

Universidade Federal do Paraná

Paulo Domingos Conejo

**MÉTODOS DE REGIÃO DE CONFIANÇA SEM DERIVADAS
PARA OTIMIZAÇÃO RESTRITA**

Curitiba

2012

Paulo Domingos Conejo

**MÉTODOS DE REGIÃO DE CONFIANÇA SEM DERIVADAS
PARA OTIMIZAÇÃO RESTRITA**

Tese apresentada ao Programa de Pós-Graduação em Métodos Numéricos em Engenharia, Área de Concentração em Programação Matemática, Setores de Tecnologia e de Ciências Exatas da Universidade Federal do Paraná, como requisito parcial à obtenção do título de Doutor em Ciências.

Orientadora

Prof.^a. Dr.^a. Elizabeth Wegner Karas

Coorientador

Prof. Dr. Lucas Garcia Pedroso

Curitiba

2012

TERMO DE APROVAÇÃO

PAULO DOMINGOS CONEJO

MÉTODOS DE REGIÃO DE CONFIANÇA SEM DERIVADAS PARA OTIMIZAÇÃO RESTRITA

Tese aprovada como requisito parcial para a obtenção do grau de Doutor em Ciências, no Programa de Pós-Graduação em Métodos Numéricos em Engenharia - Programação Matemática - da Universidade Federal do Paraná, pela seguinte banca examinadora:

Prof^a. Dr^a. Elizabeth Wegner Karas - orientadora
Programa de Pós-Graduação em Métodos Numéricos
em Engenharia - PPGMNE, UFPR

Prof. Dr. Lucas Garcia Pedroso - coorientador
Departamento de Matemática, UFPR

Prof. Dr. Clóvis Caesar Gonzaga
Departamento de Matemática, UFSC

Prof. Dr. Ernesto Julián Goldberg Birgin
Instituto de Matemática e Estatística - USP

Prof. Dr. Luiz Carlos Matioli
Departamento de Matemática, UFPR

Prof^a. Dr^a. Neida Maria Patias Volpi
Departamento de Engenharia de Produção, UFPR

Curitiba, 14 de dezembro de 2012.

Aos meus pais Vicente e Rita.

Agradecimentos

À Professora Elizabeth pela orientação e dedicação no desenvolvimento deste trabalho. Obrigado pela paciência e os bons conselhos.

Ao meu coorientador, Professor Lucas, pelo apoio e valiosas contribuições.

Ao Professor Ademir pelas inegáveis contribuições. À Professora Neida por me apresentar ao grupo de pesquisa em otimização do Departamento de Matemática da UFPR e ao Professor Mario Martínez por sugerir o tema de pesquisa.

Aos meus colegas de pesquisa Adriano, Priscila e Professora Mael.

À Maristela Bandil, pela prontidão e eficiência com as quais realiza seu trabalho.

Ao Programa de Pós graduação em Métodos Numéricos da UFPR pela oportunidade, à UNIOESTE por oferecer o tempo suficiente para conclusão deste curso e à Fundação Araucária pelo apoio financeiro.

Aos meus pais e irmãos pelo apoio de sempre. À minha esposa pelo companheirismo e à minha filha pelo amor e carinho que sempre manifestou.

Resumo

Otimização sem derivadas tem sido de grande interesse nos últimos anos, principalmente pela necessidade em se resolver problemas definidos por funções cujos valores são calculados por simulação. Muitos métodos sem derivadas para problemas restritos já foram desenvolvidos, alguns baseados em região de confiança, com bons resultados numéricos. Métodos de região de confiança sem derivadas para o caso irrestrito têm consagradas provas de convergência. No entanto, para o caso restrito, até onde sabemos não há na literatura resultados teóricos de convergência. Nesta tese apresentamos dois algoritmos sem derivadas para o problema de otimização não linear com restrições. O primeiro algoritmo é baseado em região de confiança sem derivadas aplicado à problemas com restrições convexas. Em cada iteração é construído um modelo quadrático da função objetivo que deve ser minimizado na interseção da região de confiança com o conjunto viável. Com hipóteses razoáveis em otimização sem derivadas, provamos que todo ponto de acumulação da sequência gerada pelo algoritmo é estacionário de primeira ordem. O segundo algoritmo é desenvolvido e aplicado a problemas com restrições gerais. Também com a estrutura de região de confiança, o algoritmo é baseado em interpolação polinomial e utiliza ideias propostas por Michael Powell nas atualizações dos modelos. Experimentos numéricos mostram a eficiência e robustez do algoritmo proposto, cuja complexidade algorítmica em número de operações permite resolver problemas de grandes dimensões quando tratamos de otimização sem derivadas.

Palavras chave: algoritmos sem derivadas; interpolação polinomial; otimização restrita

Abstract

In this work we discuss trust-region derivative-free algorithms for constrained problems. In the first part of the work we propose a trust-region algorithm for the problem of minimizing a function within a convex closed domain. We assume that the objective function is differentiable but no derivatives are available. The algorithm has a very simple structure and allows a great deal of freedom in the choice of the models. Under reasonable assumptions for derivative-free schemes, we prove global convergence, that is to say, that all accumulation points of the sequence generated by the algorithm are stationary. In the second part we develop a trust-region derivative-free algorithm for problems with general constraints. The models are constructed by polynomial interpolation and updated using ideas proposed by Powell in his many works on this subject. Numerical experiments show the efficiency and robustness of the algorithm, whose good performance allows to solve problems considered large in the context of derivative-free optimization.

Keywords: derivative-free optimization; trust-region methods; constrained optimization; polynomial interpolation

Lista de Figuras

1.1	Caso em que $\Delta_k > \alpha\pi_k$	9
1.2	Caso em que $\Delta_k \leq \alpha\pi_k$ e $r_k \geq \eta$	9
3.1	$2n + 1$ pontos de interpolação para $n = 2$	23
3.2	6 pontos de interpolação para $n = 2$, com reordenamento de y^3 e y^5	25
3.3	Escolha entre dois modelos gerados pelo BOBYQA na mesma iteração	41
4.1	$(n + 1)(n + 2)/2$ pontos de interpolação para $n = 2$	50
4.2	Configurações para os 4 primeiros conjuntos interpoladores	52
5.1	Comparação de desempenho entre os Algoritmos m_1 , m_2 , utilizando todos os 216 problemas restritos da coleção HS	65
5.2	Comparação de desempenho entre os Algoritmos m_1 , m_2 , Restauração Inexata e HOPSPACK, utilizando 105 problemas restritos da coleção HS	68
5.3	Comparação de desempenho entre os Algoritmos m_1 , m_2 e DFO, utilizando 28 problemas restritos da coleção HS	70
5.4	Solução obtida com o Algoritmo m_1 . Área 2.1586 e 1643 avaliações da função objetivo	71
5.5	Solução obtida com o Algoritmo m_2 . Área 3.1415 e 259 avaliações da função objetivo	71
5.6	Solução obtida com o Algoritmo m_2 , com melhor precisão no cálculo das interseções das elipses. Área 2.8504 e 1461 avaliações da função objetivo	71
5.7	Solução obtida com o Algoritmo m_1 . Área 3.7177 e 1070 avaliações da função objetivo	72
5.8	Solução obtida com o Algoritmo m_2 . Área 3.6052 e 3572 avaliações da função objetivo	72
5.9	Solução obtida com o Algoritmo m_1 , com melhor precisão no cálculo das interseções das elipses. Área 3.7109 e 1378 avaliações da função objetivo	72

Lista de Tabelas

5.1	Tempo de CPU em segundos pelos Algoritmos m_1 e m_2	65
5.2	Resultados dos 105 problemas da coleção HS obtidos pelos Algoritmos m_1 , m_2 , Restauração Inexata e HOPSPACK	68
5.3	Resultados numéricos obtidos pelos Algoritmos m_1 , m_2 e DFO para os 28 problemas com restrições retirados de [20]	69
5.4	Resultados numéricos obtidos pelo Algoritmo m_1 aplicado ao problema (5.3)	74
5.5	Resultados numéricos obtidos pelo Algoritmo m_1 aplicado ao problema (5.4)	75
5.6	Resultados numéricos obtidos pelo Algoritmo m_1 aplicado ao problema (5.5)	76
5.7	Dados dos 216 problemas com restrições de igualdade ou desigualdade da coleção HS	88

Sumário

Introdução	1
1 Convergência global	4
1.1 Método de região de confiança clássico	4
1.2 Região de confiança para o caso convexo e sem derivadas	5
1.2.1 O algoritmo	6
1.2.2 Análise de convergência	9
2 Interpolação polinomial	16
2.1 Interpolação quadrática	16
2.2 Modelos quadráticos subdeterminados	20
3 Algoritmo BOBYQA	22
3.1 Construções	23
3.2 Atualizações	26
3.3 Cálculo do passo	36
3.4 Atualização do conjunto interpolador	37
3.5 Mudanças no modelo	38
3.6 Outros procedimentos	42
3.6.1 Atualizações das regiões	42
3.6.2 Critério de parada	43
3.7 O Algoritmo	44
4 Algoritmo implementado	48
4.1 Ideia geral	48
4.2 Conjunto interpolador e modelo	49
4.2.1 Construção	50
4.2.2 Atualização	51
4.3 Lagrangiano Aumentado	53
4.3.1 Descrição do algoritmo	53
4.3.2 Convergência	55
4.3.3 ALGENCAN	56
4.4 O algoritmo implementado	56
4.5 Detalhes da implementação	58
4.6 Resultados teóricos	59

5 Testes numéricos	62
5.1 Problemas da coleção Hock-Schittkowski	63
5.1.1 Desempenho de variações do algoritmo	64
5.1.2 Comparação com algoritmos disponíveis na literatura	66
5.2 Aplicação	70
5.3 Problemas de dimensões maiores	73
Conclusões	77
Referências Bibliográficas	77
Apêndice A	84

Introdução

Métodos de região de confiança são uma classe de algoritmos para resolver problemas de otimização não linear. São baseados em modelos quadráticos ou lineares, que aproximam a função objetivo em uma vizinhança de um ponto corrente. No caso em que as derivadas da função objetivo não estão disponíveis, tais modelos são obtidos somente com informações dos valores da função e podem ser construídos por interpolação, regressão ou qualquer outra técnica de aproximação.

Quando a função objetivo permite aproximações de derivadas por algum método numérico, em geral, um método com derivadas pode utilizar este artifício e resolver problemas de otimização com restrições sem utilizar explicitamente derivadas. No entanto, existem problemas [20, 45] em que a função objetivo é oriunda de simulação e apresenta ruídos, situações em que não é adequado aproximar derivadas. Nestes casos, pode não ser possível aplicar métodos com derivadas e a utilização de métodos sem derivadas é uma alternativa.

Desde a década de 60, Powell trabalha com otimização sem derivadas e tem hoje substancial influência no desenvolvimento da área. Os métodos NEWUOA [52] e BOBYQA [54] desenvolvidos por Powell constroem modelos quadráticos satisfatórios, mesmo deixando liberdade em parâmetros do modelo. Os modelos quadráticos são obtidos por interpolação polinomial de Lagrange, com as atualizações dos modelos feitas a partir da minimização da norma de Frobenius [49]. Em NEWUOA a otimização é irrestrita, enquanto que BOBYQA resolve problemas em caixa.

Existem na literatura diversos métodos para otimização sem derivadas e com estruturas diferentes. Para a estrutura de região de confiança, no caso irrestrito podemos citar os trabalhos [19, 22, 23, 40, 47, 48, 57, 67], contendo inclusive prova de convergência para métodos baseados em interpolação polinomial [22]. Métodos de região de confiança para problemas com restrições de caixas são considerados em Arouxet, Echebest e Pilotta [4] e Powell [54]. Problemas com restrições lineares são tratados em Powell [56], Conn, Scheinberg e Toint [20]. O algoritmo em [20] proposto por Conn, Scheinberg e Toint foi o primeiro a considerar métodos de região de confiança sem derivadas para problemas com restrições gerais. Utiliza aproximações quadráticas em regiões de confiança e as restrições originais do problema são consideradas nos subproblemas. A técnica utilizada para atualização do conjunto interpolador trabalha com o conceito de *bom posicionamento* [19, 22]. A estratégia para atualizar o modelo é minimizar a norma de Frobenius da Hessiana do modelo como sugerido em [23], mas sem resultados de convergência para o caso restrito. Mais tarde, os autores provam em [19] a convergência para o caso irrestrito. Em [20] são apresentados resultados numéricos de problemas da coleção CUTE, e comparados com COBYLA [47] e com LANCELOT [16]. Os subproblemas são resolvidos por NPSOL [33], que utiliza programação quadrática sequencial. Outro algoritmo que considera restrições gerais incorporando-as

aos subproblemas é o CONDOR, descrito em [6], onde os autores fazem uma extensão do Algoritmo UOBYQA [48] para problemas restritos.

Outra estrutura para otimização sem derivadas é o Algoritmo GSS (*Generating Set Search*), que engloba muitas estratégias, incluindo o GPS (*Generalized Pattern Search*) [38]. Algoritmos com esta estrutura desfrutam de teoria de convergência. Em [37, 39], os autores identificam as restrições ativas para gerar conjuntos de buscas (GSS) e apresentam resultados de convergência quando as restrições são lineares. Para restrições em caixa há o trabalho de Lewis e Torczon [38]. Em [27], Audet e Dennis consideram problemas com restrições de igualdade e desigualdade, onde é introduzida uma classe de algoritmos, denominada MADS (*Mesh Adaptive Direct Search*), para otimização não linear. Sob certas hipóteses [60], MADS trata as restrições utilizando barreiras e considera o problema como irrestrito. MADS é uma generalização da busca padrão, usa um conjunto denso de direções e possui resultados de convergência para problemas não suaves. Em [10], Bueno, Friedlander, Martínez e Sobral propõem um esquema de restauração inexata, onde o algoritmo GSS é utilizado na fase de otimalidade, evitando assim avaliações do gradiente da função objetivo.

A estrutura sem derivadas também está presente em algoritmos do tipo Lagrangianos Aumentados. Quando as restrições são de igualdades e desigualdades, encontramos na literatura os trabalhos de Kolda, Lewis e Torczon [36] e de Diniz-Ehrhardt, Martínez e Pedroso [26], onde os autores apresentam um método Lagrangiano Aumentado sem derivadas para otimização contínua, com restrições gerais no nível superior e restrições gerais e lineares no nível inferior. Sob diferentes condições de qualificações, provam convergência a ponto estacionário de primeira ordem.

No que diz respeito a convergência de métodos de região de confiança com derivadas, tanto para problemas restritos como irrestritos, podemos encontrar na literatura algoritmos bem estabelecidos e com sólidos resultados de convergência, por exemplo [15, 17, 18, 44, 59]. Para métodos de região de confiança sem derivadas irrestritos a convergência já está estabelecida por exemplo nos trabalhos [19, 22, 57]. No entanto, quando se trata de métodos de região de confiança sem derivadas para problemas restritos, apesar de algoritmos com bons desempenhos práticos terem sido desenvolvidos [4, 16, 54, 56, 61, 66], permanecem as dificuldades em estabelecer resultados teóricos de convergência.

Nesta tese consideramos a classe de métodos de região de confiança sem derivadas, que tem como pioneiro Winfield [68] e tem sido exaustivamente estudado por Powell [47, 52, 53, 54, 57], Conn e Toint [23], Conn, Scheinberg e Toint [19], Conn, Scheinberg e Vicente [22], Fasano, Morales e Nocedal [32], Gratton, Toint e Tröltzsch [35].

Contribuição: esta tese apresenta essencialmente duas contribuições na área da otimização sem derivadas. Propomos um algoritmo globalmente convergente para o caso em que o conjunto viável é convexo e fechado e a função objetivo não tem derivadas disponíveis. O algoritmo é bastante geral pois permite o uso de qualquer técnica para obtenção dos modelos, desde que sejam aproximações locais da função objetivo. Além disto, o algoritmo permite o uso de qualquer método para a resolução dos subproblemas de minimização do modelo da função objetivo sujeito à região de confiança e às restrições verdadeiras, desde que o decréscimo obtido no modelo seja pelo menos uma fração do decréscimo fornecido pelo ponto de Cauchy. Com estas considerações, além de hipóteses clássicas, provamos a convergência global do algoritmo. Até onde

sabemos, nossa abordagem é a primeira em apresentar teoria de convergência. A outra contribuição está relacionada ao desenvolvimento de um algoritmo que pode ser aplicado para resolver problemas de otimização sem derivadas da função objetivo em domínios gerais. O algoritmo, disponível em www.ufpr.br/~ewkaras/doutorado/ckp, foi implementado em Fortran 77. Não tem prova de convergência. Combina importantes características do Algoritmo BOBYQA (*Bound Optimization By Quadratic Approximation*) proposto por Powell [54] na construção e atualização dos modelos e utiliza o Algoritmo ALGENCAN (*Lagrangiano Aumentado*) introduzido por Andreani, Birgin, Martínez e Schuverdt [2, 3, 70] nas soluções dos subproblemas. Cada modelo é minimizado sujeito às restrições do problema original e à região de confiança.

Assim podemos explicitar os seguintes objetivos:

- Propor um algoritmo de região de confiança sem derivadas da função objetivo em domínios convexos e fechados.
- Provar convergência global do algoritmo proposto.
- Desenvolver e implementar um algoritmo de região de confiança baseado em interpolação polinomial e que pode ser aplicado a problemas com restrições gerais.
- Exibir testes numéricos.

O texto está organizado em cinco capítulos. No primeiro apresentamos e discutimos resultados de convergência de um algoritmo de região de confiança sem derivadas. No segundo revisamos a teoria de interpolação polinomial. No terceiro fazemos um estudo do Algoritmo BOBYQA. No quarto capítulo, descrevemos o algoritmo desenvolvido e que foi implementado. Finalmente são apresentados experimentos numéricos no quinto capítulo.

Notações

$[x^k]_i$ indica a i -ésima componente do vetor $x^k \in \mathbb{R}^n$.

$\|\cdot\| = \|\cdot\|_2$ é a norma euclidiana.

$B(y, \Delta) = \{x \in \mathbb{R}^n \mid \|x - y\| \leq \Delta\}$.

$B_\infty(y, \Delta) = \{x \in \mathbb{R}^n \mid \|x - y\|_\infty \leq \Delta\}$.

P_Ω denota o operador projeção euclidiana sobre Ω .

$e^i \in \mathbb{R}^n$ representa o i -ésimo vetor canônico do \mathbb{R}^n .

$f_i = f(y^i)$ para $y^i \in \mathbb{R}^n$.

$\|\cdot\|_F$ indica a norma de Frobenius $\|S\|_F^2 = \sum_{i=1}^n \sum_{j=1}^n S_{i,j}^2$, $S \in \mathbb{R}^{n \times n}$.

Capítulo 1

Convergência global

Neste capítulo apresentamos uma das contribuições desta tese. Propomos um algoritmo de região de confiança sem derivadas para problemas restritos e fazemos um estudo teórico sobre sua convergência. Os resultados obtidos estão disponíveis em [13].

Na Seção 1.1 deste capítulo discutimos a estrutura geral de algoritmos de região de confiança. Na Seção 1.2 abordamos um algoritmo de região de confiança sem derivadas para restrições convexas. Com hipóteses razoáveis, provamos que todo ponto de acumulação gerado pelo algoritmo proposto é estacionário de primeira ordem.

De um modo geral, um método de região de confiança [44] define um modelo da função objetivo e uma região em torno do ponto corrente na qual confiamos no modelo. Então, é calculado um minimizador aproximado do modelo na região de confiança. Caso o ponto forneça uma redução razoável no valor da função objetivo, este é aceito e o processo repetido. Caso contrário, o ponto é recusado e o tamanho da região de confiança reduzido.

1.1 Método de região de confiança clássico

Considere o problema de otimização irrestrito

$$\begin{array}{ll} \text{minimizar} & f(x) \\ \text{sujeita a} & x \in \mathbb{R}^n, \end{array}$$

com a função $f : \mathbb{R}^n \rightarrow \mathbb{R}$ diferenciável.

Em cada iteração k do algoritmo de região de confiança, é considerado o ponto corrente x^k e o modelo quadrático na forma

$$q_k(d) = f(x^k) + (g^k)^T d + \frac{1}{2} d^T G_k d,$$

onde g^k e G_k são, respectivamente, o gradiente em $d = 0$ e a Hessiana do modelo. Minimizando aproximadamente q_k no conjunto

$$\{d \in \mathbb{R}^n \mid \|d\| \leq \Delta_k\}$$

com Δ_k o raio da região de confiança, é obtido um ponto tentativo $x^k + d^k$. Espera-se que o ponto tentativo proporcione uma redução na função objetivo que seja no mínimo

uma fração da redução do modelo. Para formalizar este conceito definimos a redução real na função objetivo e a redução predita pelo modelo como

$$ared = f(x^k) - f(x^k + d^k) \quad \text{e} \quad pred = q_k(0) - q_k(d^k). \quad (1.1)$$

O ponto tentativo é aceito como novo iterando se para um dado $\eta > 0$, $ared \geq \eta pred$. Quando vale a desigualdade anterior o modelo é atualizado e o tamanho do raio Δ_k mantido ou incrementado. Caso contrário, o ponto tentativo é recusado e o raio Δ_k decrementado.

A prova de convergência para algoritmo de região de confiança com derivadas para o caso de restrições convexas é estabelecida em [18, Seção 12.2]. O estudo trabalha com o fato de que os gradientes do modelo e da função objetivo coincidem no iterando atual. Este é, por exemplo, o caso da aproximação por Taylor. Em região de confiança sem derivadas, esta propriedade dos gradientes não necessariamente é verdadeira, mas pode ser controlada, dependendo da técnica utilizada para a construção dos modelos. Na próxima seção, a relação entre o gradiente do modelo e da função objetivo em um ponto não são necessariamente coincidentes. Os modelos podem ser construídos por qualquer técnica que não utilize informações da derivada da função objetivo, desde que mantenha determinadas hipóteses sobre a qualidade dos modelos, dentre elas, um controle da relação entre os gradientes do modelo e da função objetivo.

1.2 Região de confiança para o caso convexo e sem derivadas

Discutimos a convergência global de um algoritmo de região de confiança sem derivadas, para resolver o problema de programação não linear

$$\begin{aligned} &\text{minimizar} && f(x) \\ &\text{sujeita a} && x \in \Omega, \end{aligned} \quad (1.2)$$

com $\Omega \subset \mathbb{R}^n$ não vazio, fechado, convexo e $f : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função diferenciável. Apesar da função objetivo ser diferenciável, assumimos que suas derivadas não estão disponíveis. Esta situação é comum em várias aplicações [20], particularmente quando a função objetivo é fornecida por um pacote de simulação ou uma caixa preta. Tais situações práticas têm motivado a pesquisa em otimização sem derivadas nos últimos anos [20, 22].

Consideramos a classe de métodos de região de confiança sem derivadas, que tem como pioneiro Winfield [68] e tem sido exaustivamente estudado, para problemas irrestritos e com restrições em caixa, por Powell [47, 52, 53, 54, 57], Conn e Toint [23], Conn, Scheinberg e Toint [19], Conn, Scheinberg e Vicente [22], Fasano, Morales e Nocedal [32], Gratton, Toint e Tröltzsch [35]. No caso irrestrito, se os modelos são baseados em interpolação polinomial e o conjunto interpolador substancialmente modificado em cada iteração, resultados de convergência são obtidos (ver [22] para uma cobertura completa). Em [35, 57] resultados de convergência para problemas irrestritos são estabelecidos para algoritmos que substituem apenas um ponto do conjunto interpolador por iteração, desde que o novo ponto e o ponto a ser substituído sejam adequadamente escolhidos. Para o caso derivável e com restrições convexas, a prova de convergência é estabelecida em [18, Seção 12.2].

Neste capítulo estamos particularmente interessados em problemas em que é fácil projetar um ponto sobre o conjunto viável Ω . O algoritmo proposto considera modelos quadráticos (ou lineares) que aproximam a função objetivo e são baseados em informações de ordem zero. Os modelos devem ser construídos tais que seus gradientes representem adequadamente o gradiente da função objetivo. Esta propriedade pode ser atingida por muitas técnicas sem derivadas, a maioria delas baseadas em interpolação polinomial [12, 21, 22, 25]. Em cada iteração o modelo corrente é minimizado na interseção de Ω com a região de confiança, de modo que a solução deste subproblema satisfaça uma condição do tipo Cauchy. Este ponto deve ser aceito ou rejeitado como um novo iterando, com base na razão entre as reduções predita e real, clássica em região de confiança. Temos o problema irrestrito como caso particular do algoritmo proposto. Nesta situação, a projeção é reduzida ao gradiente do modelo e o passo clássico de Cauchy [18, 22, 44] pode ser utilizado nas soluções dos subproblemas.

Os resultados de convergência são obtidos do fato de que um ponto $x \in \Omega$ satisfaz as condições de Karush-Kuhn-Tucker (KKT) se, e somente se, a norma do gradiente projetado da função objetivo sobre Ω é zero neste ponto [18, Theorem 12.1.2].

1.2.1 O algoritmo

Propomos um algoritmo geral de região de confiança para resolver o problema (1.2) e que gera uma sequência de minimizadores aproximados para os subproblemas restritos. O algoritmo permite grande liberdade nas construções e resoluções dos subproblemas.

Em cada iteração $k \in \mathbb{N}$, consideramos o iterando atual $x^k \in \Omega$ e o modelo quadrático

$$q_k(d) = f(x^k) + (g^k)^T d + \frac{1}{2} d^T G_k d, \quad (1.3)$$

onde $g^k = \nabla q_k(0) \in \mathbb{R}^n$ e $G_k \in \mathbb{R}^{n \times n}$ é uma matriz simétrica. Qualquer modelo quadrático desta forma pode ser utilizado, desde que forneça uma aproximação suficientemente precisa da função objetivo. Assumimos pouco sobre G_k , somente simetria e limitação uniforme, permitindo que modelos lineares possam ser utilizados. Não utilizamos modelos obtidos por Taylor pois nos interessa o caso onde, apesar de existentes, as derivadas não estão disponíveis.

Consideramos a medida de estacionariedade do problema de minimizar o modelo sobre o conjunto convexo e fechado Ω em x^k , definido por

$$\pi_k = \|P_\Omega(x^k - g^k) - x^k\|,$$

onde P_Ω denota a projeção ortogonal sobre o conjunto Ω . Cabe ressaltar que consideramos o caso em que é fácil projetar um ponto sobre o conjunto viável Ω . Relacionamos a medida π_k com a medida de estacionariedade do problema original. Note que o ponto $x^* \in \Omega$ é estacionário para o problema original (1.2) se, e somente se,

$$\|P_\Omega(x^* - \nabla f(x^*)) - x^*\| = 0.$$

Para provar convergência a pontos estacionários, assumimos que as soluções aproximadas, $d^k \in \mathbb{R}^n$, dos subproblemas de região de confiança

$$\begin{aligned} & \text{minimizar} && q_k(d) \\ & \text{sujeita a} && x^k + d \in \Omega \\ & && \|d\| \leq \Delta_k, \end{aligned} \quad (1.4)$$

satisfaçam a condição de decréscimo

$$q_k(0) - q_k(d^k) \geq c_1 \pi_k \min \left\{ \frac{\pi_k}{1 + \|G_k\|}, \Delta_k, 1 \right\}, \quad (1.5)$$

com $c_1 > 0$ uma constante independente de k . Consideramos que conhecemos um algoritmo que resolve aproximadamente o subproblema (1.4).

Condições do tipo (1.5) são bem conhecidas nas abordagens com regiões de confiança e utilizadas por vários autores, em diferentes situações. No caso irrestrito, em que $\Omega = \mathbb{R}^n$, a medida de estacionariedade π_k é simplesmente $\|g^k\|$ e o clássico passo de Cauchy d_c^k satisfaz a condição

$$q_k(0) - q_k(d_c^k) \geq c_1 \|g^k\| \min \left\{ \frac{\|g^k\|}{1 + \|G_k\|}, \Delta_k \right\},$$

como provado em [44, Lema 4.5] e [22, Teorema 10.1] com e sem derivadas da função objetivo, respectivamente. Condições do tipo (1.5) também aparecem ao longo do livro [18], em diferentes contextos. Em [34] os autores provam convergência global de um método de filtro para programação não linear, assumindo que as soluções aproximadas dos subproblemas satisfazem uma condição similar a (1.5). Para o caso de otimização não linear com restrições de caixa e sem derivadas, Tröltzsch [66] também assume esta condição quando trata o problema (1.2) com Ω uma caixa, também para o caso sem derivadas. Neste trabalho, que é a tese de doutorado da Anke Tröltzsch sob orientação de Gratton e Toint, o autor sugere a possibilidade de provar convergência para um algoritmo de região de confiança baseado em interpolação polinomial.

Depois de calculada uma solução aproximada do subproblema, analisamos se a solução fornece um decréscimo satisfatório no modelo. Como usual em métodos de região de confiança, o passo tentativo é avaliado por meio da razão

$$r_k = \frac{ared}{pred}, \quad (1.6)$$

onde *ared* e *pred* são definidas em (1.1).

Apresentamos o Algoritmo 1.1 de região de confiança sem derivadas, sem especificar como o modelo é atualizado e qual algoritmo interno é utilizado para resolver os subproblemas.

Algoritmo 1.1 *Algoritmo geral*

Dados: $x^0 \in \Omega$, $\alpha > 0$, $\Delta_0 > 0$, $0 < \tau_1 < 1 \leq \tau_2$, $\eta \in (0, \frac{3}{4})$.

Defina $k = 0$.

REPITA

Obtenha o modelo q_k (1.3).

SE $\Delta_k > \alpha \|P_\Omega(x^k - g^k) - x^k\|$, ENTÃO

$$\Delta_{k+1} = \tau_1 \Delta_k, \quad d^k = 0 \quad \text{e} \quad x^{k+1} = x^k.$$

SENÃO

Determine uma solução d^k de (1.4) satisfazendo (1.5).

SE $r_k \geq \eta$, ENTÃO

$$x^{k+1} = x^k + d^k \quad \text{e} \quad \Delta_{k+1} = \tau_2 \Delta_k.$$

SENÃO

$$x^{k+1} = x^k \quad \text{e} \quad \Delta_{k+1} = \tau_1 \Delta_k.$$

$k = k + 1$.

O modelo pode ser modificado em toda iteração, pois deve se ajustar à função objetivo em uma vizinhança do ponto corrente. Provamos na Seção 1.2.2 que $\Delta_k \rightarrow 0$ quando $k \rightarrow \infty$, o que é importante nas provas de convergência de métodos de região de confiança sem derivadas. Isto sugere também que, dada uma tolerância $\varepsilon > 0$ e parâmetros $\beta_1, \beta_2 > 0$, a combinação de $\Delta_k \leq \beta_1 \varepsilon$ e $\pi_k \leq \beta_2 \varepsilon$ pode ser utilizada como critério de parada na implementação do algoritmo. Quando π_k é pequeno, o iterando está provavelmente perto de uma solução do problema de minimizar o modelo dentro do conjunto viável Ω . Por outro lado, se Δ_k é grande, não podemos garantir que o modelo represente adequadamente a função objetivo. Então, quando $\Delta_k > \alpha \pi_k$, o raio da região de confiança é reduzido objetivando encontrar modelos mais precisos. Embora possamos tomar $\alpha = 1$, este parâmetro deve ser utilizado para balancear a magnitude de π_k e Δ_k de acordo com o problema.

Pelo Algoritmo 1.1 a razão dada em (1.6) está bem definida, pois, na iteração k em que o algoritmo a calcula, vale $\Delta_k \leq \alpha \pi_k$. Logo, $\pi_k \neq 0$ e pela condição de decréscimo (1.5), temos que

$$q_k(0) - q_k(d^k) \neq 0.$$

Nas Figuras 1.1 e 1.2 mostramos o comportamento do algoritmo em duas situações particulares para uma mesma iteração. Apresentamos a região de confiança delimitada pela circunferência pontilhada, o conjunto convexo Ω (linear e limitado), as curvas de nível do modelo (elipses) e da função objetivo. Nestas figuras, o vetor u representa o gradiente projetado $P_\Omega(x^k - g^k) - x^k$. Na Figura 1.1, $\Delta_k > \alpha \pi_k$ para $\alpha = 1$. Neste caso, $x^{k+1} = x^k$ e o raio da região é reduzido pela metade. Na situação representada pela Figura 1.2, estamos com $\Delta_k \leq \alpha \pi_k$. Neste caso, quando ocorre o decréscimo $r_k \geq \eta$, o iterando é atualizado para $x^{k+1} = x^k + d^k$ e o raio da região de confiança é mantido.

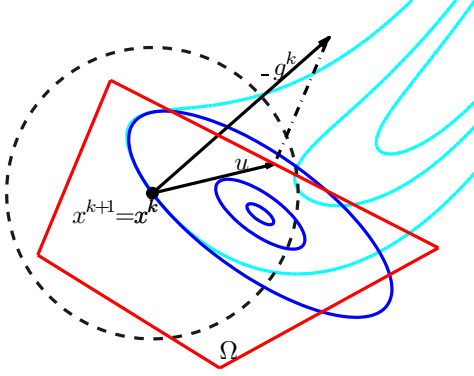


Figura 1.1: Caso em que $\Delta_k > \alpha\pi_k$

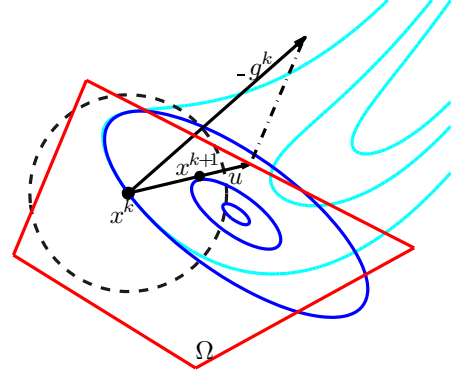


Figura 1.2: Caso em que $\Delta_k \leq \alpha\pi_k$ e $r_k \geq \eta$

1.2.2 Análise de convergência

De agora em diante, assumimos que o algoritmo gera uma sequência infinita $\{x^k\} \subset \Omega$. Provamos que todo ponto de acumulação da sequência gerada pelo algoritmo é estacionário.

Consideramos as seguintes hipóteses para a análise de convergência.

H1 f é diferenciável e ∇f é Lipschitziana com constante de Lipschitz $L > 0$.

H2 f é limitada inferiormente no conjunto Ω .

H3 As matrizes G_k são limitadas superiormente, isto é, existe $\beta \geq 1$ tal que para todo $k \in \mathbb{N}$, $\|G_k\| \leq \beta - 1$.

H4 Existe uma constante $c_2 > 0$ tal que para todo $k \in \mathbb{N}$,

$$\|g^k - \nabla f(x^k)\| \leq c_2 \Delta_k.$$

As Hipóteses H1 e H2 impõem condições sobre a função objetivo, enquanto H3 e H4 descrevem propriedades que devem ser satisfeitas pelos modelos. As três primeiras hipóteses são comuns em análise de convergência para algoritmos de região de confiança com e sem derivadas. A Hipótese H4 impõe que o modelo represente adequadamente a função objetivo próximo ao ponto corrente. Existem algoritmos capazes de encontrar modelos com tal propriedade sem calcular $\nabla f(x^k)$, por exemplo [22, Capítulo 6]. No próximo capítulo fazemos uma revisão sobre como a Hipótese H4 pode ser mantida quando utilizamos interpolação polinomial. O Algoritmo 1.1 permite utilizar qualquer técnica para satisfazer a Hipótese H4, porém ressaltamos que na literatura o mais comum é interpolação polinomial [22, 32, 35, 52, 61]. A Hipótese H4 é omitida em alguns algoritmos práticos e com bons resultados numéricos [20, 32], mas em nosso algoritmo é uma propriedade fundamental para estabelecer convergência.

