se } x_i > u_i, \end{cases} \quad (4.2.3)$$

calcula-se o ponto de Cauchy generalizado x^c , determinando-se o primeiro minimizador local de

$$q_k(t) = m_k(x(t)). \quad (4.2.4)$$

(ver [4])

Os índices das variáveis que chegam a seu limite superior ou inferior em x^c são então armazenados em $\mathcal{A}(x^c)$. O problema torna-se

$$\min \{m_k(x) : x_i = x_i^c, \forall i \in \mathcal{A}(x^c)\} \quad (4.2.5)$$

$$\text{s. a } l_i \leq x_i \leq u_i, \forall i \notin \mathcal{A}(x^c). \quad (4.2.6)$$

Resolve-se exata ou aproximadamente (4.2.5), ignorando-se as restrições definidas em (4.2.6), através de um método direto ou iterativo no subespaço das variáveis livres, ou por uma aproximação dual, trabalhando-se os limites ativos em (4.2.5) através dos multiplicadores de Lagrange. Quando um método iterativo é aplicado, x^c é usado como aproximação inicial e as novas aproximações para a solução do subproblema devem satisfazer (4.2.6).

Com a solução aproximada \bar{x}_{k+1} obtida, calcula-se o novo ponto x_{k+1} , através de uma busca linear ao longo da direção $d_k = \bar{x}_{k+1} - x_k$, satisfazendo-se a condição de

decréscimo suficiente

$$f(x_{k+1}) \leq f(x_k) + \alpha \lambda_k g_k^\top d_k \quad (4.2.7)$$

e a condição de curvatura

$$|g_{k+1}^\top d_k| \leq \beta |g_k^\top d_k| \quad (4.2.8)$$

onde λ_k é o tamanho do passo e $\alpha, \beta \in (0, 1)$ são parâmetros a serem definidos.

Assim,

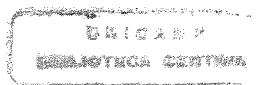
$$x_{k+1} = x_k + \lambda_k d_k. \quad (4.2.9)$$

Calculando-se o gradiente em x_{k+1} e a nova aproximação da Hessiana B_{k+1} , repete-se o processo descrito acima.

No algoritmo, cada aproximação da matriz Hessiana é definida positiva. Portanto, a solução aproximada \bar{x}_{k+1} do problema quadrático (4.2.5) e (4.2.6) define uma direção de descida $d_k = \bar{x}_{k+1} - x_k$ para a função objetivo f . Para ver isto, nota-se primeiro que o ponto de Cauchy generalizado x^c , que é um minimizador de $m_k(x)$ na direção de máxima descida projetada, satisfaz $m_k(x_k) > m_k(x^c)$ se o gradiente projetado é não-nulo. Desde que \bar{x}_{k+1} está no caminho a partir de x^c até o minimizador de (4.2.5), ao longo do qual m_k decresce, o valor de m_k em \bar{x}_{k+1} não é maior que seu valor em x^c . Assim,

$$f(x_k) = m_k(x_k) > m_k(x^c) \geq m_k(\bar{x}_{k+1}) = f(x_k) + g_k^\top d_k + \frac{1}{2} d_k^\top B_k d_k. \quad (4.2.10)$$

Esta desigualdade implica que $g_k^\top d_k < 0$ se B_k é definida positiva e d_k é não-nulo. Em [3] não é apresentada nenhuma análise de convergência ou estudo da possibilidade de “zigzag”, mas os autores afirmam que, dado o uso do método de projeção do gradiente no cálculo do passo, a análise deve ser similar à feita em [5] e [7] e que “zigzag” é somente um problema em casos degenerados.



As próximas seções abordam a obtenção da matriz B_{k+1} , o cálculo do ponto de Cauchy e um método para minimização no subespaço das variáveis livres.

4.3 Aproximações BFGS com memória limitada

No modelo quadrático (4.2.1), a Hessiana de f em x_k é aproximada por uma matriz B_k , a qual, a partir de B_0 dada, é atualizada a cada iteração k pela fórmula quase-Newton BFGS proposta por Broyden, Fletcher, Goldfarb e Shanno, em 1970 (ver [10]).

A atualização BFGS é dada por

$$B_{k+1} = B_k - \frac{B_k s_k s_k^\top B_k}{s_k^\top B_k s_k} + \frac{y_k y_k^\top}{y_k^\top s_k}, \quad (4.3.1)$$

onde

$$s_k = x_{k+1} - x_k, \quad y_k = g_{k+1} - g_k \quad (4.3.2)$$

para $k = 0, 1, \dots$

No algoritmo proposto na seção 2, as aproximações (4.3.1) são obtidas usando-se memória limitada. Para não armazenar a matriz B_k a cada iteração, o método armazena um certo número m dos pares mais recentes $\{s_i, y_i\}$, $i = k - m, \dots, k - 1$, $k > m$, como definidos em (4.3.2), os quais contém informações sobre a curvatura da função. Para $k \leq m$, temos o BFGS clássico. Com estes m pares, é possível definir a matriz B_k sem precisar explicitá-la.

Sejam as $n \times k$ matrizes de correção definidas por

$$S_k = [s_0, \dots, s_{k-1}], \quad Y_k = [y_0, \dots, y_{k-1}] \quad (4.3.3)$$

e D_k uma matriz diagonal $k \times k$ dada por

$$D_k = \text{diag}[s_0^\top y_0, \dots, s_{k-1}^\top y_{k-1}]. \quad (4.3.4)$$

Alguns teoremas, enunciados ao longo deste capítulo, serão necessários na implementação do método de memória limitada. As demonstrações podem ser encontradas em [4].

Teorema 4.3.1. *Seja B_0 uma matriz simétrica e definida positiva e suponha que os k pares $\{s_i, y_i\}_{i=0}^{k-1}$ satisfazem $s_i^\top y_i > 0$. Seja B_k uma matriz obtida pela atualização de B_0 k vezes usando a fórmula BFGS dada pela equação (4.3.1) e os pares $\{s_i, y_i\}_{i=0}^{k-1}$. Então*

$$B_k = B_0 - \begin{bmatrix} B_0 S_k & Y_k \end{bmatrix} \begin{bmatrix} S_k^\top B_0 S_k & L_k \\ L_k^\top & -D_k \end{bmatrix}^{-1} \begin{bmatrix} S_k^\top B_0 \\ Y_k^\top \end{bmatrix}, \quad (4.3.5)$$

onde L_k é uma matriz $k \times k$

$$(L_k)_{i,j} = \begin{cases} s_{i-1}^\top y_{j-1} & \text{se } i > j, \\ 0 & \text{caso contrário.} \end{cases} \quad (4.3.6)$$

A matriz $2k \times 2k$, indefinida,

$$\begin{bmatrix} S_k^\top B_0 S_k & L_k \\ L_k^\top & -D_k \end{bmatrix}, \quad (4.3.7)$$

que aparece na equação (4.3.5), pode ser reescrita a partir de uma reordenação de blocos, como:

$$\begin{bmatrix} -D_k & L_k^\top \\ L_k & S_k^\top B_0 S_k \end{bmatrix} = \begin{bmatrix} D_k^{\frac{1}{2}} & 0 \\ -L_k D_k^{-\frac{1}{2}} & J_k \end{bmatrix} \begin{bmatrix} -D_k^{\frac{1}{2}} & D_k^{-\frac{1}{2}} L_k^\top \\ 0 & J_k \end{bmatrix}, \quad (4.3.8)$$

onde

$$J_k J_k^\top = S_k^\top B_0 S_k + L_k D_k^{-1} L_k^\top. \quad (4.3.9)$$

O teorema enunciado abaixo mostra que J_k existe e é não-singular.

