

ANA PAULA OENING

**UM ALGORITMO DE FILTRO GLOBALMENTE CONVERGENTE
PARA PROGRAMAÇÃO NÃO LINEAR**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre, pelo Programa de Pós-Graduação em Métodos Numéricos em Engenharia, Setores de Ciências Exatas e Tecnologia, Universidade Federal do Paraná.

Orientadora:

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

Co-orientador:

Prof. Dr. Ademir Alves Ribeiro

Curitiba

2006

TERMO DE APROVAÇÃO

*Há perguntas que não têm resposta,
o que é uma lição difícilíssima de aprender.*

Katharine Graham

*Para
Galilei e Lucas.*

Agradecimentos

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

Aos meus orientadores, Elizabeth e Ademir, pela paciência em me ensinar muitos conceitos que deveriam ser básicos, pelo esforço e dedicação que demonstraram, não somente me orientando, mas participando de todo o desenvolvimento dessa pesquisa. E, principalmente, pelo exemplo dos maravilhosos professores que são.

Aos membros da banca, professor Clóvis e professor Matioli, obrigada por aceitarem o convite de avaliar esse trabalho e por fazerem sugestões que somente o enriqueceram.

Minha querida amiga Débora. Ninguém melhor do que você para saber quais foram nossos obstáculos e o quão importante é a nossa vitória. Obrigada por entender minhas manias, aceitar minha teimosia, escutar minhas histórias, participar das minhas crises. Você é a minha companheira inseparável no estudo e amiga indispensável na vida.

Aos meus amigos e colegas do programa. Durante o primeiro ano, nossos grupos de estudo, com as intermináveis listas, foram essenciais para vencer o primeiro trimestre. E, durante todo o curso, a pausa para o cafézinho foi o que muitas vezes refrescou minha cabeça e me deu um novo ânimo. Maris, a melhor secretária que eu conheço, pelo permanente alto astral que a todos contagia.

Aos meus amigos e familiares, de alguma forma todos vocês contribuíram. À minha mãe, que sempre me apoiou. Adriana, por me ouvir e realmente acreditar na minha "inteligência". Lucas e Galilei, que aceitaram a minha ausência e justificaram a minha presença. Meus sinceros agradecimentos.

E, finalmente, mas não menos importante, à Deus.

Resumo

Propomos neste trabalho um algoritmo de filtro globalmente convergente para programação não linear. O algoritmo aqui apresentado combina as idéias de obtenção do passo do algoritmo proposto por Gonzaga, Karas e Vanti, com a definição de filtro segundo Fletcher, Leyffer e Toint. Fazemos a análise de convergência e provamos que a sequência gerada pelo algoritmo tem pelo menos um ponto de acumulação estacionário.

Os métodos de filtro foram desenvolvidos por Fletcher e Leyffer. O filtro define uma região proibida guardando pares ordenados que representam a função objetivo e a inviabilidade e evitam o uso de funções de mérito. Fletcher, Leyffer e Toint usam uma definição modificada do filtro que incorpora uma inclinação na região proibida. Em ambos os trabalhos, a obtenção do passo é baseada em programação quadrática sequencial. Gonzaga, Karas e Vanti propuseram um algoritmo de filtro baseado no método de restauração inexata, no sentido de Martínez e Pilotta. Cada iteração é composta de duas fases. Na primeira, chamada de fase de viabilidade, é reduzida uma medida de inviabilidade. A segunda fase é a fase de otimalidade, onde se reduz a função objetivo numa aproximação tangencial do conjunto viável, evitando perder muito a viabilidade conseguida na primeira fase. Essas fases são independentes e se tem grande liberdade na escolha dos algoritmos usados em cada uma. De maneira geral, pode ser usado qualquer algoritmo interno, desde que satisfaça algumas condições razoáveis sobre sua eficiência.

Abstract

We propose in this work a globally convergent filter algorithm for non-linear programming. The algorithm presented here combines the ideas of the step computation of the algorithm by Gonzaga, Karas and Vanti, with the filter definition given by Fletcher, Leyffer and Toint. We present the convergence analysis and prove that the sequence generated by the algorithm has at least one stationary accumulation point.

Filter methods were introduced by Fletcher and Leyffer, whose aim is to dispense the need for a merit function, a common tool in most algorithms for constrained optimization. The filter defines a forbidden region by memorizing pairs that represent the objective and infeasibility functions. Fletcher, Leyffer and Toint use a modified definition of the filter referred to as the slanting envelope. In these works, the step computation is based on sequential quadratic programming. Gonzaga, Karas and Vanti have proposed a filter algorithm based on inexact restoration methods in the sense of Martínez and Pilotta. Each iteration is composed of a restoration phase, which reduces a measure of infeasibility, and an optimality phase, which reduces the objective function in a tangential approximation of the feasible set. These two phases are totally independent, and the only coupling between them is provided by the filter. The method does not depend on the internal algorithms used in each iteration, as long as these algorithms satisfy reasonable assumptions on their efficiency.

Sumário

Introdução	1
1 Método de região de confiança para problemas irrestritos	6
1.1 O Problema	6
1.2 Método de Região de Confiança	7
1.2.1 Método Dogleg	12
1.2.2 Método de Steihaug	16
2 Problemas com restrições	19
2.1 O Problema	20
2.2 Obtenção do passo	21
2.2.1 Programação Quadrática Sequencial	21
2.2.2 Restauração inexata	28
2.3 Critérios de aceitação do passo	31
3 Métodos de filtro	34
3.1 O conceito de filtro	35
3.1.1 Margens para o Filtro	36
3.2 Algoritmo geral	38
3.3 Alguns algoritmos de filtro existentes	41
3.3.1 Algoritmo de PQS com filtro reto	41
3.3.2 Algoritmo de PQS com filtro inclinado	43
3.3.3 Algoritmo de restauração inexata com filtro reto	44
3.4 Proposta de algoritmo	46
3.4.1 Algoritmo	47
3.4.2 Convergência global	48
3.4.3 Discussão sobre os algoritmos internos	54
Conclusões	64
Referências Bibliográficas	65

Lista de Figuras

1.1	Um passo de região de confiança.	10
1.2	Ponto de Cauchy.	11
1.3	Minimizador é o passo cheio d_B^k	15
1.4	Minimizador é a interseção.	15
1.5	Minimizador é o passo de Cauchy d_C^k	15
2.1	O problema com restrições.	20
2.2	Incompatibilidade do subproblema (PQ^{RC})	26
2.3	Decomposição do passo.	29
2.4	Um passo de restauração inexata.	30
2.5	Comparação entre restauração inexata e PQS.	31
2.6	Função de mérito.	32
2.7	Filtro.	33
2.8	Comparação entre filtro e função de mérito.	33
3.1	Dominância e região proibida.	36
3.2	Margem reta.	37
3.3	Margem inclinada.	37
3.4	Margens.	38
3.5	Vértice do filtro reto.	39
3.6	Vértice do filtro inclinado.	39
3.7	Folga do filtro.	49
3.8	Ponto de Cauchy e melhores que ele.	57

Introdução

Neste trabalho discutimos uma classe de métodos para resolver problemas de programação não linear do tipo

$$(P) \quad \begin{cases} \text{minimizar} & f_0(x) \\ \text{sujeito a :} & f_{\mathcal{E}}(x) = 0 \\ & f_{\mathcal{I}}(x) \leq 0, \end{cases}$$

onde $f_i : \mathbb{R}^n \rightarrow \mathbb{R}, i \in \{0\} \cup \mathcal{E} \cup \mathcal{I}$, são funções continuamente diferenciáveis, com $f_{\mathcal{E}} : \mathbb{R}^n \rightarrow \mathbb{R}^p$ e $f_{\mathcal{I}} : \mathbb{R}^n \rightarrow \mathbb{R}^q$. O problema consiste basicamente em minimizar uma função objetivo sujeito a p restrições de igualdade e q de desigualdade, sendo $m = p + q$. Poderemos considerar $\mathcal{E} = \emptyset$ ou $\mathcal{I} = \emptyset$, quando quisermos estudar apenas problemas com restrições de desigualdades ou de igualdades, respectivamente. Podemos, ainda, considerar $\mathcal{E} = \emptyset$ e $\mathcal{I} = \emptyset$, e estudar um problema irrestrito.

Chamamos de conjunto viável o conjunto dos pontos que satisfazem as restrições, isto é, o conjunto $\Omega = \{x \in \mathbb{R}^n \mid f_{\mathcal{E}}(x) = 0, f_{\mathcal{I}}(x) \leq 0\}$. Qualquer ponto $x \in \Omega$ é chamado de ponto viável. Um ponto x^* é uma solução do problema (P) se é viável e assume menor valor para a função objetivo que os outros pontos pertencentes ao conjunto viável. Encontrar um ponto com tal propriedade é um objetivo ambicioso. Em otimização, nos contentamos em encontrar minimizadores locais, ou seja, pontos $x^* \in \Omega$ que minimizam a função objetivo, em relação a pontos viáveis, numa vizinhança de x^* . Esse conceito não é operacional. Uma forma de identificar minimizadores locais é fazer uso de condições que devem ser necessariamente satisfeitas por tais pontos. Essas condições estabelecem critérios sobre as derivadas da função objetivo e das restrições. Pontos que satisfazem tais condições serão chamados de pontos estacionários. Em nosso trabalho não vamos discutir a teoria geral de otimização e sim alguns métodos que resolvem esse problema. O estudo dessa teoria pode ser encontrado em várias referências, por exemplo [2, 5, 27, 39, 54].

Em casos simples encontrar pontos estacionários do problema é fácil. Mas a medida que o problema se torna mais complexo, pelas características da função objetivo e pelo grande número de variáveis ou de restrições, esse trabalho se torna difícil. Assim fazem-se necessários os métodos numéricos, ou seja, os algoritmos.

Os algoritmos são processos iterativos que geram uma sequência de pontos a partir de um ponto inicial x^0 . Espera-se que tal sequência tenha pelo menos um ponto de acumulação que seja ponto estacionário do problema em questão. Podemos dizer que uma iteração desses algoritmos é composta por duas etapas: obtenção de um passo tentativo e avaliação desse passo.

Problema irrestrito. Quando trabalhamos com um problema irrestrito, caso particular do problema (P) em que $\mathcal{E} = \emptyset$ e $\mathcal{I} = \emptyset$, queremos encontrar um ponto tentativo que diminua o valor da função objetivo em relação ao ponto corrente. Métodos clássicos que resolvem esse problema são os métodos com buscas direcionais. Nestes métodos é definida uma direção, chamada de direção de descida, e feita uma busca linear nessa direção para encontrar o tamanho do passo. Escolhas usuais para a direção do passo são as direções de Cauchy, Newton e quase-Newton. Para encontrar o tamanho do passo existem vários métodos, dentre os quais citamos as buscas de Armijo e Wolfe. Uma discussão detalhada sobre esses métodos, e muitos outros, pode ser encontrada em [2, 5, 39, 54].

Uma outra classe de métodos para resolver o problema irrestrito são os métodos de região de confiança, onde a direção e o tamanho do passo são obtidos simultaneamente. Nesses métodos minimiza-se uma aproximação quadrática da função objetivo sujeita a uma região de confiança, que controla o tamanho do passo. O desenvolvimento desses métodos remonta aos trabalhos de Levenberg [38] e Marquardt [40] para problemas não lineares de mínimos quadrados. Os trabalhos de Powell [56, 57] são de grande importância nesta classe de métodos, pois neles foi estabelecida, pela primeira vez, a convergência global dos métodos de região de confiança. Moré [49] e Moré e Sorensen [50] fazem um tratado sobre os algoritmos de região de confiança. Byrd, Schnabel e Schultz [9, 10] discutem a teoria geral desses métodos. Dennis e Schnabel [15] também discutem esse método em seu trabalho, fornecendo pontos para desenvolvimentos importantes na literatura.

Problema com restrições. Desenvolver métodos para resolver o problema (P) é uma tarefa difícil e que faz parte de contínuas pesquisas, com trabalhos novos surgindo constantemente. Podemos considerar que existem basicamente três categorias de métodos que resolvem esse problema: os métodos de penalidade ou barreira, métodos do gradiente reduzido generalizado ou restauração inexata e programação quadrática sequencial.

Os métodos de penalidade e barreira são procedimentos que resolvem problemas com restrições através de uma sequência de problemas irrestritos, formados pela função objetivo somada a um múltiplo das restrições. A função objetivo do problema irrestrito é chamada de função penalizada, e depende de um parâmetro de penalidade. No clássico livro de Fiacco e McCormick [17] é feito um estudo sobre

essa classe de métodos, de forma geral.

Nos métodos de barreira (ou métodos de penalidade interna), o algoritmo caminha sempre na região viável do problema com restrições, penalizando os pontos viáveis que se aproximem da fronteira do conjunto. Tais métodos são tratados em livros clássicos de otimização, como por exemplo [5, 54]. O interesse por estes métodos ressurgiu na década de 90, com uma explosão de atividades nessa área, após o importante trabalho de Karmakar [36] (veja [14, 34, 37, 53, 63, 70]).

Nos algoritmos que usam a penalidade externa, os passos são dados sempre fora do conjunto viável do problema. Em [39], Luenberger apresenta as propriedades teóricas destes métodos. Nessa classe de métodos, incluem-se também os métodos de lagrangeano aumentado [18].

Nos métodos de restauração inexata a função objetivo e as restrições são tratadas como dois objetivos independentes. Cada iteração do método é composta de uma fase de viabilidade, que diminui uma medida de inviabilidade, e uma fase de otimalidade, que reduz a função objetivo. Tais métodos remontam ao método de gradiente projetado de Rosen [64], gradiente reduzido generalizado (GRG) de Abadie e Carpentier [1] e SRGA de Miele [46, 47, 48]. Uma descrição e discussão das propriedades do método GRG pode se encontrada em [39]. Martínez [41] e Martínez e Pilotta [42] desenvolveram um método de restauração inexata onde as fases viabilidade e otimalidade se tornam mais independentes, com grande liberdade na escolha dos algoritmos usados para esses passos.