Para os propósitos de nossa análise, consideramos o conjunto de índices de iterações de *sucesso* definido por

$$\mathcal{S} = \{k \in \mathbb{N} \mid r_k \geq \eta\}.$$

No lema a seguir as constantes c_1 , L , β e c_2 são definidas em (1.5) e nas Hipóteses H1, H3 e H4, respectivamente. O lema garante que se o raio da região de confiança é suficientemente pequeno, então o algoritmo deve executar uma iteração de sucesso.

Lema 1.2 *Suponha que valham as Hipóteses H1, H3 e H4. Considere o conjunto*

$$\mathcal{K} = \left\{ k \in \mathbb{N} \mid \Delta_k \leq \min \left\{ \frac{\pi_k}{\beta}, \frac{\pi_k}{4c}, \alpha\pi_k, 1 \right\} \right\}, \quad (1.7)$$

onde $c = \frac{L + c_2 + \frac{\beta}{2}}{c_1}$. Se $k \in \mathcal{K}$, então $k \in \mathcal{S}$.

Demonstração. Considere $k \in \mathcal{K}$ arbitrário. Pelo Teorema do Valor Médio, existe $t_k \in (0, 1)$ tal que

$$f(x^k + d^k) = f(x^k) + \nabla f(x^k + t_k d^k)^T d^k. \quad (1.8)$$

Assim, pela definição de q_k em (1.3) e a igualdade (1.8),

$$\begin{aligned} |ared - pred| &= |f(x^k) - f(x^k + d^k) - q_k(0) + q_k(d^k)| \\ &= \left| -(\nabla f(x^k + t_k d^k) - g^k)^T d^k + \frac{1}{2}(d^k)^T G_k d^k \right|. \end{aligned}$$

Somando e subtraindo $\nabla f(x^k)$, utilizando as desigualdades triangular e de Cauchy-Schwarz, obtemos

$$\begin{aligned} |ared - pred| &= \left| -(\nabla f(x^k + t_k d^k) - g^k - \nabla f(x^k) + \nabla f(x^k))^T d^k + \frac{1}{2}(d^k)^T G_k d^k \right| \\ &\leq (\|\nabla f(x^k + t_k d^k) - \nabla f(x^k)\| + \|\nabla f(x^k) - g^k\|) \|d^k\| + \frac{1}{2}\|d^k\|^2 \|G_k\|. \end{aligned}$$

Pelas Hipóteses H1, H3 e H4, temos

$$|ared - pred| \leq (t_k L \|d^k\| + c_2 \Delta_k) \|d^k\| + \frac{1}{2} \beta \|d^k\|^2.$$

Visto que $\|d^k\| \leq \Delta_k$ e $t_k \in (0, 1)$,

$$|f(x^k) - f(x^k + d^k) - q_k(0) + q_k(d^k)| \leq c_0 \Delta_k^2, \quad (1.9)$$

onde $c_0 = L + c_2 + \frac{\beta}{2}$.

De (1.7), para todo $k \in \mathcal{K}$ temos que $\Delta_k \leq \alpha\pi_k$ e conseqüentemente $\pi_k > 0$. Então, segue de (1.5) que

$$q_k(0) - q_k(d^k) \neq 0.$$

Mas, das expressões (1.6), (1.9) e (1.5), para todo $k \in \mathcal{K}$,

$$\begin{aligned} |r_k - 1| &= \left| \frac{f(x^k) - f(x^k + d^k) - q_k(0) + q_k(d^k)}{q_k(0) - q_k(d^k)} \right| \\ &\leq \frac{c_0 \Delta_k^2}{c_1 \pi_k \min \left\{ \frac{\pi_k}{\beta}, \Delta_k, 1 \right\}} \\ &= \frac{c \Delta_k^2}{\pi_k \min \left\{ \frac{\pi_k}{\beta}, \Delta_k, 1 \right\}}. \end{aligned}$$

Por (1.7),

$$\Delta_k = \min \left\{ \frac{\pi_k}{\beta}, \Delta_k, 1 \right\} \quad \text{e} \quad \frac{c\Delta_k}{\pi_k} \leq \frac{1}{4}.$$

Logo, $|r_k - 1| \leq \frac{1}{4}$ e consequentemente $r_k \geq \frac{3}{4} > \eta$. Portanto a iteração k é de sucesso, completando a demonstração. □

O próximo lema fornece um resultado de convergência fraco para o problema de minimizar o modelo dentro do conjunto viável Ω . Provamos que a sequência de gradientes projetados do modelo em Ω tem uma subsequência convergindo para zero.

Lema 1.3 *Suponha que valham as Hipóteses H1, H2, H3 e H4. Então $\liminf_{k \rightarrow \infty} \pi_k = 0$.*

Demonstração. Suponha por contradição que existe $\varepsilon > 0$ tal que para todo $k \in \mathbb{N}$, $\pi_k \geq \varepsilon$. Tome

$$\tilde{\Delta} = \min \left\{ \frac{\varepsilon}{\beta}, \frac{\varepsilon}{4c}, \alpha\varepsilon, 1 \right\},$$

onde β é a constante da Hipótese H3, c definida no Lema 1.2 e $\alpha > 0$ o parâmetro dado no Algoritmo 1.1.

Se $\Delta_k \leq \tilde{\Delta}$, então $k \in \mathcal{K}$, com \mathcal{K} dado na expressão (1.7). Pelo Lema 1.2 a iteração k é de sucesso e então $\Delta_{k+1} = \tau_2 \Delta_k$. Distto, segue que o raio da região de confiança somente pode decrescer se $\Delta_k > \tilde{\Delta}$, e neste caso,

$$\Delta_{k+1} = \tau_1 \Delta_k > \tau_1 \tilde{\Delta}.$$

Logo, para todo $k \in \mathbb{N}$,

$$\Delta_k \geq \min \left\{ \tau_1 \tilde{\Delta}, \Delta_0 \right\}. \tag{1.10}$$

Se \mathcal{S} é finito, então Δ_k converge para zero, contradizendo (1.10). Por outro lado, se \mathcal{S} é infinito, temos de (1.5) que para todo $k \in \mathcal{S}$,

$$\begin{aligned} f(x^k) - f(x^{k+1}) &= f(x^k) - f(x^k + d^k) \\ &\geq \eta [q_k(0) - q_k(d^k)] \\ &\geq \eta c_1 \pi_k \min \left\{ \frac{\pi_k}{\beta}, \Delta_k, 1 \right\}. \end{aligned}$$

Então, utilizando (1.10) e a hipótese de contradição $\pi_k \geq \varepsilon$, concluímos que para todo $k \in \mathcal{K}$, existe

$$\tilde{\delta} = \min \left\{ \tau_1 \tilde{\Delta}, \Delta_0, \frac{\varepsilon}{\beta}, 1 \right\} > 0$$

tal que

$$f(x^k) - f(x^{k+1}) \geq \tilde{\delta}. \tag{1.11}$$

Por outro lado, pela Hipótese H2 a sequência $\{f(x^k)\}$ é limitada inferiormente. Como também é não crescente, temos que $f(x^k) - f(x^{k+1}) \rightarrow 0$, fornecendo uma contradição com (1.11) e concluindo a prova. □

Na Hipótese H4 dizemos que quanto menor Δ_k , melhor o gradiente do modelo deve representar o gradiente da função objetivo. Logo, é razoável para a análise de convergência que o raio da região de confiança convirja para zero. No lema a seguir mostramos que o algoritmo proposto tem esta propriedade.

Lema 1.4 *Suponha que valham as Hipóteses H2 e H3. Então a sequência $\{\Delta_k\}$ converge para zero.*

Demonstração. Se \mathcal{S} é finito, então existe $k_0 \in \mathbb{N}$ tal que para todo $k \geq k_0$, $\Delta_{k+1} \leq \tau_1 \Delta_k$. Logo, $\{\Delta_k\}$ converge para zero. Se \mathcal{S} é infinito, para algum $k \in \mathcal{S}$, utilizando (1.5) e a Hipótese H3 temos

$$f(x^k) - f(x^{k+1}) \geq \eta (Q_k(0) - Q_k(d^k)) \geq \eta c_1 \pi_k \min \left\{ \frac{\pi_k}{\beta}, \Delta_k, 1 \right\}.$$

Como $k \in \mathcal{S}$, temos que $\Delta_k \leq \alpha \pi_k$ e assim

$$f(x^k) - f(x^{k+1}) \geq \eta c_1 \frac{\Delta_k}{\alpha} \min \left\{ \frac{\Delta_k}{\alpha \beta}, \Delta_k, 1 \right\}.$$

Visto que $\{f(x^k)\}$ é não crescente e, pela Hipótese H2, limitada inferiormente, o lado esquerdo da expressão imediatamente anterior converge para zero. Então,

$$\lim_{k \in \mathcal{S}} \Delta_k = 0. \tag{1.12}$$

Considere o conjunto

$$\mathcal{U} = \{k \in \mathbb{N} \mid k \notin \mathcal{S}\}.$$

Se \mathcal{U} é finito, então por (1.12) temos que $\lim_{k \rightarrow \infty} \Delta_k = 0$. Agora suponha que \mathcal{U} é infinito. Considere $k \in \mathcal{U}$ e defina ℓ_k o índice da última iteração de sucesso antes de k . Do mecanismo do Algoritmo 1.1, $\Delta_k \leq \tau_2 \Delta_{\ell_k}$, o que implica

$$\lim_{k \in \mathcal{U}} \Delta_k \leq \tau_2 \lim_{k \in \mathcal{U}} \Delta_{\ell_k} = \tau_2 \lim_{\ell_k \in \mathcal{S}} \Delta_{\ell_k}.$$

Por (1.12) segue que $\lim_{k \in \mathcal{U}} \Delta_k = 0$ o que completa a prova.

□

Utilizando os lemas anteriores podemos provar que não somente existe uma subsequência de $\{\pi_k\}$ convergindo para zero como dito no Lema 1.3, mas que a convergência é em toda sequência.

No Algoritmo 1.1 o modelo pode ser modificado mesmo em uma iteração k onde o iterando permanece o mesmo. Isto faz com que a medida π_k também seja modificada nesta iteração. Este fato é observado no lema a seguir, onde provamos convergência a zero para a sequência $\{\pi_k\}$.

Lema 1.5 *Suponha que valham as Hipóteses H1, H2, H3 e H4. Então*

$$\lim_{k \rightarrow \infty} \pi_k = 0.$$

Demonstração. Suponha por contradição que para algum $\varepsilon > 0$ o conjunto

$$\mathbb{N}' = \{k \in \mathbb{N} \mid \pi_k \geq \varepsilon\} \quad (1.13)$$

é infinito.

Pelo Lema 1.4, a sequência $\{\Delta_k\}$ converge para zero. Então, existe $k_0 \in \mathbb{N}$ tal que para todo $k \geq k_0$ com $k \in \mathbb{N}'$,

$$\Delta_k \leq \min \left\{ \frac{\varepsilon}{\beta}, \frac{\varepsilon}{4c}, \alpha\varepsilon, 1 \right\}, \quad (1.14)$$

onde β é dado na Hipótese H3, c é a constante do Lema 1.2 e $\alpha > 0$ é dado no algoritmo.

Por (1.13),

$$\Delta_k \leq \min \left\{ \frac{\pi_k}{\beta}, \frac{\pi_k}{4c}, \alpha\pi_k, 1 \right\} \quad (1.15)$$

e pelo Lema 1.2, $k \in \mathcal{S}$.

Dado $k \in \mathbb{N}'$ com $k \geq k_0$, considere ℓ_k o primeiro índice tal que $\ell_k > k$ e $\pi_{\ell_k} \leq \varepsilon/2$. A existência de ℓ_k é assegurada pelo Lema 1.3. Assim,

$$\pi_k - \pi_{\ell_k} \geq \frac{\varepsilon}{2}.$$

Utilizando a definição de π_k , a desigualdade triangular e a propriedade de contração das projeções, temos

$$\begin{aligned} \frac{\varepsilon}{2} &\leq \|P_\Omega(x^k - g^k) - x^k\| - \|P_\Omega(x^{\ell_k} - g^{\ell_k}) - x^{\ell_k}\| \\ &\leq \|P_\Omega(x^k - g^k) - x^k - P_\Omega(x^{\ell_k} - g^{\ell_k}) + x^{\ell_k}\| \\ &\leq 2\|x^k - x^{\ell_k}\| + \|g^k - g^{\ell_k}\|. \end{aligned}$$

Somando e subtraindo $\nabla f(x^k)$ e $\nabla f(x^{\ell_k})$ e novamente pela desigualdade triangular,

$$\begin{aligned} \frac{\varepsilon}{2} &\leq 2\|x^k - x^{\ell_k}\| + \|g^k - \nabla f(x^k) + \nabla f(x^k) - \nabla f(x^{\ell_k}) + \nabla f(x^{\ell_k}) - g^{\ell_k}\| \\ &\leq 2\|x^k - x^{\ell_k}\| + \|g^k - \nabla f(x^k)\| + \|\nabla f(x^k) - \nabla f(x^{\ell_k})\| + \|\nabla f(x^{\ell_k}) - g^{\ell_k}\|. \end{aligned}$$

Utilizando as Hipóteses H1 e H4,

$$\frac{\varepsilon}{2} \leq 2\|x^k - x^{\ell_k}\| + c_2\Delta_k + L\|x^k - x^{\ell_k}\| + c_2\Delta_{\ell_k},$$

isto é,

$$\frac{\varepsilon}{2} \leq (2 + L)\|x^k - x^{\ell_k}\| + c_2(\Delta_k + \Delta_{\ell_k}). \quad (1.16)$$

Considere $C_k = \{i \in \mathcal{S} \mid k \leq i < \ell_k\}$. Note que por (1.15) $k \in \mathcal{S}$, logo $C_k \neq \emptyset$. Para cada $i \in C_k$, utilizando o fato de que $i \in \mathcal{S}$, a condição (1.5) e a Hipótese H3, concluímos que

$$\begin{aligned} f(x^i) - f(x^{i+1}) &\geq \eta(Q_i(0) - Q_i(d^i)) \\ &\geq \eta c_1 \pi_i \min \left\{ \frac{\pi_i}{\beta}, \Delta_i, 1 \right\}. \end{aligned}$$

Pela definição de ℓ_k , temos que $\pi_i > \varepsilon/2$ para todo $i \in C_k$. Como $i \geq k$, por (1.14) $\Delta_i \leq \varepsilon/\beta$ e $\Delta_i \leq 1$. Desta forma,

$$\frac{\Delta_i}{2} \leq \frac{\varepsilon}{2\beta} \leq \frac{\pi_i}{\beta}.$$

Segue que

$$f(x^i) - f(x^{i+1}) > \frac{\eta c_1 \varepsilon \Delta_i}{4},$$

e assim

$$\Delta_i < \frac{4}{\eta c_1 \varepsilon} (f(x^i) - f(x^{i+1})). \quad (1.17)$$

Por outro lado,

$$\|x^k - x^{\ell_k}\| \leq \sum_{i \in C_k} \|x^i - x^{i+1}\| \leq \sum_{i \in C_k} \Delta_i,$$

que combinado com (1.17) fornece

$$\|x^k - x^{\ell_k}\| < \frac{4}{\eta c_1 \varepsilon} (f(x^k) - f(x^{\ell_k})).$$

Pela Hipótese H2 a sequência $\{f(x^k)\}$ é limitada inferiormente, e como é monótona não crescente,

$$f(x^k) - f(x^{\ell_k}) \rightarrow 0.$$

Logo, $\{\|x^k - x^{\ell_k}\|\}$ converge para zero, e junto com o Lema 1.4 contradiz (1.16), completando a prova. □

Temos agora todos os ingredientes para provar convergência global a pontos estacionários de primeira ordem. No teorema a seguir, estabelecemos a relação entre a medida de estacionariedade do problema original e a medida de estacionariedade dada no Lema 1.5, obtendo um resultado forte de convergência.

Teorema 1.6 *Suponha que valham as Hipóteses H1, H2, H3 e H4, então*

$$\lim_{k \rightarrow \infty} \|P_\Omega(x^k - \nabla f(x^k)) - x^k\| = 0.$$

Demonstração. Pela desigualdade triangular, a propriedade de contração das projeções e a Hipótese H4, temos que

$$\begin{aligned} \|P_\Omega(x^k - \nabla f(x^k)) - x^k\| &= \|P_\Omega(x^k - \nabla f(x^k)) - P_\Omega(x^k - g^k) + P_\Omega(x^k - g^k) - x^k\| \\ &\leq \|P_\Omega(x^k - \nabla f(x^k)) - P_\Omega(x^k - g^k)\| + \|P_\Omega(x^k - g^k) - x^k\| \\ &\leq \|\nabla f(x^k) - g^k\| + \|P_\Omega(x^k - g^k) - x^k\| \\ &\leq c_2 \Delta_k + \pi_k. \end{aligned}$$

Utilizando os Lemas 1.4 e 1.5, completamos a prova. □

Do Teorema 1.6 concluímos que se o Algoritmo 1.1 gera uma sequência $\{x^k\}$ com algum ponto de acumulação x^* , então o ponto x^* é estacionário de primeira ordem [18, 59]. Uma maneira de garantir a existência de um ponto de acumulação é supondo que o conjunto de nível $\{x \in \mathbb{R}^n \mid f(x) \leq f(x^0)\}$ é limitado. Note que assim toda a sequência $\{x^k\}$ é limitada e logo possui uma subsequência convergente.

As Hipóteses H3 e H4 podem ser obtidas quando utilizamos na construção dos modelos a interpolação polinomial de Lagrange. No próximo capítulo apresentamos como estas hipóteses podem ser supridas a partir da teoria existente em [22] e um resultado disponível em [18].

Capítulo 2

Interpolação polinomial

Discutimos na Seção 1.2 que, para garantir resultados de convergência para o Algoritmo 1.1, é necessário que haja um controle da relação entre os gradientes dos modelos e da função objetivo, como sugere a Hipótese H4, e que as Hessianas dos modelos sejam uniformemente limitadas superiormente, como sugere a Hipótese H3. Neste capítulo, apresentamos na primeira seção a existência de algoritmos capazes de construir modelos satisfazendo estas hipóteses. O estudo foi feito com base em [18, 22]. Na segunda seção discutimos sobre a possibilidade de construir modelos quadráticos com liberdade em seus parâmetros. São os denominados modelos subdeterminados ou subquadráticos.

2.1 Interpolação quadrática

Objetivamos nesta seção mostrar que a literatura trata as Hipóteses H3 e H4 como possíveis de serem atingidas quando utilizamos interpolação quadrática.

Em métodos de região de confiança sem derivadas, a técnica usual disponível na literatura para construção de modelos é a baseada em interpolação polinomial [22, 20, 35, 47, 57, 61]. Algoritmos utilizam bases distintas na construção de modelos, por exemplo, base de Newton no Algoritmo DFO de Conn, Scheinberg e Toint [20] e base de Lagrange no Algoritmo BOBYQA [57]. As boas propriedades da interpolação polinomial a consagraram como uma forte parceira na resolução de problemas de otimização sem derivadas.

Desejamos construir um modelo quadrático para uma função $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Para isto, considere um conjunto interpolador $Y = \{y^1, y^2, \dots, y^m\}$, onde o inteiro positivo m define o número de pontos de interpolação do conjunto.

Definição 2.1 Dizemos que um polinômio quadrático (ou modelo quadrático) $Q : \mathbb{R}^n \rightarrow \mathbb{R}$ interpola a função f no conjunto Y quando para todo $y^i \in Y$

$$Q(y^i) = f(y^i). \quad (2.1)$$

A igualdade (2.1) é denominada condição de interpolação.

Um modelo quadrático tem $p = (n + 1)(n + 2)/2$ parâmetros independentes. São n parâmetros para a parcela linear, 1 parâmetro para o termo constante e pela simetria da Hessiana são $(n + 1)n$ parâmetros para a parcela quadrática. Logo, são necessários p pontos de interpolação para determinar inteiramente um modelo quadrático. Nesta seção tomamos $m = p$.

Seja \mathbb{P}_n^2 o espaço dos polinômios de grau menor ou igual a 2 em \mathbb{R}^n . Considere $\phi_i \in \mathbb{P}_n^2$, $i = 1, \dots, p$, de modo que

$$\phi = \{\phi_1(x), \dots, \phi_p(x)\}$$

seja uma base para o espaço \mathbb{P}_n^2 . Assim, todo polinômio quadrático Q pode ser escrito unicamente como

$$Q(x) = \sum_{i=1}^p \alpha_i \phi_i(x),$$

onde para todo i , α_i é um coeficiente real. O modelo quadrático interpolador dado na Definição 2.1 fica determinado quando se conhece os coeficientes α_i no sistema linear

$$M(\phi, Y)\alpha = \begin{bmatrix} \phi_1(y^1) & \phi_2(y^1) & \cdots & \phi_p(y^1) \\ \phi_1(y^2) & \phi_2(y^2) & \cdots & \phi_p(y^2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(y^m) & \phi_2(y^m) & \cdots & \phi_p(y^m) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_p \end{bmatrix} = \begin{bmatrix} f(y^1) \\ f(y^2) \\ \vdots \\ f(y^m) \end{bmatrix}. \quad (2.2)$$

O conjunto interpolador Y é dito *posicionado* para interpolação polinomial em \mathbb{R}^n quando a matriz $M(\phi, Y)$ é não singular, ou seja, o sistema (2.2) tem solução única. O posicionamento não depende da base, pois bases em espaço de dimensão finita são equivalentes. Utilizamos em definições posteriores e no Algoritmo 4.8 a base de Lagrange definida a seguir. A base de Lagrange é utilizada por Conn, Scheinberg e Vicente [22, Seção 6.2] em algoritmos que permitem construir conjuntos posicionados para interpolação polinomial.

Definição 2.2 Dado o conjunto interpolador $Y = \{y^1, y^2, \dots, y^m\}$, denominamos de base de Lagrange para \mathbb{P}_n^2 o conjunto de polinômios $\lambda = \{\lambda_1(x), \dots, \lambda_m(x)\}$ tal que

$$\lambda_j(y^i) = \begin{cases} 1, & \text{se } j = i, \\ 0, & \text{se } j \neq i. \end{cases}$$

Lema 2.3 Se Y é posicionado para interpolação, então a base de polinômios de Lagrange existe e é única.

Demonstração. [22, Lema 3.4].

□

Podemos expressar o único polinômio que interpola f no conjunto Y como sendo

$$Q(x) = \sum_{i=1}^m f(y^i) \lambda_i(x).$$

Uma propriedade dos polinômios de Lagrange é a seguinte. Dado o conjunto posicionado $Y = \{y^1, y^2, \dots, y^m\} \subset \mathbb{R}^n$, podemos expressar $\phi(x) = [\phi_1(x), \dots, \phi_p(x)]^T$ unicamente em termos dos vetores $\phi(y^i)$, $i = 1, \dots, m$, como

$$\sum_{i=1}^m \lambda_i(x) \phi(y^i) = \phi(x),$$

ou em notação matricial,

$$M(\phi, Y)^T \lambda(x) = \phi(x), \quad (2.3)$$

onde $\lambda(x) = [\alpha_1(x), \dots, \alpha_m(x)]^T$. Considere agora o conjunto

$$Y_i(x) = Y \setminus \{y^i\} \cup \{x\}, \quad i = 1, \dots, m.$$

Aplicando a regra de Cramer em (2.3) temos explicitamente cada elemento da base de Lagrange,

$$\lambda_i(x) = \frac{\det(M(\phi, Y_i(x)))}{\det(M(\phi, Y))}. \quad (2.4)$$

Considere $\phi(Y) = \{\phi(y^i), i = 1, \dots, m\}$. Considere $\text{vol}(\phi(Y))$ o volume da envoltória convexa do conjunto $\phi(Y)$ dado por

$$\text{vol}(\phi(Y)) = \frac{\det(M(\phi, Y))}{(m-1)!}.$$

Portanto, por (2.4) para todo $i = 1, \dots, m$

$$|\lambda_i(x)| = \frac{\text{vol}(\phi(Y_i(x)))}{\text{vol}(\phi(Y))}, \quad (2.5)$$

ou seja, o valor absoluto do i -ésimo polinômio de Lagrange em um ponto $x \in \mathbb{R}^n$ é a mudança no volume da envoltória convexa do conjunto $\phi(Y)$, quando y^i é substituído por x .

Os polinômios de Lagrange podem ser utilizados para limitar o erro entre as derivadas do modelo quadrático e da função objetivo. Para garantir a Hipótese H4 no Capítulo 1, é preciso garantir que nos atuais iterandos o erro entre os gradientes da função f e do modelo Q converge para zero quando a distância entre os pontos do conjunto interpolador vai para zero. O conceito a seguir de Λ -posicionamento [22] auxilia em uma fórmula para expressar um limitante superior para o erro entre os gradientes do modelo e da função objetivo.

Definição 2.4 *Dados $\Lambda > 0$ e $C \subset \mathbb{R}^n$ um conjunto qualquer, dizemos que o conjunto posicionado $Y = \{y^1, y^2, \dots, y^m\}$ é Λ -posicionado em C se para a base polinomial de Lagrange λ associada a Y , vale*

$$\Lambda \geq \max_{1 \leq j \leq m} \max_{x \in C} |\lambda_j(x)|.$$

Esta definição de Λ -posicionamento é em relação a uma base de polinômios de Lagrange. O lema a seguir encontrado em [22, 61], mostra como é possível obter o Λ -posicionamento em processos finitos.

Lema 2.5 *Dados uma bola B fechada, um conjunto interpolador*

$$Y = \{y^1, y^2, \dots, y^m\} \subset B$$

e uma constante $\Lambda > 1$, considere o seguinte procedimento: encontre $j \in \{1, \dots, m\}$ e um ponto $x \in B$ tal que $|\lambda_j(x)| \geq \Lambda$ (se este ponto existe) e substitua y^j por x para obter um novo conjunto Y e sua correspondente base de Lagrange λ . Então este procedimento termina em um número finito de iterações com um modelo que é Λ -posicionado em B .

Demonstração. Fixe uma base ϕ para o conjunto \mathbb{P}_n^2 , e considere o volume $vol(\phi(Y))$. Pela expressão (2.5),

$$|\lambda_j(x)| = \frac{vol(\phi(Y_j(x)))}{vol(\phi(Y))} \quad \Rightarrow \quad vol(\phi(Y_j(x))) \geq \Lambda vol(\phi(Y)),$$

logo, cada vez que um ponto é substituído, o volume é incrementado em pelo menos $\Lambda > 1$. Portanto o procedimento é finito pois o volume $vol(\phi(Y_j(x)))$ é uniformemente limitado superiormente, uma vez que o conjunto inicial $Y \subset B$ e permanece em B depois de cada mudança.

□

Assumindo a possibilidade em obter o máximo global de um polinômio quadrático sobre a bola fechada B em um número finito de iterações (por exemplo, como em [42]), o procedimento estabelecido no lema anterior de fato garante o Λ -posicionamento em B com um número finito de iterações.

Em [22] é apresentado um algoritmo capaz de gerar bases de Lagrange satisfazendo a Definição 2.4. Quando o conjunto interpolador é Λ -posicionado, é possível determinar modelos interpoladores que satisfazem hipóteses razoáveis para resultados teóricos de convergência, em métodos de região de confiança sem derivadas irrestritos [22, 61].

Com a definição de Λ -posicionamento podemos apresentar um resultado que garante a Hipótese H4. No lema a seguir é dito que tomando um conjunto interpolador Λ -posicionado, cuja existência é garantida pelo Lema 2.5, o erro entre os gradientes (ou Hessianas) do modelo interpolador e da função objetivo é limitado por um múltiplo positivo do raio do conjunto interpolador.

Lema 2.6 *Considere $m = (n + 1)(n + 2)/2$ pontos posicionados para interpolação e o conjunto interpolador $Y = \{y^1, y^2, \dots, y^m\} \subset B(y^1, \Delta)$ e $\{\lambda_i(x)\}_{i=1}^m$ a base de Lagrange associada a Y . Suponha f duas vezes continuamente diferenciável em um aberto contendo $B(y^1, \Delta)$ e que a Hessiana de f é Lipschitz contínua. Então existem constantes $\kappa_1, \kappa_2 > 0$ tais que, para todo polinômio quadrático interpolador Q e para todo $x \in B(y^1, \Delta)$*

1. $\|\nabla f(x) - \nabla Q(x)\| \leq \kappa_1 \Lambda \Delta$, onde $\Lambda \geq \max_{1 \leq j \leq m} \max_{x \in B(y^1, \Delta)} |\lambda_j(x)|$,
2. $\|\nabla^2 f(x) - \nabla^2 Q(x)\| \leq \kappa_2 \Lambda \Delta$.

Demonstração. Em [22, Teorema 3.16].

□

Considere a hipótese adicional de que a Hessiana da f é uniformemente limitada. Logo, o segundo item do Lema 2.6 garante que a Hipótese H3 seja satisfeita, isto é, a Hessiana do modelo Q é limitada. Observe que a hipótese que limita uniformemente a Hessiana da f é também uma maneira de garantir que ∇f seja Lipschitz e assim a Hipótese H1 também é satisfeita.

No Lema 2.6 $m = (n + 1)(n + 2)/2$, logo não existe liberdade nos parâmetros do modelo. Em recente artigo [5], os autores trabalham com hipótese de manter o Lema 2.6 utilizando menos do que $(n + 1)(n + 2)/2$ pontos de interpolação.

O resultado do lema anterior vale para todo x na bola $B(y^1, \Delta)$, com $\Delta > 0$. No entanto, a Hipótese H4 exige apenas que o resultado (1) do lema seja válido no ponto y^1 . Discutimos a seguir outra maneira de garantir que a Hipótese H4 seja satisfeita.

Um importante resultado é demonstrado por Ciarlet e Raviart em [12] e tem como caso particular que para todo x na envoltória convexa do conjunto $Y = \{y^1, y^2, \dots, y^m\}$,

$$|f(x) - Q(x)| \leq \frac{1}{3!} M \sum_{j=1}^m |\lambda_j(x)| \|x - y^j\|^3, \quad (2.6)$$

onde $M > 0$ é um limitante para $\|\nabla f(x)\|$. Pela expressão (2.6), para todo x em uma bola de diâmetro Δ e contendo a envoltória convexa do conjunto Λ -posicionado Y ,

$$\begin{aligned} |f(x) - Q(x)| &\leq \frac{1}{6} M \sum_{j=1}^m |\lambda_j(x)| \|x - y^j\|^3 \\ &\leq \frac{1}{6} M \sum_{j=1}^m \max_{1 \leq j \leq m} \max_{x \in B} |\lambda_j(x)| \|x - y^j\|^3 \\ &\leq \frac{1}{6} M m \Lambda \Delta^3. \end{aligned} \quad (2.7)$$

Logo, a diferença entre a função objetivo e o modelo quadrático interpolador é limitada superiormente por constantes que dependem do Λ -posicionamento e do diâmetro do conjunto interpolador. O próximo lema estabelece que a Hipótese H4 também pode ser obtida a partir de (2.7).

Lema 2.7 *Dado $z \in \mathbb{R}^n$, suponha f duas vezes continuamente diferenciável, as Hessianas da f e do modelo Q uniformemente limitadas e que existe $\kappa_3 \in \mathbb{R}$ independente de x tal que para todo $x \in B(z, \Delta)$ vale $|f(x) - Q(x)| \leq \kappa_3 \Delta^2$. Então existe $\kappa_4 \in \mathbb{R}$ tal que*

$$\|\nabla f(z) - \nabla Q(z)\| \leq \kappa_4 \Delta.$$

Demonstração. Em [18, Teorema 9.1.1].

□

Considere uma base de Lagrange e tome a constante κ_3 do Lema 2.7 como sendo

$$\kappa_3 = \frac{1}{6} M m \Lambda.$$

Se $\Delta \leq 1$, por (2.7) a hipótese do Lema 2.7 é satisfeita. Logo, com a hipótese adicional de que $\Delta \leq 1$, a Hipótese H4 do estudo da convergência no Capítulo 1 é satisfeita.

2.2 Modelos quadráticos subdeterminados

Uma característica importante em métodos de região de confiança sem derivadas é a necessidade em resolver problemas com um número pequeno de avaliações de função objetivo, pois estas avaliações são consideradas caras. Quando o número de pontos de interpolação é $n + 1$, podemos construir modelos lineares que são econômicos em avaliação de função objetivo. No entanto, algoritmos que trabalham com modelos

lineares podem não fazer tanto progresso quanto um algoritmo que utiliza modelos quadráticos obtidos com $(n+1)(n+2)/2$ pontos de interpolação. Mas esta falta de liberdade nos parâmetros dos modelos faz com que o número de avaliações de função seja alto para obtenção de modelos. Para ter um equilíbrio entre a economia de um modelo linear e a eficiência de um modelo quadrático, os modelos quadráticos subdeterminados são possíveis e populares em métodos de região de confiança sem derivadas.

Quando o número m de pontos do conjunto interpolador Y é

$$n+1 < m < \frac{1}{2}(n+1)(n+2),$$

os modelos são denominados de modelos quadráticos subdeterminados ou modelos quadráticos incompletos. Como $m < (n+1)(n+2)/2$, e a base ϕ tem $(n+1)(n+2)/2$ elementos, a solução do sistema (2.2) não é necessariamente única. No que segue apresentamos duas abordagens que visam buscar unicidade em modelos quadráticos obtidos por interpolação incompleta. Ambas as abordagens precisam de hipóteses adicionais para manterem unicidade. A primeira é a utilizada em nosso algoritmo prático e os detalhes estão no Capítulo 3. A técnica trabalha com uma quantidade fixa de pontos de interpolação e, para determinar os parâmetros do modelo quadrático, é calculada a solução mínima na norma de Frobenius da mudança nas Hessianas, como no problema

$$\begin{aligned} & \text{minimizar}_{c,g,\nabla^2 Q^+} && \frac{1}{4} \|\nabla^2 Q^+ - \nabla^2 Q\|^2 \\ & \text{sujeita a} && Q^+(Y) = f(Y). \end{aligned} \quad (2.8)$$

onde Q é o modelo atual e Q^+ o modelo a ser obtido com os parâmetros $c, g, \nabla^2 Q^+$.

Esta técnica foi desenvolvida por Powell em [49] e utilizada nos algoritmos NEWUOA e BOBYQA. Ainda, em BOBYQA, é mantida como padrão a técnica anterior, combinada com uma segunda técnica que é a utilizada por Conn, Scheinberg e Toint em [20]. O modelo linear ou subdeterminado é obtido minimizando a norma de Frobenius da matriz Hessiana do novo modelo como discutido em [20], resolvendo o problema

$$\begin{aligned} & \text{minimizar}_{c,g,\nabla^2 Q^+} && \frac{1}{4} \|\nabla^2 Q^+\|^2 \\ & \text{sujeita a} && Q^+(Y) = f(Y). \end{aligned} \quad (2.9)$$

Se a cardinalidade de Y é $(n+1)(n+2)/2$ e a matriz $M(\phi, Y)$ em (2.2) é não singular, temos o caso da interpolação quadrática completa. Os problemas (2.8) e (2.9) são de programação quadrática convexa, e portanto têm soluções. No entanto, em ambos os problemas, para garantir unicidade nas atualizações dos modelos interpoladores são necessárias hipóteses adicionais. Para o problema (2.8) abordamos estas hipóteses no Capítulo 3 e que também são úteis em nossa proposta do algoritmo implementado apresentado no Capítulo 4.

O Algoritmo BOBYQA [54] proposto por Powell foi projetado para resolver problemas sem derivadas com restrições em caixa. Objetiva resolver problemas com um número pequeno de avaliações de função objetivo. Teve forte influência em nossa proposta e apresentamos no próximo capítulo o estudo que fizemos deste eficiente algoritmo.

Capítulo 3

Algoritmo BOBYQA

O Algoritmo BOBYQA (Bound Optimization By Quadratic Approximation) descrito em [54] remonta os trabalhos [48, 49, 50, 51, 52, 53], onde constam os argumentos para a construção do algoritmo. BOBYQA é a base para a construção dos modelos quadráticos que utilizamos no algoritmo que implementamos e discutimos no próximo capítulo. Neste capítulo descrevemos os procedimentos deste eficiente, e nada simples, algoritmo.