Teorema 4.3.2. *Se B_0 é definida positiva e $s_i^\top y_i > 0$, $i = 1, \dots, k-1$, então a matriz $S_k^\top B_0 S_k + L_k D_k^{-1} L_k^\top$ é definida positiva.*

Calcula-se então J_k através da decomposição de Cholesky de $S_k^\top B_0 S_k + L_k D_k^{-1} L_k^\top$.

Com estes resultados, pode-se descrever a implementação do método de memória limitada (MML).

4.4 Implementação do MML

Tirando-se proveito da decomposição (4.3.8), a equação (4.3.5) pode ser reescrita como

$$B_k = \theta_k I - \begin{bmatrix} Y_k & \theta_k S_k \end{bmatrix} \begin{bmatrix} -D_k^{\frac{1}{2}} & D_k^{-\frac{1}{2}} L_k^\top \\ 0 & J_k^\top \end{bmatrix}^{-1} \begin{bmatrix} D_k^{\frac{1}{2}} & 0 \\ -L_k D_k^{-\frac{1}{2}} & J_k \end{bmatrix}^{-1} \begin{bmatrix} Y_k \\ \theta_k S_k^\top \end{bmatrix}, \quad (4.4.1)$$

onde

$$(L_k)_{i,j} = \begin{cases} s_{(k-m-1+i)}^\top y_{(k-m-1+j)} & \text{se } i > j, \\ 0 & \text{caso contrário,} \end{cases} \quad (4.4.2)$$

e S_k , Y_k e D_k são definidas como em (4.3.3) e (4.3.4), substituindo-se s_0 e y_0 por s_{k-m} e y_{k-m} , respectivamente. Assim, o conjunto de m pares $\{s_i, y_i\}$ será renovado a cada iteração, retirando-se o para mais antigo e substituindo-o pelo mais recente. O fator θ_k pode ser calculado por

$$\theta_k = \frac{y_k^\top y_k}{s_k^\top y_k}, \quad \text{se } s_k^\top y_k > 0.$$

Para se obter os fatores de B_k da forma acima definida, realiza-se $2mn + \mathcal{O}(m^3)$ multiplicações e $(4m+1)n + \mathcal{O}(m^2)$ multiplicações para o produto $B_k v$ ser calculado, como pode ser visto em [3] e [4]. Nota-se que, para m pequeno, o número de multiplicações será linear em n .

Para manter a matriz B_k definida positiva, descarta-se o par de correção $\{s_k, y_k\}$ quando a desigualdade

$$s_k^\top y_k > eps \|y_k\|^2 \quad (4.4.3)$$

não for satisfeita para um dado ϵ positivo, próximo de zero. Neste caso, não é retirado o par mais antigo.

Seguindo-se o mesmo raciocínio feito para a matriz B_k acima, pode-se definir a matriz BFGS com memória limitada H_k , que aproxima a inversa da matriz Hessiana, como

$$H_k \equiv \frac{1}{\theta} I + \overline{W}_k \overline{M}_k \overline{W}_k^\top, \quad (4.4.4)$$

onde

$$\overline{W}_k \equiv \begin{bmatrix} \frac{1}{\theta} Y_k & S_k \end{bmatrix},$$

$$\overline{M}_k \equiv \begin{bmatrix} 0 & -R_k^{-1} \\ -R_k^{-T} & R_k^{-T} (D_k + \frac{1}{\theta} Y_k^\top Y_k) R_k^{-1} \end{bmatrix}$$

e

$$(R_k)_{i,j} = \begin{cases} (s_{k-m-1+i}^\top y_{k-m-1+j}) & \text{se } i \leq j, \\ 0 & \text{caso contrário.} \end{cases} \quad (4.4.5)$$

Maiores detalhes podem ser encontrados em [4].

Existem também fórmulas semelhantes à que foi descrita para representação de matrizes com memória limitada baseadas nas atualizações SR1 e DFP (ver em [4]). No algoritmo proposto na seção 2, utiliza-se a fórmula BFGS, uma vez que esta tem mostrado melhor desempenho computacional (ver [4]).

4.5 O Ponto de Cauchy generalizado

O procedimento aqui descrito determinará o primeiro minimizador local do modelo quadrático (4.2.1), através do caminho linear por partes obtido pela projeção dos pontos ao longo da direção de máxima descida, $x_k - tg_k$, na região factível. Definindo

$x^0 = x_k$, determina-se os “breakpoints” em cada coordenada da direção como

$$t_i = \begin{cases} \frac{(x_i^0 - u_i)}{g_i} & \text{se } g_i < 0, \\ \frac{(x_i^0 - l_i)}{g_i} & \text{se } g_i > 0, \\ \infty & \text{caso contrário.} \end{cases} \quad (4.5.1)$$

O conjunto $\{t_i, i = 1, \dots, n\}$ com os pontos em ordem crescente, gera o conjunto $\{t^j : t^j \leq t^{j+1}, j = 1, \dots, n\}$.

Reescreve-se $P(x^0 - tg, l, u)$ como

$$x_i(t) = \begin{cases} x_i^0 - tg_i & \text{se } t \leq t_i, \\ x_i^0 - t_i g_i & \text{caso contrário.} \end{cases} \quad (4.5.2)$$

Considerando o intervalo $[t^{j-1}, t^j]$, define-se

$$x^{j-1} = x(t^{j-1}),$$

$$x(t) = x^{j-1} + \Delta t d^{j-1},$$

onde

$$\Delta t = t - t^{j-1}$$

e

$$d_i^{j-1} = \begin{cases} -g_i & \text{se } t^{j-1} < t_i \\ 0 & \text{caso contrário} \end{cases} \quad (4.5.3)$$