Os métodos de programação quadrática sequencial resolvem um problema de programação não linear através de uma sequência de problemas de programação quadrática, cuja função objetivo é uma aproximação quadrática da função objetivo original e as restrições são aproximações lineares das restrições originais. Os métodos de PQS foram propostos, inicialmente, por Wilson [69] em 1963, na sua tese de doutorado, mas foi na década de 70 que esses métodos se tornaram mais conhecidos. Veja Garcia-Palomares e Mangasarian [23], Han [32, 33] e Powell [58, 59, 60]. Em [4], Boggs e Tolle fazem um estudo bem detalhado sobre o método. Uma boa referência em português é [43]. O método pode ser encontrado também em livros como [5, 54].

Métodos que usam função de mérito. Ao resolvermos problemas com restrições nos deparamos com dois objetivos conflitantes: reduzir o valor da função objetivo e obter viabilidade. Diferentemente do problema irrestrito, onde a própria função objetivo avalia a qualidade do passo obtido, nesse caso faz-se necessário algum critério que possa medir o ganho em otimalidade e viabilidade. Uma maneira de fazer isso é utilizar funções de mérito, como proposto por Gomes, Maciel e Martínez [26], Byrd, Gilbert e Nocedal [7] e Byrd, Hribar e Nocedal [8], em trabalhos que utilizam PQS com região de confiança para obtenção do passo. De maneira geral, a função de mérito é uma combinação entre uma medida de otimalidade e uma de inviabilidade.

Martínez [41] e Martínez e Pilotta [42] também utilizam uma função de mérito para avaliar o ponto tentativo, onde o passo é obtido por um método de restauração inexata. Outra forma de utilizar uma função de mérito é dada em [16, 24, 66, 68], onde o método de Newton é aplicado às condições de KKT do problema, obtendo-se uma direção de busca. Então é feita uma busca linear, tipo Armijo, nessa direção, para encontrar um passo que reduza suficientemente a função de mérito.

Apesar de muito utilizadas, as funções de mérito apresentam algumas desvantagens. Não há consenso nas medidas de otimalidade e viabilidade a serem usadas e algumas funções de mérito clássicas são não diferenciáveis. Outra desvantagem, talvez a mais crítica, é a escolha do parâmetro de penalidade. Esta é uma escolha bastante delicada, pois, dependendo do valor desse parâmetro, o algoritmo pode se tornar muito lento ou recusar uma solução ótima.

Métodos de filtro. Fletcher e Leyffer [20] propuseram uma alternativa ao uso das funções de mérito, introduzindo os chamados métodos de filtro que definem uma região proibida guardando pares ordenados que representam a função objetivo e a inviabilidade.

O primeiro resultado de convergência para essa classe de métodos foi estabelecido por Fletcher, Leyffer e Toint [21] e Chin e Fletcher [12], em algoritmos de programação linear sequencial.

Para os métodos de PQS com filtro, a convergência global foi obtida por Fletcher, Leyffer e Toint [22]. Nesse trabalho é assumido que uma solução global dos subproblemas quadráticos, gerados por PQS, é encontrada. Utilizando uma estratégia de decomposição do passo, baseada em Byrd [6] e Omojokun [55], Fletcher, Gould, Leyffer, Toint e Wächter [19] também estabeleceram a convergência global, exigindo apenas que soluções aproximadas dos subproblemas quadráticos sejam encontradas.

Chin [11] propôs em sua tese de doutorado uma modificação na definição do filtro, incorporando uma inclinação na região proibida. Essa modificação foi usada por Chin e Fletcher [12] no contexto de programação linear sequencial e por Fletcher, Leyffer e Toint [22] em programação quadrática sequencial.

Ulbrich, Ulbrich e Vicente [67] definem o filtro de um modo diferente do que foi proposto originalmente em [20]. As entradas do filtro, ao invés da função objetivo e medida de inviabilidade, são o lagrangeano e uma medida de inviabilidade que envolve as variáveis primais e duais. Aliando essa técnica de filtro aos algoritmos de pontos interiores, é estabelecida a convergência global do algoritmo.

Gonzaga, Karas e Vanti [28] propuseram um algoritmo de filtro globalmente convergente para programação não linear, baseado em restauração inexata [41, 42].

As idéias de filtro vem sendo utilizadas na resolução de outros tipos de

problemas. Gould, Leyffer e Toint [29] apresentam um algoritmo de filtro multi-dimensional para resolução de sistemas de equações não lineares. Gould, Sainvitu e Toint [30] estendem as idéias de filtro multi-dimensional para problemas de minimização irrestrita.

Karas, Ribeiro, Sagastizábal e Solodov [35] aplicaram as técnicas de filtro ao método de feixes proposto em [66], para a resolução de problemas de otimização convexa não diferenciável. Gomes [25] define um critério de aceitação de passo que combina filtro e função de mérito em seu trabalho.

Em [61], Ribeiro estabelece uma teoria geral para os métodos de filtro, propondo um algoritmo com grande liberdade para o cálculo do passo e provando que esse algoritmo tem convergência global.

Proposta do trabalho. Nessa dissertação propomos um algoritmo de filtro para problemas de programação não linear. O algoritmo proposto é baseado no algoritmo de Gonzaga, Karas e Vanti [28], mas usa o filtro inclinado, conforme Fletcher, Leyffer e Toint [22], como critério de aceitação do passo. Cada iteração do algoritmo é composta de uma fase de restauração, que reduz uma medida de inviabilidade, e uma fase de otimalidade, que reduz o valor da função objetivo. O método tem grande liberdade na escolha dos algoritmos internos, desde que esses algoritmos satisfaçam algumas hipóteses razoáveis sobre sua eficiência. Apresentamos a análise de convergência do algoritmo proposto.

Organização dos capítulos. No Capítulo 1 apresentamos o método de região de confiança para resolver um problema irrestrito e dois métodos para encontrar a solução do subproblema gerado, Dogleg e Steihaug. No Capítulo 2 tratamos dos problemas com restrições. Apresentamos dois métodos para obtenção do passo: programação quadrática sequencial (PQS) e restauração inexata (RI). Nesse capítulo apresentamos também uma discussão sobre os critérios de aceitação do passo. Por fim, no Capítulo 3, apresentamos o principal resultado de nosso trabalho. Discutimos alguns métodos de filtro existentes e propomos um novo algoritmo, estabelecendo a convergência global.

Capítulo 1

Método de região de confiança para problemas irrestritos

Neste capítulo apresentamos um caso particular do problema de programação não linear, onde não existem restrições sobre as variáveis, e um método para encontrar uma solução. Esse problema é chamado de problema irrestrito e o método que vamos descrever é o método de região de confiança.

Na primeira seção, descrevemos o problema de minimização irrestrita e fazemos um apanhado geral sobre os métodos existentes para resolvê-lo. Na segunda seção discutimos o método de região de confiança. Então descrevemos os métodos de Dogleg e Steihaug, que resolvem um subproblema quadrático gerado no método de região de confiança.

1.1 O Problema

Vamos considerar o problema (P) , apresentado na introdução deste trabalho, com $\mathcal{E} = 0$ e $\mathcal{I} = 0$. Assim, reescrevemos o problema,

$$(PI) \quad \begin{cases} \text{minimizar} & f_0(x) \\ & x \in \mathbb{R}^n, \end{cases}$$

onde $f_0 : \mathbb{R}^n \rightarrow \mathbb{R}$ é de classe \mathcal{C}^2 . Nesse caso o conjunto viável é todo o \mathbb{R}^n e precisamos encontrar um minimizador local do problema.

Os métodos desenvolvidos para resolver o problema irrestrito procuram encontrar uma solução através de um processo iterativo, ou seja, dado um ponto inicial x^0 eles buscam, passo a passo, pontos que diminuam o valor da função objetivo em relação ao ponto anterior.

Os métodos clássicos para resolver o problema irrestrito iniciam com o ponto inicial x^0 e determinam uma direção d que vai reduzir o valor da função objetivo.

Então, nessa direção, procuram caminhar o suficiente para que a redução obtida seja razoável.

Para obter a direção do passo temos dois métodos clássicos: Cauchy e Newton. O Método de Cauchy é um método cauteloso, que tem garantia de convergência em condições muito gerais, mas pode se tornar lento. O Método de Newton é rápido e arrojado. Quando iniciado perto de uma solução do problema se torna um método excelente, mas se iniciado em pontos ruins pode divergir. Métodos mais modernos procuram unir as vantagens de ambos e são chamados Métodos Quase-Newton.

Definida a direção, esses métodos precisam encontrar o tamanho do passo. Para isso existem vários outros métodos, chamados de métodos de busca unidirecional, que garantem que o decréscimo no valor da função objetivo seja proporcional ao comprimento do passo. Os métodos de busca podem ser divididos em duas categorias: busca exata e inexata. Na busca exata o tamanho do passo é encontrado minimizando a função objetivo na direção obtida. Exemplos de métodos de busca exata são a Seção Áurea e a Biseção. Na busca inexata o que se procura é uma boa redução na função objetivo na direção obtida, mas sem tentar minimizá-la. Os métodos de Armijo, Wolfe e Goldstein são os mais conhecidos dessa categoria.

Esses métodos, e outros não citados, são descritos em [2, 5, 27, 39, 54]. Uma outra classe de métodos para resolver problemas irrestritos são os métodos de região de confiança, onde a direção e o tamanho do passo são obtidos simultaneamente. Na seção seguinte vamos descrever esse método, pois a estratégia de região de confiança será utilizada também nos métodos para resolver o problema com restrições.

1.2 Método de Região de Confiança

O desenvolvimento dos métodos de região de confiança remonta aos trabalhos de Levenberg [38] e Marquardt [40] para problemas não lineares de mínimos quadrados. Powell [56, 57] foi o primeiro a estabelecer convergência dos métodos de região de confiança para otimização irrestrita. Moré [49] e Moré e Sorensen [50] fazem um tratado sobre o desenvolvimento desses métodos. Para escrevermos este capítulo, usamos como referência [5, 27, 54], que trazem um tratamento mais moderno sobre o assunto.

O método de região de confiança é um processo iterativo onde, a cada iteração, um passo tentativo é calculado minimizando um modelo quadrático para a função objetivo numa região em que esse modelo é confiável. Dessa maneira obtemos a direção e o tamanho do passo ao mesmo tempo. Se esse passo tentativo não é aceito, reduzimos o tamanho da região de confiança e procuramos um novo minimizador. Normalmente, quando o tamanho da região é alterado a direção do passo também se altera.

Vamos aplicar o método de região de confiança ao problema (PI). Construímos um modelo para a função objetivo em torno de um ponto corrente $x^k \in \mathbb{R}^n$, onde $k \in \mathbb{N}$. O modelo quadrático que vamos considerar é do tipo:

$$m_k^{RC}(d) = f_0(x^k) + \nabla f_0(x^k)^t d + \frac{1}{2} d^t \mathcal{B}_k d, \quad (1.1)$$

onde $d = x - x^k$ e \mathcal{B}_k é uma matriz simétrica $n \times n$, podendo ser uma aproximação da hessiana $\nabla^2 f_0(x^k)$ ou qualquer outra matriz simétrica que satisfaça

$$\|\mathcal{B}_k\| \leq \beta,$$

para alguma constante $\beta > 0$, independente de k .

Para obter d minimizamos o modelo (1.1) restrito a uma certa região. Essa região é chamada de região de confiança, pois o modelo só é um bom representante da função objetivo quando x está numa vizinhança de x^k , ou seja, na região:

$$\{d \in \mathbb{R}^n \mid \|d\| \leq \Delta\}, \quad (1.2)$$

onde $\Delta > 0$ é o raio da região de confiança e $\|\cdot\|$ indica uma norma arbitrária em \mathbb{R}^n .

Assim, o passo d é uma solução do subproblema:

$$\begin{cases} \text{minimizar} & m_k^{RC}(d) \\ \text{sujeito a:} & \|d\| \leq \Delta \end{cases} . \quad (1.3)$$

Note que o passo de região de confiança d depende do raio Δ , assim para evidenciar essa dependência vamos denotá-lo por $d(x^k, \Delta)$. Definimos a redução verdadeira na função objetivo como:

$$ared(x^k, \Delta) = f_0(x^k) - f_0(x^k + d(x^k, \Delta)), \quad (1.4)$$

e a redução predita pelo modelo:

$$pred(x^k, \Delta) = m_k^{RC}(0) - m_k^{RC}(d). \quad (1.5)$$

Podemos ver que a redução predita será sempre não negativa, pois o passo $d(x^k, \Delta)$ é obtido minimizando-se o modelo numa região em torno de x^k . De acordo com [54, Lema 4.5] a menos que x^k seja uma solução do problema original, a expressão 1.5 é positiva. Assim definimos a razão:

$$\rho_k = \frac{ared(x^k, \Delta)}{pred(x^k, \Delta)}. \quad (1.6)$$

Um passo $d(x^k, \Delta)$ somente será aceito se ρ_k for positivo, ou seja, se houver uma redução na função objetivo. Se essa redução acontecer e ρ_k for próximo de 1, aceitamos o passo e aumentamos a região de confiança, pois a redução na função objetivo foi tão boa quanto a redução no modelo. Se essa razão for positiva e longe de 1, aceitamos o passo e mantemos o raio da região. Por outro lado, se ρ_k for negativo, é porque a função objetivo aumentou, e assim o modelo não está representando bem a função objetivo. Nesse caso, rejeitamos o passo e diminuimos o raio da região de confiança. A seguir temos o algoritmo que descreve esse processo.