BOBYQA é um algoritmo de região de confiança sem derivadas para resolver problemas de otimização em caixa

$$\begin{array}{ll} \text{minimizar} & f(x) \\ \text{sujeita a} & x \in \mathbb{A}, \end{array}$$

onde $f : \mathbb{R}^n \rightarrow \mathbb{R}$ é uma função diferenciável e $\mathbb{A} = \{x \in \mathbb{R}^n | a \leq x \leq b\}$. O algoritmo não utiliza informações de derivadas, somente avaliações da função f e retorna um ponto viável na caixa \mathbb{A} . Foi desenvolvido a partir de algumas modificações em NEWUOA [52] e está implementado em Fortran 77.

Considere a k -ésima iteração do algoritmo e o conjunto interpolador $Y_k = \{y^1, \dots, y^m\}$, onde

$$n + 2 \leq m \leq \frac{1}{2}(n + 1)(n + 2).$$

Cada iteração trabalha com uma aproximação quadrática Q_k para f satisfazendo a condição de interpolação $Q_k(y^i) = f(y^i)$, com $y^i \in Y_k$.

O número de pontos de interpolação, m , garante que o modelo seja quadrático, pois pelo menos uma entrada da matriz Hessiana é fornecida quando $m = n + 2$. Powell sugere utilizar $m = 2n + 1$, pois os testes mostram que com esse valor, o algoritmo resolve bem uma boa classe de problemas, e há liberdade na escolha da Hessiana do modelo Q_k . O forte apelo do Algoritmo BOBYQA é resolver problemas com o menor número de avaliações de função objetivo quanto possível e executar, em geral, cada iteração dentro de $O(m^2)$ operações.

A atualização do modelo quadrático segue a técnica de minimizar a norma de Frobenius da mudança na segunda derivada do modelo, sujeita à hipótese de interpolação. Em trabalho recente [55], Powell sugere uma seminorma de Frobenius no processo de atualização do modelo quadrático, mas não disponibiliza resultados numéricos a respeito e [69] faz um tratado teórico sobre este processo.

O algoritmo consiste em uma sequência de minimizações aproximadas de sub-problemas quadráticos restritos à caixa \mathbb{A} e à região de confiança. O modelo na k -ésima iteração tem a forma

$$Q_k(x) = c_k + (g^k)^T(x - x^0) + \frac{1}{2}(x - x^0)^T \nabla^2 Q_k(x - x^0), \quad (3.1)$$

onde c_k é uma constante, $g^k \in \mathbb{R}^n$ é o gradiente do modelo em $x = x^0$, a matriz simétrica $\nabla^2 Q_k \in \mathbb{R}^{n \times n}$ é a Hessiana e x^0 é um vetor auxiliar que sofre mudanças esporádicas durante as atualizações em BOBYQA conforme veremos adiante.

3.1 Construções

Nesta seção descrevemos a construção do primeiro modelo Q_1 e do primeiro conjunto interpolador Y_1 .

Começamos adequando os pontos à caixa \mathbb{A} , para permitir caminharmos nas direções coordenadas sem sair de \mathbb{A} . Considere $y^1 \in \mathbb{A}$, $\Delta > 0$, $a = (a_1, a_2, \dots, a_n)^T$, $b = (b_1, b_2, \dots, b_n)^T$ e suponha o diâmetro de \mathbb{A} maior ou igual a 2Δ . Para $i = 1, 2, \dots, n$,

se $a_i < [y^1]_i < a_i + \Delta$, então redefina $[y^1]_i = a_i + \Delta$;
 se $b_i - \Delta < [y^1]_i < b_i$, então redefina $[y^1]_i = b_i - \Delta$.

Escolha dos primeiros $2n + 1$ pontos de Y_1

Para $i = 1, \dots, n$, definimos

$$\begin{aligned} y^{i+1} &= y^1 + \Delta e^i \quad \text{e} \quad y^{n+i+1} = y^1 - \Delta e^i, & \text{se } a_i < [y^1]_i < b_i, \\ y^{i+1} &= y^1 + \Delta e^i \quad \text{e} \quad y^{n+i+1} = y^1 + 2\Delta e^i, & \text{se } [y^1]_i = a_i, \\ y^{i+1} &= y^1 - \Delta e^i \quad \text{e} \quad y^{n+i+1} = y^1 - 2\Delta e^i, & \text{se } [y^1]_i = b_i. \end{aligned} \quad (3.2)$$

A Figura 3.1 exemplifica no plano os cinco pontos obtidos por (3.2) quando y^1 está na borda da caixa \mathbb{A} , figura da esquerda, e quando y^1 está no interior da caixa, figura da direita.

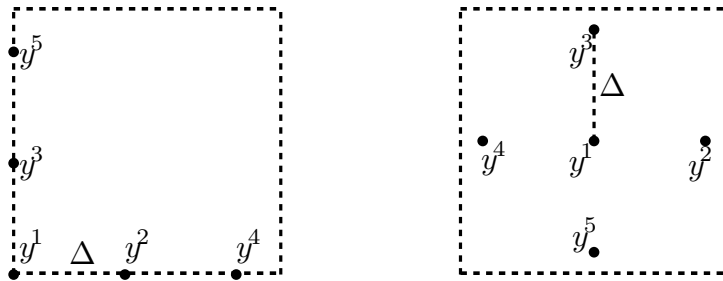


Figura 3.1: $2n + 1$ pontos de interpolação para $n = 2$

O Algoritmo BOBYQA permite que o ponto inicial $x^1 = y^1$ seja tomado fora da caixa \mathbb{A} . Neste caso y^1 é adequado à caixa por um procedimento que o projeta na fronteira de \mathbb{A} .

Escolha dos demais $m - (2n + 1)$ pontos de Y_1

Primeiramente reordenamos os pontos de interpolação y^i obtidos pela expressão (3.2) como segue: considere $i = 1, 2, \dots, n$. Se

$$f(y^{i+1}) < f(y^{n+i+1}), \quad (3.3)$$

então y^{i+1} é trocado por y^{n+i+1} . Este reordenamento analisa a tendência de decréscimo da função objetivo f e estas direções são consideradas nas escolhas dos $m - 2n - 1$ pontos interpoladores restantes como discutimos no próximo parágrafo.

Considere $2n + 2 \leq j \leq m$ e

$$u(j) = \begin{cases} j - 2n - 1, & 2n + 2 \leq j \leq 3n + 1, \\ u(j - n), & 3n + 2 \leq j \leq m. \end{cases} \quad (3.4)$$

A sequência finita $\{u(j)\}$ tem $\left\lfloor \frac{m - 2n - 2}{n} \right\rfloor + 1$ ciclos. O último ciclo têm comprimento menor ou igual a n e os demais têm comprimento n . Tome

$$1 \leq c \leq \left\lfloor \frac{m - 2n - 2}{n} \right\rfloor + 1$$

um número que indica o c -ésimo ciclo em $\{u(j)\}$ e para $2n + 2 \leq j \leq m$, considere

$$v(j) = \begin{cases} u(j) + c & \text{se } (u(j) + c) \in \{1, 2, \dots, n\} \\ u(j) + c - n & \text{se } (u(j) + c) \notin \{1, 2, \dots, n\}. \end{cases} \quad (3.5)$$

Finalmente, os $m - 2n - 1$ pontos de interpolação para $j \in \{2n + 2, \dots, m\}$ são dados por

$$y^j = y^{u(j)+1} + y^{v(j)+1} - y^1. \quad (3.6)$$

Segue um exemplo de como são construídas as sequências $\{u(j)\}$ e $\{v(j)\}$.

Exemplo 3.1 Considere $n = 6$ e $m = \frac{1}{2}(n + 1)(n + 2) = 28$, então temos $c = 3$ ciclos e para $j = 14, \dots, 28$, os $C_6^2 = 15$ pares distintos são

$$\left\{ \begin{pmatrix} u(j) \\ v(j) \end{pmatrix} \right\} = \underbrace{\left\{ \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 3 \\ 4 \end{pmatrix}, \begin{pmatrix} 4 \\ 5 \end{pmatrix}, \begin{pmatrix} 5 \\ 6 \end{pmatrix}, \begin{pmatrix} 6 \\ 1 \end{pmatrix} \right\}}_{c=1}, \\ \underbrace{\left\{ \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \begin{pmatrix} 2 \\ 4 \end{pmatrix}, \begin{pmatrix} 3 \\ 5 \end{pmatrix}, \begin{pmatrix} 4 \\ 6 \end{pmatrix}, \begin{pmatrix} 5 \\ 1 \end{pmatrix}, \begin{pmatrix} 6 \\ 2 \end{pmatrix} \right\}}_{c=2}, \underbrace{\left\{ \begin{pmatrix} 1 \\ 4 \end{pmatrix}, \begin{pmatrix} 2 \\ 5 \end{pmatrix}, \begin{pmatrix} 3 \\ 6 \end{pmatrix} \right\}}_{c=3}.$$

A Figura 3.2 exemplifica no plano os seis pontos determinados por (3.2) e por (3.6). Neste caso houve o reordenamento de y^3 e y^5 para a obtenção de y^6 .

O algoritmo inicia com um conjunto interpolador Y_1 determinado por (3.2) e (3.6), e com um modelo quadrático obtido por uma fórmula fechada em virtude da disposição geométrica deste conjunto interpolador. Em geral, todas as demais iterações do Algoritmo BOBYQA altera o conjunto interpolador em apenas um ponto de uma iteração para outra.

No teorema a seguir demonstramos que o conjunto interpolador Y_1 permite construir um modelo quadrático que os interpola, através de uma fórmula fechada.

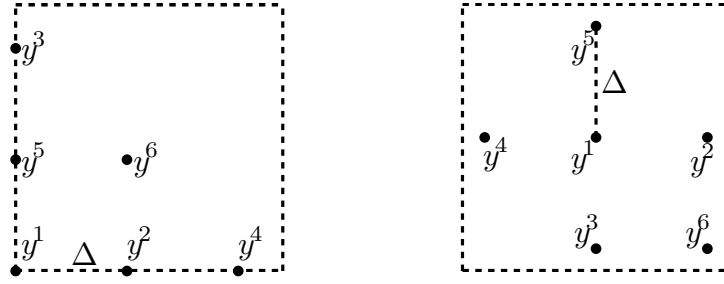


Figura 3.2: 6 pontos de interpolação para $n = 2$, com reordenamento de y^3 e y^5

Teorema 3.2 Construção do primeiro modelo

Considere $f : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função e $m \geq 2n + 1$ pontos de interpolação. Considere $y^1 \in \mathbb{A}$, $\Delta > 0$ o raio da região de confiança, $Y_1 = \{y^1, y^2, \dots, y^m\}$ o conjunto interpolador obtido por (3.2) e (3.6) com os pontos reordenados como em (3.3), e os parâmetros do modelo quadrático (3.1), com $x^0 = y^1$, dados por

1. $Q_1(y^1) = c_1 = f_1$;
2. $[\nabla Q_1(y^1)]_i = [g^1]_i = \frac{1}{2\Delta} (f_{i+1} - f_{i+1+n})$, $i = 1, 2, \dots, n$;
3. $[\nabla^2 Q_1]_{i,i} = \frac{1}{\Delta^2} (f_{i+1} + f_{i+1+n} - 2f_1)$, $i = 1, 2, \dots, n$;
4. $[\nabla^2 Q_1]_{u(j),v(j)} = \frac{1}{\Delta^2} (f_j - f_{u(j)+1} - f_{v(j)+1} + f_1)$, $j = 2n + 2, \dots, m$;
5. $[\nabla^2 Q_1]_{u(j),v(j)} = 0$, $j = m + 1, m + 2, \dots, (n + 1)(n + 2)/2$.

Então Q_1 satisfaz a condição de interpolação

$$Q_1(Y_1) = f(Y_1).$$

Demonstração. A prova será feita considerando somente a primeira parte de (3.2) e sem o reordenamento (3.3).

Segue diretamente do item 1 que $Q_1(y^1) = f(y^1)$. Considere $i = 1, \dots, n$, por (3.2) segue que $y^{i+1} = y^1 + \Delta e^i$. Logo

$$\begin{aligned} Q_1(y^{i+1}) &= Q_1(y^1 + \Delta e^i) \\ &= f_1 + [g^1]_i \Delta + \frac{\Delta^2}{2} [\nabla^2 Q_1]_{i,i}. \end{aligned}$$

Substituindo o gradiente e a Hessiana dados em 2 e 3, segue que

$$\begin{aligned} Q_1(y^{i+1}) &= f_1 + \frac{1}{2\Delta} (f_{i+1} - f_{i+1+n}) \Delta + \frac{\Delta^2}{2\Delta^2} (f_{i+1} + f_{i+1+n} - 2f_1) \\ &= f_{i+1} = f(y^{i+1}). \end{aligned}$$

Para $i = n + 1, \dots, 2n$, por (3.2) segue que $y^{i+1} = y^1 - \Delta e^{i-n}$. Logo

$$Q_1(y^{i+1}) = f_1 - [g^1]_{i-n} \Delta + \frac{\Delta^2}{2} [\nabla^2 Q_1]_{i-n, i-n}.$$

Substituindo o gradiente e a Hessiana dados em 2 e 3, segue que

$$\begin{aligned} Q_1(y^{i+1}) &= f_1 - \frac{1}{2\Delta} (f_{i-n+1} - f_{i-n+n+1}) \Delta + \frac{\Delta^2}{2\Delta^2} (f_{i-n+1} + f_{i-n+1+n} - 2f_1) \\ &= f_{i+1}. \end{aligned}$$

Finalmente, para $j = 2n + 2, \dots, m$, mostramos que $Q_1(y^j) = f_j$. Considere o par $(u(j), v(j))^T$ como nas expressões (3.4) e (3.5). Para $j = 2n + 2, \dots, m$, por (3.6)

$$y^j = y^{u(j)+1} + y^{v(j)+1} - y^1.$$

Como $u(j)$ e $v(j)$ pertencem ao conjunto $\{1, 2, \dots, n\}$, podemos substituir a primeira parte de (3.2) na igualdade anterior e segue que

$$\begin{aligned} y^j &= y^1 + \Delta e^{u(j)} + y^1 + \Delta e^{v(j)} - y^1 \Rightarrow \\ y^j &= y^1 + \Delta(e^{u(j)} + e^{v(j)}). \end{aligned}$$

Assim,

$$\begin{aligned} Q_1(y^j) &= Q_1(y^1 + \Delta(e^{u(j)} + e^{v(j)})) \\ &= f_1 + ([g^1]_{u(j)} + [g^1]_{v(j)}) \Delta + \frac{\Delta^2}{2} \{[\nabla^2 Q_1]_{u(j),u(j)} + 2[\nabla^2 Q_1]_{u(j),v(j)} + \\ &\quad + [\nabla^2 Q_1]_{v(j),v(j)}\} \\ &= f_1 + \frac{1}{2} (f_{u(j)+1} - f_{u(j)+1+n} + f_{v(j)+1} - f_{v(j)+1+n}) + \frac{1}{2} \{f_{u(j)+1} + f_{u(j)+1+n} \\ &\quad - 2f_1 + 2(f_j - f_{u(j)+1} - f_{v(j)+1} + f_1) + f_{v(j)+1} + f_{v(j)+1+n} - 2f_1\} \\ &= f_j. \end{aligned}$$

□

O Teorema anterior é demonstrado de forma análoga para o caso em que houve o reordenamento (3.3), ou quando Y_1 é obtido pela segunda ou terceira fórmulas em (3.2).

3.2 Atualizações

Nesta seção descrevemos a atualização do conjunto interpolador e do modelo quadrático.

Dado o modelo quadrático Q_k como em (3.1), considere a condição de interpolação $Q_k(Y_k) = f(Y_k)$ onde

$$Y_k = \{y^1, y^2, \dots, y^m\}$$

é atualizado na k -ésima iteração do algoritmo para

$$Y^+ = Y_k \setminus \{y^t\} \cup \{x^k + d^k\}.$$

O modelo Q^+ associado a Y^+ é obtido atualizando Q_k . Para isto é utilizada a técnica de Broyden que minimiza

$$\|\nabla^2 Q^+ - \nabla^2 Q_k\|_F,$$

sujeita à simetria da Hessiana do modelo e às restrições $Q^+(Y^+) = f(Y^+)$.

As propriedades a seguir são exigências sobre Y^+ para garantir a **unicidade** do modelo atualizado pela técnica que minimiza a norma de Frobenius da diferença entre as Hessianas dos modelos novo e corrente.

Propriedade P1 *O conjunto interpolador Y^+ deve ser tal que a dimensão do espaço*

$$\mathbb{P}^+ = \{p : \mathbb{R}^n \rightarrow \mathbb{R} \mid p \text{ é polinômio quadrático e } p(y^{i+}) = 0, \forall y^{i+} \in Y^+\}$$

seja $\frac{1}{2}(n+1)(n+2) - m$.

As condições $p(y^{i+}) = 0$, para todo $y^{i+} \in Y^+$, resultam em um sistema linear homogêneo de m equações e $(n+1)(n+2)/2$ variáveis. Pela Propriedade P1 a dimensão do conjunto solução desse sistema coincide com a dimensão de \mathbb{P}^+ . Logo as linhas da matriz dos coeficientes são linearmente independentes e conseqüentemente um sistema do tipo $p(Y^+) = f(Y^+)$ tem espaço solução de dimensão coincidente com a dimensão de \mathbb{P}^+ . Logo, a Propriedade P1 deixa as condições de interpolação consistentes, ou seja, é possível escolher em \mathbb{P}^+ um polinômio quadrático que satisfaça tais condições.

Propriedade P2 *O conjunto interpolador Y^+ deve ser tal que se $p : \mathbb{R}^n \rightarrow \mathbb{R}$ é um polinômio linear tal que $p(Y^+) = 0$, então $p \equiv 0$.*

Esta propriedade garante a não existência de dois polinômios quadráticos distintos que se anulam em Y^+ com a mesma Hessiana, pois do contrário a diferença entre estes polinômios seria um polinômio linear não nulo, contradizendo a Propriedade P2.

Considere na k -ésima iteração o conjunto interpolador Y_k e o novo conjunto interpolador

$$Y^+ = Y_k \setminus \{y^t\} \cup \{x^k + d^k\}$$

obtido de Y_k modificando apenas um ponto. O índice t é reservado para o ponto y^t que deve deixar o conjunto interpolador.

Tome um modelo quadrático $D \neq 0$, $D \in \mathbb{P}^+$ escrito na forma

$$D(x) = c + g^T(x - x^0) + \frac{1}{2}(x - x^0)^T \nabla^2 D(x - x^0), \quad x \in \mathbb{R}^n \quad (3.7)$$

e o modelo corrente Q_k (3.1) na forma

$$Q_k(x) = c_k + (g^k)^T(x - x^0) + \frac{1}{2}(x - x^0)^T \nabla^2 Q_k(x - x^0).$$

O **novo modelo** Q^+ é obtido a partir do corrente por

$$Q^+(x) = Q_k(x) + D(x) \quad (3.8)$$

tal que D fornece o menor valor para a norma de Frobenius $\|\nabla^2 Q^+ - \nabla^2 Q_k\|_F$ e satisfaz a condição

$$D(Y^+) = f(Y^+) - Q_k(Y^+). \quad (3.9)$$

O segundo membro da expressão (3.9) tem apenas a t -ésima coordenada não nula e as expressões (3.8) e (3.9) fornecem as condições de interpolação para o novo modelo Q^+ , ou seja,

$$\begin{aligned} Q^+(Y^+) &= Q_k(Y^+) + D(Y^+) \\ &= Q_k(Y^+) + f(Y^+) - Q_k(Y^+) \\ &= f(Y^+). \end{aligned}$$

Para identificar os parâmetros c , g e $\nabla^2 D$ que determinam o polinômio quadrático D , resolvemos o problema

$$\begin{aligned} &\text{minimizar } \|\nabla^2 Q^+ - \nabla^2 Q_k\|_F \\ &\text{sujeita a } D(Y^+) = f(Y^+) - Q_k(Y^+), \end{aligned}$$

que é equivalente a

$$\begin{aligned} &\text{minimizar } \frac{1}{4} \|\nabla^2 D\|_F^2 \\ &\text{sujeita a } D(Y^+) = f(Y^+) - Q_k(Y^+). \end{aligned} \tag{3.10}$$

O problema (3.10) é de programação quadrática convexa [7], logo a condição de primeira ordem é necessária e suficiente para otimalidade. Sejam m multiplicadores de Lagrange φ_j e considere a função Lagrangiana

$$L : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^{n^2} \times \mathbb{R}^m \longrightarrow \mathbb{R}$$

dada por

$$\begin{aligned} L(c, g, \nabla^2 D, \varphi) &= \frac{1}{4} \|\nabla^2 D\|_F^2 - \sum_{j=1}^m \varphi_j \left\{ Q_k(y^{j+}) - f(y^{j+}) + c + g^T(y^{j+} - x^0) + \right. \\ &\quad \left. + \frac{1}{2} (y^{j+} - x^0)^T \nabla^2 D (y^{j+} - x^0) \right\}, \end{aligned}$$

onde $y^{i+} \in Y^+$ para todo i . Utilizando a definição da norma de Frobenius podemos reescrever

$$\begin{aligned} L(c, g, \nabla^2 D, \varphi) &= \frac{1}{4} \sum_{i=1}^n \sum_{j=1}^n [\nabla^2 D]_{i,j}^2 - \sum_{j=1}^m \varphi_j \left\{ Q_k(y^{j+}) - f(y^{j+}) + c \right. \\ &\quad \left. + g^T(y^{j+} - x^0) + \frac{1}{2} (y^{j+} - x^0)^T \nabla^2 D (y^{j+} - x^0) \right\}. \end{aligned}$$

Calculando o gradiente de L em relação a c , g , $\nabla^2 D$ e igualando a zero, temos respectivamente

$$\left\{ \begin{array}{l} \sum_{j=1}^m \varphi_j = 0 \\ \sum_{j=1}^m \varphi_j (y^{j+} - x^0) = 0 \\ \nabla^2 D - \sum_{j=1}^m \varphi_j (y^{j+} - x^0)(y^{j+} - x^0)^T = 0. \end{array} \right. \tag{3.11}$$

Assim,

$$\nabla^2 D = \sum_{j=1}^m \varphi_j (y^{j+} - x^0)(y^{j+} - x^0)^T. \quad (3.12)$$

Utilizando a expressão (3.12) é possível eliminar ∇D^2 em (3.7) e segue que

$$\begin{aligned} D(x) &= c + g^T(x - x^0) + \frac{1}{2}(x - x^0)^T \nabla^2 D(x - x^0) \\ &= c + g^T(x - x^0) + \frac{1}{2} \sum_{j=1}^m \varphi_j (x - x^0)^T \{(y^{j+} - x^0)(y^{j+} - x^0)^T\} (x - x^0) \\ &= c + g^T(x - x^0) + \frac{1}{2} \sum_{j=1}^m \varphi_j \{(x - x^0)^T (y^{j+} - x^0)\}^2. \end{aligned} \quad (3.13)$$

Substituindo esta última igualdade em (3.8) segue que

$$Q^+(x) = Q_k(x) + c + g^T(x - x^0) + \frac{1}{2} \sum_{j=1}^m \varphi_j \{(x - x^0)^T (y^{j+} - x^0)\}^2. \quad (3.14)$$

Portanto, o novo modelo Q^+ é obtido calculando as $1 + n + m$ componentes $c \in \mathbb{R}$, $g \in \mathbb{R}^n$ e $\varphi \in \mathbb{R}^m$, respectivamente. Pela expressão (3.9), as duas primeiras igualdades em (3.11) e por (3.13), o modelo atualizado Q^+ é obtido pela solução do sistema

$$\begin{cases} c + g^T(y^{i+} - x^0) + \frac{1}{2} \sum_{j=1}^m \varphi_j \{(y^{i+} - x^0)^T (y^{j+} - x^0)\}^2 = f(y^{i+}) - Q_k(y^{i+}) \\ \sum_{j=1}^m \varphi_j = 0 \\ \sum_{j=1}^m \varphi_j (y^{j+} - x^0) = 0. \end{cases} \quad (3.15)$$

O sistema (3.15) pode ser escrito na forma matricial em blocos como

$$\left(\begin{array}{c|c} A & M^T \\ \hline M & 0 \end{array} \right) \begin{pmatrix} \varphi \\ c \\ g \end{pmatrix} = \begin{pmatrix} (f - Q_k)(Y^+) \\ 0 \\ 0 \end{pmatrix}, \quad (3.16)$$

onde a matriz $A \in \mathbb{R}^{n \times n}$ é dada por

$$A_{i,j} = \frac{1}{2} ((y^{i+} - x^0)^T (y^{j+} - x^0))^2, \quad 1 \leq i, j \leq m$$

e $M \in \mathbb{R}^{(n+1) \times m}$

$$M = \begin{pmatrix} 1 & 1 & \dots & 1 \\ y^{1+} - x^0 & y^{2+} - x^0 & \dots & y^{m+} - x^0 \end{pmatrix}.$$

Denotamos por

$$W^+ = \left(\begin{array}{c|c} A & M^T \\ \hline M & 0 \end{array} \right) \quad (3.17)$$

a matriz $(m + n + 1) \times (m + n + 1)$ associada ao sistema (3.16) e por

$$H^+ = \left(\begin{array}{c|c} \Phi & E^T \\ \hline E & \Upsilon \end{array} \right)$$

a matriz $(m + n + 1) \times (m + n + 1)$ inversa de W^+ . Como A é simétrica, segue que as matrizes W^+ e H^+ são simétricas.

O segundo membro do sistema (3.16) tem apenas a componente t não nula. Logo pode ser reescrito na forma

$$\begin{pmatrix} \varphi \\ c \\ g \end{pmatrix} = H^+ \begin{pmatrix} (f - Q_k)(Y^+) \\ 0 \\ 0 \end{pmatrix} = [(f - Q_k)(Y^+)]_t H^+ e^t. \quad (3.18)$$

Assim, resolvendo o sistema (3.18), pela expressão (3.12) os parâmetros c , g e $\nabla^2 D$ do modelo (3.7) ficam determinados. Finalmente o modelo atualizado Q^+ é dado por (3.8) somando o modelo anterior Q_k com o modelo D .

Em BOBYQA o modelo quadrático é atualizado pela expressão (3.14), mas a Hessiana da atualização precisa ser armazenada, e portanto a expressão (3.12) é utilizada. Assim,

$$Q^+(x) = c_k + c + (g + g^k)^T(x - x^0) + \frac{1}{2}(x - x^0)^T(\nabla^2 D + \nabla^2 Q_k)(x - x^0). \quad (3.19)$$

A expressão (3.19) ainda sofre três mudanças. Uma relacionada à decomposição da Hessiana, consequência da necessidade em trabalhar dentro da complexidade algorítmica $O(m^2)$. Outra mudança envolve um procedimento para minimizar erros inerentes ao processo de atualização dos modelos. A terceira está relacionada à eliminação do termo constante. Isto é abordado mais adiante nesta seção. Primeiramente mostramos como os parâmetros c , g e $\nabla^2 D$ em (3.19) são calculados.

A matriz H^+ inversa de W^+ pode ser obtida a partir da matriz H inversa de W dada na iteração anterior (Teorema 3.5 a seguir). Desta forma é necessário conhecermos a matriz inversa associada ao primeiro modelo dado pelo Teorema 3.2, para que haja as atualizações destes modelos.

Dado o conjunto interpolador Y_1 com a estrutura obtida por (3.2) e (3.6), o Teorema 3.2 fornece um modelo quadrático com uma fórmula fechada para Q_1 . A partir de Y_1 , a matriz W_1 é tomada análoga à igualdade em (3.17) com sua inversa

$$H_1 = \left(\begin{array}{c|c} \Phi & E^T \\ \hline E & \Upsilon \end{array} \right) \quad (3.20)$$

com $\Phi \in \mathbb{R}^{m \times m}$, $E \in \mathbb{R}^{(n+1) \times m}$ e $\Upsilon \in \mathbb{R}^{(n+1) \times (n+1)}$ obtida pelo Teorema 3.3 a seguir.

Teorema 3.3 Matriz inversa

Suponha W_1 a matriz como em (3.17) associada ao primeiro modelo obtido pelo Teorema 3.2. Considere H_1 na forma (3.20), $m \geq 2n + 1$ e $\Delta > 0$ o raio da região de confiança. Então, H_1 é a inversa de W_1 quando

1. $E_{1,1} = 1$ e $E_{1,j} = 0$ para todo $j = 2, \dots, m$;
2. $E_{i,i} = \frac{1}{2\Delta}$ e $E_{i,i+n} = -\frac{1}{2\Delta}$ para todo $i = 2, \dots, n+1$ e os demais elementos de E são nulos;
3. a matriz Υ é nula;
4. A matriz Φ pode ser escrita como o produto ZZ^T , onde para $1 \leq l \leq n$,

$$Z_{1,l} = -\frac{\sqrt{2}}{\Delta^2}, \quad Z_{l+1,l} = \frac{\sqrt{2}}{2\Delta^2} \quad e \quad Z_{n+l+1,l} = \frac{\sqrt{2}}{2\Delta^2},$$

e para $n+1 \leq l \leq m-n-1$,

$$Z_{1,l} = Z_{n+l+1,l} = \frac{1}{\Delta^2} \quad e \quad Z_{u(j)+1,l} = Z_{v(j)+1,l} = -\frac{1}{\Delta^2}$$

onde $j = n+l+1$ e $u(j), v(j)$ como nas expressões (3.4) e (3.5).

Demonstração. Em [53].

□

A seguir exemplificamos como ocorre o procedimento de construção da inversa no plano e com cinco pontos de interpolação.

Exemplo 3.4 Construção da matriz inversa

Considere $n = 2$, $m = 5$, $\Delta > 0$ e $y^1 \in \mathbb{R}^2$. O conjunto interpolador Y_1 definido pela expressão (3.2) é

$$\begin{aligned} Y_1 &= \{ y^1, y^2, y^3, y^4, y^5 \} \\ &= \left\{ \begin{pmatrix} y_1^1 \\ y_2^1 \end{pmatrix}, \begin{pmatrix} y_1^1 + \Delta \\ y_2^1 \end{pmatrix}, \begin{pmatrix} y_1^1 \\ y_2^1 + \Delta \end{pmatrix}, \begin{pmatrix} y_1^1 - \Delta \\ y_2^1 \end{pmatrix}, \begin{pmatrix} y_1^1 \\ y_2^1 - \Delta \end{pmatrix} \right\}. \end{aligned}$$

Então a matriz W_1 como em (3.17) fica

$$W_1 = \left(\begin{array}{c|ccc} A & M^T \\ \hline M & 0 \end{array} \right) = \left(\begin{array}{ccccc|ccc} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & \frac{\Delta^4}{2} & 0 & \frac{\Delta^4}{2} & 0 & 1 & \Delta & 0 \\ 0 & 0 & \frac{\Delta^4}{2} & 0 & \frac{\Delta^4}{2} & 1 & 0 & \Delta \\ 0 & \frac{\Delta^4}{2} & 0 & \frac{\Delta^4}{2} & 0 & 1 & -\Delta & 0 \\ 0 & 0 & \frac{\Delta^4}{2} & 0 & \frac{\Delta^4}{2} & 1 & 0 & -\Delta \\ \hline 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & \Delta & 0 & -\Delta & 0 & 0 & 0 & 0 \\ 0 & 0 & \Delta & 0 & -\Delta & 0 & 0 & 0 \end{array} \right).$$

Pelo Teorema 3.3,

$$Z = \begin{pmatrix} -\frac{\sqrt{2}\Delta^{-2}}{2} & -\frac{\sqrt{2}\Delta^{-2}}{2} \\ \frac{\sqrt{2}\Delta^{-2}}{2} & 0 \\ 0 & \frac{\sqrt{2}\Delta^{-2}}{2} \\ \frac{\sqrt{2}\Delta^{-2}}{2} & 0 \\ 0 & \frac{\sqrt{2}\Delta^{-2}}{2} \end{pmatrix}$$

e a inversa da matriz W_1 é

$$H_1 = \left(\begin{array}{cc|cc} ZZ^T & E^T & & \\ E & \Upsilon & & \end{array} \right) = \left(\begin{array}{ccccc|ccc} 4\Delta^{-4} & -\Delta^{-4} & -\Delta^{-4} & -\Delta^{-4} & -\Delta^{-4} & 1 & 0 & 0 \\ -\Delta^{-4} & \frac{\Delta^{-4}}{2} & 0 & \frac{\Delta^{-4}}{2} & 0 & 0 & \frac{\Delta^{-1}}{2} & 0 \\ -\Delta^{-4} & 0 & \frac{\Delta^{-4}}{2} & 0 & \frac{\Delta^{-4}}{2} & 0 & 0 & \frac{\Delta^{-1}}{2} \\ -\Delta^{-4} & \frac{\Delta^{-4}}{2} & 0 & \frac{\Delta^{-4}}{2} & 0 & 0 & -\frac{\Delta^{-1}}{2} & 0 \\ -\Delta^{-4} & 0 & \frac{\Delta^{-4}}{2} & 0 & \frac{\Delta^{-4}}{2} & 0 & 0 & -\frac{\Delta^{-1}}{2} \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\Delta^{-1}}{2} & 0 & -\frac{\Delta^{-1}}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\Delta^{-1}}{2} & 0 & -\frac{\Delta^{-1}}{2} & 0 & 0 & 0 \end{array} \right).$$

O teorema a seguir fornece uma fórmula fechada para inverter a matriz W^+ dada em (3.17). O teorema afirma que é possível resolver o sistema (3.16) sem a necessidade de expressar explicitamente a matriz W^+ .

Teorema 3.5 *Considere o conjunto interpolador*

$$Y_k = \{y^1, \dots, y^m\},$$

H não singular e simétrica, $H = W^{-1}$ com W dada como em (3.17) em relação a Y_k . Suponha $\sigma \neq 0$, $x^+ = x^k + d^k$ o ponto que deve substituir y^t em Y_k . Tome o novo conjunto interpolador

$$Y^+ = Y_k \setminus \{y^t\} \cup \{x^+\}$$

e W^+ dada em (3.17) em relação a Y^+ . Então a inversa da matriz W^+ é

$$H^+ = H + \frac{1}{\sigma} \left\{ \alpha(e^t - Hw)(e^t - Hw)^T - \beta H e^t (e^t)^T H + \tau [H e^t (e^t - Hw)^T + (e^t - Hw)(e^t)^T H] \right\}, \quad (3.21)$$

onde $e^t \in \mathbb{R}^{m+n+1}$ é o t -ésimo vetor coordenado e w tem as componentes

$$\begin{aligned} w_i &= \frac{1}{2} \{(y^i - x^0)^T (x^+ - x^0)\}^2, & i = 1, 2, \dots, m, \\ w_{m+1} &= 1, \\ w_{i+m+1} &= [x^+ - x^0]_i, & i = 1, 2, \dots, n, \end{aligned} \quad (3.22)$$

com os parâmetros α, β, τ e σ tomando os valores

$$\begin{aligned}\alpha &= \alpha(t) &= (e^t)^T H e^t, \\ \beta &= \beta(x^+) &= \frac{1}{2} \|x^+ - x^0\|^4 - w^T H w, \\ \tau &= \tau(t) &= (e^t)^T H w, \\ \sigma &= \alpha\beta + \tau^2.\end{aligned}\tag{3.23}$$

Demonstração. Em [49].

□

A proposição a seguir afirma que o denominador σ da fórmula (3.21) é não negativo.

Proposição 3.6 *Os parâmetros α e β na fórmula de atualização (3.23) são não negativos.*

Demonstração. Em [49, Lema 2].

□

A relação entre σ e a não singularidade de W^+ é dada pela proposição a seguir.

Proposição 3.7 *O parâmetro σ em (3.23) tem o valor*

$$\sigma = \frac{\det W^+}{\det W}.$$

Demonstração. Em [51, Lema 1].