O modelo quadrático pode então ser reescrito como

$$\begin{aligned} m(x) &= f + g^\top (x - x^0) + \frac{1}{2} (x - x^0)^\top B (x - x^0) \\ &= f + g^\top (z^{j-1} + \Delta t d^{j-1}) + \frac{1}{2} (z^{j-1} + \Delta t d^{j-1})^\top B (z^{j-1} + \Delta t d^{j-1}) \end{aligned}$$

onde

$$z^{j-1} = x^{j-1} - x^0. \quad (4.5.4)$$

Escrevendo

$$f_{j-1} = f + g^\top z^{j-1} + \frac{1}{2} z^{j-1 \top} B z^{j-1} \quad (4.5.5)$$

$$f'_{j-1} = g^\top d^{j-1} + d^{j-1 \top} B z^{j-1} \quad (4.5.6)$$

$$f''_{j-1} = d^{j-1 \top} B d^{j-1} \quad (4.5.7)$$

representa-se $m(x)$ como

$$\hat{m}(\Delta t) \equiv f_{j-1} + f'_{j-1} \Delta t + \frac{1}{2} f''_{j-1} \Delta t^2. \quad (4.5.8)$$

Diferenciando $\hat{m}(\Delta t)$ e igualando a zero, obtém-se $\Delta t^* = -\frac{f'_{j-1}}{f''_{j-1}}$. Sendo B definida positiva, isto define um minimizador se $t^{j-1} + \Delta t^*$ está em $[t^{j-1}, t^j]$. Caso contrário, o ponto de Cauchy generalizado está em $x(t^{j-1})$ se $f'_{j-1} \geq 0$ e além de ou em $x(t^j)$ se $f'_{j-1} < 0$.

Se no intervalo $[t^{j-1}, t^j]$ o ponto de Cauchy não foi encontrado, escreve-se

$$x^j = x^{j-1} + \Delta t^{j-1} d^{j-1}, \quad \Delta t^{j-1} = t^j - t^{j-1} \quad (4.5.9)$$

e atualiza-se as derivadas direcionais f'_j e f''_{j+1} , prosseguindo a busca no próximo intervalo.

Usando a fórmula BFGS de memória limitada, o custo total do procedimento acima descrito é dado por $(2m + 2)n + \mathcal{O}(m^2) \times n_{int} + n \log n$ multiplicações, onde n_{int} denota o número de intervalos explorados (ver em [3]).

Na próxima seção, mostra-se como completar a resolução do subproblema considerando-se agora o subespaço das variáveis livres.

4.6 Um método direto para minimização no subespaço das variáveis livres

Encontrado o ponto de Cauchy x^c , minimiza-se o modelo quadrático m_k agora sobre o espaço das variáveis livres, forçando os limites do problema. Neste processo de minimização, pode-se aplicar um método direto, do tipo primal, que descreve-se a seguir. Outras duas alternativas de algoritmos para a minimização nesse subespaço são descritas em [3].

Denota-se t como o número de variáveis livres no ponto de Cauchy x^c e \mathcal{F} como o conjunto de índices correspondendo às variáveis livres. Define-se também Z_k como a matriz $n \times t$ cujas colunas são aquelas da matriz identidade que gera o subespaço das variáveis livres em x^c e A_k , matriz $n \times (n - t)$, dos gradientes das restrições ativas em x^c .

Fixa-se as $n - t$ variáveis que estão nos seus limites em x^c , e resolve-se o problema quadrático (4.2.5) no subespaço das t variáveis livres restantes, começando de x^c e impondo os limites (4.2.6) a estas variáveis.

Assim, dados $x \in \mathbb{R}^n$ e $\hat{d} \in \mathbb{R}^t$, considera-se apenas os pontos da forma

$$x = x^c + Z_k \hat{d}. \quad (4.6.1)$$

O modelo quadrático é reescrito como

$$m_k(x) = \hat{d}^\top \hat{r}^c + \frac{1}{2} \hat{d}^\top \hat{B}_k \hat{d} + \gamma \quad (4.6.2)$$

onde γ é uma constante,

$$\hat{B}_k = Z_k^\top B_k Z_k \quad (4.6.3)$$

é a Hessiana reduzida de m_k , e

$$\hat{r}^c = Z_k^\top (g_k + B_k(x^c - x_k)) \quad (4.6.4)$$

é o gradiente reduzido de m_k em x^c . Considerando-se

$$W_k = \begin{bmatrix} Y_k & \theta S_k \end{bmatrix}, \quad (4.6.5)$$

$$M_k = \begin{bmatrix} -D_k & L_k^\top \\ L_k & \theta S_k^\top S_k \end{bmatrix}^{-1}, \quad (4.6.6)$$

com L_k como em (4.4.2) e denotando-se $c = W_k^\top (x^c - x_k)$, reescreve-se a equação (4.6.4) como

$$\hat{r}^c = Z_k^\top (g_k + \theta(x^c - x_k) - W_k M_k c). \quad (4.6.7)$$

Reformula-se assim o problema como

$$\min \hat{m}_k(\hat{d}) = \hat{r}^c + \frac{1}{2} \hat{d}^\top \hat{B}_k \hat{d} + \gamma \quad (4.6.8)$$

$$\text{s. a } l_i - x_i^c \leq \hat{d}_i \leq u_i - x_i^c, \quad i \in \mathcal{F}, \quad (4.6.9)$$

onde o índice i denota a i -ésima componente de um vetor.

O problema (4.6.8) pode ser resolvido por um método direto como o que se descreve aqui, e, em seguida, impõe-se (4.6.9) por “backtracking”. A solução irrestrita de (4.6.8) é dada por

$$\hat{d}^u = -\hat{B}_k^{-1} \hat{r}^c. \quad (4.6.10)$$

onde B_k^{-1} é obtida através da fórmula de Sherman-Morrison-Woodbury [10].

Para realizar o “backtracking”, se este for necessário, calcula-se

$$\alpha^* = \max \{ \alpha : \alpha \leq 1, \quad l_i - x_i^c \leq \alpha \hat{d}_i^u \leq u_i - x_i^c, \quad i \in \mathcal{F} \}, \quad (4.6.11)$$

obtendo

$$\hat{d}^* = \alpha^* \hat{d}^u. \quad (4.6.12)$$

A solução aproximada obtida é dada então por

$$\bar{x}_i = \begin{cases} x_i^c & \text{se } i \notin \mathcal{F}, \\ x_i^c + (Z_k \hat{d}^*)_i & \text{se } i \in \mathcal{F}. \end{cases} \quad (4.6.13)$$

O custo total deste processo é de

$$2m^2t + 6mt + 4t + \mathcal{O}(m^3) \quad (4.6.14)$$

multiplicações (ver em [3]).

Capítulo 5

Experimentos Numéricos

Neste capítulo, são apresentados os experimentos numéricos realizados com o objetivo de se analisar o desempenho de BOX-QUACAN com as atualizações de banda para aproximar a Hessiana (BOX-QUACAN Modificado). Em [16], onde esta versão de BOX-QUACAN foi introduzida, poucos testes numéricos foram feitos, pois o objetivo principal era comparar os resultados obtidos usando diferentes números de diagonais, nas aproximações para $\nabla^2 f(x)$; não houve uma comparação deste algoritmo com outros voltados ao mesmo fim, seguindo a idéia de economizar memória na aproximação da Hessiana. Assim, neste trabalho, os resultados obtidos com BOX-QUACAN Modificado foram comparados com os de L-BFGS-B, um pacote que implementa o método BFGS com memória limitada, descrito no capítulo 4. Todos os testes foram também rodados com EASY, uma versão de BOX-QUACAN na qual a aproximação para a Hessiana é obtida por diferenças finitas.