Algoritmo 1.2.1 *Região de Confiança*

Dado: $x^0, \bar{\Delta} > 0, \Delta \in (0, \bar{\Delta}), \eta \in (0, 1), \varepsilon > 0$

$k = 0$;

REPITA enquanto $\|\nabla f_0(x^k)\| > \varepsilon$

Calcule $d(x^k, \Delta)$ resolvendo o subproblema (1.3);

$ared(x^k, \Delta) = f_0(x^k) - f_0(x^k + d(x^k, \Delta))$;

$pred(x^k, \Delta) = m_k^{RC}(0) - m_k^{RC}(d)$;

$\rho_k = \frac{ared(x^k, \Delta)}{pred(x^k, \Delta)}$;

SE $\rho_k < \eta$

$\Delta = \frac{1}{2}\Delta$;

SENÃO

$x^{k+1} = x^k + d(x^k, \Delta)$;

Atualize \mathcal{B}_k ;

$k = k + 1$;

SE $\rho_k > \frac{3}{4}$ e $\|d(x^k, \Delta)\| = \Delta$

$\Delta = \min\{2\Delta, \bar{\Delta}\}$;

FIM

FIM

FIM

Na Figura 1.1 temos a ilustração de um passo do algoritmo de região de confiança. As linhas tracejadas representam as curvas de nível da função objetivo do problema original. As curvas de nível do modelo m_k^{RC} estão representadas pelas linhas cheias. O ponto x^{k+1} é o minimizador do modelo na região de confiança centrada no ponto corrente x^k .

Podemos notar que o raio da região de confiança só será aumentado se ρ_k estiver próximo de 1 e o passo $d(x^k, \Delta)$ estiver na fronteira da região de confiança, o que significa que não encontramos o minimizador global do modelo. Também definimos um limite superior para o raio da região de confiança, $\bar{\Delta}$, que é necessário para que não percamos o controle sobre o tamanho da região. Por outro lado, para

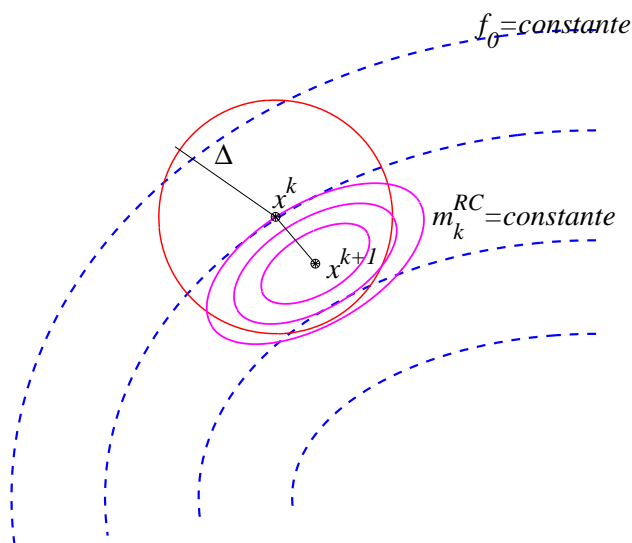


Figura 1.1: Um passo de região de confiança.

que o raio não diminua estamos exigindo que a redução na função objetivo seja de, pelo menos, uma fração da redução obtida no modelo.

Podemos ver também que quanto menor o valor de η mais tolerante estamos sendo na aceitação do passo. Se esse valor é muito próximo de zero significa que qualquer redução, por menor que seja, na função objetivo já é suficiente. Uma escolha razoável é $\eta = \frac{1}{4}$, assim a redução obtida na função objetivo é maior ou igual a 25% da redução obtida no modelo.

Com os critérios de aceitação do passo e de atualização do raio da região de confiança bem estipulados, nosso problema agora se resume a minimizar a aproximação quadrática na bola, ou seja, resolver o problema (1.3). Quando a matriz \mathcal{B}_k é positiva definida e $\|\mathcal{B}_k^{-1}\nabla f_0(x^k)\| \leq \Delta$, podemos facilmente encontrar uma solução desse problema, resolvendo o sistema:

$$\nabla f_0(x^k) + \mathcal{B}_k d(x^k, \Delta) = \nabla m_k^{RC}(d(x^k, \Delta)) = 0,$$

obtendo, deste modo,

$$d(x^k, \Delta) = -(\mathcal{B}_k)^{-1}\nabla f_0(x^k).$$

Vamos denotar essa solução por d_B^k e a chamaremos de “passo cheio”. Essa é uma solução exata para o problema (1.3), mas na maioria dos casos não é possível encontrá-la e precisamos de uma solução aproximada para esse problema. Resolver aproximadamente o problema (1.3) significa encontrar uma solução tal que o ponto x^{k+1} seja tão bom quanto o ponto de Cauchy, para que tenhamos a garantia da convergência global do método de região de confiança. A demonstração dessa convergência é feita com detalhes em [13, 51, 54].

O ponto de Cauchy, que denotaremos por x_C^k , é o minimizador do modelo

(1.1) ao longo da direção oposta ao gradiente $-\nabla f_0(x^k)$, limitada pela região de confiança.

A Figura 1.2 mostra o ponto de Cauchy numa iteração k . A bola representa a região de confiança com um raio Δ e as elipses representam as curvas de nível do modelo. Na região hachurada temos o conjunto de pontos que são melhores que o ponto de Cauchy.

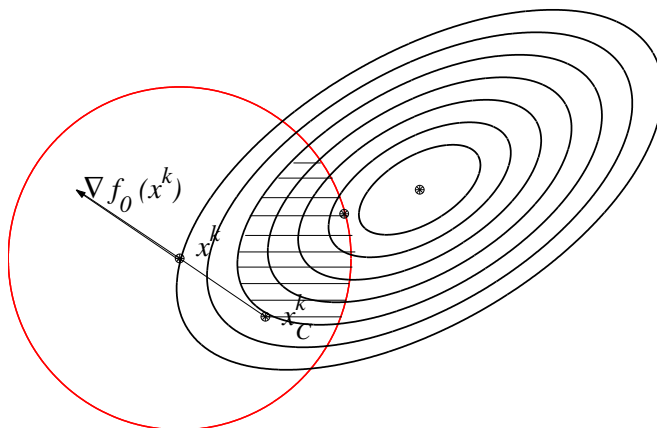


Figura 1.2: Ponto de Cauchy.

Para calcular o ponto de Cauchy, precisamos primeiramente encontrar o vetor d_s^k , que é dado por:

$$d_s^k = \arg \min_{d \in \mathbb{R}^n} \{ f_0(x^k) + \nabla f_0(x^k)^t d \mid \|d\| \leq \Delta \}.$$

Podemos verificar que d_s^k é o vetor com norma Δ e direção oposta ao gradiente, ou seja,

$$d_s^k = -\Delta \frac{\nabla f_0(x^k)}{\|\nabla f_0(x^k)\|}.$$

Agora precisamos calcular o escalar $\nu_k > 0$ que minimiza a aproximação quadrática em νd_s^k :

$$\nu_k = \arg \min_{\nu > 0} \{ m_k^{RC}(\nu d_s^k) \mid \|\nu d_s^k\| \leq \Delta \}. \quad (1.7)$$

Assim, o passo de Cauchy é dado por:

$$d_C^k = \nu_k d_s^k.$$

Se o ponto de Cauchy estiver estritamente no interior da região de confiança, temos que ν_k é o minimizador irrestrito de (1.7), ou seja,

$$\nu_k = \frac{\|\nabla f_0(x^k)\|^3}{\Delta (\nabla f_0(x^k)^t \mathcal{B}_k \nabla f_0(x^k))}.$$

Caso contrário, $\nu_k = 1$ e o passo de Cauchy é o mesmo que o obtido ao minimizar a aproximação linear. Resumidamente, temos:

$$d_C^k = -\nu_k \Delta \frac{\nabla f_0(x^k)}{\|\nabla f_0(x^k)\|}, \quad (1.8)$$

onde

$$\nu_k = \min \left\{ 1, \frac{\|\nabla f_0(x^k)\|^3}{\Delta(\nabla f_0(x^k)^t \mathcal{B}_k \nabla f_0(x^k))} \right\}.$$

Assim, temos que o ponto de Cauchy é o ponto obtido pelo passo de d_C^k , $x_C^k = x^k + d_C^k$. A seguir vamos ver dois métodos que satisfazem a condição de decréscimo suficiente dada pelo ponto de Cauchy, ou seja, que determinam um ponto x tal que

$$m_k^{RC}(x^k) - m_k^{RC}(x) > m_k^{RC}(x^k) - m_k^{RC}(x_C^k).$$

1.2.1 Método Dogleg

O Método Dogleg encontra uma solução aproximada para o problema (1.3) fazendo a interseção entre a fronteira da região de confiança e o caminho dogleg. O caminho dogleg é composto por dois segmentos de reta: o primeiro parte da origem e vai até o minimizador irrestrito ao longo da direção oposta ao gradiente e o segundo parte desse minimizador e vai até o passo cheio d_B^k .

O minimizador irrestrito ao longo da direção oposta ao gradiente é definido por

$$d_U^k = -\frac{\nabla f_0(x^k)^t \nabla f_0(x^k)}{\nabla f_0(x^k)^t \mathcal{B}_k \nabla f_0(x^k)} \nabla f_0(x^k),$$

que é igual ao passo de Cauchy quando a matriz \mathcal{B}_k é positiva definida e não se chega até a fronteira da região de confiança.

Vamos denotar essa trajetória por $\tilde{p}(\tau)$, com $\tau \in [0, 2]$, e definí-la como

$$\tilde{p}(\tau) = \begin{cases} \tau d_U^k & 0 \leq \tau \leq 1 \\ d_U^k + (\tau - 1)(d_B^k - d_U^k) & 1 \leq \tau \leq 2. \end{cases} \quad (1.9)$$

O método escolhe o passo d minimizando o modelo (1.1) ao longo desse caminho. De fato, esse minimizador será o ponto de interseção entre o caminho dogleg (1.9) e a fronteira da região de confiança. Para comprovar essa afirmação temos o teorema a seguir. A demonstração dada aqui é a apresentada em [54], enriquecida com alguns detalhes.

Teorema 1.2.2 *Seja \mathcal{B}_k positiva definida. Então*

- (i) $\|\tilde{p}(\tau)\|$ é uma função não crescente de τ , e
- (ii) $m_k^{RC}(\tilde{p}(\tau))$ é uma função decrescente de τ .

Demonstração. Nessa demonstração vamos utilizar as seguintes notações:

$$\nabla f_0 = \nabla f_0(x^k) \text{ e } f_0 = f_0(x^k).$$

Para $0 \leq \tau \leq 1$ definimos $h(\tau)$ por

$$h(\tau) = \frac{1}{2} \|\tilde{p}(\tau)\|^2 = \frac{1}{2} \|\tau d_U^k\|^2 = \frac{1}{2} (\tau d_U^k)^t (\tau d_U^k) = \frac{1}{2} \tau^2 d_U^{k_t} d_U^k.$$

Para provar (i) basta mostrar que $h'(\tau) \geq 0$.

$$\begin{aligned} h'(\tau) &= \tau d_U^{k_t} d_U^k \\ &= \tau \left(-\frac{\nabla f_0^t \nabla f_0}{\nabla f_0^t \mathcal{B}_k \nabla f_0} \nabla f_0 \right)^t \left(-\frac{\nabla f_0^t \nabla f_0}{\nabla f_0^t \mathcal{B}_k \nabla f_0} \nabla f_0 \right) \\ &= \frac{\tau \|\nabla f_0\|^6}{(\nabla f_0^t \mathcal{B}_k \nabla f_0)^2} \geq 0. \end{aligned}$$

Agora temos que

$$\begin{aligned} m_k^{RC}(\tilde{p}(\tau)) &= m_k^{RC}(\tau d_U^k) \\ &= f_0 + \nabla f_0^t (\tau d_U^k) + \frac{1}{2} (\tau d_U^k)^t \mathcal{B}_k (\tau d_U^k) \\ &= f_0 - \tau \frac{\nabla f_0^t \nabla f_0 \nabla f_0^t \nabla f_0}{\nabla f_0^t \mathcal{B}_k \nabla f_0} + \frac{\tau^2}{2} \frac{(\nabla f_0^t \nabla f_0) \nabla f_0^t \mathcal{B}_k \nabla f_0 (\nabla f_0^t \nabla f_0)}{\nabla f_0^t \mathcal{B}_k \nabla f_0 \nabla f_0^t \mathcal{B}_k \nabla f_0} \\ &= f_0 + \left(\frac{\tau^2}{2} - \tau \right) \frac{(\nabla f_0^t \nabla f_0)^2}{\nabla f_0^t \mathcal{B}_k \nabla f_0}, \end{aligned}$$

que demonstra o item (ii), pois $\left(\frac{\tau^2}{2} - \tau \right)$ é uma parábola decrescente em τ ($\tau \in (0, 1)$).