□

Pela Proposição 3.7, a mudança de um elemento no conjunto interpolador causa dependência linear nas condições de interpolação se, e somente se, o denominador σ na fórmula de atualização da matriz H é nulo. Assim, o algoritmo trabalha com uma técnica que escolhe pontos para o conjunto interpolador visando aumentar o denominador σ . Quando em uma iteração k conhecemos o ponto que deve entrar no conjunto interpolador Y_k , é possível retirar um ponto do conjunto interpolador mantendo as Propriedades P1 e P2 e obter o denominador $\sigma \neq 0$ na atualização da inversa H do Teorema 3.5. Para provar isto, consideramos para todo $\ell = 1, 2, \dots, m$, a ℓ -ésima função de Lagrange de ordem 2, como na Definição 2.2.

Como λ_ℓ é quadrática e o número de pontos de interpolação pode ser menor do que $(n+1)(n+2)/2$, a liberdade nos parâmetros é retomada com a estratégia de Broyden, minimizando $\|\Delta^2 \lambda_\ell\|_F$, sujeita a $\lambda_\ell(y^j) = \delta_{j\ell}$. Logo, os parâmetros de λ_ℓ são definidos pelo sistema (3.16), mas no segundo membro aparece o vetor $e^\ell \in \mathbb{R}^{m+n+1}$.

Portanto, considerando

$$\lambda_\ell(x) = c + g^T(x - x^0) + \frac{1}{2} \sum_{i=1}^m \varphi_i \{(x - x^0)^T (y^i - x^0)\}^2, \tag{3.24}$$

segue de modo análogo à determinação da função D em (3.7), que os parâmetros c, g e φ da forma de Lagrange λ_ℓ estão definidos na ℓ -ésima coluna de H^+ , ou seja,

$$\begin{pmatrix} \varphi \\ c \\ g \end{pmatrix} = H^+ e^\ell. \quad (3.25)$$

Tomando

$$\begin{aligned} [w(x)]_i &= \frac{1}{2} \{(x - x^0)^T (y^i - x^0)\}^2, & i = 1, 2, \dots, m, \\ [w(x)]_{m+1} &= 1, \\ [w(x)]_{i+m+1} &= [x - x^0]_i, & i = 1, 2, \dots, n, \end{aligned} \quad (3.26)$$

podemos escrever por (3.24), (3.26) e (3.25)

$$\begin{aligned} \lambda_\ell(x) &= \frac{1}{2} \sum_{i=1}^m \varphi_i \{(x - x^0)^T (y^i - x^0)\}^2 + c + g^T (x - x^0) \\ &= \sum_{i=1}^m \varphi_i [w(x)]_i + cw[x - x^0]_{m+1} + \sum_{i=1}^m g^i [w(x)]_{m+1+i} \\ &= (He^\ell)^T w(x). \end{aligned}$$

Proposição 3.8 *Considere λ_ℓ para $\ell = 1, 2, \dots, m$ as funções de Lagrange e x^+ um novo ponto diferente dos pontos do conjunto interpolador Y_k . Então*

1. *existe pelo menos um índice $t \in \{1, 2, \dots, m\}$ tal que $\lambda_t(x^+) \neq 0$;*
2. *o número τ dado em (3.23) pode ser escrito como $\tau = \lambda_t(x^+)$;*
3. *considere t escolhido tal que $\lambda_t(x^+) \neq 0$, então as Propriedades P1 e P2 são mantidas.*

Demonstração. Em [49].

□

A Proposição 3.8 estabelece que sempre é possível retirar um ponto do conjunto interpolador e o substituir por outro, mantendo as Propriedades P1 e P2. Portanto, como a técnica da norma de Frobenius fornece uma única matriz Hessiana para o modelo atualizado, as Propriedades P1 e P2 garantem a unicidade deste modelo.

Complexidade algorítmica

Faremos comentários de como a estrutura que foi produzida pode atualizar o modelo quadrático em $O(m^2)$ operações.

A partir da inversa H , o algoritmo gera a matriz H^+ com a expressão (3.21) em $O(m^2)$ operações [51, pág.13]. Os parâmetros c, g e φ que são utilizados na atualização do modelo são obtidos em $O(m)$ operações utilizando o sistema (3.18). O modelo Q^+ é restaurado com a expressão (3.19) em $O(mn^2)$ operações.

Assim, para fazer todo o procedimento de atualização e deixar explícito o novo modelo, a expressão (3.19) sofre mudanças algébricas (apenas na parcela quadrática)

para permitir que possa ser feita a restauração de Q^+ em $O(m^2)$ operações e não $O(mn^2)$. Esta modificação na parcela quadrática do novo modelo consiste em decompor a Hessiana como a soma [49, pág.13]

$$\nabla^2 Q^+ = \Gamma^+ + \sum_{j=1}^m \mu_j^+ (y^{j+} - x^0)(y^{j+} - x^0)^T, \quad (3.27)$$

onde

$$\mu_j^+ = \mu_j(1 - \delta_{jt}) + \{f(x^+) - Q_k(x^+)\} H_{jt}^+$$

com δ_{jt} o delta de Kronecker e

$$\Gamma^+ = \Gamma + \mu_t (y^t - x^0)(y^t - x^0)^T.$$

Para a primeira iteração, $\Gamma = 0$ e $\mu_j = \varphi_j$ para $j = 1, 2, \dots, m$. Apenas a matriz Γ^+ e os parâmetros μ_j^+ precisam ser armazenados para resgatar a Hessiana do novo modelo, ou seja, a matriz $\nabla^2 Q^+$ é armazenada implicitamente no formato (3.27). Desta forma, em $O(m)$ operações é possível resgatar a Hessiana de Q^+ . Ainda, aproveitando a simetria da Hessiana, a decomposição (3.27) permite multiplicar $\nabla^2 Q^+$ por um vetor do \mathbb{R}^n em $O(mn)$ operações.

Portanto, a atualização do modelo quadrático e a avaliação deste modelo em um vetor do \mathbb{R}^n fica dentro de $O(m^2)$ operações, pois $m > n$. Não entraremos no mérito, mas os outros procedimentos do BOBYQA que serão discutidos à frente, não interferem nesta complexidade.

Eliminação do termo constante

BOBYQA trabalha com o modelo na k -ésima iteração sem a parcela constante, ou seja,

$$Q_k(x) = (g^k)^T (x - x^0) + \frac{1}{2} (x - x^0)^T \nabla^2 Q_k (x - x^0).$$

Considere s na k -ésima iteração tal que $x^k = y^s$ e t a posição do ponto no conjunto interpolador que será substituído na iteração atual. Tome w e v a t -ésima e s -ésima coluna da matriz W dada no Teorema 3.5, respectivamente. Para eliminar o termo constante do modelo, o algoritmo substitui no Teorema 3.5 as ocorrências de

$$\begin{aligned} e^t - Hw & \text{ por } e^t - e^s + H(w - v), \\ (e^t)^T Hw & \text{ por } (e^t)^T H(w - v), \\ w^T Hw & \text{ por } (w - v)^T H(w - v) + 2[w]_s - [v]_s. \end{aligned} \quad (3.28)$$

A constante c_k no modelo (3.1) é eliminada com as substituições (3.28) pois a $(m+1)$ -ésima componente de $w - v$ é nula.

A justificativa para retirar o termo constante do modelo é que isto reflete na redução de erros de arredondamentos que ocorrem na atualização da inversa H . Mais detalhes em [52, pág. 29].

O Teorema 3.5 possibilita determinar a inversa H e isto é suficiente para atualizar o modelo. No entanto, é possível que o acúmulo de erros de arredondamentos introduzam autovalores negativos na submatriz $\Phi \in \mathbb{R}^{m \times m}$ de H . BOBYQA trabalha

com a decomposição $\Phi = ZZ^T$ e atualiza a matriz $Z \in \mathbb{R}^{m \times (m-n-1)}$ modificando apenas sua primeira coluna com a fórmula: para $i = 1, 2, \dots, m$,

$$Z_{i,1} = \frac{1}{\sqrt{\sigma}} \{ \tau Z_{i,1} + (e^t - e^s - H[w - v])_i Z_{t,1} \}.$$

Detalhes em [54]. Esta decomposição de Φ é uma herança do UOBYQA que utiliza esta estrutura para trabalhar com autovalores negativos em Φ [48], mas BOBYQA não precisaria desta estrutura pois sua resposta a autovalores negativos é dada com o procedimento descrito na próxima seção, e que se relaciona ao denominador σ da fórmula para a inversa H . O cálculo deste denominador não utiliza a decomposição de Φ .

Na implementação de BOBYQA, Z é denominada de $ZMAT$ e as últimas n colunas de H constituem a matriz $BMAT \in \mathbb{R}^{(m+n) \times n}$ sem a linha $m + 1$ de H . Armazenando $ZMAT$ e $BMAT$ é possível resgatar H (sem a linha e a coluna $m + 1$), o que permite atualizar o modelo quadrático sem a parcela constante.

3.3 Cálculo do passo

A partir de um ponto $x^k \in \mathbb{A}$, calculamos um passo $d^k \in \mathbb{R}^n$ por uma iteração do tipo região de confiança ou por uma iteração denominada alternativa. Em ambos os casos o passo d^k deve satisfazer

$$a \leq x^k + d^k \leq b \text{ e } \|d^k\| \leq \Delta_k,$$

onde Δ_k é o raio da região em que confiamos no modelo Q_k na k -ésima iteração.

Iteração de região de confiança

Em uma iteração de região de confiança, o passo d^k é obtido resolvendo o subproblema

$$\begin{aligned} & \text{minimizar} && Q_k(x) \\ & \text{sujeita a} && a \leq x \leq b \\ & && \|x - x^k\| \leq \Delta_k, \end{aligned}$$

utilizando uma versão do método Gradiente Conjugado Truncado. Este procedimento gera direções de buscas que iremos utilizar na Subseção 3.6.2, sendo os detalhes omitidos neste texto e disponíveis em [52, 54].

Iteração alternativa

Uma iteração alternativa é executada após uma tentativa de região de confiança em que foi calculado um passo d^k ou pequeno ou que não forneceu decréscimo suficiente na função objetivo e os pontos de interpolação estão longe do iterando atual x^k . O passo alternativo tem a finalidade de melhorar a estrutura posicional do conjunto interpolador, modificando apenas um elemento deste conjunto e trabalhando com a parcela τ do denominador σ da expressão de atualização da matriz inversa H (3.21).

Suponha que o Algoritmo BOBYQA tenha a necessidade de executar uma iteração alternativa. Neste caso, primeiro é escolhido y^t mais distante do atual x^k . Pela Proposição 3.8, item 2, o número τ é maior quanto maior for a função de Lagrange λ_t em x^+ . Sendo assim, o algoritmo escolhe x^+ em uma iteração alternativa entre dois tipos de pontos $x^k + d^k$ ou $x^k + d_a^k$ que visam aumentar o denominador σ . O passo d_a^k é obtido a partir de x^k com a solução tipo Cauchy (detalhes em [54]) do problema

$$\begin{aligned} & \text{maximizar} && |\lambda_t(x)| \\ & \text{sujeita a} && \|x - x^k\| \leq \Delta_k \end{aligned} \quad (3.29)$$

e o passo d^k é obtido maximizando o denominador $\sigma(t)$ ao longo das $m - 1$ direções a partir de x^k e os demais pontos interpoladores. A decisão entre d^k e d_a^k é tomada a partir do seguinte critério. Se

$$(\lambda_t(x^k + d_a^k))^2 > H_{tt}\beta(x^k + d^k) + (\lambda_t(x^k + d^k))^2,$$

então d_a^k é o passo escolhido, caso contrário, d^k . Este procedimento é executado em BOBYQA na subrotina ALTMOV e antes de uma chamada a esta subrotina o raio da região de confiança é temporariamente reduzido.

Observação 3.9 *O passo d^k alternativo é obtido, em geral, maximizando a t -ésima função de Lagrange, que por sua vez está relacionada ao número τ como na Proposição 3.8, item 2. Logo, o que buscamos com uma iteração alternativa é aumentar o denominador σ na fórmula de atualização da matriz H (3.21). Assim, de acordo com a Proposição 3.7, a disposição geométrica do conjunto interpolador é modificada visando melhorar o posicionamento.*

3.4 Atualização do conjunto interpolador

Nesta seção descrevemos como é escolhido o ponto que deixa o conjunto interpolador em uma iteração de região de confiança e em uma iteração alternativa.

Em cada iteração do algoritmo apenas um ponto do conjunto interpolador Y_k é modificado (exceto quando a iteração envolve o *resgate do modelo*, detalhado na Seção 3.5). O novo ponto é calculado por uma iteração de região de confiança ou por uma iteração alternativa.

Suponha que $x^k + d^k$ foi obtido por uma **iteração de região de confiança** e que o ponto $x^k + d^k$ deve entrar no conjunto Y_k . Para determinar quem deixa Y_k , o algoritmo escolhe o índice t que auxilia na não singularidade do sistema (3.16) e levando em consideração o ponto mais distante de x^k .

Seja \bar{t} a posição em Y_k onde $x^k = y^{\bar{t}}$ ocorre e para cada $j \in J = \{1, 2, \dots, m\} \setminus \{\bar{t}\}$, considere

$$\sigma(j) = \left\{ H_{j,j} \left(\frac{1}{2} \|x^+ - x^0\|^4 - w^T H w \right) + ((e^j)^T H w)^2 \right\}.$$

O ponto y^t que deixa Y_k é tal que

$$t = \arg \max_{j \in J} \left\{ \max \left\{ \sigma(j), \frac{\sigma(j) \|y^j - x^k\|^2}{\Delta_k^2} \right\} \right\}, \quad (3.30)$$

onde Hw é calculado com a modificação (3.28) pois a parcela constante do modelo quadrático não é considerada em BOBYQA.

Por outro lado, se o algoritmo irá executar uma **iteração alternativa**, então antes de calcular o passo a partir de x^k o algoritmo escolhe y^t que deixa Y_k buscando o ponto mais distante de x^k . A ideia de retirar os pontos distantes é para manter o modelo Q_k como uma aproximação adequada para f dentro da região de confiança. Assim, o índice t é escolhido como

$$t = \arg \max_{1 \leq j \leq m} \{ \|y^j - x^k\| \}, \quad (3.31)$$

ou seja, é o ponto mais distante de x^k .

As decisões em executar uma iteração de região de confiança ou alternativa são apresentadas na última seção deste capítulo.

Observação 3.10 *A geometria do conjunto interpolador é uma preocupação constante no algoritmo BOBYQA. Para uma iteração de região de confiança, primeiro é determinado $x^k + d^k$ e no momento de escolher o ponto y^t que sai de Y_k buscamos aumentar o denominador σ , que tem reflexo na geometria de Y_k como sugere a Proposição 3.23. Em uma iteração alternativa, primeiro é escolhido y^t pela expressão (3.31), que leva em consideração a geometria do conjunto interpolador Y_k . Em seguida, o ponto $x^k + d^k$ que entra no conjunto interpolador, também objetiva aumentar o valor de σ , aumentando o valor de τ como sugere o problema de otimização (3.29).*

3.5 Mudanças no modelo

Nesta seção apresentamos as modificações que o modelo quadrático pode sofrer durante uma iteração. Primeiro discutimos a necessidade em se resgatar o modelo quando o denominador na atualização da inversa H fica pequeno. Em seguida, discutimos a translação do modelo Q_k quando x^k está longe de x^0 . Finalmente, discutimos como BOBYQA determina um modelo adicional, Q_{ad} , incorporado à estrutura que produz o modelo Q_k e que pode ser utilizado caso a norma da Hessiana do modelo Q_k insista em permanecer grande.

Resgate do modelo

Quando vale a desigualdade $\sigma = \alpha\beta + \tau^2 < \tau^2/2$, o algoritmo faz um resgate no modelo quadrático (subrotina RESCUE na implementação do BOBYQA). Este procedimento tem o objetivo de restaurar a inversa H , corrigindo possíveis erros computacionais acumulados durante as iterações anteriores.

Considere a k -ésima iteração do algoritmo e Y_k o conjunto interpolador na chamada do procedimento de resgate. Então primeiramente é construído um conjunto interpolador \widehat{Y}_k de maneira semelhante ao conjunto interpolador construído na primeira iteração, mas agora utilizando em (3.2) e (3.6) o valor do atual raio da região de confiança Δ_k ao invés de Δ . Assim, a matriz H e o modelo quadrático interpolador deste novo conjunto \widehat{Y}_k ficam inteiramente determinados por (3.2) e (3.6) e o Teorema 3.5. No entanto, com o intuito de diminuir o número de chamadas da função objetivo f , o conjunto \widehat{Y}_k recebe alguns valores de Y_k quando um determinado critério é satisfeito.

Note que em Y_k a função objetivo já é conhecida. Em cada modificação de um ponto no conjunto \widehat{Y}_k , a matriz H é atualizada com a fórmula (3.21). Este custo de atualização da H é compensado com a não necessidade de avaliar a função objetivo em um novo ponto.

Consideramos o primeiro conjunto \widehat{Y}_k e sua respectiva matriz inversa H . Considere $y^{\bar{i}} = x^k$, para cada $y^\ell \in Y_k \setminus \{y^{\bar{i}}\}$, tome $I = \{i \in \{1, 2, \dots, m\} \mid \widehat{y}^i \notin Y_k\}$. Para decidir se y^ℓ substitui algum ponto em \widehat{Y}_k e qual, é feito o seguinte cálculo:

$$\widehat{\sigma} = \max_{i \in I} \{\alpha\beta + \tau^2\} = \max_{i \in I} \left\{ H_{i,i} \left(\frac{1}{2} \|y^\ell - x^k\|^4 - w^T H w \right) + ((e^i)^T H w)^2 \right\}$$

com $w \in \mathbb{R}^{m+n+1}$ como na expressão (3.22) e tomando $x^+ = x^\ell$.

Se $\widehat{\sigma} > 10^{-3} \max_{j \in I} \{((e^j)^T H w)^2\}$, então y^ℓ assume o lugar de \widehat{y}^i em \widehat{Y}_k . Se houve mudança em \widehat{Y}_k , então a matriz H é atualizada cada vez que um novo y^ℓ substitui um ponto de \widehat{Y}_k .

Ao final da construção de \widehat{Y}_k , o modelo interpolador é atualizado pelas funções de Lagrange que são armazenadas nas colunas de H .

Minimizando erros

Para evitar o alto grau de instabilidade numérica da técnica de minimizar a norma de Frobenius, o modelo quadrático na k -ésima iteração do algoritmo é tomado na forma

$$Q_k(x) = (g^k)^T (x - x^b) + \frac{1}{2} (x - x^b)^T \nabla^2 Q_k (x - x^b),$$

onde $x^b \in \mathbb{R}^n$. O primeiro valor do vetor x^b é o ponto inicial x^1 e recebe o iterando atual x^k toda vez que o algoritmo executa a *translação* discutida a seguir. Note que neste momento estamos trocando na definição do modelo (3.1), x^0 por x^b .

Translação para a origem

Este procedimento (executado na subrotina RESCUE de BOBYQA) consiste em transladar o centro da quadrática de x^b para x^k na iteração k . Assim, o modelo quadrático

$$Q_k(x) = \nabla Q_k(x^b)^T (x - x^b) + \frac{1}{2} (x - x^b)^T \nabla^2 Q_k (x - x^b)$$

é reescrito na forma

$$Q_k(x) = \nabla Q_k(x^k)^T (x - x^k) + \frac{1}{2} (x - x^k)^T \nabla^2 Q_k (x - x^k),$$

onde, como já mencionado, o termo constante não é considerado.

Powell apresenta uma justificativa da necessidade desta translação com um exemplo numérico que mostra a instabilidade na atualização da matriz H [51, Seção 3]. Este exemplo é longo e por isso omitimos. Consideramos uma situação hipotética para exemplificar a importância da translação.

Suponha que $\|x^k - x^b\|$ seja um número de grande magnitude e que as distâncias $\|x^+ - x^k\| = \|d^k\|$ e $\|y^i - x^k\|$, para $i = 1, 2, \dots, m$ tenham magnitude um. Pela expressão que define o vetor w em (3.22), as m primeiras coordenadas são

$$w_i \approx \frac{1}{2} \|x^k - x^b\|^4. \quad (3.32)$$

Como $\beta = \frac{1}{2} \|x^+ - x^b\|^4 - w^T H w$, a contribuição de x^b na primeira parcela em β é cancelada pela segunda parcela em β . Por (3.32) esta segunda parcela inclui o termo

$$w_i H_{i,j} w_j \approx \frac{1}{4} \|x^k - x^b\|^8 H_{i,j}, \quad 1 \leq i, j \leq m.$$

Portanto, erros acumulados em $H_{i,j}$ são incrementados pela presença da potência oitava no número de grande magnitude $\|x^k - x^b\|$.

A translação não ocorre em toda iteração do Algoritmo BOBYQA. Se

$$\|d^k\|^2 \leq 10^{-3} \|x^k - x^b\|^2, \quad (3.33)$$

então a posição de x^b é transladada para x^k . Esta translação pode ser feita em $O(m^2)$ operações [54]. O exemplo dado em [51, Seção 3] sugere a escolha da constante 10^{-3} na expressão (3.33).

Substituição do modelo

Em um algoritmo que trabalha com sequência de subproblemas quadráticos, existem várias formas de se atualizar o modelo quadrático interpolador quando há liberdade nos parâmetros do modelo. Por exemplo, o modelo pode ser atualizado minimizando a norma de Frobenius da diferença da Hessiana do novo e do modelo corrente, ou simplesmente, minimizando a norma de Frobenius da Hessiana do novo modelo. Em BOBYQA, estas duas técnicas são aplicadas, e a segunda somente substitui a primeira em alguns casos. O fato de se priorizar a primeira técnica é justificado por Powell com o teorema que apresentamos a seguir. É provado que quando a função objetivo é quadrática, à medida em que atualizamos os modelos, não aumenta o erro entre as Hessianas do modelo e da função objetivo.

Teorema 3.11 *Considere f uma função quadrática e para todo k o modelo Q_k atualizado pela norma mínima de Frobenius da diferença entre as Hessianas do novo modelo Q^+ e do modelo corrente Q_k . Então*

$$\|\nabla^2 Q^+ - \nabla^2 f\| \leq \|\nabla^2 Q_k - \nabla^2 f\|.$$

Demonstração. Em [49].

□

Suponha que na k -ésima iteração o modelo Q^+ já foi obtido e assim a matriz inversa H^+ também já existe. Para evitar que se trabalhe com a Hessiana do modelo $\nabla^2 Q^+$ com norma grande, o modelo quadrático adicional $Q_{ad}^+ : \mathbb{R}^n \rightarrow \mathbb{R}$,

$$Q_{ad}^+ = g_{ad}^T (x - x^b) + \frac{1}{2} (x - x^b)^T \nabla^2 Q_{ad}^+ (x - x^b),$$

é produzido em toda iteração, aproveitando a estrutura já construída. Este modelo adicional é obtido resolvendo o problema

$$\begin{aligned} & \text{minimizar} && \|\nabla^2 Q_{ad}^+\|_F \\ & \text{sujeita a} && Q_{ad}^+(Y^+) = f(Y^+). \end{aligned}$$

Logo, a matriz inversa H^+ carrega informações suficientes para determinar os parâmetros deste novo modelo Q_{ad}^+ .

O teste para decidir se o novo modelo Q^+ construído minimizando $\|\nabla^2 Q_k - \nabla^2 Q^+\|_F$ deve ser substituído pelo modelo Q_{ad}^+ , é feito utilizando a projeção dos gradientes destes modelos na caixa \mathbb{A} . Assim, Q^+ é substituído por Q_{ad}^+ se a condição

$$\|Proj_{\mathbb{A}} \nabla Q_{ad}^+(x^+)\|^2 \leq 10^{-1} \|Proj_{\mathbb{A}} \nabla Q^+(x^+)\|^2 \quad (3.34)$$

vale em **três** iterações consecutivas de **região de confiança**. Segundo Powell, a escolha da constante 10^{-1} em (3.34) tem somente a finalidade de relutar em fugir ao procedimento usual que é a construção do modelo pela norma de Frobenius da diferença das Hessianas.

A Figura 3.3 exemplifica, no plano, quais são as dificuldades provocadas na sequência de iterandos quando a norma da Hessiana é grande. Considere que a desigualdade (3.34) seja satisfeita pelos modelos Q^+ e Q_{ad}^+ que interpolam os dois pontos y^1 e y^2 . Suponha que x^* seja o único minimizador para o problema e considere a única restrição sendo $x \in [a, b]$. Pela figura é possível constatar que a solução do subproblema

$$\begin{aligned} & \text{minimizar} && Q^+(x) \\ & \text{sujeita a} && x \in [a, b] \end{aligned}$$

está no intervalo $[y^1, y^2]$. O modelo da próxima iteração pode herdar a característica da norma grande de Q^+ e assim, a próxima solução do subproblema irá permanecer no intervalo $[y^1, y^2]$, impossibilitando a aproximação dos iterandos ao minimizador x^* . Neste exemplo, o modelo adicional Q_{ad}^+ ameniza estas dificuldades de aproximação ao minimizador.

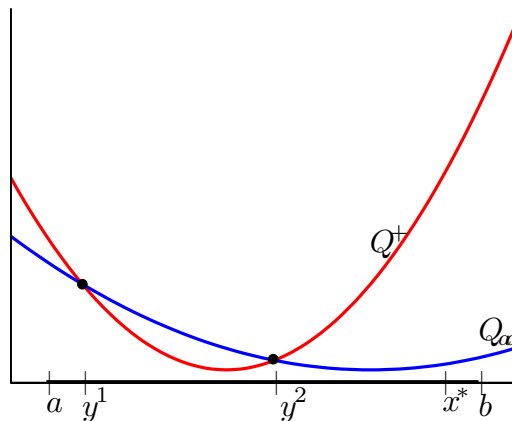


Figura 3.3: Escolha entre dois modelos gerados pelo BOBYQA na mesma iteração

3.6 Outros procedimentos

O Algoritmo BOBYQA trabalha com duas regiões. Uma de raio Δ_k que é a região em que confiamos no modelo, e outra de raio ρ_k relacionada ao conjunto interpolador. Nesta seção discutimos os procedimentos de atualização dos raios destas duas regiões e o critério de parada que o algoritmo utiliza.

3.6.1 Atualizações das regiões

Descrevemos a seguir como os parâmetros Δ_k e ρ_k são atualizados na k -ésima iteração do algoritmo. A escolha destes parâmetros é feita de forma diferente caso a iteração seja de região de confiança ou alternativa.

Atualização do parâmetro Δ_k em uma iteração de região de confiança

Considere x^k o iterando atual e d^k o passo obtido na iteração k . A escolha do raio Δ_{k+1} para a próxima iteração depende da razão entre a redução real da função objetivo f e a redução predita pelo modelo Q_k , ou seja,

$$r = \frac{f(x^k) - f(x^k + d^k)}{Q_k(x^k) - Q_k(x^k + d^k)}. \quad (3.35)$$

A partir desta razão r , o raio da região de confiança é atualizado por

$$\bar{\Delta}_{k+1} = \begin{cases} \min \{0.5\Delta_k, \|d^k\|\}, & \text{se } r \leq 0.1, \\ \max \{0.5\Delta_k, \|d^k\|\}, & \text{se } 0.1 < r \leq 0.7, \\ \max \{0.5\Delta_k, 2\|d^k\|\}, & \text{se } r > 0.7. \end{cases} \quad (3.36)$$

Se $\bar{\Delta}_{k+1} \leq 1.5\rho_k$, então $\Delta_{k+1} = \rho_k$. Senão $\Delta_{k+1} = \bar{\Delta}_{k+1}$.

Atualização do parâmetro Δ_k em uma iteração alternativa

Se a iteração corrente é do tipo alternativa, então o raio da região de confiança Δ_k é reduzido temporariamente para

$$\Delta_k = \max \{ \min \{0.1\delta_k, \Delta_k\}, \rho_k \}, \quad (3.37)$$

onde $\delta_k = \max_{1 \leq i \leq m} \|y^i - x^k\|$. Ao final da iteração alternativa o raio é atualizado por

$$\Delta_{k+1} = \min \{0.1\Delta_k, 0.5\delta_k\}. \quad (3.38)$$

Se $\Delta_{k+1} \leq 1.5\rho_k$, então $\Delta_{k+1} = \rho_k$.

Atualização do parâmetro ρ_k

Considere $\rho_{end} > 0$ o parâmetro de entrada do algoritmo. A atualização de ρ_k ocorre simultaneamente com a redução de Δ_k pela fórmula

$$\begin{aligned} \bar{\Delta}_{k+1} &= 0.5\rho_k. \\ \rho_{k+1} &= \begin{cases} \rho_{end}, & \text{se } \rho_k \leq 16\rho_{end}, \\ \sqrt{\rho_k \rho_{end}}, & \text{se } 16\rho_{end} < \rho_k \leq 250\rho_{end}, \\ 0.1\rho_k, & \text{se } \rho_k > 250\rho_{end}. \end{cases} \\ \Delta_{k+1} &= \max \{ \bar{\Delta}_{k+1}, \rho_{k+1} \}. \end{aligned} \quad (3.39)$$

3.6.2 Critério de parada

A sequência $\{\rho_k\}$ inicia com o parâmetro de entrada $\rho_{beg} > 0$ e é atualizada pela fórmula (3.39). O algoritmo para quando na k -ésima iteração $\rho_k = \rho_{end}$.

A seguir apresentamos os testes que permitem atualizar ρ_k no algoritmo BOBYQA [54]. A grosso modo, quando nas três últimas iterações o passo é pequeno e não houve mudanças significativas nos modelos, o raio da região ρ_k é atualizado pela fórmula (3.39). Caso contrário, é calculado o passo por uma iteração alternativa. No entanto, os testes em BOBYQA são mais elaborados.

Mais especificamente. Considere

$$\eta = \max \{|f(x^\ell + d^\ell) - Q_\ell(x^\ell + d^\ell)| : \ell \in \{k-3, k-2, k-1\}\}.$$

Dizemos que η é “utilizável” na iteração k quando $\|d^\ell\| \leq \rho_k$, $k-3 \leq \ell \leq k-1$.

Defina S o conjunto de direções de buscas obtidas pelo método do gradiente conjugado truncado na k -ésima iteração de região de confiança (subrotina TRSBOX) e que não são limitadas pela caixa \mathbb{A} . Tome para $i = 1, 2, \dots, n$,

$$\mathcal{V}^i = \begin{cases} \rho_k e^i & \text{se } [x^k + d^k]_i = a_i \\ -\rho_k e^i & \text{se } [x^k + d^k]_i = b_i, \end{cases}$$

onde a e b definem a caixa \mathbb{A} . Em [54, pág. 31] existe a justificativa para os testes que apresentamos a seguir.

Suponha que para todo $s \in S$ valem as desigualdades

$$\eta \leq \frac{1}{8} \rho_k^2 \|s\|^{-2} s^T \nabla^2 Q_k s \quad \text{e} \quad \|s\|^{-2} s^T \nabla^2 Q_k s < 0 \quad (3.40)$$

e que para todo $v \in \mathcal{V} = \{\mathcal{V}^i; i = 1, 2, \dots, n\}$,

$$\eta \leq \max \left\{ v^T \nabla Q_k(x^k + d^k), v^T \nabla Q_k(x^k + d^k) + \frac{1}{2} v^T \nabla^2 Q_k v \right\}. \quad (3.41)$$

Suponha inicialmente que $\|d^k\| < 0.5\rho_k$. Neste caso, se

$$\eta \text{ é utilizável e valem (3.40) e (3.41),} \quad (3.42)$$

ou $\delta_k \leq 10\rho_k$, então ρ_k é atualizado pela fórmula (3.39) ou o término ocorre no caso em que $\rho_k = \rho_{end}$. Caso contrário é executado uma iteração alternativa.

Finalmente, o último critério que permite atualizar ρ_k pela fórmula (3.39) é quando $\|d^k\| \geq 0.5\rho_k$ e valem as desigualdades

$$\|d^k\| \leq \rho_k, \quad r \leq 0 \quad \text{e} \quad \Delta_{k+1} \leq \rho_k, \quad (3.43)$$

onde r é a razão entre a redução real e predita pelo modelo dada em (3.35).

3.7 O Algoritmo

Apresentamos como trabalha o BOBYQA a partir dos quadros dados no fluxograma a seguir. O fluxograma foi construído com um estudo na implementação do algoritmo. Omitimos o índice k dos raios ρ_k , Δ_k , do modelo Q_k e do conjunto interpolador Y_k . Esta apresentação tem caráter de estabelecer as principais características do algoritmo.