Os algoritmos foram escritos em FORTRAN 77 com precisão dupla. Os testes numéricos foram realizados em uma estação de trabalho SUN Ultra1 Creator, com a opção de compilador -O para todos os algoritmos.

5.1 Apresentação dos problemas

Para as funções objetivo do problema (1.0.1), foram escolhidas 7 funções que serão descritas abaixo. As funções 1, 2 e 4 podem ser encontradas em [15], a função 3, em [8], a função 7, em [18] e, finalmente, as funções 5 e 6, em [17]. É importante destacar que a matriz Hessiana das funções 1, 2, 3 e 5 possui estrutura de banda. A princípio, isto torna estas funções interessantes para o algoritmo BOX-QUACAN Modificado. As demais funções possuem matriz Hessiana densa.

1. Função de Rosenbrock (R)

$$f(x) = \sum_{i=1}^{\frac{n}{2}} [100(x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2]$$

2. Função de Broyden (BROYD)

$$f(x) = \sum_{i=1}^n [(3 - 2x_i)x_i - x_{i-1} - 2x_{i+1} + 1]^2,$$

onde $x_0 = x_{n+1} = 0$

3. Função 7-diagonal de Toint (7D)

$$f(x) = 1 + \sum_{i=1}^n |x_{i-1} - (3 - \frac{x_i}{2})x_i + 2x_{i+1} - 1|^p + \sum_{i=1}^{\frac{n}{2}} |x_i + x_{i+\frac{n}{2}}|^p,$$

onde n é par, $p = \frac{7}{3}$ e $x_0 = x_{n+1} = 0$.

4. Função Penalty (P)

$$f(x) = \sum_{i=1}^n a(x_i - 1)^2 + ((\sum_{i=1}^n x_i^2) - \frac{1}{4})^2,$$

onde $a = 10^{-5}$.

5. Função de Discretização de um Problema de Valor de Contorno (PVC)

$$f(x) = \sum_{i=1}^n [2x_i - x_{i-1} - x_{i+1} + \frac{h^2}{2}(x_i + hi + 1)^3]^2,$$

onde $h = \frac{1}{n+1}$ e $x_0 = x_{n+1} = 0$.

6. Função de Discretização de uma Integral (I)

$$f(x) = \sum_{i=1}^n (x_i + h[(1 - t_i) \sum_{j=1}^i t_j (x_j + t_j + 1)^3 + t_i \sum_{j=i+1}^n (1 - t_j)(x_j + t_j + 1)^3]/2)^2,$$

onde $h = \frac{1}{n+1}$, $t_i = ih$ e $x_0 = x_{n+1} = 0$.

7. Função de Wolfe (W)

$$f(x) = \left(\sum_{i=1}^n x_i^2 \right) \left(\sum_{i=1}^n x_i^4 \right) \left(\sum_{i=1}^n x_i^3 \right)^2$$

Com estas funções como objetivo, foram resolvidos, inicialmente, problemas de minimização sem restrições, como sugerido nas referências citadas, com exceção da função 7. Para esta, a referência sugere a caixa $0 \leq x \leq 1$ e, assim, dá-se o nome de Problema de Wolfe.

Com a solução irrestrita obtida, são propostas 3 novas caixas que serão definidas na seção 5.3. Gerou-se então um total de 25 problemas. A seção 5.3 traz também a dimensão dos problemas, os pontos iniciais utilizados e os resultados obtidos.

5.2 Definição de parâmetros e critérios de parada

Para o algoritmo BOX-QUACAN, descrito no capítulo 2, tanto para a versão BOX-QUACAN Modificado como para EASY, os valores adotados para os principais parâmetros analisados foram $\eta = 0.9$, $\Delta_{min} = 10^{-4}$, $\tau = 0.5$ e $\alpha = 0.1$.

Para determinar o melhor valor para τ , foram feitos testes com $\tau = 0.1$ e $\tau = 0.5$. Foram calculadas as médias geométricas do número total de iterações, do número de avaliações de função e do tempo gasto para os problemas irrestritos que têm como funções objetivo R, BROVD, 7D e P e para o problema de Wolfe. Observou-se então que os melhores resultados foram obtidos com $\tau = 0.5$, como pode ser visto nas tabelas 5.1, 5.2 e 5.3 abaixo.

$\eta = 0.1$ e $acc = 0.1$	$\eta = 0.9$ e $acc = 0.1$	$\eta = 0.1$ e $acc = 0.5$	$\eta = 0.9$ e $acc = 0.5$
95	106	73	65

Tabela 5.1: Média geométrica do total de iterações

$\eta = 0.1$ e $acc = 0.1$	$\eta = 0.9$ e $acc = 0.1$	$\eta = 0.1$ e $acc = 0.5$	$\eta = 0.9$ e $acc = 0.5$
284	251	182	202

Tabela 5.2: Média geométrica do número de avaliações de função

$\eta = 0.1$ e $acc = 0.1$	$\eta = 0.9$ e $acc = 0.1$	$\eta = 0.1$ e $acc = 0.5$	$\eta = 0.9$ e $acc = 0.5$
5.94	4.86	3.06	3.21

Tabela 5.3: Média geométrica do tempo de execução(em segundos)

No algoritmo L-BFGS-B, apresentado no capítulo 4, fez-se testes preliminares para a escolha de m . Estes testes consistiram em resolver problemas de minimização irrestrita, onde as funções objetivo foram 1 e 4, com m tomando os valores 5, 10 e 15. Os melhores resultados foram obtidos com $m = 15$. Assim, nos testes realizados neste trabalho, adotou-se este valor para m .

Os critérios de parada para os algoritmos descritos são os seguintes:

- (A) - o gradiente projetado é suficientemente pequeno.

Para os algoritmos BOX-QUACAN Modificado e EASY, usa-se

$$\|g_{p_{cont}}(x)\| \leq \varepsilon_g,$$

onde

$$g_{p_{cont}}(x) = P((x - g(x)) - x, l, u),$$

para $P(x, l, u)_i$ definido como em 4.2.3.

Em L-BFGS-B, esta condição é satisfeita quando

$$\|g_p(x)\| \leq \varepsilon_g$$

ou

$$\frac{|g_p(x)|}{(1 + |f(x)|)} < 10^{-10},$$

onde $g_p(x) = P(g(x), l, u)$. Trabalhou-se com $\varepsilon_g = 10^{-6}$;

- (B) - é atingido o número máximo de avaliações de função permitido (este número variou de 10000 a 100000, de acordo com a característica do problema);
- (C) - é atingido o número máximo de iterações permitido (usou-se 1000 ou 10000, dependendo do problema).

Os algoritmos BOX-QUACAN Modificado e EASY também podem ser interrompidos se o raio da região de confiança torna-se extremamente pequeno ($\varepsilon_d \leq 10^{-10}$). Isto será denotado como critério (D).