Para $1 \leq \tau \leq 2$, definimos $h(\alpha)$ como

$$\begin{aligned} h(\alpha) &= \frac{1}{2} \|\tilde{p}(1 + \alpha)\|^2 \\ &= \frac{1}{2} \|d_U^k + \alpha(d_B^k - d_U^k)\|^2 \\ &= \frac{1}{2} \|d_U^k\|^2 + \alpha d_U^{k_t} (d_B^k - d_U^k) + \frac{\alpha^2}{2} \|d_B^k - d_U^k\|^2. \end{aligned}$$

Para provar (i) basta mostrar que $h'(\alpha) \geq 0$ para $\alpha \in (0, 1)$:

$$\begin{aligned} h'(\alpha) &= d_U^{k_t} (d_B^k - d_U^k) + \alpha \|d_B^k - d_U^k\|^2 \\ &= -d_U^{k_t} (d_U^k - d_B^k) + \alpha \|d_B^k - d_U^k\|^2 \\ &\geq -d_U^{k_t} (d_U^k - d_B^k) \\ &= \frac{\|\nabla f_0\|^2 \nabla f_0^t \mathcal{B}_k^{-1} \nabla f_0}{\nabla f_0^t \mathcal{B}_k \nabla f_0} \left[1 - \frac{\|\nabla f_0\|^4}{(\nabla f_0^t \mathcal{B}_k \nabla f_0)(\nabla f_0^t \mathcal{B}_k^{-1} \nabla f_0)} \right] \geq 0. \end{aligned}$$

A última inequação na demonstração vem do fato de

$$\frac{\|\nabla f_0\|^4}{(\nabla f_0^t \mathcal{B}_k \nabla f_0)(\nabla f_0^t \mathcal{B}_k^{-1} \nabla f_0)} \leq 1,$$

cuja prova pode ser encontrada em [51, Lema 2.11]. Para provar (ii), definimos

$$\begin{aligned}\hat{h}(\alpha) &= m_k^{RC}(\tilde{p}(1 + \alpha)) = m_k^{RC}(d_U^k + \alpha(d_B^k - d_U^k)) \\ &= f_0 + \nabla f_0^t(d_U^k + \alpha(d_B^k - d_U^k)) + \frac{1}{2}(d_U^k + \alpha(d_B^k - d_U^k))^t \mathcal{B}_k(d_U^k + \alpha(d_B^k - d_U^k)).\end{aligned}$$

Agora basta mostrar que $\hat{h}'(\alpha) \leq 0$, para $\alpha \in (0, 1)$:

$$\begin{aligned}\hat{h}'(\alpha) &= \nabla f_0^t(d_B^k - d_U^k) + \frac{1}{2}((2d_U^k)^t \mathcal{B}_k(d_B^k - d_U^k) + 2\alpha(d_B^k - d_U^k)^t \mathcal{B}_k(d_B^k - d_U^k)) \\ &= (d_B^k - d_U^k)^t (\nabla f_0 + \mathcal{B}_k d_U^k) + \alpha(d_B^k - d_U^k)^t \mathcal{B}_k(d_B^k - d_U^k) \\ &\leq (d_B^k - d_U^k)^t (\nabla f_0 + \mathcal{B}_k d_U^k + \mathcal{B}_k(d_B^k - d_U^k)) \\ &= -\nabla f_0^t(\mathcal{B}_k^{-1t} \nabla f_0 + \nabla f_0^t(\mathcal{B}_k^{-1t} \nabla f_0) = 0.\end{aligned}$$

□

Segue desse teorema que o caminho $\tilde{p}(\tau)$ intercepta a fronteira da região em somente um ponto se $\|d_B^k\| \geq \Delta$ e em nenhum ponto caso isso não aconteça. Como o modelo m_k^{RC} é decrescente ao longo desse caminho, uma boa escolha para o passo $d(x^k, \Delta)$ é exatamente o ponto de interseção se $\|d_B^k\| > \Delta$ ou o próprio d_B^k no caso contrário.

Para o caso em que ocorre a interseção, precisamos encontrar o valor apropriado para τ resolvendo a seguinte equação:

$$\|d_U^k + (\tau - 1)(d_B^k - d_U^k)\|^2 = \Delta^2. \quad (1.10)$$

Uma solução de (1.10) é facilmente encontrada, resolvendo (1.10) como uma equação de segundo grau em função de $(\tau - 1)$.

Nas figuras a seguir temos a ilustração dos três casos que podem ocorrer. Na Figura 1.3 o minimizador irrestrito da quadrática, d_B^k , está no interior da região de confiança e é portanto a solução escolhida pelo Método Dogleg.

Na Figura 1.4 o minimizador ao longo da direção oposta ao gradiente está no interior da região de confiança e a solução encontrada pelo método é exatamente a interseção entre a fronteira da região de confiança e o caminho dogleg (1.9).

Por último, na Figura 1.5 o minimizador ao longo da direção oposta ao gradiente está na fronteira da região de confiança, ou seja, é a própria direção de Cauchy d_C^k , e esta será a solução encontrada pelo método.

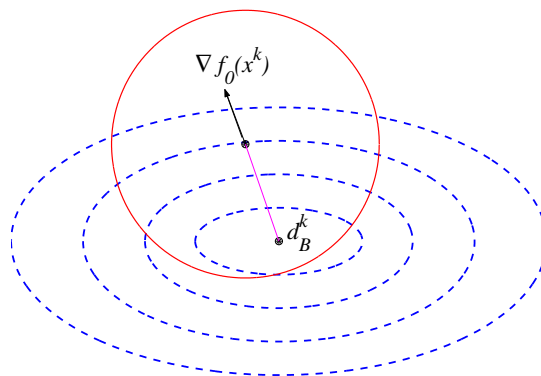


Figura 1.3: Minimizador é o passo cheio d_B^k .

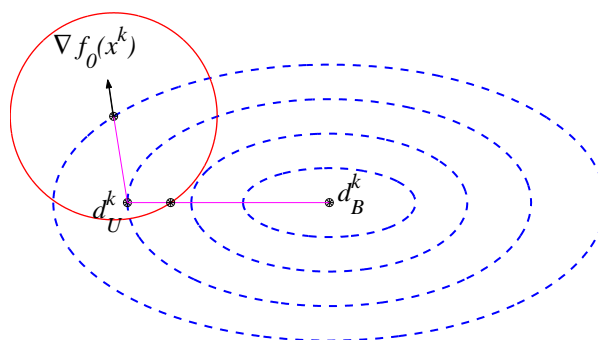


Figura 1.4: Minimizador é a interseção.

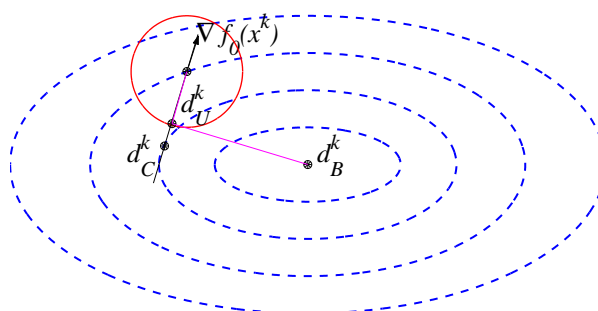


Figura 1.5: Minimizador é o passo de Cauchy d_C^k .

Apresentamos a seguir o algoritmo do Método Dogleg para resolver aproximadamente o problema 1.3.

Algoritmo 1.2.3 *Método Dogleg*

Dados: x^k , \mathcal{B}_k , $g = \nabla f_0(x^k)$

Calcule $d_B^k = -(\mathcal{B}_k)^{-1}g$

SE $\|d_B^k\| \leq \Delta$

$d(x^k, \Delta) = d_B^k$; (minimizador irrestrito da quadrática)

SENÃO

Calcule $d_U^k = -\frac{g^t g}{g^t \mathcal{B}_k g} g$;

SE $\|d_U^k\| \geq \Delta$

Calcule d_C^k por (1.8);

$d(x^k, \Delta) = d_C^k$; (minimizador na direção de Cauchy)

SENÃO

Calcule τ resolvendo (1.10);

$d(x^k, \Delta) = d_U^k + (\tau - 1)(d_B^k - d_U^k)$

FIM

FIM

1.2.2 Método de Steihaug

O Método Dogleg, descrito anteriormente, nos dá uma boa aproximação de uma solução de (1.3), mas necessita que seja resolvido um sistema simples que envolve a inversa da matriz \mathcal{B}_k , o que pode ser impossível quando essa matriz não tem inversa. Assim, motivados por essa dificuldade, vamos ver agora um método que utiliza outras técnicas para aproximar uma solução de (1.3). Steihaug propôs um método com essas características, que é baseado no Algoritmo do Gradiente Conjugado.

O Método do Gradiente Conjugado (GC) é um algoritmo iterativo para resolver problemas quadráticos com matriz simétrica positiva definida. O método transforma o problema quadrático em um sistema linear e usa direções conjugadas, fazendo sucessivas minimizações unidimensionais ao longo de cada direção escolhida. Uma importante propriedade diz que cada nova direção pode ser calculada usando somente a direção anterior e será automaticamente conjugada a todas as outras direções usadas no processo. Cada nova direção no algoritmo é escolhida como uma combinação linear entre a direção de máximo declive $-\nabla f_0(x^k)$ e a direção obtida anteriormente. Uma explicação mais detalhada desse método pode ser encontrada em [54].

O Método de Steihaug é chamado de Gradiente Conjugado Steihaug, pois usa as mesmas idéias do GC para resolver o problema quadrático na bola. A seguir

vamos descrever o Algoritmo de Steihaug e comentar sobre as diferenças essenciais entre ele e o algoritmo original do GC.

Algoritmo 1.2.4 *Steihaug GC*

Dados: $\varepsilon > 0$, $x^k \in \mathbb{R}^n$, \mathcal{B}_k

Faça $d^0 = 0$, $r^0 = \nabla f_0(x^k)$, $p^0 = -r^0$;

SE $\|r^0\| \leq \varepsilon$

$d(x^k, \Delta) = d^0$;

sai com $d(x^k, \Delta)$;

FIM

$j = 0$;

REPITA

SE $p^{j^t} \mathcal{B}_k p^j \leq 0$

Encontre $\tau = \operatorname{argmin}\{m_k^{RC}(d^j + \tau p^j) \mid \|d^j + \tau p^j\| = \Delta\}$;

$d(x^k, \Delta) = d^j + \tau p^j$;

sai com $d(x^k, \Delta)$;

FIM

$\alpha_j = \frac{r^{j^t} r^j}{p^{j^t} \mathcal{B}_k p^j}$; $d^{j+1} = d^j + \alpha_j p^j$;

SE $\|d^{j+1}\| \geq \Delta$

Encontre $\tau \geq 0$ tal que $\|d^j + \tau p^j\| = \Delta$;

$d(x^k, \Delta) = d^j + \tau p^j$;

sai com $d(x^k, \Delta)$;

FIM

$r^{j+1} = r^j + \alpha_j \mathcal{B}_k p^j$;

SE $\|r^{j+1}\| < \varepsilon \|r^0\|$

$d(x^k, \Delta) = d^{j+1}$;

sai com $d(x^k, \Delta)$;

FIM

$\beta_{j+1} = \frac{r^{j+1^t} r^{j+1}}{r^{j^t} r^j}$; $p^{j+1} = -r^{j+1} + \beta_{j+1} p^j$;

$j = j + 1$;

FIM

Esse algoritmo difere do algoritmo do GC em dois critérios extras de parada. O primeiro “SE”, no enlace do “REPITA”, termina o método se a direção atual p^j é uma direção de curvatura zero ou negativa ao longo de \mathcal{B}_k . No segundo “SE” o método pára se d^{j+1} atinge o limite da região de confiança. Em ambos os casos a direção $d(x^k, \Delta)$ é encontrada pela interseção da direção de busca atual com a fronteira da região de confiança.

Inicializar $d^0 = 0$ é um passo fundamental no algoritmo. Depois da primeira iteração, se $\|r^0\|_2 \geq \varepsilon$, temos

$$d^1 = \alpha_0 p^0 = \frac{r^{0t} r^0}{p^{0t} \mathcal{B}_k p^0} p^0 = -\frac{\nabla f_0(x^k)^t \nabla f_0(x^k)}{\nabla f_0(x^k)^t \mathcal{B}_k \nabla f_0(x^k)} \nabla f_0(x^k),$$

que é exatamente o ponto de Cauchy. Como a cada iteração o método do gradiente conjugado reduz o valor de m_k^{RC} , esse algoritmo satisfaz a condição de convergência, ou seja, encontra um ponto melhor que o ponto de Cauchy.

Outra propriedade importante do método é que cada direção d^j é maior em norma que a direção anterior. Isso significa dizer que é aceitável parar o método quando a fronteira da região de confiança é interceptada, pois não existem iterações anteriores que tenham um valor para o modelo m_k^{RC} menor que o da iteração atual. O enunciado e demonstração formal dessa propriedade podem ser encontrados em [54].

Capítulo 2

Problemas com restrições

Neste capítulo vamos discutir problemas com restrições de igualdade e desigualdade. Ao resolver problemas com restrições precisamos simultaneamente obter viabilidade e reduzir o valor da função objetivo. Esses dois objetivos muitas vezes são conflitantes e essa é uma grande dificuldade. Uma forma de combinar esses dois objetivos é fazer uso de funções de mérito ou Lagrangiano aumentado.

Byrd, Gilbert e Nocedal [7] propuseram um algoritmo, que foi implementado por Byrd, Hribar e Nocedal [8], que utiliza programação quadrática sequencial (PQS) com região de confiança para a obtenção do passo. O método de PQS, resolve um problema não linear através de uma seqüência de problemas de programação quadrática, onde minimiza-se a aproximação quadrática do lagrangeano sujeito a linearização das restrições. Pode ser provado [5, 13, 54] que, sob certas condições, este método tem convergência local quadrática. Mas, para um ponto inicial qualquer, não se pode garantir a convergência sem que se use algum critério de aceitação do passo.

Em [7] e [8], o critério utilizado foi uma função de mérito, que é uma combinação entre uma medida de inviabilidade e uma medida de otimalidade.