O algoritmo inicia no **quadro 1** onde recebe $x^1 = y^1 = x^b \in \mathbb{R}^n$, os parâmetros $\rho_{end} > \rho_{beg} > 0$, o número m de pontos de interpolação, a e b os limites inferior e superior da caixa \mathbb{A} , respectivamente. Atribui $\Delta = \rho_{beg}$, $\rho = \rho_{beg}$ e executa o **quadro 2**. Neste momento é construído o primeiro conjunto interpolador Y e o primeiro modelo utilizando as expressões (3.2), (3.6) e o Teorema 3.5, respectivamente. No **quadro 4** é obtido o passo por uma iteração de região de confiança, que consiste em calcular por uma versão do método Gradiente Conjugado Truncado o passo $d^k \in \mathbb{R}^n$ a partir do iterando atual x^k , solução do problema

$$\begin{aligned} & \text{minimizar} && Q_k(x) \\ & \text{sujeita a} && \|x - x^k\| \leq \Delta_k \\ & && a \leq x \leq b. \end{aligned}$$

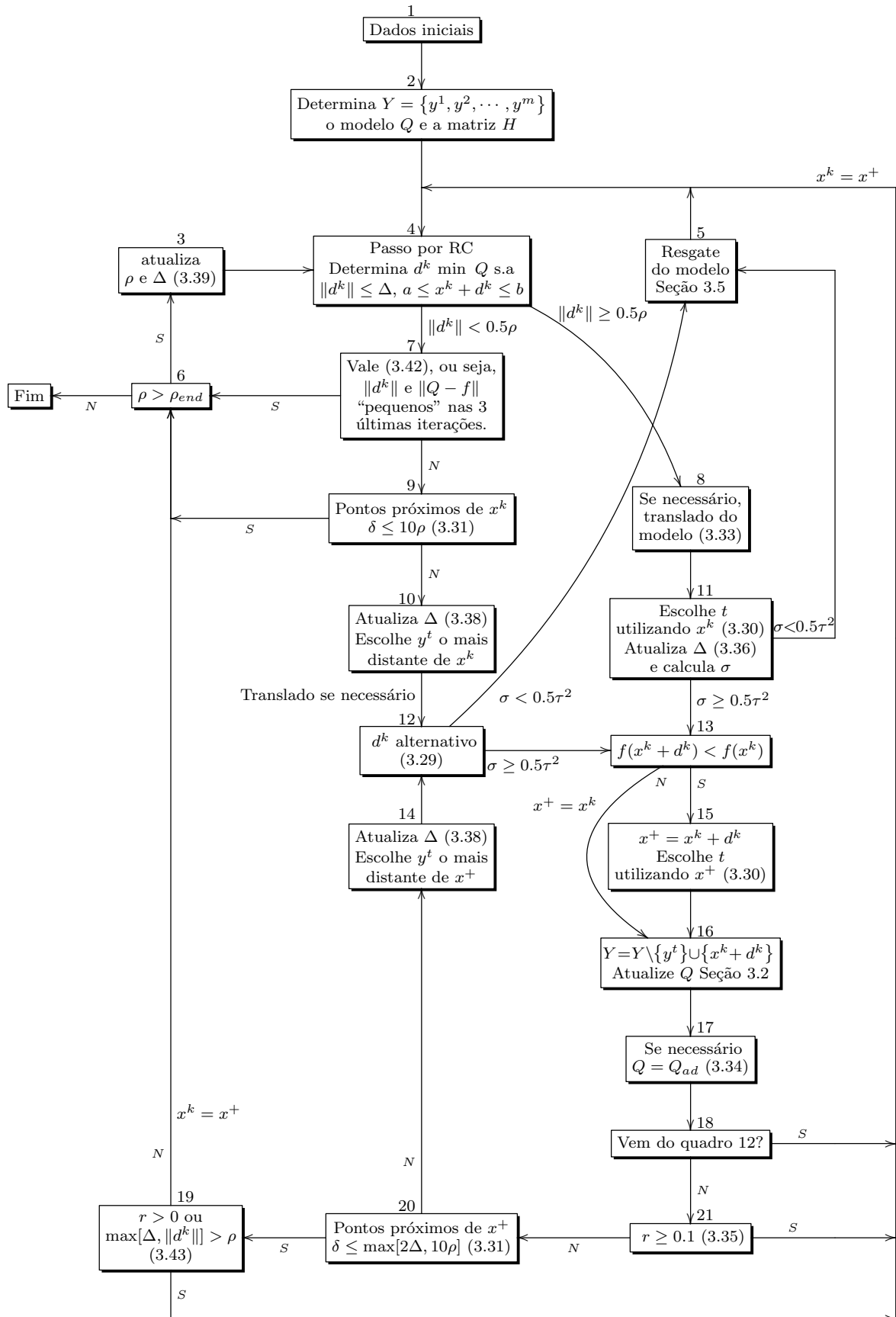
Se o passo $\|d^k\|$ não é pequeno, ou seja, $\|d^k\| \geq 0.5\rho_k$, então no **quadro 8** o algoritmo verifica se há necessidade em transladar a origem x^b para o melhor ponto x^k atual. A translação é executada quando x^b está longe de x^k e o teste é o da expressão (3.33). Em seguida é executado o **quadro 11** onde escolhemos o índice t do ponto y^t que deve deixar o conjunto interpolador levando em consideração o ponto mais distante do atual x^k e o aumento no denominador σ , utilizando a expressão (3.30). Neste quadro também é atualizada a região de confiança com (3.36). Para o t escolhido neste quadro, o valor do denominador $\sigma(t) = \alpha\beta + \tau^2$ indica se há necessidade de resgatar o modelo, o que é feito no **quadro 5** com o procedimento descrito na Seção 3.5 ou é verificado se houve decréscimo na função objetivo f no **quadro 13**. O decréscimo é verificado tanto para um passo $\|d^k\| \geq 0.5\rho_k$ quanto para um passo alternativo obtido no **quadro 12**. Se não houve decréscimo no valor da função f , então $x^+ = x^{k+1} = x^k$. Caso contrário $x^+ = x^k + d^k$ e é executado o **quadro 15**, onde calculamos novamente o ponto y^t que deixa Y considerando a fórmula (3.30) em relação ao novo ponto x^+ . No **quadro 16**, a atualização do conjunto interpolador $Y^+ = Y \setminus \{y^t\} \cup \{x^k + d^k\}$ afirma que houve mudança no conjunto. Quer dizer que, mesmo o ponto não produzindo decréscimo na f , o fato do passo não ter sido pequeno faz com que o ponto entre no conjunto interpolador. E o outro caso, quando vem do quadro 12, o ponto sempre entra no conjunto interpolador, pois em 12 é feito o cálculo alternativo do passo, que tem a finalidade de melhorar a geometria do conjunto interpolador como descrito na Seção 3.3. No **quadro 16**, como houve mudança no conjunto interpolador, o modelo é atualizado com o procedimento da Subseção 3.2, que atualiza a matriz inversa H com a fórmula (3.21). No **quadro 17** é efetuado o teste (3.34) para verificar se o atual modelo deve ser substituído pelo modelo adicional, com o objetivo de trabalhar com um modelo que possui menor valor na norma da Hessiana. O **quadro 18** indica que depois de uma iteração alternativa, a próxima iteração é sempre de região de confiança. No **quadro 21** é feito o teste do decréscimo suficiente pela razão r entre a redução real e predita, com a fórmula (3.35). Se houve decréscimo suficiente, ou seja, $r \geq 0.1$,

o algoritmo inicia outra iteração de região de confiança. Caso contrário, executa o **quadro 20** que verifica se os pontos interpoladores estão próximos de x^+ pela fórmula (3.31). Neste quadro, $\delta = \max_{1 \leq i \leq m} \|y^i - x^k\|$. Com base no teste efetuado no próprio quadro 20, se os pontos interpoladores não estão próximos de x^+ , então é executado o **quadro 14**, que determina quem deixa o conjunto interpolador antes de calcular um passo alternativo pelo quadro 12. Tanto no quadro 14 como no quadro 10, o raio da região de confiança é atualizado pela fórmula (3.38), que reduz o raio da região de confiança para a próxima iteração. Na execução do **quadro 12**, em que determinamos um passo d^k alternativo, o raio da região de confiança é temporariamente modificado pela expressão (3.37). Se com base no quadro 20 os pontos estão próximos de x^+ , então é executado o **quadro 19**, que é outro critério para reduzir o raio ρ_k como indicado na expressão (3.43).

Se o passo d^k no quadro 4 é pequeno, ou seja, $\|d^k\| < 0.5\rho_k$, então o algoritmo evita uma avaliação da função objetivo e executa o **quadro 7**, que tem importância para caracterizar se o algoritmo deve reduzir ρ_k como descrito na Subseção 3.6.2. Se o quadro 7 for satisfeito, então as duas regiões ρ e Δ são atualizadas no **quadro 3** pela fórmula (3.39) ou o algoritmo atinge o fim. Se o quadro 7 não for satisfeito, então no **quadro 9** é verificada a proximidade dos pontos de interpolação com o atual x^k e comparado com o valor $10\rho_k$. Se os pontos de Y estão próximos de x^k , então o algoritmo atualiza as duas regiões com a fórmula (3.39), caso contrário o algoritmo se prepara no **quadro 10** para um passo alternativo executado no quadro 12, verificando antes se há necessidade em fazer a translação.

O algoritmo para quando ρ_k atinge o parâmetro ρ_{end} .

Fluxograma do Algoritmo BOBYQA



Sequência nos tipos de iterações

Apresentamos quais decisões são tomadas para escolher qual a sequência de iterações que o algoritmo executa. São apenas dois tipos de iterações, e a decisão entre região de confiança e alternativa utiliza os critérios a seguir:

- A primeira iteração do algoritmo e toda iteração após uma atualização em ρ_k é do tipo região de confiança.
- Depois de uma iteração alternativa, a próxima é de região de confiança.
- Se foi feito um resgate do modelo, então a próxima iteração é de região de confiança.
- Quando $\|d^k\| \geq 0.5\rho_k$ e houve decréscimo suficiente na função objetivo, ou seja, a razão r dada em (3.35) é maior do que 0.1, então um novo passo é obtido por uma iteração de região de confiança. Se não houve decréscimo suficiente na função objetivo, então a próxima iteração pode ser de região de confiança ou alternativa dependendo dos testes feitos nos quadros 19 e 20 no fluxograma do algoritmo.
- Quando $\|d^k\| < 0.5\rho_k$, então é executada uma iteração alternativa, ou ρ_k é atualizado, o que obriga o algoritmo (primeiro item) a executar uma iteração de região de confiança. Estas decisões são tomadas nos quadros 7 e 9 no fluxograma de acordo com os testes na Subseção 3.6.2.
- O algoritmo para quando o raio da região ρ_k for reduzido ao parâmetro inicial $\rho_{end} > 0$.

A estrutura do Algoritmo BOBYQA não permite utilizá-lo para resolver problemas com restrições gerais, uma vez que o algoritmo não se resume a resolver subproblemas para obter um candidato a novo iterando. No próximo capítulo apresentamos nossa proposta de algoritmo para resolver problemas com restrições gerais, que aproveita o procedimento descrito em BOBYQA para construção e atualização dos modelos.

Capítulo 4

Algoritmo implementado

Apresentamos neste capítulo outra contribuição desta tese. Propomos um algoritmo de região de confiança sem derivadas para problemas de otimização com restrições gerais. O algoritmo é baseado em interpolação polinomial e tem forte influência do Algoritmo BOBYQA [54] nas construções e atualizações dos modelos. Na primeira seção abordamos os aspectos gerais do algoritmo. Na segunda seção os detalhes de como o modelo e o conjunto interpolador são construídos e atualizados. O Algoritmo ALGENCAN é utilizado nos subproblemas e sua estrutura com resultados de convergência é apresentada na terceira seção. Na quarta seção descrevemos o algoritmo proposto, que não possui resultados de convergência mas onde estabelecemos sua boa definição. A implementação foi feita em Fortran 77 e experimentos numéricos são discutidos no próximo capítulo.

4.1 Ideia geral

A técnica para atualização dos modelos quadráticos é da norma mínima da diferença entre as Hessianas do novo e do modelo corrente. Não podemos relacionar esta proposta de algoritmo prático que iremos apresentar com o Algoritmo 1.1 do Capítulo 1, pois não temos garantias de que os modelos determinados pela norma mínima de Frobenius cumprem a Hipótese H4 sobre a qualidade do modelo. A principal concepção da otimização sem derivadas é de que o custo em avaliar a função objetivo é alto. Assim, utilizar a técnica apresentada na Seção 2.1 para construir modelos que garantem a Hipótese H4 não é interessante, pois cada vez que o modelo é construído, é exigido avaliar a função objetivo em todos os pontos do conjunto interpolador como discutido em [22]. Nossa proposta trabalha, em geral, com apenas uma avaliação de função objetivo cada vez que obtém um novo modelo.

Consideramos o problema de otimização

$$\begin{aligned} & \text{minimizar} && f(x) \\ & \text{sujeita a} && x \in \Omega_1 \end{aligned} \tag{4.1}$$

onde $f : \mathbb{R}^n \rightarrow \mathbb{R}$ não possui derivadas disponíveis e $\Omega_1 \subset \mathbb{R}^n$ é qualquer conjunto viável não vazio e com derivadas disponíveis. A definição da função objetivo em todo espaço \mathbb{R}^n é necessária, pois, apesar do algoritmo proposto caminhar sobre pontos viáveis, a construção do modelo ocorre em vizinhanças de pontos viáveis.

O algoritmo inicia em um ponto $x^1 \in \Omega_1$ e consiste em uma sequência de resoluções aproximadas de subproblemas quadráticos restritos. O modelo quadrático na k -ésima iteração do algoritmo tem a forma

$$Q_k(x) = c_k + (g^k)^T(x - x^b) + \frac{1}{2}(x - x^b)^T \nabla^2 Q_k(x - x^b), \quad (4.2)$$

onde $c_k \in \mathbb{R}$, $g^k \in \mathbb{R}^n$, $\nabla^2 Q_k \in \mathbb{R}^{n \times n}$ e x^b é um vetor que recebe na primeira iteração o ponto inicial x^1 e é modificado toda vez que x^b está longe do iterando atual. A existência do vetor x^b é para evitar acúmulos de erros inerentes ao processo adotado para atualizar os modelos, como discutido na Seção 3.5. O modelo quadrático (4.2) tem $(n+1)(n+2)/2$ parâmetros livres. São n parâmetros para o gradiente, 1 para a constante e $(1+n)n/2$ para a Hessiana. A técnica que utilizamos permite trabalhar com liberdade nestes parâmetros. Dados m de pontos de interpolação,

$$2n + 1 \leq m \leq \frac{1}{2}(n+1)(n+2)$$

e o conjunto interpolador

$$Y_k = \{y^1, y^2, \dots, y^m\},$$

o modelo (4.2) deve satisfazer a condição de interpolação

$$Q_k(y^i) = f(y^i),$$

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

A liberdade nos parâmetros do modelo foi considerada como sugere Powell em [54] obtendo, na k -ésima iteração, Q_{k+1} a partir da norma de Frobenius da diferença entre as Hessianas do novo e do modelo corrente, ou seja, resolvendo o problema

$$\begin{aligned} &\text{minimizar} \quad \|\nabla^2 Q_{k+1} - \nabla^2 Q_k\|_F \\ &\text{sujeita a} \quad f(Y_{k+1}) = Q_{k+1}(Y_{k+1}). \end{aligned} \quad (4.3)$$

Detalhes são descritos na Seção 3.2.

As restrições do problema original e a região de confiança de raio $\Delta_k > 0$ são consideradas em cada subproblema, que toma a forma

$$\begin{aligned} &\text{minimizar} \quad Q_k(x) \\ &\text{sujeita a} \quad x \in \Omega_1 \\ &\quad \quad \quad \|x - x^k\|_\infty \leq \Delta_k, \end{aligned} \quad (4.4)$$

onde x^k é o iterando atual no passo k e Q_k é dado em (4.2).

Aplicar ALGENCAN ao problema original não é possível pois as derivadas da função objetivo não estão disponíveis e não podem ser aproximadas. No entanto, as derivadas das restrições e do modelo Q_k estão disponíveis e ALGENCAN pode, assim, ser aplicado aos subproblemas (4.4).

4.2 Conjunto interpolador e modelo

Em cada iteração o modelo (4.2) é obtido por interpolação polinomial. No algoritmo ele pode ser construído ou atualizado. A construção ocorre na primeira iteração e sempre que não há progresso no algoritmo, enquanto nas demais iterações o modelo é atualizado a partir das informações do modelo anterior.

4.2.1 Construção

Para a construção do modelo, a escolha dos primeiros $2n+1$ pontos do conjunto interpolador é feita de forma similar à apresentada na Seção 3.1. A partir de um dado ponto inicial $y^1 \in \Omega_1$ e do raio Δ da região de confiança, para $i = 1, \dots, n$ são definidos

$$y^{i+1} = y^1 + \Delta e^i \quad \text{e} \quad y^{n+i+1} = y^1 - \Delta e^i. \quad (4.5)$$

Note que os pontos do conjunto interpolador podem ser inviáveis, enquanto que no Capítulo 3 eles eram viáveis em relação à restrição de caixa.

Os demais $m - 2n - 1$ pontos de interpolação são construídos de modo análogo ao apresentado na expressão (3.6), mas sem o reordenamento dos pontos. O reordenamento analisa a tendência de decréscimo da função objetivo, no entanto, para otimização restrita esta técnica é menos atraente, pois ela considera apenas a otimalidade e não a viabilidade. Os pontos de interpolação para $j \in \{2n + 2, \dots, m\}$ são obtidos por

$$y^j = y^{u(j)+1} + y^{v(j)+1} - y^1, \quad (4.6)$$

onde $u(j)$ e $v(j)$ são dados por (3.4) e (3.5), respectivamente.

A Figura 4.1 exemplifica no plano os seis pontos obtidos por (4.5) e (4.6). Somente há exigências de que $y^1 \in \Omega_1$. Os demais pontos interpoladores podem ficar fora do conjunto viável.

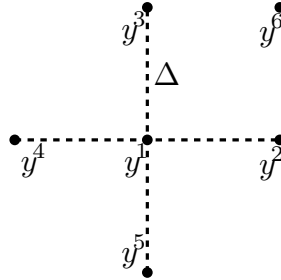


Figura 4.1: $(n + 1)(n + 2)/2$ pontos de interpolação para $n = 2$

O algoritmo inicia com um conjunto interpolador construído pelas expressões (4.5) e (4.6), e com um modelo quadrático construído com pouco esforço computacional em virtude da disposição geométrica deste conjunto interpolador. Enquanto é obtido um decréscimo suficiente na função objetivo, o conjunto interpolador é alterado em apenas um ponto de uma iteração para outra. Quando o decréscimo suficiente não é atingido, em geral, todo o conjunto interpolador pode ser modificado pelas expressões (4.5) e (4.6), tomando o ponto y^1 como sendo o melhor ponto obtido até o momento.

O resultado a seguir estabelece uma fórmula fechada para construção do modelo quadrático, a partir do conjunto interpolador gerado pelas expressões (4.5) e (4.6). É análogo ao Teorema 3.2, mas sem o reordenamento dos pontos no conjunto interpolador.

Teorema 4.1 Construção do modelo

Considere $f : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função e $m \geq 2n + 1$ pontos de interpolação. Considere $y^1 \in \Omega_1$, $\Delta > 0$ o raio da região de confiança, $Y_k = \{y^1, y^2, \dots, y^m\}$ o conjunto interpolador obtido por (4.5) e (4.6) e os parâmetros do modelo quadrático Q_k (4.2), $x^b = y^1$, dados por

1. $Q_k(y^1) = c_k = f_1$;
2. $[\nabla Q_k(y^1)]_i = g^i = \frac{1}{2\Delta} (f_{i+1} - f_{i+1+n}), \quad i = 1, 2, \dots, n$;
3. $[\nabla^2 Q_k]_{i,i} = \frac{1}{\Delta^2} (f_{i+1} + f_{i+1+n} - 2f_1), \quad i = 1, 2, \dots, n$;
4. $[\nabla^2 Q_k]_{u(j),v(j)} = \frac{1}{\Delta^2} (f_j - f_{u(j)+1} - f_{v(j)+1} + f_1), \quad j = 2n + 2, \dots, m$;
5. $[\nabla^2 Q_k]_{u(j),v(j)} = 0, \quad j = m + 1, m + 2, \dots, (n + 1)(n + 2)/2$.

Então Q_k satisfaz a condição de interpolação $Q_k(Y_k) = f(Y_k)$.

Demonstração. Análoga à do Teorema 3.2. □

Pelo Teorema 4.1 a matriz Hessiana $\nabla^2 Q_k$ do modelo quadrático é uma matriz com estrutura de banda. Isto é, pelo item 3, são gerados os elementos da diagonal da matriz Hessiana $\nabla^2 Q_k$ e em seguida cada uma das demais diagonais de acordo com o número de pontos do conjunto interpolador. Note que com $m = 2n + 1$ pontos de interpolação, o modelo quadrático tem determinados os parâmetros do gradiente, do termo constante e da diagonal principal da matriz Hessiana. Os demais parâmetros são nulos.

4.2.2 Atualização

Em toda iteração que o algoritmo fornece uma redução suficiente da função objetivo, o modelo para a nova iteração é atualizado a partir das informações do modelo corrente. Dados o conjunto interpolador Y_k e o modelo quadrático Q_k , o Teorema 3.3 fornece a inversa H_k associada ao sistema (3.16) em relação a Y_k . O novo modelo quadrático Q_{k+1} é obtido a partir da inversa H_{k+1} que é atualizada através do Teorema 3.5, utilizando a informação da inversa H_k e do novo conjunto interpolador Y_{k+1} .

Discutimos a seguir como o conjunto interpolador é atualizado.

Escolha do ponto que deixa o conjunto interpolador

Suponha que na k -ésima iteração conhecemos o ponto x^+ que deve entrar no conjunto interpolador e que o iterando atual ocupa a posição \bar{t} em Y_k , ou seja, $x^k = y^{\bar{t}}$. A posição t do ponto y^t que deixa Y_k é dada por

$$t = \arg \max_{j \in J} \{ \sigma(j) \}, \quad (4.7)$$

onde $J = \{1, 2, \dots, m\} \setminus \{\bar{t}\}$ e

$$\sigma(j) = \left\{ [H_k]_{j,j} \left(\frac{1}{2} \|x^+ - x^b\|^4 - w^T [H_k] w \right) + ((e^j)^T [H_k] w)^2 \right\},$$

onde w é definido no Teorema 3.5. O novo iterando toma a posição de um ponto do conjunto interpolador, diferente do iterando atual. Sempre que $\sigma(t) \neq 0$, a Proposição 3.8 garante que o novo ponto assume uma posição geométrica que mantém o denominador σ diferente de zero na fórmula de atualização da matriz inversa H^+ dada no

Teorema 3.5 e o Teorema 3.7 garante que o volume da envoltória convexa obtida pelo conjunto Y_{k+1} é maximizado.

O exemplo a seguir ilustra a escolha feita por (4.7), quando a restrição Ω_1 no problema original (4.1) é linear.

Exemplo 4.2 Dado o problema de minimização com a função objetivo

$$f(x_1, x_2)^T = (x_1 - 1)^2 + (x_2 - 1)^2$$

e a restrição de igualdade $x_2 = 1$, considere o ponto inicial $x^1 = y^1 = (-1.5, 1)^T$ e $\Delta = 0.5$. O conjunto interpolador Y_1 obtido por (4.5) e (4.6) com $m = 6$ pontos é mostrado no primeiro gráfico da Figura 4.2. Esta figura mostra os conjuntos interpoladores Y_1, Y_2, Y_3 e Y_4 da esquerda para direita e de cima para baixo.

Suponha que o algoritmo gera os iterandos $x^2 = (-1, 1)^T$, $x^3 = (0, 1)^T$ e $x^4 = (1, 1)^T$. Note que $x^* = x^4$ é o minimizador do problema. O conjunto Y_2 é igual a Y_1 pois o iterando x^2 é um ponto do conjunto interpolador Y_1 . Os dois últimos gráficos mostram como os conjuntos interpoladores Y_3 e Y_4 são atualizados a partir de (4.7). Observamos que a melhor escolha para o ponto x^3 assumir no conjunto interpolador Y_3 é tomando o lugar de y^1 , uma vez que não pode assumir a posição do iterando anterior $x^2 = y^2$ e deve maximizar o volume da envoltória convexa determinada por Y_3 . Interpretação análoga é feita em relação à posição que x^4 assume no conjunto interpolador Y_4 . A busca pela não colinearidade dos pontos interpoladores neste exemplo, é uma necessidade para garantir unicidade na atualização dos modelos. Note que 6 pontos colineares no plano não garantem a unicidade do polinômio quadrático interpolador.

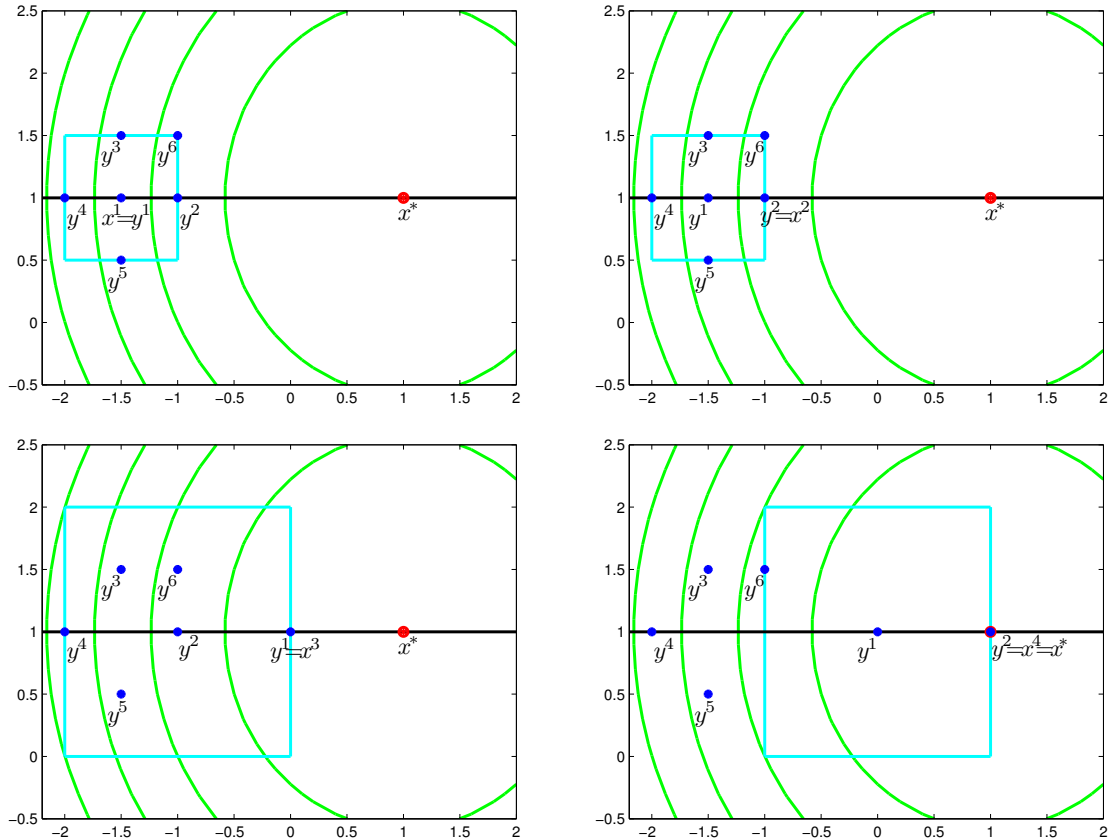


Figura 4.2: Configurações para os 4 primeiros conjuntos interpoladores

4.3 Lagrangiano Aumentado

Neste trabalho, os subproblemas da forma (4.4) são resolvidos pelo Algoritmo ALGENCAN [70], que é uma implementação do método Lagrangiano Aumentado proposto em [2, 3]. Escolhemos ALGENCAN pelos bons resultados de convergência do método Lagrangiano Aumentado e por seu desempenho computacional [8]. Nesta seção mantemos a notação de [2], principal referência adotada, que tendo em vista o contexto não causará confusão.

Considere o problema de programação não linear

$$\begin{aligned} & \text{minimizar} && f(x) \\ & \text{sujeita a} && h_1(x) = 0 \\ & && g_1(x) \leq 0 \\ & && h_2(x) = 0 \\ & && g_2(x) \leq 0, \end{aligned} \tag{4.8}$$

onde $f: \mathbb{R}^n \rightarrow \mathbb{R}$, $h_1: \mathbb{R}^n \rightarrow \mathbb{R}^{m_1}$, $h_2: \mathbb{R}^n \rightarrow \mathbb{R}^{m_2}$, $g_1: \mathbb{R}^n \rightarrow \mathbb{R}^{p_1}$ e $g_2: \mathbb{R}^n \rightarrow \mathbb{R}^{p_2}$ são funções com derivadas primeiras contínuas em um domínio aberto suficientemente grande. Definimos os conjuntos

$$\begin{aligned} \Omega_1 &= \{x \in \mathbb{R}^n \mid h_1(x) = 0, g_1(x) \leq 0\}, \\ \Omega_2 &= \{x \in \mathbb{R}^n \mid h_2(x) = 0, g_2(x) \leq 0\}. \end{aligned}$$

Consideramos que Ω_2 contém as restrições fáceis, ou seja, aquelas para as quais há um algoritmo que resolve eficientemente um problema de otimização sujeito a estas restrições. Por outro lado Ω_1 contém as restrições difíceis que serão incorporadas à função objetivo pela função Lagrangiano Aumentado que definimos a seguir.

Definição 4.3 Função Lagrangiano Aumentado

A função Lagrangiano Aumentado com respeito a Ω_1 é

$$L: \mathbb{R}^n \times \mathbb{R}^{m_1} \times \mathbb{R}_+^{p_1} \times \mathbb{R}_{++} \rightarrow \mathbb{R}$$

definida por

$$L(x, \lambda, \mu, \rho) = f(x) + \frac{\rho}{2} \sum_{i=1}^{m_1} \left([h_1(x)]_i + \frac{\lambda_i}{\rho} \right)^2 + \frac{\rho}{2} \sum_{i=1}^{p_1} \left(\max \left\{ 0, [g_1(x)]_i + \frac{\mu_i}{\rho} \right\} \right)^2.$$

Os argumentos λ e μ da função Lagrangiano Aumentado são denominados de multiplicadores de Lagrange.

4.3.1 Descrição do algoritmo

O Algoritmo Lagrangiano Aumentado apresentado em [2] consiste na resolução de uma sequência de subproblemas da forma

$$\begin{aligned} & \text{minimizar} && L(x, \lambda, \mu, \rho) \\ & \text{sujeita a} && x \in \Omega_2. \end{aligned} \tag{4.9}$$

Ressaltamos que as restrições em Ω_1 são incorporadas na função *Lagrangiano Aumentado*, enquanto que as restrições em Ω_2 ficam nos subproblemas. Cada minimização aproximada da função L é denominada de *iteração externa*.

A medida de inviabilidade da restrição $h_1(x) = 0$ é representada por $\|h_1(x)\|$, enquanto que a medida de inviabilidade e complementaridade da desigualdade $g_1(x) \leq 0$ é dada por $|\sigma(x, \mu, \rho)|_i = |\max\{[g_1(x)]_i, -\mu_i/\rho\}|$, $i = 1, \dots, p_1$.

Após a resolução de cada subproblema, são atualizados as estimativas de Lagrange λ e μ e o parâmetro de penalidade ρ que definirão a função de Lagrange para o próximo subproblema.

Consideramos soluções aproximadas dos subproblemas aquelas que satisfazem as condições (4.10) a (4.13) dadas a seguir. Estas expressões são relaxamentos das condições KKT para o problema (4.9). As aproximações de primeira ordem dos multiplicadores são calculadas na Etapa 3 do algoritmo. As estimativas dos multiplicadores de Lagrange são denotadas por λ_k e μ_k , enquanto suas salvaguardadas por $\bar{\lambda}_k$ e $\bar{\mu}_k$. Na Etapa 4 é atualizado o parâmetro de penalidade de acordo com o progresso em termos de viabilidade e complementaridade. Para análise teórica, o algoritmo é apresentado sem critério de parada, mas na prática é necessário estabelecer critérios de parada para que a execução termine.

Algoritmo 4.4 *Lagrangiano Aumentado*

Dados $x^0 \in \mathbb{R}^n$ e os parâmetros $\tau \in [0, 1)$, $\gamma > 1$, $\rho_1 > 0$, $-\infty < [\bar{\lambda}_{min}]_i \leq [\bar{\lambda}_{max}]_i < \infty$ para $i = 1, \dots, m_1$, $0 \leq [\bar{\mu}_{max}]_i < \infty$ para $i = 1, \dots, p_1$, $[\bar{\lambda}_1]_i \in [[\bar{\lambda}_{min}]_i, [\bar{\lambda}_{max}]_i]$ para $i = 1, \dots, m_1$, $[\bar{\mu}_1]_i \in [0, [\bar{\mu}_{max}]_i]$ para $i = 1, \dots, p_1$. Tome a sequência $\{\varepsilon_k\} \subset \mathbb{R}_+$ de parâmetros de tolerância tal que $\lim_{k \rightarrow \infty} \varepsilon_k = 0$.

Etapa 1. (Inicialização)

Faça $k = 1$. Para $i = 1, \dots, p_1$, calcule $[\sigma_0]_i = \max\{0, [g_1(x_0)]_i\}$.

Etapa 2. (Solução do Subproblema)

Calcule (se possível) $x^k \in \mathbb{R}^n$ tal que exista $v^k \in \mathbb{R}^{m_2}$, $u^k \in \mathbb{R}^{p_2}$ satisfazendo

$$\|\nabla L(x^k, \bar{\lambda}_k, \bar{\mu}_k, \rho_k) + \sum_{i=1}^{m_2} [v^k]_i \nabla [h_2(x^k)]_i + \sum_{i=1}^{p_2} [u^k]_i \nabla [g_2(x^k)]_i\| \leq \varepsilon_{k,1}, \quad (4.10)$$

$$[u^k]_i \geq 0 \text{ e } [g_2(x^k)]_i \leq \varepsilon_{k,2}, \quad \forall i = 1, \dots, p_2, \quad (4.11)$$

$$[g_2(x^k)]_i < -\varepsilon_{k,2} \Rightarrow [u^k]_i = 0, \quad \forall i = 1, \dots, p_2, \quad (4.12)$$

$$\|h_2(x^k)\| \leq \varepsilon_{k,3}, \quad (4.13)$$

onde $\varepsilon_{k,1}, \varepsilon_{k,2}, \varepsilon_{k,3} \geq 0$ são tais que $\max\{\varepsilon_{k,1}, \varepsilon_{k,2}, \varepsilon_{k,3}\} \leq \varepsilon_k$. Se não for possível encontrar x^k satisfazendo (4.10) a (4.13), pare a execução do algoritmo.

Etapa 3. (Estimativas para os multiplicadores)

Para todo $i = 1, \dots, m_1$, calcule

$$[\lambda_{k+1}]_i = [\bar{\lambda}_k]_i + \rho_k [h_i(x^k)]_i \text{ e}$$

$$[\bar{\lambda}_{k+1}]_i = \min\{\max\{[\lambda_{k+1}]_i, [\bar{\lambda}_{min}]_i\}, [\bar{\lambda}_{max}]_i\},$$

Para todo $i = 1, \dots, p_1$, calcule

$$\begin{aligned} [\mu_{k+1}]_i &= \max \{0, [\bar{\mu}_k]_i + \rho_k [g_1(x^k)]_i\}, \\ [\sigma_k]_i &= \max \left\{ [g_1(x^k)]_i, -\frac{[\bar{\mu}_k]_i}{\rho_k} \right\} \text{ e} \\ [\bar{\mu}_{k+1}]_i &= \min \{[\mu_{k+1}]_i, [\bar{\mu}_{max}]_i\}. \end{aligned}$$

Etapa 4. (Atualização do parâmetro de penalidade)

Se $\max \{\|h_1(x^k)\|_\infty, \|\sigma_k\|_\infty\} \leq \tau \max \{\|h_1(x^{k-1})\|_\infty, \|\sigma_{k-1}\|_\infty\}$, então defina $\rho_{k+1} = \rho_k$. Caso contrário, $\rho_{k+1} = \gamma \rho_k$.

Etapa 5. (Início de uma nova iteração externa)

Faça $k = k + 1$. Vá para a Etapa 2.

4.3.2 Convergência

A convergência deste algoritmo é discutida em [2]. A prova é baseada na condição CPLD, cujos conceitos são apresentados a seguir.

Definição 4.5 [3] (CPLD)

Considere $x^* \in \Omega_2$ e $I = \{i \in \{1, \dots, p_2\} \mid [g_2(x^*)]_i = 0\}$. Dizemos que x^* satisfaz a condição CPLD se a existência de $J_1 \subset \{1, \dots, m_2\}$, $J_2 \subset I$, $\{\lambda_i\}_{i \in J_1} \subset \mathbb{R}$, $\{\mu_i\}_{i \in J_2} \subset \mathbb{R}_+$ tais que

$$\sum_{i \in J_1} \lambda_i [\nabla h_2(x^*)]_i + \sum_{i \in J_2} \mu_i [\nabla g_2(x^*)]_i = 0$$

e

$$\sum_{i \in J_1} |\lambda_i| + \sum_{i \in J_2} \mu_i > 0,$$

implica que os gradientes $\{\nabla [h_2(x)]_i\}_{i \in J_1} \cup \{\nabla [g_2(x)]_i\}_{i \in J_2}$ sejam Linearmente Dependentes para todo x em alguma vizinhança de x^* .

Considerando que o algoritmo não para na Etapa 2, temos os seguintes resultados.

Teorema 4.6 Considere $\{x^k\}$ a sequência gerada pelo Algoritmo Lagrangiano Aumentado e x^* um ponto limite de $\{x^k\}$. Então, se a sequência de parâmetros de penalidade $\{\rho_k\}$ é limitada, o ponto limite x^* é viável. Senão, uma das seguintes possibilidades é verificada:

1. x^* é um ponto KKT para o problema

$$\begin{aligned} \text{minimizar} \quad & \frac{1}{2} \left[\sum_{i=1}^{m_1} [h_1(x)]_i^2 + \sum_{i=1}^{p_1} \max \{0, [g_1(x)]_i\}^2 \right] \\ \text{sujeita a} \quad & x \in \Omega_2. \end{aligned}$$

2. x^* não satisfaz a condição de qualificação CPLD associada a Ω_2 .

Demonstração. Em [2, Teorema 4.1].

O Teorema 4.6 mostra que todo ponto limite é viável, ou é um ponto *KKT* da soma de quadrado das inviabilidades, ou não satisfaz *CPLD*. Sob a hipótese *CPLD* do teorema a seguir mostra que os pontos limites são *KKT*.

Teorema 4.7 *Considere $\{x^k\}$ uma sequência gerada pelo Algoritmo Lagrangiano Aumentado. Assuma que $x^* \in \Omega_1 \cap \Omega_2$ é um ponto limite que satisfaz a condição de qualificação CPLD com respeito a $\Omega_1 \cap \Omega_2$. Então x^* é um ponto KKT do problema (4.8).*

Demonstração. Em [2, Teorema 4.2].