Nas tabelas 5.6 a 5.8, 5.10 a 5.19 e 5.21 a 5.26, a coluna *RP* indica as razões de parada dos algoritmos para os testes realizados, de acordo com a notação acima.

Todos os algoritmos requerem rotinas em FORTRAN para o cálculo do valor da função objetivo e do gradiente em um ponto x_k , de uma certa iteração k do algoritmo.

Segue, então, a seção 5.3 com os resultados numéricos obtidos.

5.3 Resultados numéricos

Na tabela 5.4 estão descritos alguns parâmetros utilizados que variaram de acordo com o problema testado. As colunas *itmax* e *nafmax* contêm, respectivamente, o número máximo de iterações e avaliações de função permitido para cada problema.

No algoritmo L-BFGS-B, B_0 (a aproximação inicial da Hessiana) está definida como a matriz identidade, $n \times n$, enquanto em EASY, $B_0 \approx \nabla^2 f(x_0)$ por diferenças finitas, como visto no capítulo 1.

Em BOX-QUACAN Modificado, o usuário tem a opção de definir a aproximação inicial para a Hessiana B_0 . Uma vez que, nos *softwares* L-BFGS-B e EASY, B_0 é definida de formas diferentes, decidiu-se tirar proveito da opção deixada em BOX-QUACAN Modificado para a escolha de B_0 . Assim, em problemas nos quais uma matriz mais próxima de $\nabla^2 f(x_0)$ pôde ser calculada sem um esforço computacional demasiado, usou-se B_0 como a matriz formada pelas $2d + 1$ diagonais principais de $\nabla^2 f(x_0)$. Isto ocorreu nos problemas que utilizaram as funções 1, 2, 3 e 6. No caso da função 1, B_0 coincidiu com $\nabla^2 f(x_0)$. Para os demais problemas, utilizou-se $B_0 = I$.

	n	d	itmax	nafmax
R	5000	1	1000	10000
BROYD	5000	2	1000	10000
7D	200	1	1000	10000
P	1000	1	1000	10000
PVC	5000	2	10000	100000
I	500	2	10000	100000
W	100	0	1000	10000

Tabela 5.4: Parâmetros utilizados

5.3.1 Tabelas gerais

Nas tabelas 5.6 a 5.8 e 5.10 a 5.19, os resultados numéricos obtidos são introduzidos. Nestas tabelas, a coluna *Tit* denota o número de iterações realizadas; a coluna *Naf*

registra o número total de avaliações de função; F informa o valor da função no ponto encontrado; NGP , a norma do gradiente projetado e a coluna T , o tempo total gasto pelo algoritmo, em segundos.

Como já mencionado, a coluna RP contém as razões de parada dos algoritmos, de acordo com os critérios que estão descritos na seção 5.2.

Como colocado na seção 5.1, foram resolvidos inicialmente problemas irrestritos com as funções 1 a 6 descritas. Conhecendo-se a solução irrestrita x_* dos problemas, 3 novas caixas são definidas para cada um deles, da seguinte forma:

canalização 1 :

$$1 + (x_*)_i \leq x_i \leq 10 + (x_*)_i;$$

canalização 2 :

$$-10 + (x_*)_i \leq x_i \leq -1 + (x_*)_i;$$

canalização 3 :

$$\begin{aligned} 1 + (x_*)_i \leq x_i \leq 10 + (x_*)_i, & \text{ se } i \text{ é ímpar,} \\ -0.95(x_*)_i \leq x_i \leq 0.95(x_*)_i, & \text{ se } i \text{ é par,} \end{aligned}$$

para $i = 1, \dots, n$.

Os pontos iniciais para os problemas irrestritos (x_0^{Irr}) são apresentados na tabela 5.5 abaixo. Para os problemas que utilizaram a canalização 1, os pontos iniciais foram $(x_0^1)_i = (x_*)_i + 10$, para $i = 1, \dots, n$, quando utilizou-se as funções 2, 3, 4 e 6. Para as funções 1 e 5, os pontos iniciais foram os mesmos utilizados nos problemas irrestritos. Para os problemas com tipo de canalização 2 e 3, os pontos iniciais foram também os mesmos utilizados nos problemas irrestritos, com as seguintes exceções: para a função

1 com canalização do tipo 2 e para as funções 1 e 4 com canalização 3, o ponto inicial foi $(x_*)_i - 10$ e $(x_*)_i + 10$, para $i = 1, \dots, n$, respectivamente.

Para o Problema de Wolfe, o ponto inicial foi $x_0 = (\frac{1}{2}, \dots, \frac{1}{i+1}, \dots, \frac{1}{n+1})^\top$.

função	x_0^{lrr}
R	$(3, \dots, 3)^\top$
BROYD	$(-1, \dots, -1)^\top$
7D	$(-1, \dots, -1)^\top$
P	$(-1, \dots, -1)^\top$
PVC	$(10^{-3}, \dots, 10^{-3})^\top$
I	$(\frac{1}{n+1}(\frac{1}{n+1} - 1), \dots, i\frac{1}{n+1}(i\frac{1}{n+1} - 1), \dots, n\frac{1}{n+1}(n\frac{1}{n+1} - 1))^\top$

Tabela 5.5: Pontos iniciais para os problemas irrestritos

Seguem então abaixo as tabelas com os resultados obtidos.

5.3.2 Problemas irrestritos

Problema	RP	Tit	Naf	F	NGP	T
R	A	186	561	2.01D-13	6.48D-07	79.36
BROYD	A	17	18	6.76D-14	2.14D-06	1.19
7D	A	629	5627	3.94	2.17D-06	15.95
P	A	18	25	9.69D-03	1.15D-06	0.18
PVC	A	9	10	3.37D-10	4.57D-06	3.79
I	B	6462	100008	1.65D-05	1.65D-05	6657.26

Tabela 5.6: Irrestrito: BOX-QUACAN Modificado

Problema	RP	Tit	Naf	F	NGP	T
R	A	40	60	7.92D-17	5.45D-08	4.72
BROYD	A	48	56	1.41	1.81D-06	5.71
7D	C	1001	1024	1.02D+02	3.35D-03	7.06
P	A	11	19	9.69D-03	6.27D-07	0.13
PVC	A	198	205	4.86D-09	9.63D-07	5.22
I	A	124	152	2.68D-12	9.65D-07	11.31

Tabela 5.7: Irrestrito: L-BFGS-B

<i>Problema</i>	η	<i>RP</i>	<i>Tit</i>	<i>Naf</i>	<i>F</i>	<i>NGP</i>	<i>T</i>
R	0.9	A	76	130	2.47D-15	2.21D-06	3.62
BROYD	0.9	A	23	53	1.12D-14	7.36D-07	1.97
7D	0.9	A	248	337	5.57D+01	8.27D-07	8.07
P	0.9	A	14	22	1.03D-02	8.94D-08	0.12
PVC	0.9	A	13	11	7.12D-11	7.63D-07	20.91
I	0.9	A	71	107	1.58D-14	1.83D-07	79.54

Tabela 5.8: Irrestrito: EASY

Com os resultados dados nas tabelas acima para os problemas irrestritos, calculou-se a média geométrica do número de iterações, do número de avaliações de função e do tempo, para cada algoritmo. Isto está representado na tabela 5.9 abaixo. As colunas *MI*, *MNaf* e *MT*, representam as médias geométricas do número de iterações, do número de avaliações de função e do tempo, respectivamente.