O método de restauração inexata (RI), proposto originalmente por Martínez [41] e Martínez e Pilotta [42], também utiliza uma função de mérito para avaliar o ponto tentativo. Nestes algoritmos cada iteração é dividida em duas fases: uma de viabilidade e uma de otimalidade. Na primeira fase procura-se reduzir a inviabilidade do problema e na fase de otimalidade a intenção é reduzir o valor da função objetivo sem perder muito o ganho conseguido na fase de viabilidade. Para a segunda fase é aplicado um método de região de confiança, que tem como centro o ponto obtido depois da fase de viabilidade e não o ponto corrente.

Fletcher e Leyffer [20] propuseram um critério de aceitação do passo chamado de filtro, como uma alternativa ao uso das funções de mérito. O filtro define uma região proibida guardando pares ordenados que representam a função objetivo e a inviabilidade. Nesse trabalho, o passo é obtido por um método do tipo PQS e

avaliado por um filtro.

Nos trabalhos de Fletcher, Leyffer e Toint [22] e Fletcher *et al* [19], que usam estratégias diferentes para resolver os subproblemas quadráticos gerados no método PQS, foi dada a prova de convergência global dos métodos de filtro.

Em seu trabalho, Gonzaga, Karas e Vanti [28] propuseram um algoritmo de filtro globalmente convergente que encontra o passo através de um método do tipo restauração inexata [41, 42].

Neste capítulo discutiremos os métodos de PQS e restauração inexata para obtenção do passo, bem como os critérios de função de mérito e filtro para avaliar o passo obtido.

2.1 O Problema

Neste capítulo voltamos ao problema original, com restrições de igualdade e desigualdade. Por comodidade enunciaremos o problema novamente:

$$(P) \quad \begin{cases} \text{minimizar} & f_0(x) \\ \text{sujeito a :} & f_{\mathcal{E}}(x) = 0 \\ & f_{\mathcal{I}}(x) \leq 0, \end{cases}$$

onde $\mathcal{E} = \{1, 2, \dots, p\}$ e $\mathcal{I} = \{p+1, p+2, \dots, p+q\}$.

Na Figura 2.1 temos uma ilustração do problema (P). A linha cheia representa o conjunto viável e as linhas tracejadas representam as curvas de nível da função objetivo. O ponto x^* representa a solução do problema, isto é, dos pontos pertencentes ao conjunto viável esse é o que está sobre a curva de nível de menor valor. Se o problema fosse irrestrito a solução estaria perto do canto direito inferior da figura, com valor para a função objetivo menor que o do ponto x^* .

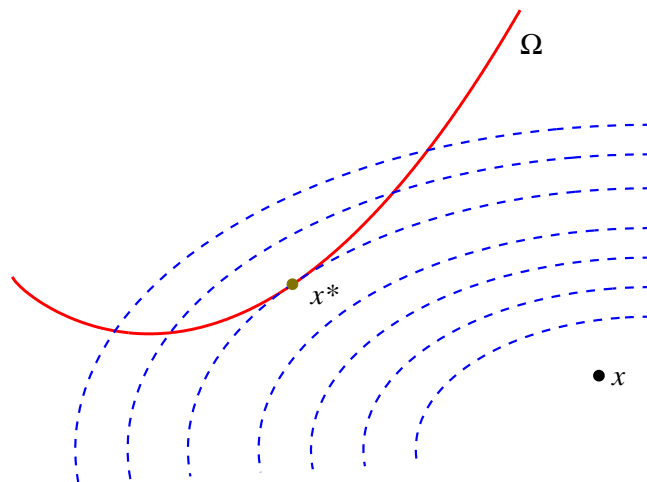


Figura 2.1: O problema com restrições.

A idéia dos processos iterativos é, dado um ponto inicial x^0 , encontrar um passo para obter um novo ponto x^1 , e assim sucessivamente. O novo ponto deve, de alguma forma, ser melhor que o anterior. Discutiremos, na Seção 2.3, critérios para decidir quando um ponto é melhor que o outro. Os métodos que vamos apresentar a seguir são métodos para obtenção do passo.

2.2 Obtenção do passo

Resolver problemas de programação não linear com restrições, tipo o problema (P), é uma tarefa difícil. Os algoritmos desenvolvidos para isso precisam balancear esforços entre viabilidade e otimalidade e trabalhos novos surgem constantemente.

Discutiremos aqui dois métodos para obtenção do passo: programação quadrática sequencial e restauração inexata.

2.2.1 Programação Quadrática Sequencial

A idéia do método de programação quadrática sequencial é resolver uma seqüência de problemas quadráticos, substituindo, a cada iteração, a função objetivo por uma aproximação quadrática do lagrangiano e as restrições por aproximações lineares, tornando assim o problema relativamente mais simples que o original.

Vamos aplicar o método de PQS ao problema (P) numa dada iteração k , ou seja, vamos considerar um subproblema quadrático cuja minimização vai definir um novo ponto x^{k+1} .

Usaremos $A_{\mathcal{E}}(x)$ para denotar a matriz jacobiana das restrições de igualdade no ponto x , isto é,

$$A_{\mathcal{E}}^t(x) = [\nabla f_1(x) \quad \nabla f_2(x) \quad \dots \quad \nabla f_p(x)].$$

A jacobiana das restrições de desigualdade vai ser denotada por $A_{\mathcal{I}}(x)$,

$$A_{\mathcal{I}}^t(x) = [\nabla f_{p+1}(x) \quad \nabla f_{p+2}(x) \quad \dots \quad \nabla f_{p+q}(x)].$$

O lagrangiano associado ao problema (P) é dado por:

$$\ell(x, \lambda, \mu) = f_0(x) + \lambda^t f_{\mathcal{E}}(x) + \mu^t f_{\mathcal{I}}(x), \tag{2.1}$$

onde $\lambda \in \mathbb{R}^p$ e $\mu \in \mathbb{R}^q$ são chamados multiplicadores de Lagrange. A matriz hessiana do lagrangiano, $\nabla_{xx}^2 \ell(x, \lambda, \mu)$, é denotada por $W(x, \lambda, \mu)$. Além disso vamos denotar $m = p + q$.

O método de PQS puro. Apresentamos inicialmente, para simplificar a notação, o método puro de PQS para problemas com restrições de igualdade ($\mathcal{I} = \emptyset$).

$$(P_E) \quad \begin{cases} \text{minimizar} & f_0(x) \\ \text{sujeito a} & : f(x) = 0, \end{cases}$$

onde $f_0 : \mathbb{R}^n \rightarrow \mathbb{R}, f : \mathbb{R}^n \rightarrow \mathbb{R}^m$.

Isto corresponde a aplicar o método de Newton às condições de KKT do problema original, como veremos mais adiante.

Como falamos no início da seção o método PQS considera a cada iteração um problema quadrático cuja função objetivo é uma aproximação quadrática do lagrangiano e as restrições são aproximações lineares das restrições originais. Então, dado um ponto $x^k \in \mathbb{R}^n$ considere a aproximação linear das restrições

$$f(x^k) + A(x^k)d = 0. \quad (2.2)$$

Dado $(x^k, \lambda^k) \in \mathbb{R}^{n+m}$ o modelo quadrático do lagrangiano é dado por:

$$m_k^{PQS}(d) = \ell(x^k, \lambda^k) + \nabla \ell(x^k, \lambda^k)^t d + \frac{1}{2} d^t W_k d,$$

ou seja,

$$m_k^{PQS}(d) = \frac{1}{2} d^t W_k d + \nabla f_0(x^k)^t d + c_1, \quad (2.3)$$

onde $c_1 = f_0(x^k) + \lambda^{k^t} (f(x^k) + A(x^k)d)$ e $W_k = W(x^k, \lambda^k)$. Note que, se considerarmos as linearizações dadas em (2.2), c_1 é uma constante. Assim, podemos excluir essa expressão ao minimizarmos o modelo (2.3), pois a solução encontrada será a mesma.

Agora, podemos reescrever o subproblema quadrático. A função objetivo passa a ser o modelo quadrático do lagrangeano (2.3), sem o termo constante, e as restrições são as linearizações dadas em (2.2). Abaixo temos a formulação do problema

$$(P_{Q_E}) \quad \begin{cases} \text{minimizar} & \frac{1}{2} d^t W_k d + \nabla f_0(x^k)^t d \\ \text{sujeito a} & : f(x^k) + A(x^k)d = 0. \end{cases}$$

O algoritmo que descreve o método de PQS puro é descrito a seguir:

Algoritmo 2.2.1 *PQS puro*

Dado: $(x^0, \lambda^0) \in \mathbb{R}^n \times \mathbb{R}^m$.

$k = 0$;

REPITA enquanto não é solução

 Resolva o problema quadrático (PQ_E) determinando uma solução primal-dual (d^k, μ^k) ;

$$x^{k+1} = x^k + d^k;$$

$$\lambda^{k+1} = \mu^k;$$

$$k = k + 1;$$

FIM

Relação entre PQS e o método de Newton. Discutiremos, inicialmente, a aplicação do método de Newton às condições de KKT do problema (P_E) , a partir de um ponto (x^k, λ^k) . Veremos, em seguida, que isto corresponde a uma iteração do método PQS.

Escrevendo as condições necessárias de otimalidade de primeira ordem para o problema (P_E) , $\nabla \ell(x, \lambda) = 0$, temos

$$F(x, \lambda) = \begin{bmatrix} \nabla f_0(x) + A^t(x)\lambda \\ f(x) \end{bmatrix} = 0. \quad (2.4)$$

Sabemos que os candidatos à solução do problema (P_E) devem satisfazer essas condições, assim precisamos resolver o sistema (2.4), que tem $n + m$ equações e incógnitas $(x, \lambda) \in \mathbb{R}^{n+m}$. Para resolvê-lo, vamos aplicar o método de Newton:

$$\nabla F(x)d = -F(x),$$

ou seja, na iteração k , o passo de Newton (d_N^k, d_N^λ) , é uma solução do sistema:

$$\begin{bmatrix} W_k & A_k^t \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d_N^k \\ d_N^\lambda \end{bmatrix} = \begin{bmatrix} -(\nabla f_0(x^k) + A_k^t \lambda^k) \\ -f(x^k) \end{bmatrix}, \quad (2.5)$$

onde $W_k = W(x^k, \lambda^k)$ e $A_k = A(x^k)$. Vamos denotar a matriz jacobiana de F por $J(x, \lambda)$, assim $J_k = J(x^k, \lambda^k)$.

Quando a matriz J_k é não singular o passo de Newton é bem definido. Utilizando as hipóteses a seguir é possível mostrar (ver [51, Lema 3.1]) que a matriz J_k , definida acima, é não singular.

- A Jacobiana das restrições tem posto completo, isto é, $\text{posto}(A_k) = m$;
- A matriz W_k é definida positiva no núcleo da Jacobiana das restrições, isto é, $d^t W_k d > 0$ para todo $d \neq 0$ tal que $A_k d = 0$.

Sob essas condições o algoritmo de Newton para resolver sistemas não lineares converge quadraticamente para uma solução, desde que o ponto inicial esteja suficientemente perto dessa solução.

Se as condições para a matriz J_k ser não singular se verificam, o subproblema (PQ_E) tem uma única solução (d^k, μ^k) que satisfaz $\nabla \ell(d^k, \mu^k) = 0$, ou seja,

$$\begin{bmatrix} W_k d^k + \nabla f_0(x^k)^t + A_k^t \mu^k \\ A_k d^k + f(x^k) \end{bmatrix} = 0,$$

que pode ser reescrito como

$$\begin{bmatrix} W_k & A_k^t \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d^k \\ \mu^k \end{bmatrix} = \begin{bmatrix} -\nabla f_0(x^k) \\ -f(x^k) \end{bmatrix}. \quad (2.6)$$

Podemos fazer uma relação entre esse sistema e o obtido pelas equações de Newton (2.5). Se adicionarmos $A_k^t \lambda^k$ em ambos os lados da primeira equação em (2.5), obtemos

$$W_k d_N^k + A_k^t d_N^\lambda + A_k^t \lambda^k = -(\nabla f_0(x^k) + A_k^t \lambda^k) + A_k^t \lambda^k.$$

E podemos reescrever o sistema (2.5) como

$$\begin{bmatrix} W_k & A_k^t \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d_N^k \\ \lambda^k + d_N^\lambda \end{bmatrix} = \begin{bmatrix} -\nabla f_0(x^k) \\ -f(x^k) \end{bmatrix}. \quad (2.7)$$

Assim, comparando os sistemas (2.6) e (2.7) temos que

- $d^k = d_N^k$ e
- $\mu^k = \lambda^k + d_N^\lambda$,

ou seja, o passo de Newton, definido por (2.5), aplicado às condições de KKT do problema original coincide com passo de PQS definido em (2.6).

Deste modo, os resultados de convergência para o método de PQS puro são equivalentes aos do método de Newton. Em [5], Bonnans, Gilbert, Lemaréchal e Sagastizábal estabelecem que a seqüência (x^k, λ^k) , gerada pelo Algoritmo 2.2.1, converge quadraticamente para a solução, desde que o ponto inicial esteja suficientemente próximo dessa solução e mais algumas condições sejam satisfeitas.

O método de PQS para problemas com restrições de igualdade e desigualdade. Retomamos, agora, o problema original

$$(P) \quad \begin{cases} \text{minimizar} & f_0(x) \\ \text{sujeito a :} & f_{\mathcal{E}}(x) = 0 \\ & f_{\mathcal{I}}(x) \leq 0, \end{cases}$$

Neste caso, o método de PQS consiste em, dado $(x^k, \lambda^k, \mu^k) \in \mathbb{R}^{n+m}$, com $m = p + q$, resolver a cada iteração o seguinte problema quadrático

$$(PQ) \quad \begin{cases} \text{minimizar} & \frac{1}{2}d^t W_k d + \nabla f_0(x^k)^t d \\ \text{sujeito a :} & f_{\mathcal{E}}(x^k) + A_{\mathcal{E}}(x^k)d = 0 \\ & f_{\mathcal{I}}(x^k) + A_{\mathcal{I}}(x^k)d \leq 0, \end{cases}$$

sendo que (λ^k, μ^k) são os multiplicadores de Lagrange correspondentes às restrições de igualdade e desigualdade, respectivamente.