4.3.3 ALGENCAN

ALGENCAN [70] é um pacote computacional em Fortran 77 para resolver problemas de programação não linear da forma

$$\begin{aligned} &\text{minimizar} && f(x) \\ &\text{sujeita a} && h_1(x) = 0 \\ &&& g_1(x) \leq 0 \\ &&& x \in \Omega_2, \end{aligned}$$

onde $f: \mathbb{R}^n \rightarrow \mathbb{R}$, $h_1: \mathbb{R}^n \rightarrow \mathbb{R}^{m_1}$, $g_1: \mathbb{R}^n \rightarrow \mathbb{R}^{p_1}$ e $\Omega_2 = \{x \in \mathbb{R}^n \mid a \leq x \leq b\}$. O ALGENCAN é uma implementação do Algoritmo Lagrangiano Aumentado introduzido por Andreani, Birgin, Martínez e Schuverdt [2, 3], descrito na Subseção 4.3.1, mas com os subproblemas restritos à caixa. Os subproblemas são resolvidos pelo Algoritmo GENCAN apresentado em [41], que é um algoritmo de restrições ativas para resolver problemas em caixa. Dentro das faces utiliza o método de Newton Truncado com busca linear, e para abandoná-las utiliza o método do Gradiente Espectral Projetado.

Temos agora todos os ingredientes para apresentar nosso algoritmo sem derivadas para resolver o problema (4.1).

4.4 O algoritmo implementado

Apresentamos um algoritmo de região de confiança sem derivadas, para o caso de restrições gerais Ω_1 . O algoritmo é implementado em Fortran 77. Os modelos são construídos e atualizados por interpolação polinomial. O número de pontos de interpolação, m , pode ser tomado no intervalo

$$2n + 1 \leq m \leq (n + 1)(n + 2)/2.$$

Esta escolha permite desde modelos subdeterminados como no Capítulo 2, até modelos completos que visam melhor ajuste da função objetivo. Em geral, em cada iteração apenas um ponto é modificado no conjunto interpolador. Porém, no caso em que não há progresso no algoritmo e ainda não foi identificado um ponto estacionário, o conjunto interpolador é substancialmente modificado buscando modelos mais adequados. O algoritmo a seguir não tem resultado de convergência, mas estabelecemos a sua boa definição.

Algoritmo 4.8 *Algoritmo implementado*

Dados $x^1 \in \Omega_1$, $\rho_1 > \varepsilon > 0$, $\gamma \in (0, 1)$ e $m \in [2n + 1, (n + 1)(n + 2)/2]$,
 $s > 2$, $\varepsilon_1 > 0$, $\Delta_1 = \rho_1$, $k = 1$.

Etapa 1: Construção do modelo

Faça $y^1 = x^k$ e $x^b = x^k$.

Construa $Y_k = \{y^1, y^2, \dots, y^m\}$ por (4.5) e (4.6) tomando $\Delta = \rho_k$.

Construa o modelo Q_k por interpolação polinomial.

Etapa 2: Solução do subproblema

Obtenha x^+ , por ALGENCAN, solução do subproblema (4.4).

Faça $d^k = x^+ - x^k$.

Etapa 3: Atualização dos parâmetros

SE $\|d^k\| \leq 0.5\rho_k$, ENTÃO

SE $\rho_k \leq \varepsilon$, ENTÃO pare.

SE $\max_{1 \leq j \leq m} \{\|y^j - x^k\|\} > s\rho_k$, ENTÃO $\rho_{k+1} = \rho_k$.

SENÃO, $\rho_{k+1} = \gamma\rho_k$.

$x^{k+1} = x^k$, $\Delta_{k+1} = \Delta_k$, $k = k + 1$ e vá para Etapa 1.

SENÃO,

$$r = \frac{f(x^k) - f(x^k + d^k)}{Q_k(x^k) - Q_k(x^k + d^k)}.$$

$$\bar{\Delta}_{k+1} = \begin{cases} 0.5\Delta_k, & \text{se } r < 0.1, \\ \Delta_k, & \text{se } 0.1 \leq r \leq 0.7, \\ 2\Delta_k, & \text{se } r > 0.7. \end{cases}$$

SE $f(x^k + d^k) < f(x^k)$, ENTÃO

Obtenha t por (4.7).

SE $\sigma(t) \leq \varepsilon_1$,

ENTÃO $x^{k+1} = x^k$, $\rho_{k+1} = \rho_k$, $\Delta_{k+1} = \bar{\Delta}_{k+1}$, $k = k + 1$ e vá para Etapa 1.

SE $r \geq 0.1$, ENTÃO

$x^{k+1} = x^k + d^k$, $\rho_{k+1} = \rho_k$, $\Delta_{k+1} = \bar{\Delta}_{k+1}$, $Y_{k+1} = Y_k \setminus \{y^t\} \cup \{x^k + d^k\}$,

obtenha Q_{k+1} que resolve (4.3), $k = k + 1$ e vá para a Etapa 2.

SENÃO,

$x^{k+1} = x^k$.

SE $\max_{1 \leq j \leq m} \{\|y^j - x^{k+1}\|\} > s\rho_k$, ENTÃO

$\rho_{k+1} = \rho_k$, $\Delta_{k+1} = \bar{\Delta}_{k+1}$, $k = k + 1$ e vá para a Etapa 1.

SENÃO,

SE $\rho_k \leq \varepsilon$, ENTÃO pare.

SENÃO,

SE $f(x^k + d^k) < f(x^k)$, ENTÃO

$Y_{k+1} = Y_k \setminus \{y^t\} \cup \{x^k + d^k\}$ e obtenha Q_{k+1} que resolve (4.3).

SENÃO,

$Y_{k+1} = Y_k$, $Q_{k+1} = Q_k$.

$\rho_{k+1} = \gamma\rho_k$, $\Delta_{k+1} = \rho_k$, $k = k + 1$ e vá para a Etapa 2.

4.5 Detalhes da implementação

Nesta seção descrevemos os procedimentos executados em cada etapa do algoritmo implementado.

Dados de entrada

O Algoritmo 4.8 é de pontos viáveis e portanto o ponto inicial x^1 deve ser viável. Se o ponto inicial fornecido pelo usuário for não viável, o algoritmo executa ALGENCAN minimizando a inviabilidade. Esta fase de restauração da viabilidade está incorporada à implementação. Nesta fase de restauração o ponto que o ALGENCAN encontra deve ser um ponto quase viável e este é tomado como o ponto inicial x^1 . A quase viabilidade é uma característica do ALGENCAN, que caminha por pontos inviáveis e tem como critério de parada, além da quase otimalidade, a quase viabilidade.

Etapa 1

Esta etapa é executada na primeira iteração e sempre que o passo é pequeno, ou seja, $\|d^k\| \leq 0.5\rho_k$, ou a razão $r < 0.1$ e os pontos do conjunto interpolador estão longe do iterando atual. O algoritmo não obtém decréscimo suficiente quando $r < 0.1$ e este fracasso no desempenho pode ocorrer pelo fato que o iterando atual está próximo a uma solução, ou o modelo não é adequado pois os pontos do conjunto interpolador estão longe do iterando atual. A Etapa 1 tem a função de rearranjar os pontos de forma que estejam posicionados para a interpolação e próximos do iterando atual. Suponha que em uma iteração k o Algoritmo 4.8 execute a Etapa 1. Então são construídos o conjunto interpolador Y_k utilizando as fórmulas (4.5) e (4.6) tomando $y^1 = x^k$ e $\Delta = \rho_k$. Assim, os pontos de Y_k ficam dentro da bola $B_\infty(x^k, \rho_k)$. A construção do modelo segue o Teorema 4.1. Ainda, como a atualização do modelo é feita segundo a matriz inversa H_k associada ao sistema, o Teorema 3.3 permite construí-la com uma fórmula fechada.

Etapa 2

Nesta etapa é determinado o passo d^k a partir da solução do subproblema (4.4). Os subproblemas são resolvidos pelo Algoritmo ALGENCAN versão 2.2.1 [70], que é baseado em Lagrangianos Aumentados e descrito na Subseção 4.3.3.

Etapa3

Nesta etapa são atualizados o iterando, o conjunto interpolador e o modelo. O parâmetro s é utilizado para caracterizar quando o conjunto de pontos interpoladores está longe do iterando atual. Isto é, se na iteração k existe ponto no conjunto interpolador a uma distância maior do que $s\rho_k$ do iterando atual x^k , então o algoritmo constrói o modelo a partir da Etapa 1, mantendo o valor do atual ρ_k . O iterando x^k recebe o ponto $x^k + d^k$ somente quando houver decréscimo suficiente na função objetivo. A existência do decréscimo suficiente é estabelecida pela razão, $r \geq 0.1$, entre o decréscimo real e o predito pelo modelo. Na atualização do conjunto Y_k , apenas um ponto é alterado utilizando a expressão (4.7), o que garante unicidade na atualização

do modelo. Quando há decréscimo simples na função objetivo, o ponto é incorporado ao conjunto interpolador. Este procedimento garante que mesmo quando não há decréscimo suficiente na função objetivo, o conjunto interpolador é melhorado por ser um ponto com boas características. A atualização do raio da região de confiança segue o formato clássico em métodos de região de confiança. O parâmetro ρ_k está relacionado à construção do conjunto interpolador Y_k na Etapa 1. A sequência $\{\rho_k\}$ é não crescente e seu decréscimo é utilizado para estabelecer o fim das iterações do algoritmo. Sempre que em uma iteração k não houve decréscimo suficiente na função objetivo e os pontos do conjunto interpolador Y_k estão próximos do iterando atual x^k no sentido da fórmula

$$\max_{1 \leq j \leq m} \{\|y^j - x^k\|\} \leq s\rho_k,$$

o termo ρ_k é reduzido pelo fator $\gamma \in (0, 1)$. Desta forma, está na hora de reduzir ρ_k pois possivelmente estamos próximos a uma solução do problema original, ou o valor de ρ_k é grande o que colaborou para não deixar o modelo adequadamente ajustado à função objetivo em uma vizinhança do iterando corrente. Nesta etapa o modelo quadrático é atualizado com a estrutura devido a Powell e apresentada na Seção 3.2 a partir da matriz inversa dada no Teorema 3.5. A Proposição 3.8 afirma que o valor de $\sigma(t)$ é estritamente positivo. No entanto, por erros inerentes ao sistema de numeração este valor pode ser menor ou igual a zero. Neste caso, o algoritmo comporta uma salvaguarda que o leva à Etapa 1, onde são feitas novas construções. O parâmetro $\varepsilon_1 > 0$ deve ser pequeno.

4.6 Resultados teóricos

No lema a seguir mostramos que o algoritmo está bem definido.

Lema 4.9 *Suponha que na k -ésima iteração o Algoritmo 4.8 não atinge o critério de parada $\rho_k \leq \varepsilon$. Dados o iterando x^k , o raio da região de confiança Δ_k , o raio ρ_k , o conjunto interpolador Y_k , a inversa H_k e o modelo Q_k , então o algoritmo será bem sucedido em gerar os elementos x^{k+1} , Δ_{k+1} , ρ_{k+1} , Y_{k+1} , a matriz inversa H_{k+1} e o modelo Q_{k+1} .*

Demonstração. A Etapa 1 está bem definida, pois segue dos Teoremas 4.1 e 3.3.

Para mostrar que os elementos x^{k+1} , ρ_{k+1} , Δ_{k+1} e Y_{k+1} estão bem definidos, considere dois casos:

- Caso 1. Suponha o passo $\|d^k\| \leq 0.5\rho_k$. Logo $x^{k+1} = x^k$. Se

$$\max_{1 \leq j \leq m} \{\|y^j - x^k\|\} > s\rho_k,$$

então $\rho_{k+1} = \rho_k$, caso contrário $\rho_{k+1} = \gamma\rho_k$. O raio da região de confiança é atualizado por $\Delta_{k+1} = \Delta_k$.

Ainda na hipótese de $\|d^k\| \leq 0.5\rho_k$, o conjunto interpolador Y_{k+1} é construído pela Etapa 1 modificando $m - 1$ pontos de Y_k . Neste caso somente o melhor iterando até o momento é utilizado e substituí o ponto y^1 no conjunto interpolador.

- Caso 2. Suponha o passo $\|d^k\| > 0.5\rho_k$. Se $r \geq 0.1$, então $x^{k+1} = x^k + d^k$ e $\rho_{k+1} = \rho_k$ e $\Delta_{k+1} = \Delta_k$ ou $\Delta_{k+1} = 2\Delta_k$. O conjunto interpolador fica

$$Y_{k+1} = Y_k \setminus \{y^t\} \cup \{x^k + d^k\}$$

com t obtido por (4.7). Se $r < 0.1$, então $x^{k+1} = x^k$ e se

$$\max_{1 \leq j \leq m} \{\|y^j - x^k\|\} > s\rho_k,$$

então $\rho_{k+1} = \rho_k$ e $\Delta_{k+1} = 0.5\Delta_k$, do contrário $\rho_{k+1} = \gamma\rho_k$ e $\Delta_{k+1} = \rho_k$.

Ainda na hipótese de $\|d^k\| > 0.5\rho_k$ e $r < 0.1$. Se $f(x^k + d^k) < f(x^k)$, o ponto $x^k + d^k$ ocupa no conjunto interpolador o lugar do ponto y^t , onde t é dado por (4.7). Caso contrário, se

$$\max_{1 \leq j \leq m} \{\|y^j - x^k\|\} \leq s\rho_k,$$

então $Y_{k+1} = Y_k$, senão Y_{k+1} é definido pela Etapa 1.

Agora, mostramos que a inversa H_{k+1} e o modelo Q_{k+1} estão bem definidos. A primeira matriz inversa existe pela Etapa 1. Pelo Teorema 3.5, basta que $\sigma \neq 0$ para existência de $H^+ = H_{k+1}$. O Lema 3.8 garante que o ponto escolhido pelo algoritmo por (4.7), é tal que o valor de τ é não nulo. Pela Proposição 3.6, α e β são não negativos. Como $\sigma = \alpha\beta + \tau^2$ (3.23), segue que $\sigma \neq 0$. Assim, o modelo quadrático é atualizado a partir da matriz inversa H_{k+1} , o que completa a demonstração. □

O Algoritmo 4.8 é de descida e de pontos quase viáveis. A solução aproximada na Etapa 2 define a quase viabilidade.

A sequência $\{\rho_k\}$ é monótona não crescente. O algoritmo para quando em uma iteração k vale a desigualdade $\rho_k < \varepsilon$. No lema a seguir estabelecemos que se o passo obtido pelo Algoritmo 4.8 fica suficientemente pequeno, a sequência $\{\rho_k\}$ converge para zero. Particularmente, se o ponto inicial é uma solução local, o algoritmo a identifica e para com tolerância $\varepsilon > 0$.

Lema 4.10 *Suponha que exista $k_0 \in \mathbb{N}$ tal que para todo $k \geq k_0$,*

$$\|d^k\| \leq 0.5\rho_k \quad \text{ou} \quad \frac{f(x^k) - f(x^k + d^k)}{Q_k(x^k) - Q_k(x^k + d^k)} < 0.1,$$

então

$$\lim_{k \rightarrow \infty} \rho_k = 0.$$

Demonstração. Pela estrutura do algoritmo, em ambas as hipóteses do lema é testada a desigualdade

$$\max_{1 \leq j \leq m} \{\|y^j - x^k\|\} > s\rho_k. \tag{4.14}$$

No entanto, ela não é satisfeita duas vezes consecutivas pois $s > 2$ e a Etapa 1 reconstrói o conjunto Y_k de modo que

$$\max_{1 \leq j \leq m} \{\|y^j - x^k\|\} \leq \rho_k.$$

Mesmo que o novo ponto $x^k + d^k$ tenha entrado no conjunto interpolador, a distância deste ponto até o iterando atual x^k é menor do que ρ_k . Como ρ_k é reduzido por um fator $\gamma \in (0, 1)$ sempre que a desigualdade (4.14) não é satisfeita, segue pela monotonicidade que a sequência $\{\rho_k\}$ converge para zero.

□

O Lema 4.10 mostra que se o algoritmo atingiu uma solução local do problema original, então não sairá deste ponto e reduzirá ρ_k até declarar convergência. Isto ocorre pois, mesmo se o modelo não for adequado e o passo d^k for grande, o decréscimo suficiente não será atingido. Então as hipóteses do Lema 4.10 são satisfeitas e o teste $\rho_k \leq \varepsilon$ é verdadeiro para algum k .

No próximo capítulo apresentamos testes numéricos para avaliar o desempenho do algoritmo implementado.

Capítulo 5

Testes numéricos

Implementamos o Algoritmo 4.8 em Fortran 77. Utilizamos para os testes uma máquina com processador I7 2.1GHz com 8GB RAM. Compilamos com o gfortran-4.2.3 no sistema operacional Linux - Mandriva 2008.

Dividimos o capítulo em 3 seções. Na primeira seção testamos todos os 216 problemas da coleção Hock-Schittkowski [30] com alguma restrição de igualdade ou desigualdade além de caixa. A coleção apresenta 306 problemas e destes, 90 são ir-restritos ou com restrições somente em caixas. Comparamos o desempenho de duas variações do algoritmo proposto com 3 algoritmos disponíveis na literatura. Na segunda seção aplicamos o algoritmo ao problema de interseção de elipses sugerido em [45]. O objetivo foi verificar a funcionalidade do algoritmo em um problema que não envolve derivadas. O problema tem forte apelo geométrico o que permite verificar graficamente a qualidade da solução. Na terceira seção tratamos problemas de dimensões consideradas grandes para algoritmos sem derivadas. Nestes problemas discutimos a questão do tempo computacional.

Para realizar as comparações utilizamos os gráficos de perfil de desempenho [28] e data profile [43]. Como medida de desempenho escolhemos o número de avaliações de função objetivo, pois em otimização sem derivadas, avaliação de função objetivo é considerada com alto custo em relação a outras ações executadas por um algoritmo. Desta forma, a eficiência é medida como o número de avaliações de função. Entendemos como robustez a capacidade do algoritmo em resolver problemas.

Considere \mathcal{A} o conjunto de algoritmos testados. Dado um critério para definição de solução de um problema, seja $\nu \in \mathcal{A}$ e $\sigma_\nu(i)$ o número de avaliações de função objetivo gastos pelo Algoritmo ν para resolver um problema i de acordo com o critério adotado para solução. Tomemos $\sigma^*(i)$ o menor número de avaliações que um algoritmo em \mathcal{A} gastou para resolver o problema i . Construímos a função de distribuição acumulada

$$d_\nu(\alpha) = \text{card} \left(i : \frac{\sigma_\nu(i)}{\sigma^*(i)} \leq \alpha \right).$$

O número $d_\nu(\alpha)$ indica quantos problemas o Algoritmo ν resolveu, gastando até α vezes o número de avaliações de função do algoritmo que resolveu mais rapidamente (menos avaliações de função objetivo) o mesmo problema. Se tomamos $\alpha = 1$ temos o número de problemas nos quais o Algoritmo ν foi mais eficiente, ou seja, aquele que gastou menos avaliações de função objetivo. Por outro lado, se tomamos $\alpha = \infty$, temos o número total de problemas que o Algoritmo ν resolveu, ou seja, sua robustez.

Assim, nos gráficos de perfil de desempenho mostramos, simultaneamente, a eficiência e a robustez dos algoritmos testados.

Escolha dos parâmetros

Apresentamos resultados do algoritmo proposto em duas variações que denominamos de Algoritmo m_1 e Algoritmo m_2 , e que diferem no número de pontos de interpolação. Para o Algoritmo m_1 são $m = 5$ pontos de interpolação quando a dimensão do problema for $n = 2$ e $m = 2n + 3$ quando $n \geq 3$. Para o Algoritmo m_2 não deixamos liberdade nos parâmetros do modelo, isto é, o número de pontos de interpolação é

$$\frac{1}{2}(n+1)(n+2).$$

O raio da região de confiança inicial é $\Delta_1 = 10^{-1}$, e $\rho_1 = 10^{-1}$. Os algoritmos param quando a sequência ρ_k atinge o valor $\varepsilon = 10^{-4}$. Escolhemos o parâmetro $s = 10$ para caracterizar quando os pontos interpoladores estão longe do iterando atual, $\gamma = 0.1$ para o decréscimo na sequência ρ_k e $\varepsilon_1 = 10^{-10}$ na salvaguarda. O ALGENCAN permite aproximar as derivadas das restrições. Em nossos testes, definimos as derivadas dos modelos, aproximamos os gradientes das restrições e a Jacobiana da restrição. Deixamos todos os parâmetros padrões no ALGENCAN e iniciamos cada subproblema no ponto x^b .

Comparamos as duas variações de nosso algoritmo com outros três algoritmos sem derivadas encontrados na literatura.

1. HOPSPACK (Hybrid Optimization Parallel Search PACKage) [46]: resolve problemas de otimização sem derivadas utilizando um código aberto em C_{++} . É uma implementação de um algoritmo baseado em Lagrangianos Aumentados [31, 37], como o algoritmo na Seção 4.3.1, mas com o Algoritmo GSS [65] aplicado aos subproblemas. O algoritmo possui resultados de convergência.
2. Restauração Inexata [10]: utiliza o Algoritmo GSS na fase de otimalidade. Nesta fase, ocorre a minimização da função objetivo original somada a um termo de regularização e dentro de um subespaço tangente. O algoritmo possui resultados de convergência.
3. DFO (Derivative Free Optimization): é um código aberto em Fortran, da implementação de um algoritmo de região de confiança sem derivadas desenvolvido por Conn, Scheinberg e Toint [20]. Os subproblemas são resolvidos pelo Algoritmo NPSOL [33] que utiliza programação quadrática sequencial. DFO inicia com 2 pontos interpoladores viáveis e mantém o conjunto interpolador viável. Não é tratada teoria de convergência para o caso restrito. O critério de parada de DFO é o raio da região de confiança igual a 10^{-4} .

5.1 Problemas da coleção Hock-Schittkowski

Os problemas da coleção Hock-Schittkowski (HS) possuem informações de derivadas e portanto podem ser resolvidos por algoritmos já consagrados em otimização com derivadas e com desempenho melhor do que o algoritmo proposto. No entanto,

estes testes tem o objetivo de verificar como o algoritmo proposto funciona. Alguns problemas da coleção são mal condicionados e mal escalados, o que causa dificuldades na solução mesmo por algoritmos que utilizam derivadas.

Na primeira subseção avaliamos o desempenho das duas variações do algoritmo proposto, Algoritmos m_1 e m_2 . Na segunda subseção comparamos o desempenho de m_1 , m_2 , Restauração Inexata e HOPSPACK. Na terceira subseção comparamos m_1 e m_2 com DFO.

5.1.1 Desempenho de variações do algoritmo

Testamos todos os 216 problemas com restrições que envolvem alguma restrição de igualdade ou desigualdade além de caixa, e disponíveis em Hock-Schittkowsk (HS) [30]. As dimensões dos problemas variam de 2 a 50 e o número de restrições entre 1 e 45. Detalhes de dimensão, quantidade de restrições e soluções obtidas estão no Apêndice A. Comparamos o desempenho dos Algoritmos m_1 e m_2 a partir das soluções disponíveis na coleção HS. Para estes testes, consideramos o problema resolvido pelo algoritmo quando a solução \bar{x} satisfaz a tolerância 10^{-8} para viabilidade (na norma euclidiana) e

$$\frac{f(\bar{x}) - f_{HS}}{\max\{1, |f(\bar{x})|, |f_{HS}|\}} \leq 10^{-4}, \quad (5.1)$$

onde f_{HS} é a solução local obtida com o código NLPQLP [63], que é uma implementação em Fortran de um algoritmo de programação quadrática sequencial (PQS). Os valores de f_{HS} estão disponíveis em [30] e reescritos no Apêndice A. O código NLPQLP é utilizado para resolver problemas de programação não linear suaves. É uma extensão do código NLPQL [62]. Segundo os autores em [30], SQP pertence à classe dos métodos mais poderosos para resolver problemas da natureza destes que foram testados.

O Algoritmo m_1 resolveu 96.8% dos problemas de acordo com o critério (5.1). Falhou nos problemas 112, 268, 270, 348, 361, 364 e 393. No problema 268 o resultado está longe de uma solução. Para o problema 270, a solução encontrada é diferente das disponíveis na coleção HS mas a mesma encontrada pelo Algoritmo ALGENCAN. O problema 348 não está definido em uma vizinhança do primeiro ponto viável obtido, e logo não permite construir o primeiro modelo interpolador. O Algoritmo 4.8 exige a definição da função objetivo em uma vizinhança de pontos da sequência viável gerada. Esta exigência ocorre na primeira iteração e em todas as iterações que executam a Etapa 1 do algoritmo. No problema 361 o ALGENCAN falhou em determinar um ponto viável a partir do ponto inicial fornecido na coleção HS. Os problemas 112, 364 e 393 satisfazem (5.1) quando a precisão é 10^{-2} .

O Algoritmo m_2 resolveu 97.2% dos problemas de acordo com o critério (5.1). Falhou nos problemas 68, 69, 112, 270, 348 e 361. Nos problemas 112, 270, 348 e 361 observamos as mesmas características consideradas no Algoritmo m_1 . As sequências geradas para os problemas 68 e 69 convergem para a mesma solução local que o ALGENCAN.

O Algoritmo m_1 permite economia no número de avaliações de função objetivo na construção do modelo, pois deixa liberdade nos parâmetros deste. No entanto, os problemas 58, 70, 105, 117, 231, 268 e 364 mostram que a qualidade do modelo pode obrigar o algoritmo a executar mais chamadas a cálculos de subproblemas e de

construção de modelos, o que justifica o grande número de avaliações de função pelo Algoritmo m_1 para resolver estes problemas.

O perfil de desempenho e o *data profile*, Figura 5.1, mostram que a eficiência quando utilizamos o Algoritmo m_1 é grande em relação ao Algoritmo m_2 , e que a robustez de ambos é equivalente. Apesar de m_1 ser mais eficiente, a diferença entre m_1 e m_2 no que se refere ao número de avaliações de função objetivo é pequena, como mostra o primeiro gráfico na Figura 5.1.

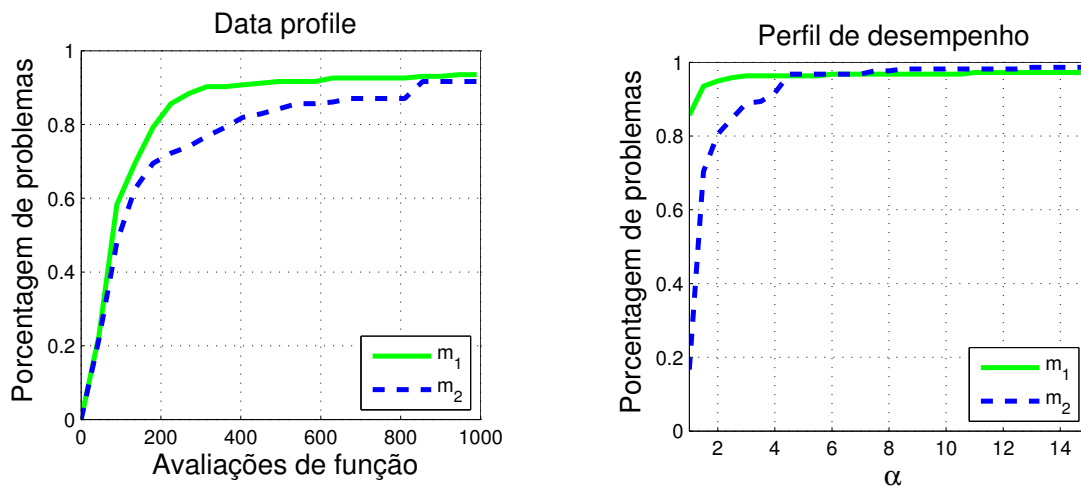


Figura 5.1: Comparação de desempenho entre os Algoritmos m_1 , m_2 , utilizando todos os 216 problemas restritos da coleção HS

A coleção [30] disponibiliza as soluções globais dos problemas até então conhecidas. Com base no critério (5.1) em relação à solução global, os Algoritmos m_1 e m_2 resolvem respectivamente 89.8% e 90.7% dos problemas.

Na Tabela 5.1 apresentamos o tempo de CPU, em segundos, que os Algoritmos m_1 e m_2 gastaram para resolver problemas com dimensões maiores do que 30. A coluna P indica o número do problema na coleção HS e n a dimensão. Estes problemas, 392, 393 e 395, aliam grande dimensão a um grande número de avaliações de função objetivo, o que justifica o tempo computacional gasto para execução. No problema 392 são executadas menos avaliações de função objetivo que no problema 393 que tem dimensão maior. No entanto, o tempo de execução do Algoritmo m_1 no problema 392 é maior do que o tempo de execução no problema 393. Logo, o tempo foi gasto essencialmente nas soluções dos subproblemas. Tratamos na última seção sobre como o tempo é consumido durante as execuções dos algoritmos.

P	n	Algoritmo m_1		Algoritmo m_2	
		#AF	tempo(s)	#AF	tempo(s)
392	30	1005	5.150E+01	2991	1.094E+02
393	48	1115	5.556E+00	12271	2.878E+03
395	50	625	9.732E-01	7963	1.571E+03

Tabela 5.1: Tempo de CPU em segundos pelos Algoritmos m_1 e m_2

5.1.2 Comparação com algoritmos disponíveis na literatura

Nesta subseção comparamos o desempenho dos Algoritmos m_1 e m_2 com 3 algoritmos disponíveis na literatura. Utilizamos os dados contidos nos trabalhos [10, 20].

Para comparar com os Algoritmos HOPSPACK e Restauração Inexata, resolvemos todos os 105 problemas que foram sugeridos pelos autores em [10]. Consideramos o problema resolvido utilizando o mesmo teste dado em [10], que relaciona o desempenho dos algoritmos comparados. Um ponto \bar{x} é solução de um problema quando satisfaz a tolerância 10^{-8} para viabilidade (na norma euclidiana) e

$$\frac{f(\bar{x}) - f_L}{\max\{1, |f(\bar{x})|, |f_L|\}} \leq 0.1, \quad (5.2)$$

onde f_L é o menor valor de função objetivo encontrado pelos algoritmos comparados.

A Tabela 5.2 mostra os resultados obtidos pelos Algoritmos m_1 , m_2 , Restauração Inexata e HOPSPACK. A letra P indica a coluna com o número do problema da coleção HS [30], #AF o número de avaliações da função objetivo e f o valor da função objetivo no ponto solução encontrado. O símbolo * indica que o problema não foi resolvido segundo o critério (5.2) e o símbolo ** que a viabilidade foi maior do que 10^{-8} . Os resultados para os algoritmos de Restauração Inexata e HOPSPACK foram retirados de [10]. Nos problemas em que não aparecem valores, o Algoritmo não conseguiu resolver o problema por questão de tempo computacional [10]. Todos os 105 problemas foram resolvidos pelos Algoritmos m_1 e m_2 em no máximo um minuto. As dimensões variam de 2 a 16 e o número de restrições entre 1 e 29.

O Algoritmo m_1 resolveu 97.1% dos problemas, o Algoritmo m_2 resolveu 95.2%, Restauração Inexata resolveu 89.5% e HOPSPACK resolveu 58.1%. Em 29 problemas HOPSPACK não encontrou viabilidade e em 3 superou o tempo de 20 minutos de CPU.

As soluções encontradas pelos Algoritmos m_1 e m_2 nos problemas 33 e 59 são locais [30]. As soluções obtidas por m_1 e m_2 para o problema 16, e as soluções por m_2 para os problemas 68 e 69, são as mesmas encontradas localmente pelo Algoritmo ALGENCAN, isto é, a solução encontrada satisfaz ao critério de parada do ALGENCAN.