<i>Software</i>	<i>MI</i>	<i>MNaf</i>	<i>MT</i>
BOX-QUACAN Modificado	113	335	13.79
L-BFGS-B	89	112	3.40
EASY	42	62	4.77

Tabela 5.9: Médias geométricas dos problemas irrestritos

5.3.3 Problemas canalizados

Restritos: canalização 1

<i>Problema</i>	<i>RP</i>	<i>Tit</i>	<i>Naf</i>	<i>F</i>	<i>NGP</i>	<i>T</i>
R	A	5	6	2.5D+3	0.0	0.27
BROYD	A	6	7	6.07D+5	0.0	0.36
7D	A	7	8	5.12D+3	0.0	0.01
P	A	7	8	2.57D+7	0.0	0.27
PVC	C	10000	30064	2.44D-1	1.57D-2	852.00
I	A	119	121	1.94D+8	0.0	12.35

Tabela 5.10: Canalização 1: BOX-QUACAN Modificado

<i>Problema</i>	<i>RP</i>	<i>Tit</i>	<i>Naf</i>	<i>F</i>	<i>NGP</i>	<i>T</i>
R	A	16	19	2.5D+3	3.55D-13	1.85
BROYD	A	1	2	6.07D+5	0.0	0.14
7D	A	1	2	5.12D+3	0.0	0.01
P	A	1	2	2.57D+7	0.0	0.17
PVC	B	10000	10513	9.80D-2	2.91D-5	669.55
I	A	1	2	1.94D+8	0.0	0.14

Tabela 5.11: Canalização 1: L-BFGS-B

<i>Problema</i>	<i>RP</i>	<i>Tit</i>	<i>Naf</i>	<i>F</i>	<i>NGP</i>	<i>T</i>
R	A	3	5	2.5D+3	3.19D-7	0.13
BROYD	A	4	5	9.32D+5	0.0	0.01
7D	A	4	5	5.12D+3	0.0	0.01
P	A	6	7	2.57D+7	0.0	0.16
PVC	A	206	207	9.40D-2	8.90D-7	5147.41
I	A	9	10	1.94D+8	0.0	5.12

Tabela 5.12: Canalização 1: EASY

Restritos: canalização 2

<i>Problema</i>	<i>RP</i>	<i>Tit</i>	<i>Naf</i>	<i>F</i>	<i>NGP</i>	<i>T</i>
R	A	8	9	2.5D+3	0.0	0.34
BROYD	A	17	18	6.76D-14	2.14D-6	1.43
7D	A	300	886	3.05D+2	1.31D-6	12.88
P	A	1	2	2.43D+7	0.0	0.02
PVC	A	8127	27058	9.33D-2	7.90D-6	9810.00
I	A	1723	20159	4.62D-13	3.05D-6	1090.10

Tabela 5.13: Canalização 2: BOX-QUACAN Modificado

<i>Problema</i>	<i>RP</i>	<i>Tit</i>	<i>Naf</i>	<i>F</i>	<i>NGP</i>	<i>T</i>
R	A	1	2	2.5D+3	0.0	0.11
BROYD	A	29	33	7.11D-14	9.10D-7	3.82
7D	C	1001	1012	5.93D+2	2.97D-3	28.19
P	A	1	2	2.43D+7	0.0	0.15
PVC	C	10000	10593	9.74D-2	3.06D-5	700.97
I	A	46	51	9.92D-13	9.06D-7	3.85

Tabela 5.14: Canalização 2: L-BFGS-B

<i>Problema</i>	<i>RP</i>	<i>Tit</i>	<i>Naf</i>	<i>F</i>	<i>NGP</i>	<i>T</i>
R	A	10	11	2.5D+3	0.0	0.30
BROYD	A	7	10	1.93	3.47D-8	0.29
7D	A	43	64	4.7D+2	7.2D-7	6.00
P	A	1	2	2.43D+7	0.0	0.04
PVC	A	168	170	9.33D-2	4.80D-7	4211.09
I	A	98	148	8.67D-15	3.04D-7	158.12

Tabela 5.15: Canalização 2: EASY

Restritos: canalização 3

<i>Problema</i>	<i>RP</i>	<i>Tit</i>	<i>Naf</i>	<i>F</i>	<i>NGP</i>	<i>T</i>
R	A	6	7	2.33D+6	0.0	0.22
BROYD	A	12	14	1.30D+5	9.5D-7	0.72
7D	A	11	13	3.70D+3	3.47D-13	0.01
P	A	8	9	1.03D+6	0.0	0.05
PVC	A	2	3	2.0d+4	0.0	0.08
I	A	1	2	4.93D+7	0.0	0.26

Tabela 5.16: Canalização 3: BOX-QUACAN Modificado

<i>Problema</i>	<i>RP</i>	<i>Tit</i>	<i>Naf</i>	<i>F</i>	<i>NGP</i>	<i>T</i>
R	A	1	2	3.01D+5	0.0	0.14
BROYD	A	10	12	1.30D+5	1.96D-7	0.49
7D	A	15	17	3.67D+3	7.53D-7	0.06
P	A	3	5	2.56D+5	2.98D-11	0.04
PVC	A	1	2	2.0D+4	0.0	0.15
I	A	1	2	4.93D+7	0.0	0.14

Tabela 5.17: Canalização 3: L-BFGS-B

<i>Problema</i>	<i>RP</i>	<i>Tit</i>	<i>Naf</i>	<i>F</i>	<i>NGP</i>	<i>T</i>
R	A	5	6	2.33D+6	0.0	0.13
BROYD	A	14	29	2.02D+5	1.9D-5	0.67
7D	A	7	12	3.70D+3	4.83D-7	0.66
P	A	6	7	1.03D+6	0.0	0.06
PVC	A	1	2	2.0D+4	0.0	0.05
I	A	3	4	4.93D+7	0.0	1.84

Tabela 5.18: Canalização 3: EASY

Problema de Wolfe

<i>Software</i>	<i>RP</i>	<i>Tit</i>	<i>Naf</i>	<i>F</i>	<i>NGP</i>	<i>T</i>
BOX-QUACAN Modificado	D	30	135	-1.36D+02	2.47D-06	0.30
L-BFGS-B	A	23	87	-1.04D+02	2.18D-06	0.06
EASY	A	13	31	-1.07D+02	1.47D-06	0.07

Tabela 5.19: Problema de Wolfe

Da mesma forma como foi feita para os problemas irrestritos, calcula-se, para os problemas restritos, as médias geométricas do número de iterações, do número de avaliações de função e do tempo. Estas estão representadas na tabela 5.20, abaixo.