Aqui também pode-se estabelecer uma relação com o método de Newton e a convergência quadrática da seqüência gerada pelo método, desde que o ponto inicial esteja suficientemente próximo da solução. Veja, por exemplo, em [5, 13].

PQS com Região de Confiança. Como comentamos anteriormente, o método PQS puro só converge para uma solução se o ponto inicial estiver suficientemente próximo desta solução. Entretanto, existem maneiras de modificar esse algoritmo de modo que uma solução seja encontrada a partir de qualquer ponto inicial. Chamamos essa modificação de globalização do método PQS. Vamos discutir neste trabalho somente a globalização com região de confiança, sendo que outras técnicas podem ser encontradas em [5, 13, 26, 31, 44, 54].

A idéia de introduzir a região de confiança no subproblema é a mesma que a do Capítulo 1, ou seja, restringir o tamanho do passo de modo que o modelo utilizado continue sendo um bom representante para a função objetivo. Assim, adicionando a restrição de região de confiança ao subproblema (PQ) , obtemos

$$(PQ^{RC}) \quad \begin{cases} \text{minimizar} & \frac{1}{2}d^t W_k d + \nabla f_0(x^k)^t d \\ \text{sujeito a :} & f_{\mathcal{E}}(x^k) + A_{\mathcal{E}}(x^k)d = 0 \\ & f_{\mathcal{I}}(x^k) + A_{\mathcal{I}}(x^k)d \leq 0 \\ & \|d\| \leq \Delta. \end{cases}$$

A inclusão da região de confiança torna o problema consideravelmente mais difícil, podendo até ser incompatível. Na Figura 2.2 temos uma ilustração do caso de incompatibilidade, para um problema em \mathbb{R}^2 com apenas uma restrição de igualdade. As linhas tracejadas representam as curvas de nível da função objetivo e as linhas cheias representam o conjunto viável e sua linearização. Podemos ver que o conjunto dos pontos viáveis não tem nenhum ponto em comum com o dos pontos que satisfazem a região de confiança.

Para resolver o caso de incompatibilidade não basta apenas aumentar o raio, pois isto vai contra a idéia de inclusão da região de confiança. Uma abordagem para contornar esse problema é dada em [54] e consiste em fazer uma translação das

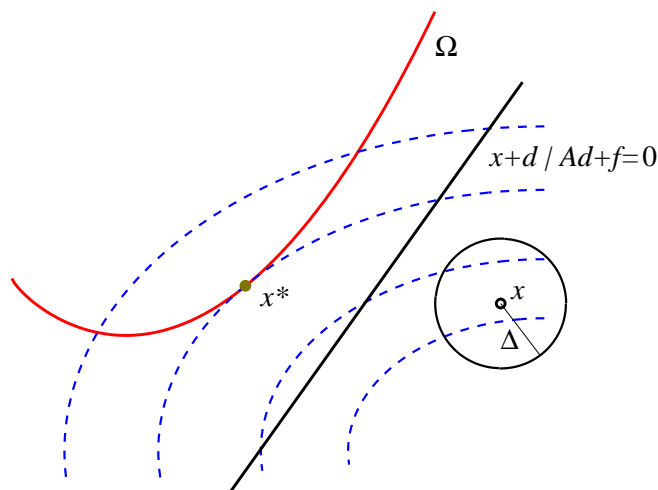


Figura 2.2: Incompatibilidade do subproblema (PQ^{RC}) .

restrições lineares de modo que o novo problema seja compatível.

A estratégia que descrevemos aqui é devida a Byrd [6] e Omojokun [55] e consiste em decompor o passo em duas componentes: uma normal e uma tangencial. O passo *normal* é um passo de viabilidade, onde preocupa-se somente em reduzir a inviabilidade do problema. No passo *tangencial* o objetivo é reduzir o valor da função objetivo, sem perder a viabilidade conseguida na primeira fase.

Resolução dos subproblemas normal e tangencial. Discutiremos agora a obtenção dos passos normal e tangencial. Para simplificar a exposição, consideraremos novamente $\mathcal{I} = \emptyset$, ou seja, o problema só com restrições de igualdade:

$$(P_E) \quad \begin{cases} \text{minimizar} & f_0(x) \\ \text{sujeito a} & : f(x) = 0. \end{cases}$$

Dado (x^k, λ^k) devemos resolver a cada iteração o problema quadrático

$$(PQ_E^{RC}) \quad \begin{cases} \text{minimizar} & \frac{1}{2}d^t W_k d + \nabla f_0(x^k)^t d \\ \text{sujeito a} & : f(x^k) + A(x^k)d = 0 \\ & \|d\| \leq \Delta, \end{cases}$$

onde $W_k = \nabla_{xx}^2 \ell(x^k, \lambda^k)$.

Nesse caso, seguindo as estratégias de Byrd e Omojokun [6, 55], o passo é decomposto em normal e tangencial. O passo normal pode ser considerado como um solução do problema

$$\begin{cases} \text{minimizar} & \|A_k n + f(x^k)\| \\ \text{sujeito a} & : \|n\| \leq \xi \Delta, \end{cases} \quad (2.8)$$

onde $\xi \in (0, 1)$ é um parâmetro que evita que o passo normal chegue até a fronteira

da região de confiança. Isto é necessário para que a próxima fase esteja bem definida.

Note que o subproblema normal (2.8) é simplesmente um problema de minimizar uma quadrática na bola. Assim, para resolvê-lo, podemos usar os algoritmos de Dogleg ou Steihaug que descrevemos nas Subseções 1.2.1 e 1.2.2.

Uma outra abordagem para o cálculo do passo normal é fazer uma projeção do ponto corrente x^k no conjunto definido pela linearização das restrições de (P_E) . Esta abordagem pode ser encontrada em [19].

Tendo calculado o passo normal n^k , voltamos ao problema (PQ_E^{RC}) , mas relaxando a restrição linear:

$$\begin{cases} \text{minimizar} & \frac{1}{2}d^t W_k d + \nabla f_0(x^k)^t d \\ \text{sujeito a :} & A_k d = A_k n^k \\ & \|d\| \leq \Delta. \end{cases} \quad (2.9)$$

Esse problema com certeza é compatível, pois o passo normal satisfaz as restrições. A diferença entre o subproblema tangencial (2.9) e o problema de região de confiança (1.3), dado no Capítulo 1, está na restrição $A_k d = A_k n^k$. Queremos então eliminar essa restrição para que o subproblema tangencial seja resolvido com um dos métodos utilizados para resolver o problema de região de confiança.

Pelo método de Byrd e Omojokum conseguimos transformar esse problema fazendo $d = n^k + t$, onde t é o *passo tangencial*. Substituindo isso na restrição de igualdade de (2.9), temos

$$\begin{aligned} A_k(n^k + t) &= A_k n^k \\ A_k t &= 0 \Rightarrow t \in N(A_k) \Rightarrow t = Z_k u, \end{aligned}$$

onde as colunas da matriz Z_k formam uma base para o espaço nulo de A_k e $u \in \mathbb{R}^{n-m}$. Com isso temos

$$d = n^k + Z_k u. \quad (2.10)$$

Vamos reescrever a função objetivo de (2.9) utilizando o passo total (2.10):

$$\begin{aligned} & \frac{1}{2}d^t W_k d + \nabla f_0(x^k)^t d = \\ &= \frac{1}{2}(n^k + Z_k u)^t W_k (n^k + Z_k u) + \nabla f_0(x^k)^t (n^k + Z_k u) \\ &= \frac{1}{2}(Z_k u)^t W_k Z_k u + (\nabla f_0(x^k) + W_k n^k)^t Z_k u + c_2. \end{aligned} \quad (2.11)$$

onde $c_2 = \frac{1}{2}n^{k^t} W_k n^k + \nabla f_0(x^k)^t n^k$ é uma constante, pois não depende de u .

Agora vamos reescrever a segunda restrição de (2.9). Elevando ao quadrado

ambos os lados da desigualdade, temos

$$\begin{aligned} \|n^k + Z_k u\|^2 &\leq \Delta^2 \\ (n^k + Z_k u)^t (n^k + Z_k u) &\leq \Delta^2 \\ (n^k)^t n^k + 2n^{k,t} Z_k u + (Z_k u)^t Z_k u &\leq \Delta^2, \end{aligned} \quad (2.12)$$

Como $Z_k u \in N(A_k)$ e $n^k \in I(A_k^t)$, sabemos que $(Z_k u)$ e n^k são ortogonais, ou seja, $(Z_k u)^t n^k = 0$. Assim, (2.12) pode ser reescrita como

$$\begin{aligned} \|n^k\|^2 + \|Z_k u\|^2 &\leq \Delta^2 \\ \|Z_k u\| &\leq \sqrt{\Delta^2 - \|n^k\|^2}. \end{aligned} \quad (2.13)$$

Com isso, o subproblema tangencial pode ser reduzido a um problema de minimizar uma quadrática na bola. Colocando (2.11), sem o termo constante c_2 , na função objetivo e (2.13) como restrição, temos o novo subproblema tangencial:

$$\begin{cases} \text{minimizar} & \frac{1}{2}(Z_k u)^t W_k Z_k u + (\nabla f_0(x^k) + W_k n^k)^t Z_k u \\ \text{sujeito a :} & \|Z_k u\| \leq \sqrt{\Delta^2 - \|n^k\|^2}. \end{cases} \quad (2.14)$$

Para resolver esse subproblema podemos usar os métodos de Dogleg e Steihaug, vistos no Capítulo 1, lembrando que o método Dogleg só pode ser usado quando se tem a garantia de que a matriz $Z_k^t W_k Z_k$ é positiva definida.

Considerando u^k uma solução do problema (2.14), definimos $t^k = Z_k u^k$ e, finalmente, por (2.10), temos $d^k = n^k + t^k$.

Na Figura 2.3 ilustramos a decomposição do passo d^k em normal e tangencial. As curvas de nível estão representadas por linhas tracejadas e as linhas cheias representam o conjunto viável e sua linearização. A região de confiança está representada pela bola maior (linha cheia) e a bola menor representa a redução no raio de região de confiança para o passo normal.

2.2.2 Restauração inexata

Os métodos de restauração remontam ao método do gradiente projetado de Rosen [64, 65], aos algoritmos de SGRA de Miele *et al* [46, 47, 48] e ao método GRG introduzido por Abadie e Carpentier [1].

Como comentamos anteriormente, uma forma de trabalhar com problemas de programação não linear consiste em utilizar processos iterativos. Assim, devido a não linearidade das restrições, um ponto viável raramente é encontrado e o processo de restauração é normalmente inexato.

A idéia do método de restauração inexata é desenvolver algoritmos que

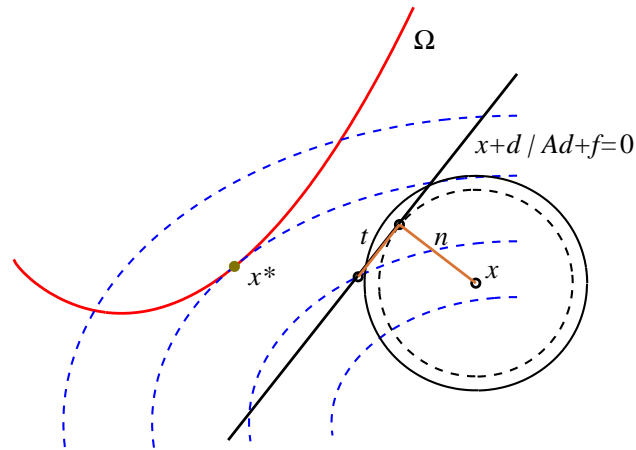


Figura 2.3: Decomposição do passo.

podem dar passos inviáveis, mas, quando próximos à solução, obtenham viabilidade. Dado o ponto corrente x^k , Martínez [41] e Martínez e Pilotta [42] desenvolveram um método onde a obtenção do passo é feita em duas fases: a primeira fase reduz uma medida de inviabilidade, obtendo um ponto z^k , e a segunda fase procura reduzir o valor da função objetivo em relação a esse ponto. Uma estratégia de região de confiança é aplicada em torno de z^k para controlar o tamanho do passo, que é dado em uma aproximação tangencial do conjunto viável. Nesses algoritmos as fases de otimalidade e viabilidade são mais independentes em relação a outros métodos que decompõem o passo, como veremos a seguir.

Fase de viabilidade. O objetivo dessa fase é encontrar um ponto z^k que se aproxime do conjunto viável do problema. Como utilizamos algoritmos iterativos que aceitam pontos inviáveis no decorrer do processo, vamos considerar uma função h , que mede de algum modo a inviabilidade.

A princípio, qualquer algoritmo que reduza essa medida de inviabilidade h pode ser usado nessa fase, desde que satisfaça algumas hipóteses razoáveis sobre a sua eficiência. Martínez [41] e Martínez e Pilotta [42] descrevem um algoritmo para problemas com restrições de igualdade não lineares e restrições de desigualdade em caixa para a fase de viabilidade. O método para essa fase pode usar também as idéias de Byrd, Gilbert e Nocedal [7] e de Byrd, Hribar e Nocedal [8], que reescrevem o problemas usando restrições de igualdade e variáveis de folga não negativas. Mukai e Polak [52] discutem tolerâncias relacionadas à viabilidade em seu trabalho. Em [62], Rom e Avriel provam teoremas de convergência para algoritmos clássicos de restauração usando o esquema Mukai-Polak e redução progressiva da tolerância de viabilidade. O algoritmo da fase de viabilidade pode falhar se h tem um ponto estacionário inviável. Neste caso, o método pára com insucesso.