P	Algoritmo m_1		Algoritmo m_2		Restauração Inexata		HOPSPACK	
	f	#AF	f	#AF	f	#AF	f	#AF
6	4.37E-18	36	4.33E-18	42	5.57E-10	366	** 4.84E+00	151
7	-1.73E+00	30	-1.73E+00	35	-1.73E+00	153	* 6.93E-01	325
8	-1.00E+00	26	-1.00E+00	31	-1.00E+00	4	** -1.00E+00	187
9	-5.00E-01	40	-5.00E-01	52	-5.00E-01	117	-5.00E-01	26
10	-1.00E+00	28	-1.00E+00	33	-1.00E+00	260	* -8.65E-01	357
11	-8.50E+00	27	-8.50E+00	32	-8.50E+00	126	-8.48E+00	523
12	-3.00E+01	41	-3.00E+01	42	-3.00E+01	685	-3.00E+01	342
13	9.96E-01	35	9.96E-01	41	1.09E+00	52326	9.97E-01	1183
14	1.39E+00	20	1.39E+00	31	1.39E+00	20	** 1.39E+00	202
15	3.06E+00	30	3.06E+00	41	3.06E+00	82	3.06E+00	451
16	* 3.98E+00	37	* 3.98E+00	33	2.58E-01	546	2.50E-01	600
17	1.00E+00	1025	1.00E+00	94	1.00E+00	118	1.00E+00	612
18	5.00E+00	39	5.00E+00	45	5.00E+00	39217	* 1.07E+01	263
19	-6.96E+03	37	-6.96E+03	43	-6.96E+03	138	** -7.47E+03	1370
20	3.82E+01	29	3.82E+01	34	4.02E+01	88	3.82E+01	393
21	-1.00E+02	35	-1.00E+02	41	-1.00E+02	153	-1.00E+02	32
22	1.00E+00	27	1.00E+00	32	1.00E+00	35	1.00E+00	276
23	2.00E+00	35	2.00E+00	41	2.00E+00	37	2.00E+00	466
24	-1.00E+00	37	-1.00E+00	42	-1.00E+00	132	-1.00E+00	27

26	4.15E-13	159	2.11E-12	170	1.58E-07	11112	* 2.12E+01	585
27	4.00E+00	107	4.00E+00	91	4.00E+00	4135	** 4.00E+00	1358
28	6.76E-18	61	2.85E-15	67	2.05E-23	510	7.70E-08	264
29	-2.26E+01	89	-2.26E+01	70	-2.26E+01	647	-2.25E+01	327
30	1.00E+00	59	1.00E+00	65	1.00E+00	790	1.00E+00	55
31	6.00E+00	58	6.00E+00	64	6.00E+00	526	6.00E+00	921
32	1.00E+00	63	1.00E+00	69	1.00E+00	86	1.00E+00	51
33	* -4.00E+00	59	* -4.00E+00	65	* -4.00E+00	54	-4.59E+00	381
34	-8.34E-01	62	-8.34E-01	68	-8.34E-01	228	* -2.28E-01	582
35	1.11E-01	50	1.11E-01	55	1.11E-01	289	1.11E-01	340
36	-3.30E+03	62	-3.30E+03	68	-3.30E+03	273	-3.30E+03	60
37	-3.46E+03	151	-3.46E+03	164	-3.46E+03	404	-3.46E+03	102
39	-1.00E+00	58	-1.00E+00	78	-9.89E-01	125	-1.00E+00	830
40	-2.50E-01	59	-2.50E-01	77	-2.50E-01	133	** -2.51E-01	897
41	1.93E+00	170	1.93E+00	81	1.93E+00	430	1.93E+00	292
42	1.39E+01	71	1.39E+01	95	1.39E+01	831	1.40E+01	779
43	-4.40E+01	72	-4.40E+01	96	-4.40E+01	1878	-4.40E+01	1134
44	-1.50E+01	84	-1.50E+01	97	* -1.30E+01	277	* -1.30E+01	57
46	9.43E-12	219	5.01E-12	370	1.42E-06	1485	* 3.34E+00	777
47	6.87E-17	122	5.99E-17	182	1.15E-08	289	* 1.25E+01	901
48	6.19E-09	112	2.55E-15	133	1.07E-24	861	1.12E-06	497
49	5.64E-12	254	8.83E-12	381	1.34E-07	20308	1.43E-04	1002
50	1.31E-17	140	4.85E-21	231	9.25E-27	620	5.29E-07	290
51	2.54E-16	84	2.54E-16	132	1.94E-27	507	1.25E-06	142
52	5.33E+00	85	5.33E+00	133	5.33E+00	307	5.33E+00	311
53	4.09E+00	84	4.09E+00	132	4.09E+00	308	4.09E+00	216
54	-9.03E-01	939	-8.87E-01	6941	* -1.54E-01	447	** -	-
55	6.67E+00	76	6.67E+00	141	6.67E+00	18	** 6.00E+00	0
56	-3.46E+00	276	-3.46E+00	300	* -1.06E-06	21267	* -1.00E+00	2075
57	2.85E-02	65	2.85E-02	60	3.06E-02	287	3.06E-02	74
58	3.19E+00	2998	3.19E+00	65	3.19E+00	102	3.19E+00	817
59	* -6.75E+00	46	* -6.75E+00	57	-7.80E+00	929	* -6.75E+00	340
60	3.26E-02	63	3.26E-02	69	3.26E-02	596	** 5.47E-02	465
61	-1.44E+02	122	-1.44E+02	133	-1.44E+02	182	-1.43E+02	621
62	-2.63E+04	63	-2.63E+04	71	-2.63E+04	848	-2.63E+04	233
63	9.62E+02	60	9.62E+02	53	9.62E+02	171	9.63E+02	317
64	6.30E+03	98	6.30E+03	138	6.30E+03	1025	** 6.30E+03	6680
65	9.54E-01	62	9.54E-01	68	9.54E-01	1840	1.01E+00	379
66	5.18E-01	49	5.18E-01	54	5.18E-01	406	5.33E-01	566
68	-9.20E-01	447	* 2.40E-05	244	-9.20E-01	10424	** -8.44E-01	1316
69	-9.57E+02	386	* 4.00E-03	240	-9.57E+02	4130	** -9.57E+02	2471
70	7.50E-03	6663	7.50E-03	414	* 2.69E-01	5563	7.74E-03	3766
71	1.70E+01	75	1.70E+01	80	1.70E+01	4586	** 1.70E+01	1939
72	7.28E+02	77	7.28E+02	102	7.28E+02	58181	** 7.28E+02	18188
73	2.99E+01	59	2.99E+01	79	2.99E+01	211	3.02E+01	223
74	5.13E+03	80	5.13E+03	104	5.13E+03	392	** 5.14E+03	46145
75	5.17E+03	78	5.17E+03	102	5.17E+03	139	5.23E+03	22678
76	-4.68E+00	73	-4.68E+00	96	-4.68E+00	505	-4.68E+00	403
77	2.42E-01	137	2.42E-01	298	2.42E-01	790	** 4.68E+00	1904
78	-2.92E+00	152	-2.92E+00	108	-2.92E+00	566	** -2.89E+00	869
79	7.88E-02	152	7.88E-02	214	7.88E-02	362	** 2.42E-01	1054
80	5.39E-02	73	5.39E-02	108	5.39E-02	658	** 1.00E+00	557
81	5.39E-02	130	5.39E-02	178	5.39E-02	770	** 1.00E+00	557
83	-3.07E+04	87	-3.07E+04	135	-3.07E+04	450	-3.07E+04	1729
84	-5.28E+01	104	-5.28E+01	135	-5.28E+01	338	-5.28E+01	2842
86	-3.23E+01	93	-3.23E+01	132	-3.23E+01	437	-3.23E+01	485
87	8.93E+03	120	8.93E+03	210	8.85E+03	860	** 9.33E+03	16244
88	1.36E+00	27	1.36E+00	32	1.37E+00	105	** 1.36E+00	1341
89	1.36E+00	48	1.36E+00	53	1.36E+00	510	** 1.36E+00	2208
90	1.36E+00	58	1.36E+00	78	1.37E+00	1304	1.36E+00	2885
91	1.36E+00	68	1.36E+00	108	1.36E+00	1305	** 1.36E+00	4470
92	1.36E+00	79	1.36E+00	144	1.37E+00	3235	** 1.36E+00	4672
93	1.35E+02	166	1.35E+02	178	1.48E+02	2006	1.37E+02	129
95	1.56E-02	77	1.56E-02	142	1.56E-02	741	1.71E-02	156
96	1.56E-02	77	1.56E-02	142	1.56E-02	668	1.71E-02	156
97	3.14E+00	94	3.14E+00	172	* 4.07E+00	5654	* 4.12E+00	145
98	3.14E+00	94	3.14E+00	172	3.14E+00	1540	* 4.12E+00	145
99	-8.31E+08	89	-8.31E+08	183	-8.31E+08	4	** -7.46E+08	729
100	6.81E+02	113	6.81E+02	264	* 5.18E+03	1314	6.84E+02	873
101	1.81E+03	595	1.81E+03	468	1.81E+03	267	1.82E+03	14614
102	9.12E+02	463	9.12E+02	313	* 1.13E+03	57697	9.20E+02	15222

103	5.44E+02	240	5.44E+02	653	* 1.55E+03	22	5.44E+02	14583
104	3.95E+00	272	3.95E+00	282	3.95E+00	28794	3.95E+00	9844
105	1.14E+03	11140	1.14E+03	2000	1.14E+03	25324	1.14E+03	10580
106	7.05E+03	133	7.05E+03	289	7.05E+03	68430	** 1.14E+04	30424
107	5.06E+03	108	5.06E+03	278	5.06E+03	265	* 5.06E+03	7232
108	-8.66E-01	135	-8.66E-01	504	* -5.00E-01	139823	* -5.00E-01	99
109	5.36E+03	163	5.36E+03	349	5.39E+03	62792	** 5.50E+03	57551
111	-4.78E+01	191	-4.78E+01	467	* -4.28E+01	2470	*** -	-
112	-4.76E+01	150	-4.77E+01	341	-4.78E+01	10907	-4.78E+01	730
113	2.43E+01	146	2.43E+01	404	2.43E+01	4469	2.54E+01	1944
114	-1.77E+03	181	-1.77E+03	1486	-1.76E+03	108605	*** -	-
116	9.76E+01	188	9.76E+01	645	9.76E+01	48155	** 5.00E+01	9131
117	3.23E+01	1928	3.23E+01	2213	3.23E+01	7951	* 5.40E+01	7190
118	6.65E+02	207	6.65E+02	825	6.65E+02	1987	6.65E+02	3762
119	2.45E+02	299	2.45E+02	1238	2.45E+02	1356	2.45E+02	944

Tabela 5.2: Resultados dos 105 problemas da coleção HS obtidos pelos Algoritmos m_1 , m_2 , Restauração Inexata e HOPSPACK

Na Figura 5.2 são apresentados o *data profile* e o perfil de desempenho para os 4 algoritmos com base nos dados da Tabela 5.2. Constatamos que m_1 e m_2 são mais eficientes e robustos que Restauração Inexata e HOPSPACK. No primeiro gráfico da Figura 5.2 observamos que a quantidade de avaliações de função objetivo é consideravelmente menor nos Algoritmos m_1 e m_2 em relação aos demais algoritmos.

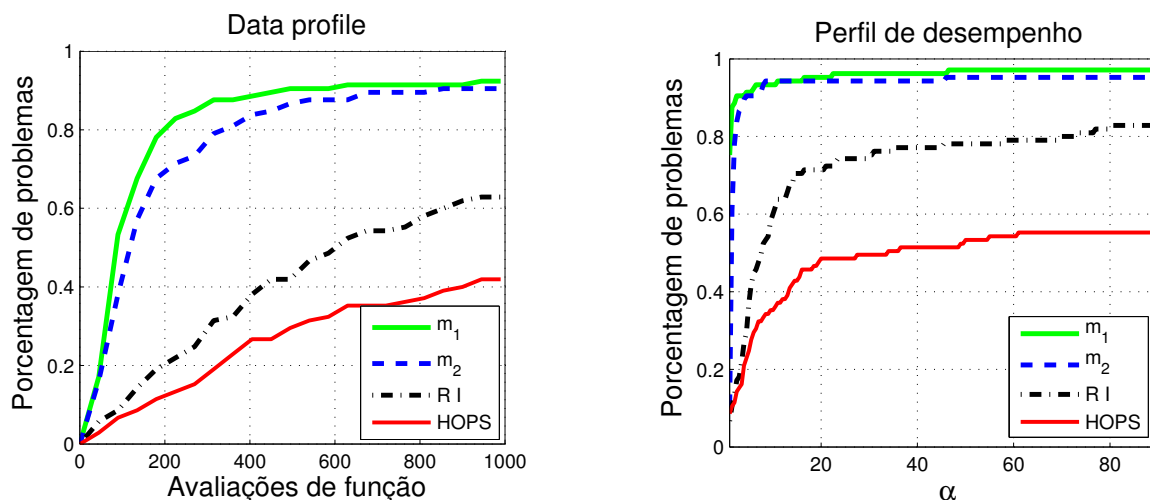


Figura 5.2: Comparação de desempenho entre os Algoritmos m_1 , m_2 , Restauração Inexata e HOPSPACK, utilizando 105 problemas restritos da coleção HS

Comparamos os algoritmos propostos com o Algoritmo DFO que utiliza a mesma estrutura de região de confiança sem derivadas que os Algoritmos m_1 e m_2 . Na Tabela 5.3 apresentamos os resultados dos 28 problemas restritos da coleção HS [29] que foram resolvidos por DFO em [20]. Constam de todos os problemas desta coleção [29] que possuem restrições somente de igualdade ou somente de desigualdade. Os dados sobre DFO (última coluna) foram retirados de [20]. O símbolo * indica o problema não resolvido segundo o critério (5.2). O símbolo ** indica que o algoritmo não encontrou viabilidade na solução do subproblema. O Algoritmo m_1 resolveu 96.4% dos problemas, o Algoritmo m_2 resolveu 92.9% dos problemas, enquanto que o Algoritmo DFO resolveu 85.7% dos problemas.

P	Algoritmo m_1		Algoritmo m_2		Algoritmo DFO	
	f	#AF	f	#AF	f	#AF
22	1.0000E+00	27	1.0000E+00	32	1.0000E+00	15
23	2.0000E+00	35	2.0000E+00	41	2.0000E+00	16
26	4.1473E-13	159	2.1108E-12	170	1.9355E-09	49
32	1.0000E+00	63	1.0000E+00	69	1.0000E+00	15
34	-8.3403E-01	62	-8.3403E-01	68	-8.3403E-01	22
44	-1.5000E+01	84	-1.5000E+01	97	-1.5000E+01	26
48	6.1937E-09	112	2.5484E-15	133	2.1132E-20	31
49	5.6396E-12	254	8.8253E-12	381	2.3516E-06	85
54	-9.0284E-01	939	-8.8699E-01	6941	* -1.5391E-01	27
56	-3.4560E+00	276	-3.4560E+00	300	-3.4560E+00	37
59	-6.7546E+00	46	-6.7546E+00	57	-6.7495E+00	32
63	9.6172E+02	60	9.6172E+02	53	9.6172E+02	12
65	9.5353E-01	62	9.5353E-01	68	9.5353E-01	35
68	-9.2043E-01	447	* 2.4000E-05	244	-9.2042E-01	127
69	-9.5671E+02	386	* 4.0000E-03	240	* -9.4134E+02	46
74	5.1265E+03	80	5.1265E+03	104	5.1265E+03	71
76	-4.6818E+00	73	-4.6818E+00	96	-4.6818E+00	29
79	7.8773E-02	152	7.8773E-02	214	7.8777E-02	25
100	6.8063E+02	113	6.8063E+02	264	6.8063E+02	127
106	7.0492E+03	133	7.0492E+03	289	7.0492E+03	63
107	5.0550E+03	108	5.0550E+03	278	5.0550E+03	29
108	-8.6603E-01	135	-8.6603E-01	504	-8.6603E-01	62
111	-4.7761E+01	191	-4.7761E+01	467	-4.7761E+01	227
112	-4.7571E+01	150	-4.7689E+01	341	-4.7761E+01	337
114	-1.7688E+03	181	-1.7688E+03	1486	* -9.1628E+02	8
116	9.7591E+01	188	9.7591E+01	645	** 9.7485E+01	87
119	2.4490E+02	299	2.4490E+02	1238	2.4490E+02	91
268	* 5.9475E-01	2642	2.3341E-12	133	-1.8190E-11	84

Tabela 5.3: Resultados numéricos obtidos pelos Algoritmos m_1 , m_2 e DFO para os 28 problemas com restrições retirados de [20]

Na Figura 5.3 são apresentados o *data profile* e o perfil de desempenho para os 3 algoritmos com base nos dados da Tabela 5.3. Constatamos que DFO é mais eficiente que m_1 e m_2 e menos robusto que ambos. No *data profile* observamos que o Algoritmo m_1 e DFO com 300 avaliações de função objetivo resolvem a mesma quantidade de problemas. A eficiência de DFO é devido ao fato de que o número de pontos de interpolação é dinâmico, e com poucos pontos interpoladores o algoritmo já é capaz de algum progresso.

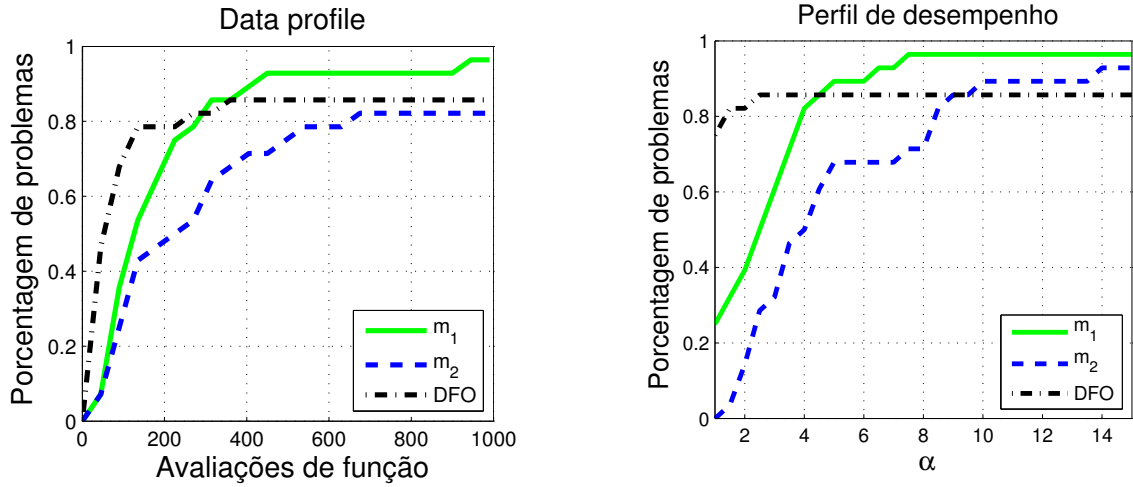


Figura 5.3: Comparação de desempenho entre os Algoritmos m_1 , m_2 e DFO, utilizando 28 problemas restritos da coleção HS

5.2 Aplicação

Consideramos o problema proposto em [45], que consiste em minimizar a área da interseção de duas elipses E_1 , E_2 e contendo pontos prefixados $c^i \in \mathbb{R}^2$, $i = 1, \dots, l$. Uma elipse fica inteiramente definida quando conhecemos seus focos e seu eixo maior. Tomamos duas elipses E_1 e E_2 definidas por um ponto $x = (x_1, x_2, \dots, x_{10})^T \in \mathbb{R}^{10}$. As cinco primeiras coordenadas de x definem a elipse E_1 , tomando para o eixo maior a coordenada x_5 e para os focos os pares $y^1 = (x_1, x_2)^T$ e $\bar{y}^1 = (x_3, x_4)^T$. As cinco últimas coordenadas de x definem a elipse E_2 , tomando para o eixo maior a coordenada x_{10} e para os focos os pares $y^2 = (x_6, x_7)^T$ e $\bar{y}^2 = (x_8, x_9)^T$. As restrições do problema são, para $i = 1, \dots, l$,

$$\|y^1 - \bar{y}^1\| \leq x_5, \quad \|c^i - y^1\| + \|c^i - \bar{y}^1\| \leq x_5, \quad \|y^2 - \bar{y}^2\| \leq x_{10} \quad \text{e} \quad \|c^i - y^2\| + \|c^i - \bar{y}^2\| \leq x_{10}.$$

O problema de interseção possui vários minimizadores locais e o mínimo encontrado pelo algoritmo depende do ponto inicial e do número de pontos de interpolação considerados na construção do modelo quadrático. Utilizamos para avaliar a função objetivo (calcular a área da interseção de elipses) um método numérico que aproxima área. Consiste em definir malhas sobre um quadrado de lado igual ao comprimento do eixo maior de uma das elipses. A solução do problema é sensível à forma com que as malhas são refinadas. Assim, não é possível boas aproximações para derivadas e portanto o problema pode ser considerado sem derivadas. Assim, os exemplos desta seção tem o objetivo de mostrar que o algoritmo é capaz de tratar problemas sem derivadas.

Exemplo 5.1 Considere $l = 4$ pontos

$$(1, 0)^T, (0, 1)^T, (-1, 0)^T, (0, -1)^T$$

e o ponto inicial viável $x^1 \in \mathbb{R}^{10}$,

$$x^1 = (0, 0, 0, 0, 4, 0, 0, 0, 0, 4)^T.$$

As elipses E_1 e E_2 definidas pelo ponto x^1 são circunferências de raio 2.

As Figuras 5.4 e 5.5 mostram as soluções obtidas pelos Algoritmos m_1 e m_2 , respectivamente. Os pontos em formato de losango são os l pontos que devem permanecer na interseção das elipses. Os demais pontos que aparecem na figura são os focos das duas elipses. Com o apelo geométrico deste problema, sabemos que a configuração dada na Figura 5.4 é de um minimizador global. As soluções fornecidas pelos Algoritmos m_1 e m_2 são distintas, mostrando que a solução é também sensível ao número de pontos de interpolação. Na Figura 5.6, melhoramos a precisão no cálculo da função objetivo (área da interseção das elipses). Observamos com esta mudança um novo minimizador local, ao invés de um refinamento no resultado obtido na Figura 5.5 que também utilizou o Algoritmo m_2 . A geometria do problema permite verificar graficamente a qualidade da solução em um número razoável de avaliações de função objetivo.

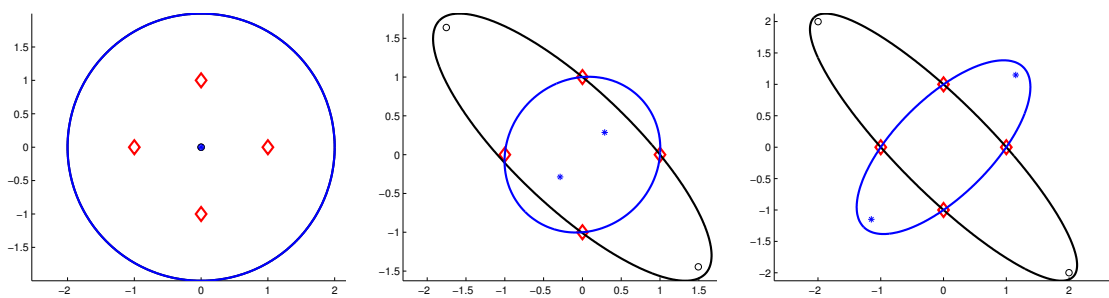


Figura 5.4: Solução obtida com o Algoritmo m_1 . Área 2.1586 e 1643 avaliações da função objetivo

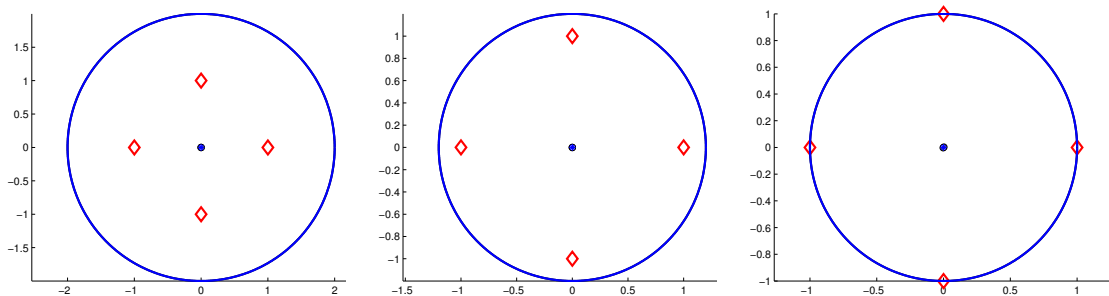


Figura 5.5: Solução obtida com o Algoritmo m_2 . Área 3.1415 e 259 avaliações da função objetivo

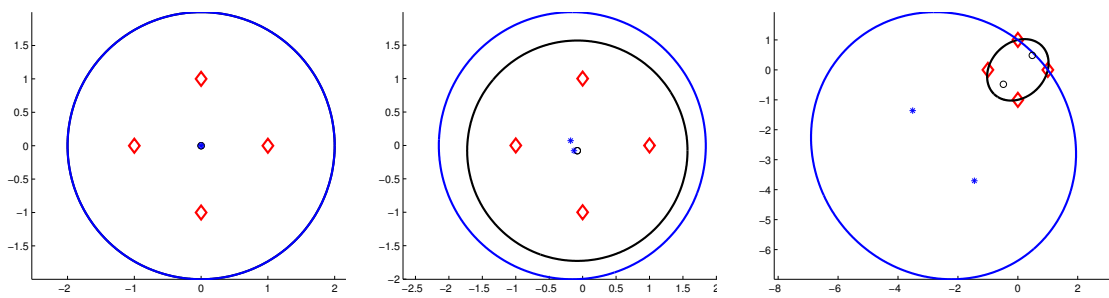


Figura 5.6: Solução obtida com o Algoritmo m_2 , com melhor precisão no cálculo das interseções das elipses. Área 2.8504 e 1461 avaliações da função objetivo

Exemplo 5.2 Considere $l = 8$ pontos

$$(-1, 0)^T, (0, -1)^T, (1, 0)^T, (0, 1)^T, (-1, -1)^T, (1, 1)^T, (0.7, 0.5)^T, (-0.1, 1.1)^T$$

e o ponto inicial viável $x^1 = (0, 0, 0, 0, 4, 0, 0, 0, 0, 4)^T$.

As Figuras 5.7 e 5.8 mostram as soluções obtidas pelos Algoritmos m_1 e m_2 , respectivamente. O número de variáveis é o mesmo em relação ao exemplo anterior, mas o número de restrições aumenta. Os pontos em formato de losango são os l pontos que devem permanecer na interseção das elipses. Os demais pontos que aparecem na figura são os focos das duas elipses. Novos minimizadores locais são encontrados e constatamos mais uma vez a sensibilidade do algoritmo em relação aos parâmetros de entrada. Na Figura 5.9 observamos novamente que o refinamento da malha provocou novo minimizador local.

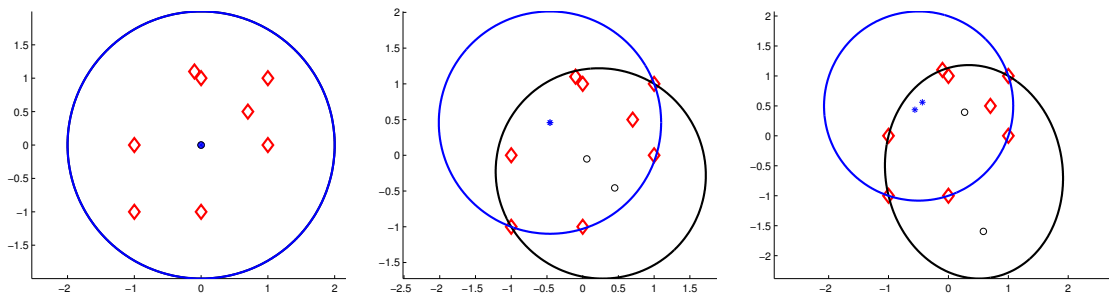


Figura 5.7: Solução obtida com o Algoritmo m_1 . Área 3.7177 e 1070 avaliações da função objetivo

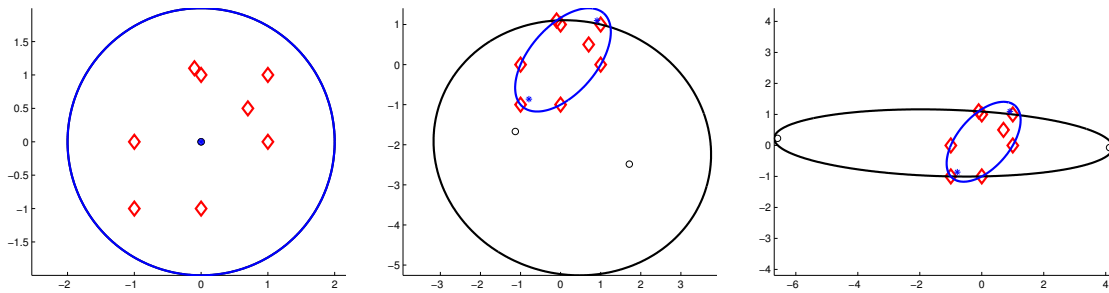


Figura 5.8: Solução obtida com o Algoritmo m_2 . Área 3.6052 e 3572 avaliações da função objetivo

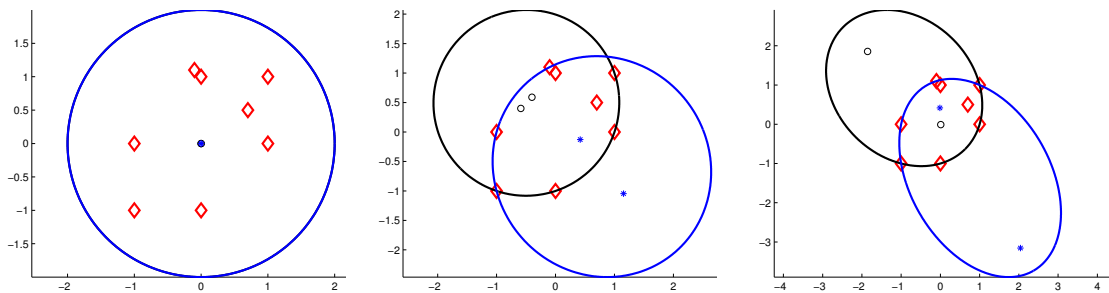


Figura 5.9: Solução obtida com o Algoritmo m_1 , com melhor precisão no cálculo das interseções das elipses. Área 3.7109 e 1378 avaliações da função objetivo

5.3 Problemas de dimensões maiores

Em otimização sem derivadas, problemas com poucas centenas de variáveis já são considerados grandes. Nesta seção aplicamos nossa proposta a problemas dessa magnitude. A complexidade algorítmica é $O(m^2)$ operações na atualização devido ao Teorema 3.5, onde m é o número de pontos de interpolação. Pelo Teorema 4.1, na construção do modelo a matriz $Z^{m \times (m-n-1)}$ deve ser multiplicada pela sua transposta. Logo, o número de pontos de interpolação $2n+3$ é mais adequado do que $(n+1)(n+2)/2$ para trabalhar com problemas de grandes dimensões.

Nesta subseção testamos três problemas que envolvem restrições de igualdade, caixa e desigualdade. Aplicamos o Algoritmo m_1 e mantivemos os mesmos parâmetros utilizados nos problemas anteriores.

Problema 1

Reescrevemos o problema 395 da coleção HS, originalmente definido em \mathbb{R}^{50} , para permitir dimensões maiores. O problema consiste em minimizar uma função na esfera unitária, ou seja,

$$\begin{aligned} \text{minimizar} \quad & f(x) = \sum_{i=1}^n i [x_i^2 + x_i^4] \\ \text{sujeita a} \quad & \sum_{i=1}^n x_i^2 = 1. \end{aligned} \tag{5.3}$$

Observamos que para $n \geq 2$ o ponto $x^* = (x_1, x_2, \dots, x_n)^T$ com coordenadas

$$x_1 = \frac{\sqrt{5}}{\sqrt{6}}, \quad x_2 = \frac{1}{\sqrt{6}}, \quad x_i = 0 \text{ para todo } i = 3, \dots, n$$

é KKT para o problema (5.3). De fato, suponha que $x_1 \neq 0 \neq x_2$ e para todo $i = 3, \dots, n$, $x_i = 0$. As condições de KKT para o problema (5.3) fornecem

$$\begin{cases} x_1 + 2x_1^3 + \bar{\lambda}x_1 & = 0 \\ 2x_2 + 4x_2^3 + \bar{\lambda}x_2 & = 0 \\ x_1^2 + x_2^2 & = 0, \end{cases}$$

onde $\bar{\lambda}$ é o multiplicador de Lagrange. Como $x_1 \neq 0 \neq x_2$ temos

$$\begin{cases} 1 + 2x_1^2 = 2 + 4x_2^2 \\ x_1^2 + x_2^2 = 0. \end{cases}$$

Segue que

$$x_1 = \frac{\sqrt{5}}{\sqrt{6}}, \quad x_2 = \frac{1}{\sqrt{6}} \quad \text{e} \quad \bar{\lambda} = \frac{8}{3}.$$

O Algoritmo m_1 convergiu para $f(x^*) = 1.916666\dots$ em todas as dimensões testadas.

Na Tabela 5.4 apresentamos resultados numéricos obtidos pelo Algoritmo m_1 aplicado ao problema (5.3). Na primeira coluna temos a dimensão do problema, na segunda o número de pontos de interpolação $m = 2n + 3$, na coluna marcada com

AF temos o número de avaliações de função objetivo, a coluna com a marca f indica o valor da função objetivo, o tempo de CPU na quinta coluna é em segundos, a sexta coluna disponibiliza o número de subproblemas que o ALGENCAN resolveu e na última coluna o número de construções de modelo (chamadas para a Etapa 1). O ponto inicial para o problema 395 da coleção HS não é viável. Iniciamos o algoritmo no ponto viável

$$x^1 = \left(\frac{\sqrt{n}}{n}, \frac{\sqrt{n}}{n}, \dots, \frac{\sqrt{n}}{n} \right)^T$$

para evitar o tempo com a restauração da viabilidade.

Nas 11 dimensões testadas o Algoritmo m_1 foi capaz de determinar um ponto KKT, com uma quantidade razoável de avaliações de função objetivo. Neste problemas, o tempo de CPU aumentou rapidamente até a dimensão $n = 400$, e manteve um tempo inferior a 2 horas inclusive para o problema em dimensão $n = 1000$. Na máquina utilizada nos testes, para $n = 1000$ são necessários 6 minutos para cada vez que o algoritmo executa a Etapa 1 que é $O(m^3)$ operações. Como o algoritmo fez isto 6 vezes (ver última coluna na Tabela 5.4), gastou pouco mais de meia hora em construções de modelos. A Etapa 1 é onde o tempo no algoritmo é de fato gasto, pois os demais processos são $O(m^2)$ operações. Assim, o principal consumo de tempo na execução é durante as soluções dos subproblemas. Na Tabela 5.4 observamos que foram 14 subproblemas resolvidos pelo ALGENCAN quando $n = 1000$.

n	m	# AF	f	tempo(s)	# ALG	# E1
100	203	1226	1.91666667E+00	3.066E+0	13	6
200	403	2832	1.91666667E+00	3.563E+1	17	7
300	603	4231	1.91666667E+00	2.926E+2	16	7
400	803	5631	1.91666666E+00	1.062E+3	16	7
500	1003	6030	1.91666666E+00	6.645E+2	17	6
600	1203	7230	1.91666667E+00	1.348E+3	17	6
700	1403	8429	1.91666667E+00	2.261E+3	16	6
800	1603	9628	1.91666667E+00	3.770E+3	15	6
900	1803	12630	1.91666667E+00	5.534E+3	15	7
1000	2003	12027	1.91666667E+00	7.084E+3	14	6
1100	2203	15432	1.91666667E+00	1.923E+04	17	7

Tabela 5.4: Resultados numéricos obtidos pelo Algoritmo m_1 aplicado ao problema (5.3)

Problema 2

Este problema teste denominado ARWHEAD foi retirado de [14]. Consiste em minimizar uma função sujeita à restrição de caixa, ou seja,

$$\begin{aligned} \text{minimizar} \quad & f(x) = \sum_{i=1}^{n-1} (x_i^2 + x_n^2)^2 - 4x_i + 3 \\ \text{sujeita a} \quad & \|x\|_\infty \leq 30. \end{aligned} \tag{5.4}$$

Na Tabela 5.5 apresentamos resultados numéricos obtidos pelo Algoritmo m_1 aplicado ao problema (5.4). O ponto viável inicial $x^1 = (0, 0, \dots, 0)^T$ foi utilizado e o algoritmo obteve em todas as dimensões testadas a solução $(1, \dots, 1, 0)^T$. É fácil ver que $\nabla f(1, \dots, 1, 0) = 0$ e que $f(1, \dots, 1, 0) = 0$.

Para $n = 1000$ o Algoritmo m_1 demorou aproximadamente 1 hora e 10 minutos para resolver o problema. Foram 8 construções pela Etapa 1, ou seja, em torno de 48 minutos, e o restante do tempo de execução foi gasto essencialmente pelo ALGENCAN nos 19 subproblemas.

n	m	# AF	f	tempo(s)	# ALG	# E1
100	203	1638	1.77635684E-15	3.225E+0	18	8
200	403	3238	0.00000000E+00	7.848E+1	18	8
300	603	4839	0.00000000E+00	1.656E+2	19	8
400	803	5635	0.00000000E+00	4.459E+2	18	7
500	1003	7035	0.00000000E+00	3.794E+2	18	7
600	1203	8435	0.00000000E+00	6.899E+2	18	7
700	1403	9835	0.00000000E+00	1.036E+3	18	7
800	1603	12839	0.00000000E+00	4.164E+3	19	8
900	1803	14439	0.00000000E+00	5.200E+3	19	8
1000	2003	16039	0.00000000E+00	4.180E+3	19	8
1100	2203	17639	0.00000000E+00	5.440E+3	19	8

Tabela 5.5: Resultados numéricos obtidos pelo Algoritmo m_1 aplicado ao problema (5.4)

Problema 3

Este problema foi sugerido por [11] para discutir o fenômeno denominado *greediness* (voracidade), que consiste na atração à minimizadores irrestritos por algoritmos que usam critérios de mérito que combinam função objetivo e penalidades da viabilidade. O problema consiste em

$$\begin{aligned}
 &\text{minimizar} && f(x) = \sum_{i=1}^n \varphi(x_i) \\
 &\text{sujeita a} && \sum_{i=1}^n x_i^2 \leq 1,
 \end{aligned} \tag{5.5}$$

onde

$$\varphi(t) = \begin{cases} \log(\cos(t)) & \text{se } \cos(t) > 0 \\ -10^{30} & \text{caso contrário.} \end{cases}$$

No artigo [11] os autores utilizam uma versão de 2007 do ALGENCAN que falha ao resolver este problema, no entanto, a versão 2.2.1 de 2009 que utilizamos em nossos subproblemas fornece a solução com valor da função objetivo disponibilizado na última coluna da Tabela 5.6, onde apresentamos também os resultados numéricos obtidos pelo Algoritmo m_1 aplicado ao problema (5.5). Em [64] os autores determinam o valor -0.61562 para o caso $n = 100$. Obtivemos aproximadamente este mesmo valor em

todas as dimensões testadas. Utilizamos o ponto inicial $x^1 = (1/n, 1/n, \dots, 1/n)^T$, que é viável e sugerido em [11].