<i>Software</i>	MIt	MNaf	MT
BOX-QUACAN Modificado	25	43	1.01
L-BFGS-B	10	15	0.59
EASY	9	11	0.82

Tabela 5.20: Médias geométricas dos problemas restritos

5.4 Um exemplo de aplicação

Com o objetivo de mostrar um problema mais prático, introduz-se um modelo no qual aloca-se um determinado número de horas de estudo para uma dada disciplina, esperando-se um retorno. Isso faz com que menos horas sejam gastas com as demais disciplinas. Este problema pode ser modelado por:

$$\begin{aligned} \text{Maximizar} \quad & \sum_{i=1}^n \left[\alpha(x_i) \prod_{j \neq i} \beta_j(x_j) \right] \\ \text{sujeita a} \quad & x_i \geq 0, \quad i = 1, \dots, n, \end{aligned} \tag{5.4.1}$$

onde $\alpha(x_i)$ é o retorno com o estudo da matéria i e $\beta_j(x_j)$ é o fator que influencia negativamente em cada matéria i enquanto se estuda a matéria j . Para o problema

em questão, definiu-se

$$\alpha(x_i) = \bar{\alpha}(1 - e^{-x_i})$$

com $\bar{\alpha} = 10$ e $\bar{\alpha} = 12$ e

$$\beta_i(x_j) = \left(\frac{0.05}{i}\right)e^{-x_j} + \left(1 - \frac{0.05}{i}\right).$$

Neste problema, a Hessiana é diagonalmente dominante, o que favoreceu muito os resultados numéricos obtidos com o BOX-QUACAN Modificado. Pode-se comprovar isto através das tabelas 5.21 a 5.26, abaixo. Para a escolha do parâmetro d , foram feitos testes onde d assumiu os valores 0, 1 e 2. Os melhores resultados foram obtidos quando $d = 0$ e estes estão registrados nas tabelas 5.21 e 5.26. Para a aproximação inicial para a Hessiana, nos algoritmos BOX-QUACAN Modificado e L-BFGS-B, utilizou-se $B_0 = I$ e o ponto inicial $x_0 = 0$ para todos os algoritmos. Os demais parâmetros estão como definidos na seção 5.2.

Software	RP	Iter	FE	NGP	F	T
BOX QUACAN Modificado	A	32	33	1665.08	1.38d-5	0.66
L-BFGS-B	A	20	23	1665.08	9.47D-07	0.38
EASY	A	10	11	1665.08	4.36D-07	0.81

Tabela 5.21: Problema prático: $n = 200$ e $\bar{\alpha} = 10$

Software	RP	Iter	FE	NGP	F	T
BOX QUACAN Modificado	A	9	10	1998.09	1.12D-5	0.20
L-BFGS-B	A	21	24	1998.09	7.0D-7	0.40
EASY	A	10	11	1998.09	5.24D-7	0.81

Tabela 5.22: $n = 200$ e $\bar{\alpha} = 12$

5.5 Comentários e conclusões

De um modo geral, o desempenho de BOX-QUACAN Modificado, para os problemas cujas Hessianas da função objetivo têm estrutura de banda (problemas 1, 2, 3 e 5),

Software	RP	Iter	FE	NGP	F	T
BOX QUACAN Modificado	A	31	65	8304.67	3.07d-5	22.72
L-BFGS-B	A	22	27	8305.00	8.15D-07	11.20
EASY	A	10	11	8304.67	8.03D-07	42.95

Tabela 5.23: Problema prático: $n = 1000$ e $\bar{\alpha} = 10$

Software	RP	Iter	FE	NGP	F	T
BOX QUACAN Modificado	A	13	14	9965.60	9.41d-6	7.80
L-BFGS-B	A	23	26	9965.00	6.61D-07	10.79
EASY	A	12	13	9965.60	2.59D-07	49.92

Tabela 5.24: Problema prático: $n = 1000$ e $\bar{\alpha} = 12$

Software	RP	Iter	FE	NGP	F	T
BOX QUACAN Modificado	A	40	41	41502.68	4.30D-5	517.560
L-BFGS-B	A	21	24	41502.67	2.99D-7	215.50
EASY	A	13	14	41502.68	5.98D-7	1762.79

Tabela 5.25: Problema prático: $n = 5000$ e $\bar{\alpha} = 10$

Software	RP	Iter	FE	NGP	F	T
BOX QUACAN Modificado	A	16	17	49803.21	2.35D-5	218.78
L-BFGS-B	A	25	29	49803.21	3.35D-7	256.60
EASY	A	12	13	49803.22	4.31D-7	1657.98

Tabela 5.26: $n = 5000$ e $\bar{\alpha} = 12$

foi bom. Isto pode ser observado na figura 5.1 abaixo. Esta apresenta o gráfico das médias geométricas do tempo total de CPU gasto, para cada um dos algoritmos, estes indicados pelos números 1, 2 e 3, no eixo que representa o tipo do problema, para BOX-QUACAN Modificado, L-BFGS-B e EASY, respectivamente. Estas médias envolvem os resultados dos problemas 1, 2, 3 e 5, para Irrestritos e Canalizados 3, e 1, 2, 3, para Canalizados 1 (onde o problema 5 não converge para BOX-QUACAN Modificado e L-BFGS-B) e 2 (onde o problema 5 não converge para L-BFGS-B).

Deve-se destacar que os problemas 1, 2 e 5 possuem 5000 variáveis, caracterizando testes de grande porte.

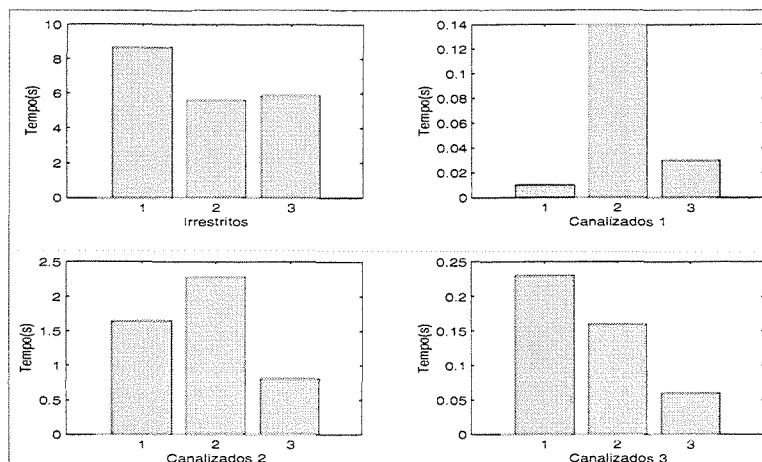


Figura 5.1: Médias geométricas do tempo de CPU.