Fase de otimalidade. Nesta fase a intenção é encontrar um ponto $z^k + d$ que

reduza o valor da função objetivo em relação ao ponto z^k , mas sem perder muito da viabilidade conseguida na primeira fase. Para isso, o passo de otimalidade é dado no conjunto linearizado

$$L(z^k) = \{z^k + d \in \mathbb{R}^n \mid A_{\mathcal{E}}(z^k)d = 0, f_{\mathcal{I}}(z^k) + A_{\mathcal{I}}(z^k)d \leq \max\{0, f_{\mathcal{I}}(z^k)\}\}. \quad (2.15)$$

Nessa fase, uma estratégia de região de confiança é aplicada em torno de z^k para controlar o tamanho do passo. Se o ponto obtido na fase de otimalidade não for um bom ponto, segundo algum critério que veremos na seção seguinte, somente essa fase é repetida.

Na Figura 2.4 temos a ilustração de um passo de um algoritmo de restauração inexata. O ponto z é o ponto obtido na fase de viabilidade e x_{RI}^+ é o ponto tentativo obtido na fase de otimalidade.

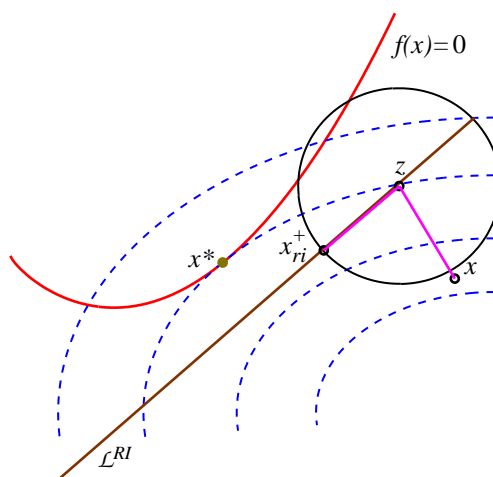


Figura 2.4: Um passo de restauração inexata.

Comparação entre restauração inexata e PQS. Vamos agora fazer uma comparação entre o método de restauração inexata e o método de PQS, que vimos nas seções anteriores. O objetivo é destacar as semelhanças e diferenças entre esses dois métodos.

A principal diferença entre os métodos está na obtenção do passo. Apesar de, em ambos, cada iteração poder ser decomposta em duas fases: uma de viabilidade e outra de otimalidade, no método de restauração inexata existe uma certa liberdade para os algoritmo internos.

Nos dois métodos é usada uma estratégia de região de confiança. Aqui, a diferença reside no centro dessa região. No método de PQS, o ponto corrente x^k é o centro da região de confiança, enquanto que no método de restauração inexata é o ponto z^k obtido na fase de viabilidade. A vantagem de centrar a região de confiança no ponto z^k é que, se o ponto tentativo obtido na fase de otimalidade for recusado,

somente a última fase é repetida.

A diferença no centro da região de confiança reflete também na linearização das restrições. Enquanto, no método de restauração inexata as restrições e a jacobiana são avaliadas em z^k , no método de PQS, são avaliadas em x^k . Note que o conjunto linearizado, no método de restauração inexata, foi definido por

$$L^{RI}(z^k) = \{z^k + d \in \mathbb{R}^n \mid A_{\mathcal{E}}(z^k)d = 0, f_{\mathcal{I}}(z^k) + A_{\mathcal{I}}(z^k)d \leq \max\{0, f_{\mathcal{I}}(z^k)\}\}.$$

De acordo com o problema quadrático (PQ^{RC}) do método PQS, a linearização das restrições no ponto corrente é dada por

$$L^{PQS} = \{x^k + d \in \mathbb{R}^n \mid f_{\mathcal{E}}(x^k) + A_{\mathcal{E}}(x^k)d = 0, f_{\mathcal{I}}(x^k) + A_{\mathcal{I}}(x^k)d \leq 0\}.$$

Na Figura 2.5 temos representada uma iteração de cada método. Os pontos tentativos de restauração inexata e PQS estão indicados por x_{RI}^+ e x_{PQS}^+ respectivamente e o ponto corrente por x . Representamos também os conjuntos linearizados L^{RI} e L^{PQS} .

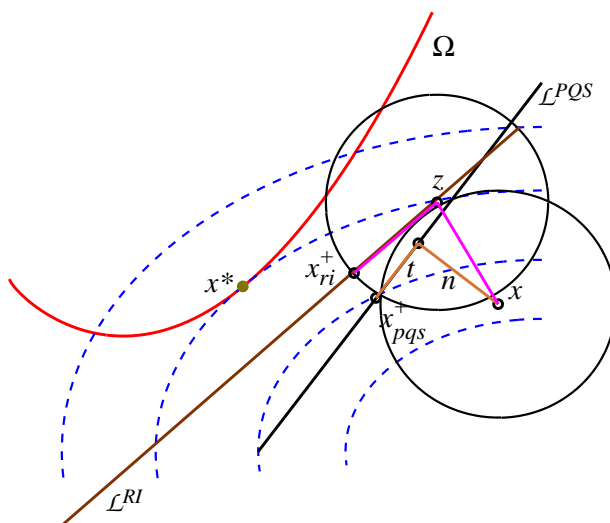


Figura 2.5: Comparação entre restauração inexata e PQS.

2.3 Critérios de aceitação do passo

Nós discutimos nesse capítulo dois métodos para encontrar o passo ao resolver problemas de programação não linear: programação quadrática sequencial e restauração inexata. Quando trabalhamos com problemas com restrições precisamos medir o progresso do algoritmo através de algum critério. Para sabermos se o ponto obtido é melhor que o anterior temos que avaliar tanto o decréscimo na função objetivo quanto na inviabilidade do problema.

Conciliar esses dois objetivos, otimalidade e viabilidade, é uma tarefa difícil. Obtido um ponto tentativo x^+ precisamos decidir se esse é um bom passo, e nem sempre obtemos resultados positivos nos dois objetivos. Para medir os esforços em otimalidade e viabilidade são usados critérios de aceitação do passo. Um critério clássico para medir o progresso do algoritmo é fazer uso de uma função de mérito, que é uma combinação entre uma medida de inviabilidade e uma de otimalidade.

O uso das funções de mérito é feito em vários algoritmos. Gomes, Maciel e Martínez [26] a utilizaram para avaliar o passo obtido através de PQS com região de confiança. Byrd, Gilbert e Nocedal [7] e Byrd, Hribar e Nocedal [8] também utilizam função de mérito para avaliar o passo obtido por um algoritmo de PQS denominado NITRO. Martínez [41] e Martínez e Pilotta [42] também fazem uso de uma função de mérito, mas em algoritmos de Restauração Inexata.

Uma função de mérito muito utilizada é dada por

$$\phi(x) = f_0(x) + \sigma h(x),$$

onde σ é chamado de parâmetro de penalidade e h é uma medida de inviabilidade. Para ser aceito, um ponto tentativo deve provocar um decréscimo suficiente em ϕ , em relação ao ponto anterior.

Na Figura 2.6 temos uma ilustração de uma função de mérito. Por abuso de notação, o ponto x representa o par $(f_0(x), h(x))$. Os pontos que pertencem à área pintada são pontos que serão recusados pelo método, segundo o critério da função de mérito. Note que a inclinação da reta é determinada pelo parâmetro σ .

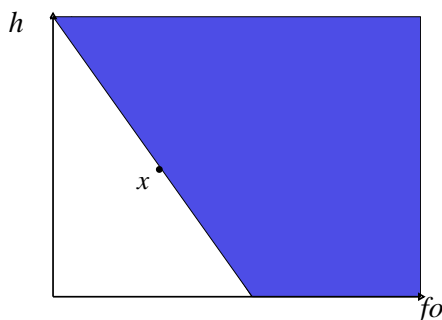


Figura 2.6: Função de mérito.

A escolha do parâmetro σ é uma questão delicada. Dependendo do valor que esse parâmetro assume, o algoritmo pode recusar uma solução ótima ou, por outro lado, ficar muito lento. Apesar de bastante utilizadas, as funções de mérito apresentam essa desvantagem. Outra desvantagem é que algumas funções clássicas não são diferenciáveis.

Como uma alternativa para o uso das funções de mérito, Fletcher e Leyffer [20] introduziram os chamados métodos de filtro, e uma prova de convergência global

foi dada em Fletcher, Gould, Leyffer, Toint e Wachter [19]. Nesses algoritmos a obtenção do passo é feita por PQS. Gonzaga, Karas e Vanti [28] propuseram um algoritmo de filtro globalmente convergente que é baseado em restauração inexata.

Os algoritmos de filtro definem uma região de pontos proibidos segundo a regra usual de dominação de Pareto:

“ x domina y se, e somente se, $f_0(y) \geq f_0(x)$ e $h(y) \geq h(x)$ ”.

Um ponto tentativo x^+ só será aceito se não for dominado pelo ponto corrente segundo esse critério. A Figura 2.7 ilustra esse critério. Novamente estamos fazendo um abuso de notação, e os pontos que pertencem à área pintada são pontos que não seriam aceitos segundo esse critério.

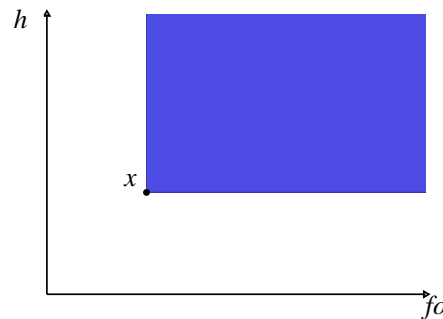


Figura 2.7: Filtro.

Na Figura 2.8 temos uma comparação entre o critério de filtro e a função de mérito. A área pintada, inteira, representa os pontos proibidos pela função de mérito. A área pintada mais clara representa os pontos proibidos segundo o critério de filtro. O ponto x representa o ponto corrente e o ponto x^+ representa um ponto tentativo. Note que, nesse caso, o ponto tentativo é recusado pela função de mérito, mas é aceito pelo filtro. Assim, a grosso modo, podemos dizer que os métodos de filtro são menos restritivos que os métodos com função de mérito.

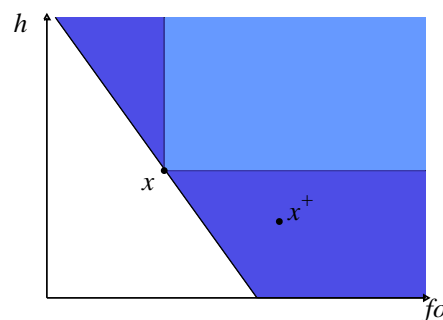


Figura 2.8: Comparação entre filtro e função de mérito.

No capítulo seguinte fazemos uma descrição detalhada dos métodos de filtro e descrevemos alguns algoritmos que utilizam esse critério.

Capítulo 3

Métodos de filtro

No Capítulo 2 vimos métodos para obtenção do passo ao resolver um problema de programação não linear com restrições e alguns critérios para avaliar a qualidade do passo obtido.

Um critério clássico para avaliar o ponto tentativo é fazer uso de funções de mérito, como vimos no capítulo anterior. Um outro critério, mais recente, consiste no uso de filtros, propostos, inicialmente, por Fletcher e Leyffer [20]. O filtro define uma região proibida guardando pares ordenados que representam a função objetivo e a inviabilidade.

O método proposto por Fletcher e Leyffer [20] calcula o passo baseado em programação quadrática sequencial (PQS). A convergência global do método foi obtida por Fletcher, Leyffer e Toint [22], assumindo que uma solução global dos subproblemas quadráticos é encontrada (subproblemas gerados no método de PQS, visto na Subseção 2.2.1). Usando técnicas de decomposição do passo baseadas em Byrd [7] e Omojokun [55], Fletcher *et al* [19] estabelecem convergência global assumindo que apenas soluções aproximadas para os subproblemas quadráticos sejam encontradas.

Gonzaga, Karas e Vanti [28] propuseram um algoritmo de filtro globalmente convergente, baseado em restauração inexata [41, 42]. Nesse algoritmo cada iteração é composta de uma fase de restauração, que reduz uma medida de inviabilidade, e de uma fase de otimalidade, que melhora o valor da função objetivo em uma aproximação tangencial do conjunto viável.

Fletcher, Leyffer e Toint [22] propuseram uma modificação na definição da região proibida dada pelo filtro original de Fletcher e Leyffer [20], incorporando uma inclinação na região proibida.

Os métodos de filtro foram aplicados no contexto de Programação Linear Sequencial (PLS) com região de confiança. Chin e Fletcher [12] e Fletcher, Leyffer e Toint [21] apresentam prova de convergência global para o método.

As técnicas de filtro foram também aliadas aos algoritmos de pontos inte-

riores por Ulbrich, Ulbrich e Vicente [67], que estabelecem a convergência global do algoritmo em seu trabalho.

Neste capítulo, que é a parte principal dessa dissertação, discutiremos alguns métodos de filtro existentes, ressaltando suas semelhanças e diferenças. A parte original dessa dissertação consiste na proposta de um algoritmo baseado em Gonzaga, Karas e Vanti [28] com o filtro dado em Fletcher, Leyffer e Toint [22]. Apresentamos a análise de convergência do algoritmo proposto.

3.1 O conceito de filtro

Relembrando o problema que estamos trabalhando, temos

$$(P) \quad \begin{cases} \text{minimizar} & f_0(x) \\ \text{sujeito a} & : \quad ex = 0 \\ & f_{\mathcal{I}}(x) \leq 0, \end{cases}$$

onde $f_i : \mathbb{R}^n \rightarrow \mathbb{R}, i \in \{0\} \cup \mathcal{E} \cup \mathcal{I}$, são funções continuamente diferenciáveis.