As soluções que encontramos foram confirmadas pelo Algoritmo ALGENCAN, mas para os testes tivemos que retirar o -10^{30} na definição da f . Quando mantivemos o -10^{30} na definição do problema, e partimos da solução que encontramos (cujo valor da função é aproximadamente -0.61562), o ALGENCAN 2.2.1 não consegue enxergar esta solução e devolve a informação de que não foi possível resolver o problema. Deve ter ocorrido a atração ao minimizador irrestrito da função Lagrangiana, ou seja, o efeito *greediness*.

n	m	# AF	f	tempo(s)	# ALG	# E1	ALGENCAN
100	203	1635	-6.15626470E-1	1.012E+1	15	8	-5.0083E-1
200	403	2831	-6.15626470E-1	2.463E+1	14	7	-5.0041E-1
300	603	4835	-6.15626470E-1	1.547E+2	15	8	-5.0027E-1
400	803	6434	-6.15626470E-1	3.319E+2	15	8	-5.0020E-1
500	1003	8033	-6.15626470E-1	4.341E+2	15	8	-5.0016E-1
600	1203	9635	-6.15626470E-1	1.486E+3	16	8	-5.0013E-1
700	1403	11234	-6.15626471E-1	1.079E+3	15	8	-5.0013E-1
800	1603	16041	-6.15626470E-1	4.153E+3	16	10	-5.0010E-1
900	1803	16237	-6.15626470E-1	4.041E+3	15	9	-5.0009E-1
1000	2003	16034	-6.15626471E-1	3.911E+3	14	8	-5.0009E-1
1100	2203	19838	-6.15626470E-1	6.641E+3	16	9	-5.0009E-1

Tabela 5.6: Resultados numéricos obtidos pelo Algoritmo m_1 aplicado ao problema (5.5)

Conclusões

Nesta tese apresentamos duas contribuições na área da otimização sem derivadas. A primeira refere-se à proposta de um algoritmo globalmente convergente de região de confiança sem derivadas para minimização em conjuntos convexos e fechados. O algoritmo tem uma estrutura simples, que deixa liberdade na construção dos modelos e na resolução dos subproblemas. Com hipóteses razoáveis em otimização sem derivadas, provamos que todo ponto de acumulação da sequência gerada pelo algoritmo é estacionário de primeira ordem. A estrutura de região de confiança sem derivadas para problemas restritos essencialmente foi abordada por Conn, Scheinberg e Toint em [20] e por Powell em [56], mas até onde sabemos na literatura não constam resultados de convergência.

A outra contribuição é relativa ao desenvolvimento de um algoritmo de região de confiança sem derivadas em que as restrições são tratadas nos subproblemas. Os modelos são obtidos por interpolação polinomial e os subproblemas são resolvidos com ALGENCAN. O algoritmo não exige convexidade do conjunto viável e mostrou-se eficiente e robusto nos experimentos numéricos. O algoritmo resolveu aproximadamente 97% de todos os 216 problemas com restrições de igualdade ou desigualdade da coleção Hock-Schittkowski, encontrando a solução global em aproximadamente 90% dos casos. A versão denominada Algoritmo m_1 , que deixa liberdade nos parâmetros do modelo, foi capaz de tratar problemas restritos considerados de grandes dimensões no contexto de otimização sem derivadas. Tais problemas foram resolvidos utilizando um número razoável de avaliações de função objetivo em aceitável tempo computacional.

Referências Bibliográficas

- [1] R. Andreani, J. M. Martínez e M. L. Schuverdt. On the relation between the constant positive linear dependence condition and quasinormality constraint qualification. *Journal of Optimization Theory and Applications*, 125:473–485, 2005.
- [2] R. Andreani, E. G. Birgin, J. M. Martínez e M. L. Schuverdt. On Augmented Lagrangian methods with general lower-level constraints. *SIAM Journal on Optimization*, 18:1286–1309, 2007.
- [3] R. Andreani, E. G. Birgin, J. M. Martínez e M. L. Schuverdt. Augmented Lagrangian methods under the constant positive linear dependence constraint qualification. *Mathematical Programming*, 111:5–32, 2008.
- [4] B. M. Arouxet, N. Echebest e A. Pilotta. Active-set strategy in Powell’s method for optimization without derivatives. *Computational and Applied Mathematics*, 30(1):171–196, 2011.
- [5] A. S. Bandeira, K. Scheinberg e L. N. Vicente. Computation of sparse low degree interpolating polynomials and their application to derivative-free optimization. Disponível em Optimization Online, 2012.
- [6] H. Bersini e F. V. Berghen. CONDOR, a new parallel, constrained extension of Powell’s UOBYQA algorithm: Experimental results and comparison with the DFO algorithm. *Journal of Computational and Applied Mathematics*, 181(1):157–175, 2005.
- [7] D. P. Bertsekas. *Convex Analysis and Optimization*. Athena Scientific, 2003.
- [8] E. G. Birgin, R. Castillo e J. M. Martínez. Numerical comparison of Augmented Lagrangian algorithms for nonconvex problems. *Computational Optimization and Applications*, 31:31–56, 2005.
- [9] E. G. Birgin e J. M. Martínez. Large-scale active-set box-constrained optimization method with spectral projected gradients. *Computational Optimization and Applications*, 23:101–125, 2002.
- [10] L. F. Bueno, A. Friedlander, J. M. Martínez e F. N. C. Sobral. Inexact restoration method for derivative-free optimization with smooth constraints. *SIAM Journal on Optimization*, to appear.

- [11] E. V. Castelani, A. L. M. Martinez, J. M. Martínez e B. F. Svaiter. Addressing the greediness phenomenon in Nonlinear Programming by means of Proximal Augmented Lagrangians. *Computational Optimization and Applications*, 46(2):229–245, 2010.
- [12] P. G. Ciarlet e P. A. Raviart. General Lagrange and Hermite interpolation in \mathbb{R}^n with applications to finite elements methods. *Arch. Rational Mech. Anal*, 46:177–199, 1972.
- [13] P. D. Conejo, E. W. Karas, L. G. Pedroso, A. A. Ribeiro e M. Sachine. Global convergence of trust-region algorithms for constrained minimization without derivatives. Disponível em Optimization Online, 2012.
- [14] A. R. Conn, N. Gould, M. Lescrenier e Ph. L. Toint. Performance of a multifrontal scheme for partially separable optimization. Em J. P. Hennart e S. Gomez, editores, *Advances in Optimization and Numerical Analysis*, páginas 79–96. Kluwer, Dordrecht, 1994.
- [15] A. R. Conn, N. I. M. Gould, A. Sartenaer e Ph. L. Toint. Convergence properties of minimization algorithms for convex constraints using a structured trust region. *SIAM Journal on Optimization*, 6(4):1059–1086, 1996.
- [16] A. R. Conn, N. I. M. Gould e Ph. L. Toint. *LANCELOT: a Fortran Package for Large-Scale Nonlinear Optimization*. Springer Series in Computational Mathematics 17. Springer Verlag, New York, 1992.
- [17] A. R. Conn, N. I. M. Gould e Ph. L. Toint. Global convergence of a class of trust region algorithms for optimization with simple bounds. *SIAM Journal on Numerical Analysis*, 25(2):433–460, 1988.
- [18] A. R. Conn, N. I. M. Gould e Ph. L. Toint. *Trust-Region Methods*. MPS-SIAM Series on Optimization. SIAM, Philadelphia, 2000.
- [19] A. R. Conn, K. Scheinberg e Ph. L. Toint. On the Convergence of Derivative-Free Methods for Unconstrained Optimization. Em M. D. Buhmann e A. Iserles, editores, *Approximation Theory and Optimization: Tributo a M. J. D. Powell*, páginas 83–108. Cambridge University Press, 1997.
- [20] A. R. Conn, K. Scheinberg e Ph. L. Toint. A Derivative Free Optimization Algorithm in Practice. Em *Proceedings of the AIAA Conference*, St Louis, 1998.
- [21] A. R. Conn, K. Scheinberg e L. N. Vicente. Geometry of interpolation sets in derivative free optimization. *Mathematical Programming*, 111:141–172, 2008.
- [22] A. R. Conn, K. Scheinberg e L. N. Vicente. *Introduction to Derivative-Free Optimization*. MPS-SIAM Series on Optimization. SIAM, Philadelphia, 2009.
- [23] A. R. Conn e Ph. L. Toint. An Algorithm using Quadratic Interpolation for Unconstrained Derivative Free Optimization. Em G. Di Pillo e F. Gianessi, editores, *Nonlinear Optimization and Applications*, páginas 27–47. Plenum, 1996.

- [24] D. Coppersmith e S. Winograd. On the asymptotic complexity of matrix multiplication. *SIAM Journal on Computing*, 11(3):472–492, 1982.
- [25] P. J. Davis. *Interpolation and Approximation*. Blaisdell, New York, 1963.
- [26] M. A. Diniz-Ehrhardt, J. M. Martínez e L. G. Pedroso. Derivative-free methods for nonlinear programming with general lower-level constraints. *Computational and Applied Mathematics*, 30:19–52, 2011.
- [27] C. Audet e J. E. Dennis. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization*, 17:188–217, 2006.
- [28] E. D. Dolan e J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2009.
- [29] W. Hock e K. Schittkowski. Test examples for nonlinear programming codes. *Lecture Notes in Economics and Mathematical Systems*, Springer, 187, 1981.
- [30] W. Hock e K. Schittkowski. An updated set of 306 test problems for nonlinear programming with validated optimal solutions - User's Guide. Disponível em Optimization Online, 2009.
- [31] J. D. Griffin e T. G. Kolda. Nonlinearly constrained optimization using heuristic penalty methods and asynchronous parallel generating set search. *Applied Mathematics Research Express*, 1:36–62, 2010.
- [32] G. Fasano, J. L. Morales e J. Nocedal. On the geometry phase in model-based algorithms for derivative-free optimization. *Optimization Methods and Software*, 24:145–154, 2009.
- [33] P. E. Gill, W. Murray, M. A. Saunders e M. H. Wright. User's guide for NPSOL (version 4.0): A fortran package for nonlinear programming. Technical Report SOL 86-2, Department of Operations Research, USA, 1986.
- [34] C. C. Gonzaga, E. W. Karas e M. Vanti. A globally convergent filter method for nonlinear programming. *SIAM J. Optimization*, 14(3):646–669, 2003.
- [35] S. Gratton, Ph. L. Toint e A. Tröltzsch. An active set trust-region method for derivative-free nonlinear bound-constrained optimization. *Optimization Methods and Software*, 26:873–894, 2011.
- [36] T. G. Kolda, R. M. Lewis e V. Torczon. A Generating Set Direct Search Augmented Lagrangian Algorithm for Optimization with a Combination of General and Linear Constraints. Technical Report SAND2006-5315, Sandia National Laboratories, 2006.
- [37] T. G. Kolda, R. M. Lewis e V. Torczon. Stationarity results for generating set search for linearly constrained optimization. *SIAM Journal on Optimization*, 17(4):943–968, 2006.
- [38] R. M. Lewis e V. Torczon. Pattern search algorithms for linearly constrained minimization. *SIAM Journal on Optimization*, 10:917–941, 2000.

- [39] R. M. Lewis e V. Torczon. Active set identification for linearly constrained minimization without explicit derivatives. *SIAM Journal on Optimization*, 20(3):1378–1405, 2009.
- [40] S. Lucidi e M. Sciandrone. On the global convergence of derivative-free methods for unconstrained optimization. *SIAM Journal on Optimization*, 13:97–116, 2002.
- [41] J. M. Martínez. *Otimização Prática Utilizando o Lagrangiano Aumentado*. IMECC, 2009.
- [42] J. J. Moré e D. C. Sorensen. Computing a trust region step. *SIAM J. Sci. Statist. Comput.*, 4:553–572, 1983.
- [43] J. J. Moré e S. M. Wild. Benchmarking derivative-free optimization algorithms. *SIAM Journal on Optimization*, 20(1):172–191, 2009.
- [44] J. Nocedal e S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer-Verlag, 1999.
- [45] L. G. Pedroso. Programação não linear sem derivadas. Tese de doutorado, Universidade de Campinas, 2009.
- [46] T. D. Plantenga. HOPSPACK 2.0 user manual. Technical Report SAND2009-6265, Sandia National Laboratories, Albuquerque, NM e Livermore, CA, 2009.
- [47] M. J. D. Powell. A Direct Search Optimization Method that Models the Objective and Constraint Functions by Linear Interpolation. Em S. Gomez e J. P. Hennart, editores, *Advances in Optimization and Numerical Analysis*, páginas 51–67. Kluwer Academic, Dordrecht, 1994.
- [48] M. J. D. Powell. UOBYQA: unconstrained optimization by quadratic approximation. *Math. Programming*, 92:555–582, 2002.
- [49] M. J. D. Powell. Least frobenius norm updating of quadratic models that satisfy interpolation conditions. *Math. Programming B*, 100:183–215, 2004.
- [50] M. J. D. Powell. On the lagrange functions of quadratic models that are defined by interpolation. páginas 56–78. Science, Beijing, 2004.
- [51] M. J. D. Powell. On updating the inverse of a KKT matrix. *Numerical Linear Algebra and Optimization*, ed. Ya-Xiang Yuan, Science Press (Beijing), páginas 56–78, 2004.
- [52] M. J. D. Powell. The NEWUOA software for unconstrained optimization without derivatives. Em G. Di Pillo e M. Roma, editores, *Large-Scale Nonlinear Optimization*, páginas 255–297. Springer, New York, 2006.
- [53] M. J. D. Powell. Developments of NEWUOA for minimization without derivatives. *IMA Journal Numerical Analysis*, 28:649–664, 2008.
- [54] M. J. D. Powell. The BOBYQA Algorithm for Bound Constrained Optimization without Derivatives. Technical Report DAMTP 2009/NA06, Department of Applied Mathematics and Theoretical Physics, Cambridge, England, August 2009.

- [55] M. J. D. Powell. Beyond symmetric Broyden for updating quadratic models in minimization without derivatives. *Mathematical Programming*. Série A, a aparecer.
- [56] M. J. D. Powell. On Derivative-Free Optimization with Linear Constraints. Em *21st ISMP*, Berlin, Germany, 2012.
- [57] M. J. D. Powell. On the convergence of trust region algorithms for unconstrained minimization without derivatives. *Computational Optimization and Applications*, 53:527-555, 2012.
- [58] L. Qi e Z. Wei. On the constant positive linear dependence condition and its application to SQP methods. *SIAM Journal on Optimization*, 10:963–981, 2000.
- [59] A. A. Ribeiro e E. W. Karas. *Um Curso de Otimização*. Cengage Learning, São Paulo, Brazil. Em Português. A aparecer.
- [60] R. T. Rockafellar. Generalized directional derivatives and subgradients of nonconvex functions. *Canadian Journal of Mathematics*, 32:157–180, 1980.
- [61] K. Scheinberg e Ph. L. Toint. Self-correcting geometry in model-based algorithms for derivative-free unconstrained optimization. *SIAM Journal on Optimization*, 20(6):3512–3532, 2010.
- [62] K. Schittkowski. NLPQL: A Fortran subroutine solving constrained non-linear programming problems. *Annals of Operations Research*, 5:485–500, 1986.
- [63] K. Schittkowski. NLPQLP: A fortran implementation of a sequential quadratic programming algorithm with distributed and non-monotone line search - User's Guide, version 2.2. Technical report, Department of Computer Science, University of Bayreuth, 2006.
- [64] F. N. C. Sobral. Otimização sem derivadas em conjuntos magros. Tese de doutorado, Universidade de Campinas, 2012.
- [65] R. M. Lewis e V. Torczon T. G. Kolda. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Review*, 45(3):385–482, 2003.
- [66] A. Tröltzsch. *An Active-set Trust-Region Method for Bound-Constrained Nonlinear Optimization without Derivatives Applied to Noisy Aerodynamic Design Problems*. Tese de doutorado, Université de Toulouse, 2011.
- [67] A. Verdério. Convergência global de um método de região de confiança sem derivadas com modelos baseados em interpolação polinomial. Dissertação de mestrado, Universidade Federal do Paraná, 2011.
- [68] D. Winfield. Function minimization by interpolation in a data table. *IMA Journal of Applied Mathematics*, 12(3):339–347, 1973.
- [69] Z. Zaikun. Sobolev seminorm of quadratic functions with applications to derivative-free optimization. Technical Report arXiv:1201.6532, Chinese Academy of Sciences, Beijing, China, 2012.

- [70] <http://www.ime.usp.br/~egbirgin/tango>. Acessado em 11/07/2011.

Apêndice A

Na Tabela 5.7 apresentamos dados de todos os 216 problemas da coleção Hock-Schittkowitz [30] com alguma restrição de igualdade ou desigualdade além de caixa. A coleção apresenta 306 problemas e destes, 90 são irrestritos ou com restrições somente em caixas. Na primeira coluna colocamos o número do problema na coleção Hock-Schittkowitz, na segunda coluna a dimensão n , na terceira coluna o número de restrições (o valor entre parênteses indica a quantidade de restrições que são de igualdade). Nas próximas colunas, colocamos o valor de função e número de avaliações para o Algoritmo m_1 e Algoritmo m_2 , e na última coluna o valor da função no ponto solução determinado pelo Algoritmo SQP - Programação Quadrática Sequencial com derivadas, utilizado para resolver os problemas da coleção Hock-Schittkowitz [30]. O símbolo * indica o problema não resolvido pelos Algoritmos m_1 ou m_2 segundo o critério (5.1).

P	n	# rest	Algoritmo m_1		Algoritmo m_2		SQP
			f	#AF	f	#AF	f
6	2	1 (1)	4.3712E-18	36	4.3282E-18	42	1.8696E-13
7	2	1 (1)	-1.7321E+00	30	-1.7321E+00	35	-1.7321E+00
8	2	2 (2)	-1.0000E+00	26	-1.0000E+00	31	-1.0000E+00
9	2	1 (1)	-5.0000E-01	40	-5.0000E-01	52	-5.0000E-01
10	2	1	-1.0000E+00	28	-1.0000E+00	33	-1.0000E+00
11	2	1	-8.4985E+00	27	-8.4985E+00	32	-8.4985E+00
12	2	1	-3.0000E+01	41	-3.0000E+01	42	-3.0000E+01
13	2	1	9.9591E-01	35	9.9591E-01	41	1.0000E+00
14	2	2 (1)	1.3935E+00	20	1.3935E+00	31	1.3935E+00
15	2	2	3.0650E+00	30	3.0650E+00	41	3.0650E+00
16	2	2	3.9821E+00	37	3.9821E+00	33	2.3145E+01
17	2	2	1.0000E+00	1025	1.0000E+00	94	1.0000E+00
18	2	2	5.0000E+00	39	5.0000E+00	45	5.0000E+00
19	2	2	-6.9618E+03	37	-6.9618E+03	43	-6.9618E+03
20	2	3	3.8199E+01	29	3.8199E+01	34	3.8199E+01
21	2	1	-9.9960E+01	35	-9.9960E+01	41	-9.9960E+01
22	2	2	1.0000E+00	27	1.0000E+00	32	1.0000E+00
23	2	5	2.0000E+00	35	2.0000E+00	41	2.0000E+00
24	2	3	-1.0000E+00	37	-1.0000E+00	42	-1.0000E+00
26	3	1 (1)	4.1473E-13	159	2.1108E-12	170	7.4474E-08
27	3	1 (1)	4.0000E+00	107	4.0000E+00	91	4.0000E+00
28	3	1 (1)	6.7635E-18	61	2.8454E-15	67	3.0998E-13
29	3	1	-2.2627E+01	89	-2.2627E+01	70	-2.2627E+01
30	3	1	1.0000E+00	59	1.0000E+00	65	1.0000E+00
31	3	1	6.0000E+00	58	6.0000E+00	64	6.0000E+00
32	3	2 (1)	1.0000E+00	63	1.0000E+00	69	1.0000E+00
33	3	2	-4.0000E+00	59	-4.0000E+00	65	-4.0000E+00

34	3	2	-8.3403E-01	62	-8.3403E-01	68	-8.3403E-01
35	3	1	1.1111E-01	50	1.1111E-01	55	1.1111E-01
36	3	1	-3.3000E+03	62	-3.3000E+03	68	-3.3000E+03
37	3	2	-3.4560E+03	151	-3.4560E+03	164	-3.4560E+03
39	4	2 (2)	-1.0000E+00	58	-1.0000E+00	78	-1.0000E+00
40	4	3 (3)	-2.5000E-01	59	-2.5000E-01	77	-2.5000E-01
41	4	1 (1)	1.9259E+00	170	1.9259E+00	81	1.9259E+00
42	4	2 (2)	1.3858E+01	71	1.3858E+01	95	1.3858E+01
43	4	3	-4.4000E+01	72	-4.4000E+01	96	-4.4000E+01
44	4	6	-1.5000E+01	84	-1.5000E+01	97	-1.5000E+01
46	5	2 (2)	9.4252E-12	219	5.0080E-12	370	5.5456E-07
47	5	3 (3)	6.8678E-17	122	5.9911E-17	182	9.4223E-10
48	5	2 (2)	6.1937E-09	112	2.5484E-15	133	1.6289E-16
49	5	2 (2)	5.6396E-12	254	8.8253E-12	381	2.2836E-05
50	5	3 (3)	1.3096E-17	140	4.8460E-21	231	3.1211E-09
51	5	3 (3)	2.5447E-16	84	2.5447E-16	132	9.6183E-17
52	5	3 (3)	5.3266E+00	85	5.3266E+00	133	5.3266E+00
53	5	3 (3)	4.0930E+00	84	4.0930E+00	132	4.0930E+00
54	6	1 (1)	-9.0284E-01	939	-8.8699E-01	6941	-7.2242E-34
55	6	6 (6)	6.6667E+00	76	6.6667E+00	141	6.6667E+00
56	7	4 (4)	-3.4560E+00	276	-3.4560E+00	300	-3.4560E+00
57	2	1	2.8460E-02	65	2.8460E-02	60	3.0646E-02
58	2	2	3.1903E+00	2998	3.1903E+00	65	3.1903E+00
59	2	3	-6.7546E+00	46	-6.7546E+00	57	-6.7546E+00
60	3	1 (1)	3.2568E-02	63	3.2568E-02	69	3.2568E-02
61	3	2 (2)	-1.4365E+02	122	-1.4365E+02	133	-1.4365E+02
62	3	1 (1)	-2.6273E+04	63	-2.6273E+04	71	-2.6273E+04
63	3	2 (2)	9.6172E+02	60	9.6172E+02	53	9.6172E+02
64	3	1	6.2998E+03	98	6.2998E+03	138	6.2998E+03
65	3	1	9.5353E-01	62	9.5353E-01	68	9.5353E-01
66	3	2	5.1816E-01	49	5.1816E-01	54	5.1816E-01
67	3	14	-1.1620E+03	117	-1.1620E+03	106	-1.1620E+03
68	4	2 (2)	-9.2043E-01	447	* 2.4000E-05	244	-9.2043E-01
69	4	2 (2)	-9.5671E+02	386	* 4.0000E-03	240	-9.5671E+02
70	4	1	7.5026E-03	6663	7.4985E-03	414	7.4985E-03
71	4	2 (1)	1.7014E+01	75	1.7014E+01	80	1.7014E+01
72	4	2	7.2768E+02	77	7.2768E+02	102	7.2768E+02
73	4	3 (1)	2.9894E+01	59	2.9894E+01	79	2.9894E+01
74	4	5 (3)	5.1265E+03	80	5.1265E+03	104	5.1265E+03
75	4	5 (3)	5.1744E+03	78	5.1744E+03	102	5.1744E+03
76	4	3	-4.6818E+00	73	-4.6818E+00	96	-4.6818E+00
77	5	2 (2)	2.4151E-01	137	2.4151E-01	298	2.4151E-01
78	5	3 (3)	-2.9197E+00	152	-2.9197E+00	108	-2.9197E+00
79	5	3 (3)	7.8773E-02	152	7.8773E-02	214	7.8777E-02
80	5	3 (3)	5.3950E-02	73	5.3950E-02	108	5.3950E-02
81	5	3 (3)	5.3950E-02	130	5.3950E-02	178	5.3950E-02
83	5	6	-3.0666E+04	87	-3.0666E+04	135	-3.0666E+04
84	5	6	-5.2803E+01	104	-5.2803E+01	135	-5.2803E+06
85	5	38	-1.9052E+00	107	-1.9052E+00	161	-1.9052E+00
86	5	10	-3.2349E+01	93	-3.2349E+01	132	-3.2349E+01

87	6	4 (4)	8.9276E+03	120	8.9276E+03	210	8.9276E+03
88	2	1	1.3627E+00	27	1.3627E+00	32	1.3627E+00
89	3	1	1.3627E+00	48	1.3627E+00	53	1.3627E+00
90	4	1	1.3627E+00	58	1.3627E+00	78	1.3627E+00
91	5	1	1.3627E+00	68	1.3627E+00	108	1.3627E+00
92	6	1	1.3627E+00	79	1.3627E+00	144	1.3627E+00
93	6	2	1.3508E+02	166	1.3508E+02	178	1.3508E+02
95	6	4	1.5620E-02	77	1.5620E-02	142	1.5620E-02
96	6	4	1.5620E-02	77	1.5620E-02	142	1.5620E-02
97	6	4	3.1358E+00	94	3.1358E+00	172	3.1358E+00
98	6	4	3.1358E+00	94	3.1358E+00	172	3.1358E+00
99	7	2 (2)	-8.3108E+08	89	-8.3108E+08	183	-8.3108E+08
100	7	4	6.8063E+02	113	6.8063E+02	264	6.8063E+02
101	7	6	1.8098E+03	595	1.8098E+03	468	1.8098E+03
102	7	6	9.1188E+02	463	9.1188E+02	313	9.1188E+02
103	7	6	5.4367E+02	240	5.4367E+02	653	5.4367E+02
104	8	6	3.9512E+00	272	3.9512E+00	282	3.9512E+00
105	8	1	1.1384E+03	11140	1.1384E+03	2000	1.1384E+03
106	8	6	7.0492E+03	133	7.0492E+03	289	7.0492E+03
107	9	6 (6)	5.0550E+03	108	5.0550E+03	278	5.0550E+03
108	9	13	-8.6603E-01	135	-8.6603E-01	504	-8.6603E-01
109	9	10 (6)	5.3621E+03	163	5.3621E+03	349	5.3621E+03
111	10	3 (3)	-4.7761E+01	191	-4.7761E+01	467	-4.7761E+01
112	10	3 (3)	* -4.7571E+01	150	* -4.7689E+01	341	-4.7761E+01
113	10	8	2.4306E+01	146	2.4306E+01	404	2.4306E+01
114	10	11 (3)	-1.7688E+03	181	-1.7688E+03	1486	-1.7688E+03
116	13	15	9.7591E+01	188	9.7591E+01	645	9.7591E+01
117	15	5	3.2349E+01	1928	3.2349E+01	2213	3.2349E+01
118	15	29	6.6482E+02	207	6.6482E+02	825	6.6482E+02
119	16	8 (8)	2.4490E+02	299	2.4490E+02	1238	2.4490E+02
215	2	1	-6.1520E-12	34	3.3367E-12	40	-2.6592E-09
216	2	1 (1)	9.9938E-01	32	9.9938E-01	36	9.9938E-01
217	2	2 (1)	-8.0000E-01	26	-8.0000E-01	31	-8.0000E-01
218	2	1	0.0000E+00	40	0.0000E+00	46	0.0000E+00
219	4	2 (2)	-1.0000E+00	57	-1.0000E+00	77	-1.0000E+00
220	2	1	1.0000E+00	49	1.0000E+00	55	3.0240E+01
221	2	1	-1.0000E+00	29	-1.0000E+00	34	-1.0000E+00
222	2	1	-1.5000E+00	34	-1.5000E+00	40	-1.5000E+00
223	2	2	-8.3403E-01	37	-8.3403E-01	43	-8.3403E-01
224	2	4	-3.0400E+02	36	-3.0400E+02	42	-3.0400E+02
225	2	5	2.0000E+00	34	2.0000E+00	40	2.0000E+00
226	2	2	-5.0000E-01	54	-5.0000E-01	33	-5.0000E-01
227	2	2	1.0000E+00	28	1.0000E+00	33	1.0000E+00
228	2	2	-3.0000E+00	35	-3.0000E+00	41	-3.0000E+00
230	2	2	3.7500E-01	27	3.7500E-01	32	3.7500E-01
231	2	2	2.9322E-07	5396	8.1299E-17	210	6.5498E-09
232	2	3	-1.0000E+00	34	-1.0000E+00	41	-1.0000E+00
233	2	1	1.8728E-05	83	4.8052E-17	78	4.5193E-09
234	2	1	-8.0000E-01	87	-8.0000E-01	33	-8.0000E-01
235	3	1 (1)	4.0000E-02	59	4.0000E-02	65	4.0000E-02

236	2	2	-5.8903E+01	40	-5.8903E+01	53	-5.8903E+01
237	2	3	-5.8903E+01	49	-5.8903E+01	54	-5.8903E+01
238	2	3	-5.8903E+01	40	-5.8903E+01	53	-5.8903E+01
239	2	1	-5.8903E+01	40	-5.8903E+01	53	-5.8903E+01
248	3	2 (1)	-8.0000E-01	59	-8.0000E-01	65	-8.0000E-01
249	3	1	1.0000E+00	58	1.0000E+00	64	1.0000E+00
250	3	2	-3.3000E+03	61	-3.3000E+03	67	-3.3000E+03
251	3	1	-3.4560E+03	157	-3.4560E+03	92	-3.4560E+03
252	3	1 (1)	4.0000E-02	86	4.0000E-02	95	4.0000E-02
253	3	1	6.9282E+01	73	6.9282E+01	79	6.9282E+01
254	3	2 (2)	-1.7321E+00	46	-1.7321E+00	51	-1.7321E+00
262	4	4 (1)	-1.0000E+01	74	-1.0000E+01	74	-1.0000E+01
263	4	4 (2)	-1.0000E+00	58	-1.0000E+00	78	-1.0000E+00
264	4	3	-4.4000E+01	71	-4.4000E+01	95	-4.4000E+01
265	4	2 (2)	1.9036E+00	55	1.9036E+00	75	1.9036E+00
268	5	5	* 5.9475E-01	2642	2.3341E-12	133	1.3919E-07
269	5	3 (3)	4.0930E+00	83	4.0930E+00	131	4.0930E+00
270	5	1	* 1.0639E-10	101	* 1.5126E-11	135	-1.0000E+00
277	4	4	5.0762E+00	59	5.0762E+00	79	5.0762E+00
278	6	6	7.8385E+00	112	7.8385E+00	174	7.8385E+00
279	8	8	1.0606E+01	120	1.0606E+01	322	1.0606E+01
280	10	10	1.3375E+01	144	1.3375E+01	604	1.3375E+01
284	15	10	-1.8400E+03	202	-1.8400E+03	820	-1.8400E+03
285	15	10	-8.2520E+03	202	-8.2520E+03	820	-8.2520E+03
315	2	3	-8.0000E-01	35	-8.0000E-01	41	-8.0000E-01
316	2	1 (1)	3.3431E+02	38	3.3431E+02	44	3.3431E+02
317	2	1 (1)	3.7247E+02	36	3.7247E+02	42	3.7247E+02
318	2	1 (1)	4.1275E+02	37	4.1275E+02	43	4.1275E+02
319	2	1 (1)	4.5240E+02	37	4.5240E+02	43	4.5240E+02
320	2	1 (1)	4.8553E+02	37	4.8553E+02	43	4.8553E+02
321	2	1 (1)	4.9611E+02	37	4.9611E+02	43	4.9611E+02
322	2	1 (1)	4.9996E+02	37	4.9996E+02	43	4.9996E+02
323	2	2	3.7989E+00	28	3.7989E+00	33	3.7989E+00
324	2	2	5.0000E+00	38	5.0000E+00	44	5.0000E+00
325	2	3 (1)	3.7913E+00	35	3.7913E+00	41	3.7913E+00
326	2	2	-7.9808E+01	34	-7.9808E+01	40	-7.9808E+01
327	2	1	2.8460E-02	71	2.8460E-02	58	3.0646E-02
329	2	3	-6.9618E+03	37	-6.9618E+03	43	-6.9618E+03
330	2	1	1.6206E+00	61	1.6206E+00	65	1.6206E+00
331	2	1	4.2584E+00	62	4.2584E+00	37	4.2584E+00
332	2	2	1.1495E+02	29	1.1495E+02	34	1.1495E+02
335	3	2 (2)	-4.4721E-03	46	-4.4721E-03	51	-4.4721E-03
336	3	2 (2)	-3.3790E-01	48	-3.3790E-01	53	-3.3790E-01
337	3	1	6.0000E+00	58	6.0000E+00	64	6.0000E+00
338	3	2 (2)	-7.2057E+00	59	-7.2057E+00	65	-7.2057E+00
339	3	1	3.3617E+00	86	3.3617E+00	94	3.3617E+00
340	3	1	-5.4000E-02	151	-5.4000E-02	64	-5.4000E-02
341	3	1	-2.2627E+01	88	-2.2627E+01	69	-2.2627E+01
342	3	1	-2.2627E+01	89	-2.2627E+01	70	-2.2627E+01
343	3	2	-5.6848E+00	74	-5.6848E+00	70	-5.6848E+00

344	3	1 (1)	3.2568E-02	61	3.2568E-02	68	3.2568E-02
345	3	1 (1)	3.2568E-02	76	3.2568E-02	173	3.2568E-02
346	3	2	-5.6848E+00	74	-5.6848E+00	70	-5.6848E+00
347	3	1 (1)	1.7375E+04	48	1.7375E+04	53	1.7375E+04
348	3	1 (1)	* -	-	* -	-	3.6971E+01
353	4	3 (1)	-3.9934E+01	56	-3.9934E+01	76	-3.9934E+01
354	4	1	1.1378E-01	93	1.1378E-01	97	1.1378E-01
355	4	1 (1)	6.9675E+01	91	6.9675E+01	99	1.2102E+02
356	4	5	2.3812E+00	84	2.3812E+00	113	2.3812E+00
359	5	14	-5.2803E+06	89	-5.2803E+06	138	-5.2803E+06
360	5	2	-5.2803E-01	103	-5.2803E-01	134	-5.2803E-01
361	5	6	*	*	*	*	-7.7641E+05
362	5	4	2.6710E-01	208	2.6760E-01	363	2.7080E-01
364	6	4	* 6.1898E-02	15566	6.0600E-02	429	6.0600E-02
365	7	5	2.3314E+01	169	2.3314E+01	257	2.3314E+01
366	7	14	7.0431E+02	118	7.0431E+02	233	7.0431E+02
367	7	5 (2)	-3.7413E+01	228	-3.7413E+01	223	-3.7413E+01
369	8	6	7.0492E+03	132	7.0492E+03	334	7.0492E+03
372	9	12	1.3390E+04	139	1.3390E+04	343	1.3390E+04
373	9	6 (6)	1.3390E+04	138	1.3390E+04	343	1.3390E+04
374	10	35	2.3326E-01	142	2.3326E-01	400	2.3326E-01
375	10	9 (9)	-1.5161E+01	144	-1.5161E+01	402	-1.5161E+01
376	10	15 (1)	-4.4301E+03	284	-4.4301E+03	1165	-4.4301E+03
377	10	3 (3)	-7.9499E+02	177	-7.9499E+02	476	-7.9499E+02
378	10	3 (3)	-4.7761E+01	852	-4.7761E+01	1223	-4.7761E+01
380	12	3	3.1687E+05	262	3.1682E+05	78304	3.2516E+05
381	13	4 (1)	1.0149E+00	149	1.0149E+00	529	1.0149E+00
382	13	4 (1)	1.0383E+00	148	1.0383E+00	528	1.0383E+00
383	14	1 (1)	7.2859E+00	229	7.2859E+00	849	7.2859E+00
384	15	10	-8.3103E+03	203	-8.3103E+03	821	-8.3103E+03
385	15	10	-8.3153E+03	203	-8.3153E+03	821	-8.3153E+03
386	15	11	-8.1644E+03	203	-8.1644E+03	821	-8.1644E+03
387	15	11	-8.2501E+03	203	-8.2501E+03	821	-8.2501E+03
388	15	15	-5.8211E+03	204	-5.8211E+03	822	-5.8211E+03
389	15	15	-5.8097E+03	204	-5.8097E+03	822	-5.8097E+03
392	30	45	-1.6961E+06	1005	-1.6961E+06	2991	-1.6961E+06
393	48	3 (2)	* 8.7393E-01	1115	8.6338E-01	12271	8.6338E-01
394	20	1 (1)	1.9167E+00	265	1.9167E+00	1860	1.9167E+00
395	50	1 (1)	1.9167E+00	625	1.9167E+00	7963	1.9167E+00

Tabela 5.7: Dados dos 216 problemas com restrições de igualdade ou desigualdade da coleção HS