Em relação aos problemas irrestritos, deve-se notar que o desempenho de BOX-QUACAN Modificado no problema 6, não levando à convergência (chegou ao número máximo de avaliações de função permitido, não convergindo pela norma do gradiente), influenciou muito no cálculo das médias geométricas da tabela 5.9. Sem este problema, obtêm-se as médias geométricas dadas na tabela 5.27 abaixo, na qual verifica-se que o número total de iterações de BOX-QUACAN Modificado passa a ser menor que o de L-BFGS-B, tornando-se também mais competitivo em termos de número de avaliações de função e tempo de CPU.

<i>Software</i>	MI _t	MN _{af}	MT
BOX-QUACAN Modificado	50	107	4.00
L-BFGS-B	84	106	2.67
EASY	38	56	2.72

Tabela 5.27: Médias geométricas dos problemas irrestritos com exceção do probl. 6

Fazendo uma análise geral dos problemas de minimização em caixas testados, mediante os resultados da tabela 5.20, conclui-se que EASY realiza menos iterações e avaliações de função do que os outros 2 algoritmos, porém gasta mais tempo de CPU.

Isto confirma resultados de desempenho (em termos de velocidade de convergência) dos métodos do tipo Newton discreto. Se levarmos em conta o tempo de execução gasto, a implementação BFGS com memória limitada mostrou-se a melhor alternativa, com algumas exceções, destacando-se, por exemplo, a função 4 com canalização 2. Neste caso, BOX-QUACAN Modificado apresentou um tempo de execução bem menor que o L-BFGS-B, mesmo sendo a Hessiana da função objetivo densa.

Do total de problemas testados, 3 não convergiram quando se utilizou o BOX-QUACAN Modificado e 4 não convergiram com o L-BFGS-B. Dentre aqueles que não convergiram com o BOX-QUACAN Modificado, o problema de Wolfe parou por gerar ponto estrangulado quando $\|g_p\| = 2.47 * 10^{-6}$. Já o EASY convergiu para todos os problemas testados, mostrando a implementação do tipo Newton discreto ser mais robusta que as implementações Quase-Newton.

Considerando agora o problema prático, nota-se que, com $\bar{\alpha} = 12$, o BOX-QUACAN Modificado teve um desempenho muito bom em relação ao L-BFGS-B e ao EASY mesmo a Hessiana da função objetivo deste problema sendo densa. Com $\bar{\alpha} = 10$, o BOX-QUACAN Modificado mostrou-se competitivo em relação ao L-BFGS-B e obteve um desempenho melhor que EASY. O fato de, como afirmado na seção 5.4, esta matriz Hessiana ser diagonalmente dominante, sendo que, com $\bar{\alpha} = 12$ este critério fica mais forte, pode ter favorecido o desempenho do BOX-QUACAN Modificado. Tem-se, assim, um problema de grande porte ($n = 1000$ e $n = 5000$), onde o BOX-QUACAN Modificado teve um comportamento computacional melhor que o dos demais algoritmos.

Com base nos resultados obtidos, pode-se concluir que BOX-QUACAN Modificado é

uma boa alternativa para os problemas cujas Hessianas da função objetivo têm estrutura de banda ou “quase-banda”. Para problemas gerais, BOX-QUACAN Modificado realizou, na maior parte dos testes, mais iterações que os demais métodos, exigindo um tempo total de CPU maior. No entanto, pode-se afirmar que mostrou-se mais robusto que L-BFGS-B, uma vez que não convergiu, de fato, somente para 2 problemas, enquanto L-BFGS-B falhou em 4 testes.

Bibliografía

- [1] M. S. BAZARAA, H. D. SHERALI, AND C. M. SHETTY, *Nonlinear Programming Theory and Algorithms*, John Wiley and Sons, Canada, 1993.
- [2] R. H. BIELSCHOWSKY, A. FRIEDLANDER, F. M. GOMES, J. M. MARTINEZ, AND M. RAYDAN, *An adaptive algorithm for bound constrained quadratic minimization*, *Investigación Operativa*, 7 (1998), pp. 67–102.
- [3] R. H. BYRD, P. LU, J. NOCEDAL, AND C. ZHU, *A limited memory algorithm for bound constrained optimization*, *Siam Journal on Scientific and Statistical Computing*, 16 (1995), pp. 1190–1208.
- [4] R. H. BYRD, J. NOCEDAL, AND R. B. SCHNABEL, *Representations of quasi-Newton matrices and their use in limited memory methods*, *Mathematical Programming*, 63 (1994), pp. 129–156.
- [5] P. H. CALAMAI AND J. J. MORÉ, *Projected gradient methods for linearly constrained problems*, *Mathematical Programming*, 39 (1987), pp. 93–116.
- [6] T. F. COLEMAN AND Y. LI, *An interior trust-region approach for nonlinear minimization*, *SIAM Journal on Optimization*, 6 (1996), pp. 418–445.

-
- [7] A. R. CONN, N. I. M. GOULD, AND P. L. TOINT, *Global convergence of a class of trust-region algorithms for optimization with simple bounds*, SIAM J. Numer. Anal., 25 (1988), pp. 433–460.
- [8] A. R. CONN, N. I. M. GOULD, AND P. L. TOINT, *Testing a class of methods for solving minimization problems with simple bounds on the variables*, Mathematics of Computation, 50 (1988), pp. 399–430.
- [9] J. E. DENNIS JR. AND R. B. SCHNABEL, *Least change secant updates for quasi-Newton methods*, SIAM Review, 21 (1979), pp. 443–459.
- [10] —, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall Englewood Cliffs, New Jersey, 1983.
- [11] M. A. DINIZ-EHRHARDT, M. A. GOMES-RUGGIERO, AND S. A. SANTOS, *Comparing the numerical performance of two trust-region algorithms for large-scale bound-constrained minimization*, Revista Latino Americana de Investigacion Operativa, 7 (1994), pp. 11–24.
- [12] —, *Numerical analysis of leaving-face parameters in bound-constrained quadratic minimization*, Optimization Methods and Software, 15 (2001), pp. 45–66.
- [13] A. FRIEDLANDER, J. M. MARTÍNEZ, AND S. A. SANTOS, *A new trust-region algorithm for bound constrained minimization*, Applied Mathematics and Optimization, 30 (1994), pp. 235–366.
- [14] G. H. GOLUB AND C. F. V. LOAN, *Matrix Computations*, The Johns Hopkins University Press, 1989.

-
- [15] J. J. MORÉ, B. S. GARBOW, AND K. E. HILLSTRON, *Testing unconstrained optimization software*, ACM Transactions on Mathematical Software, 7 (1981), pp. 17–41.
- [16] J. B. O. QUANDT, *Minimização de funções com restrições canalizadas utilizando falsas Hessianas de banda*, tese de doutorado, Departamento de Engenharia de Produção, Universidade Federal de Santa Catarina, 1996.
- [17] L. TOINT, *Some numerical results using a sparse matrix updating formula in unconstrained optimization*, Mathematics of Computation, 32 (1978), pp. 839–851.
- [18] P. WOLFE, *Explicit solution of an optimization problem*, Mathematical Programming, 2 (1972), pp. 258–260.