Queremos minimizar a função objetivo e satisfazer as restrições. Como utilizamos algoritmos iterativos que aceitam pontos inviáveis no decorrer do processo, vamos considerar a função $f^+ : \mathbb{R}^n \rightarrow \mathbb{R}^m$, dada por

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

e definir uma medida de inviabilidade $h : \mathbb{R}^n \rightarrow \mathbb{R}$ por

$$h(x) = \|f^+(x)\|, \quad (3.1)$$

onde $\|\cdot\|$ é uma norma arbitrária e m é a cardinalidade de $\mathcal{E} \cup \mathcal{I}$.

Nos métodos de filtro a otimalidade e a inviabilidade são avaliadas separadamente, utilizando a função objetivo f_0 e a medida de inviabilidade h . Estes métodos são caracterizados pelo uso do conceito de dominância:

Definição 3.1.1 *Dominância:* o par (f_0, h) é dominado pelo par (f_0^j, h^j) se, e somente se

$$f_0 \geq f_0^j \quad e \quad h \geq h^j.$$

Agora definimos o filtro, que é um conjunto de pares cujas componentes representam a função objetivo e a inviabilidade:

$$F = \{(f_0^1, h^1), (f_0^2, h^2), \dots, (f_0^r, h^r)\}.$$

O par (f_0, h) é proibido pelo filtro se ele é dominado por algum par (f_0^j, h^j) do filtro corrente. Assim, temos uma região proibida pelo filtro, ou seja, um conjunto de pares (f_0, h) que são dominados por algum par (f_0^j, h^j) pertencente ao filtro.

Na Figura 3.1 temos a ilustração do conceito de dominância e da região proibida pelo par (f_0^j, h^j) .

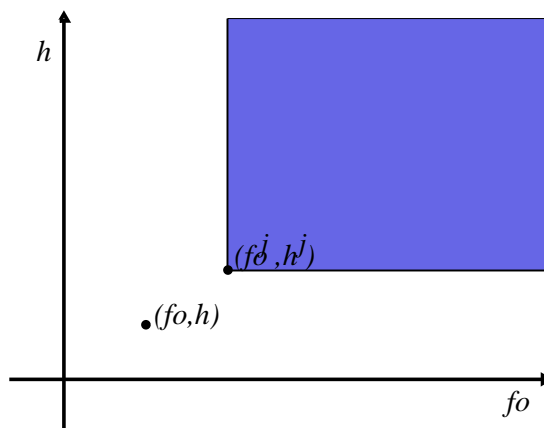


Figura 3.1: Dominância e região proibida.

Com isso, podemos definir o conjunto dos pontos x que pertencem à região proibida pelo filtro como:

$$\mathcal{F} = \{x \in \mathbb{R}^n \mid (f_0(x), h(x)) \text{ é dominado por algum } (f_0^j, h^j) \in F\}.$$

3.1.1 Margens para o Filtro

Para exigir uma redução suficiente em f_0 ou h colocamos uma margem no filtro, ou seja, passamos a definir uma região proibida um pouco maior. As margens são importantes para se garantir a convergência dessa classe de métodos.

A primeira margem que discutiremos foi introduzida por Fletcher e Leyffer em [20]. Neste trabalho vamos nos referir a essa margem como *margem reta* e ao método de filtro que utiliza essa margem como *filtro reto*.

Com essa margem definimos um outro conceito de dominância, que chamaremos de Dominância com Margem Reta, como é dado a seguir:

Definição 3.1.2 Dominância com Margem Reta: Dado $0 < \alpha < 1$. Um par (f_0, h) é dominado pelo par (f_0^j, h^j) se, e somente se,

$$h \geq (1 - \alpha)h^j \quad e \quad f_0 \geq f_0^j - \alpha h^j.$$

Note, na Figura 3.2, que a região proibida é definida por retas paralelas aos eixos.

Chin [11] propôs em sua tese de doutorado uma modificação na definição do filtro, incorporando uma margem inclinada na região proibida. Essa modificação foi

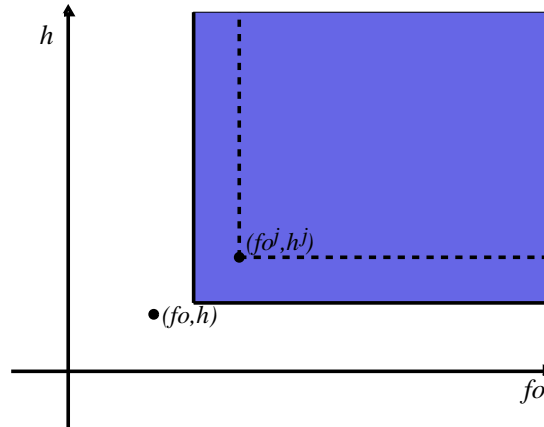


Figura 3.2: Margem reta.

usada por Chin e Fletcher [12] no contexto de programação linear sequencial e por Fletcher, Leyffer e Toint [22] em programação quadrática sequencial. Vamos chamar essa margem de *margem inclinada* e o método de filtro que utiliza essa margem de *filtro inclinado*. Assim, definimos um outro conceito de dominância:

Definição 3.1.3 *Dominância com Margem Inclinada:* Dado $0 < \alpha < 1$. Um par (f_0, h) é dominado pelo par (f_0^j, h^j) se, e somente se,

$$h \geq (1 - \alpha)h^j \quad e \quad f_0 + \alpha h \geq f_0^j.$$

A Figura 3.3 ilustra a margem inclinada. A região pintada representa o conjunto de pares (f_0, h) que são dominados pelo par (f_0^j, h^j) .

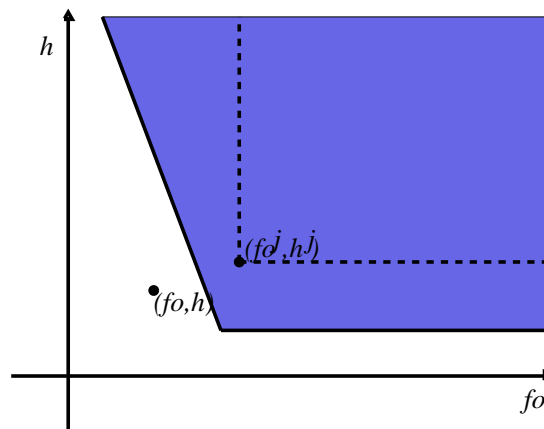


Figura 3.3: Margem inclinada.

Na Figura 3.4 comparamos esses tipos de filtro. Podemos ver que uma pequena área triangular proibida pela margem reta, deixa de ser proibida se considerarmos a margem inclinada. Por outro lado, há uma área triangular ilimitada onde ocorre o inverso, ou seja, essa área é proibida pelo filtro inclinado mas não pelo filtro reto.

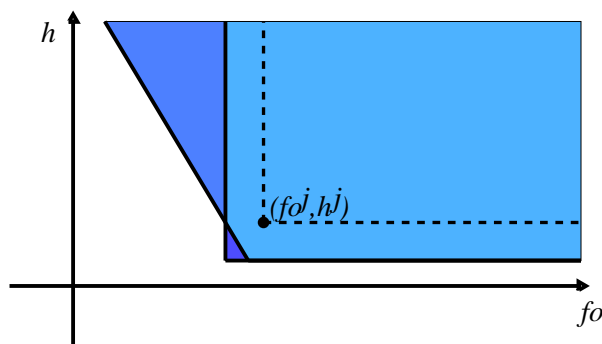


Figura 3.4: Margens.

A inclinação na margem do filtro se torna essencial para garantir alguns resultados relacionados à viabilidade que não são válidos para o filtro reto.

3.2 Algoritmo geral

Nesta seção vamos apresentar um algoritmo geral de filtro.

Algoritmo 3.2.1 Algoritmo geral de filtro

Dado: $x^0, F_0 = \emptyset, \mathcal{F}_0 = \emptyset$

$k = 0;$

REPITA enquanto não é solução

1. Defina (\tilde{f}_0, \tilde{h}) ;
2. $\bar{F}_k = F_k \cup \{(\tilde{f}_0, \tilde{h})\}$;
3. Defina $\bar{\mathcal{F}}_k$;
4. Calcule $x^{k+1} \notin \bar{\mathcal{F}}_k$;
5. Atualize o filtro;
6. $k = k + 1$;

FIM

Este algoritmo está colocado de uma forma bastante genérica e requer a explanação de alguns detalhes que caracterizarão um algoritmo específico, como veremos na Seção 3.3.

Item 1. A cada iteração um par (\tilde{f}_0, \tilde{h}) é determinado pelo filtro. Esse par é o vértice da região proibida pelo filtro, que depende do tipo de filtro utilizado: reto ou inclinado.

Dado o ponto corrente $x^k \in \mathbb{R}^n$ e $\alpha \in (0, 1)$, no caso da margem reta, o par é definido por

$$\begin{cases} \tilde{f}_0 = f_0(x^k) - \alpha h(x^k) \\ \tilde{h} = (1 - \alpha)h(x^k), \end{cases} \quad (3.2)$$

e é ilustrado na Figura 3.5.

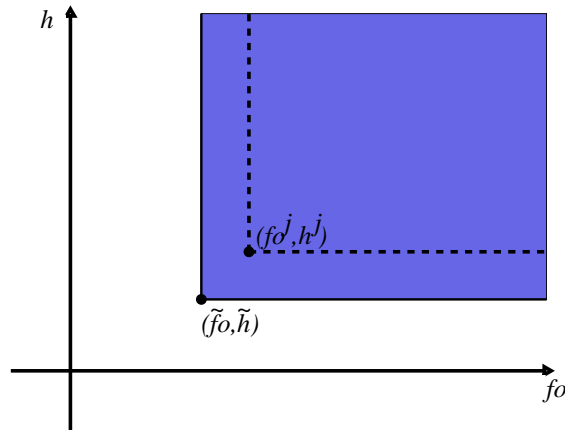


Figura 3.5: Vértice do filtro reto.

No caso da margem inclinada cabe ver que a equação da reta inclinada definida pelo filtro é dada por:

$$f_0(x) + \alpha h(x) = f_0(x^k),$$

e o vértice da região proibida:

$$\begin{cases} \tilde{f}_0 = f_0(x^k) - \alpha(1 - \alpha)h(x^k) = f_0(x^k) - \alpha\tilde{h} \\ \tilde{h} = (1 - \alpha)h(x^k), \end{cases} \quad (3.3)$$

que é ilustrado na Figura 3.6.

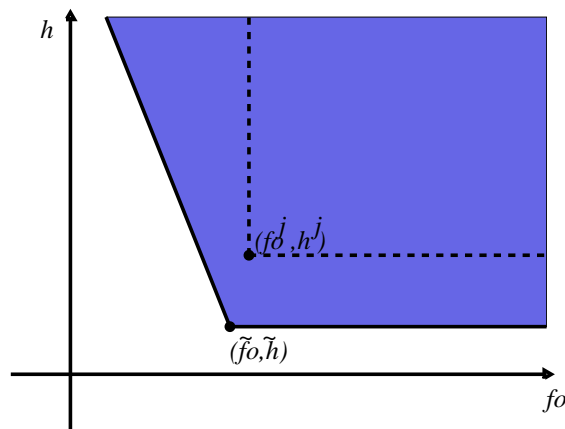


Figura 3.6: Vértice do filtro inclinado.

Item 2. O item 2 define o filtro temporário \bar{F}_k como a união do filtro corrente F_k e o par (\tilde{f}_0, \tilde{h}) . O filtro temporário se tornará permanente ou não conforme o critério de atualização do filtro estabelecido no item 5.

Item 3. O item 3 define uma região proibida $\bar{\mathcal{F}}_k$, no espaço \mathbb{R}^n , determinada pelo filtro temporário \bar{F}_k . A região proibida $\bar{\mathcal{F}}_k$ é a união da região proibida corrente \mathcal{F}_k e a região proibida pelo par (\tilde{f}_0, \tilde{h}) .

Devemos ter em mente que a região proibida $\bar{\mathcal{F}}_k$ nunca é construída, mas nos ajuda a entender o processo.

No caso da margem reta, temos que

$$\bar{\mathcal{F}}_k = \mathcal{F}_k \cup \{x \in \mathbb{R}^n \mid f_0(x) \geq \tilde{f}_0, h(x) \geq \tilde{h}\},$$

onde (\tilde{f}_0, \tilde{h}) é definido em (3.2).

No caso da margem inclinada, temos que

$$\bar{\mathcal{F}}_k = \mathcal{F}_k \cup \{x \in \mathbb{R}^n \mid f_0(x) + \alpha h(x) \geq \tilde{f}_0 + \alpha \tilde{h}, h(x) \geq \tilde{h}\},$$

onde (\tilde{f}_0, \tilde{h}) é definido em (3.3).

Item 4. O item 4 consiste em, a partir do ponto corrente x^k , calcular o ponto x^{k+1} não proibido pelo filtro temporário.

É nesse item que reside o maior trabalho do algoritmo: o cálculo do passo.

Nos trabalhos de [19, 20, 22] o passo é calculado por PQS, enquanto que em [28] o passo é calculado por restauração inexata.

Na próxima seção discutiremos alguns algoritmos de filtro existentes que utilizam métodos distintos para obtenção do passo e definição da margem do filtro.

Item 5. No item 5, atualizamos ou não o filtro, decidindo se o filtro e a região proibida temporários devem tornar-se permanentes ou não.

A grosso modo, o filtro temporário torna-se permanente se não houver um decréscimo na função objetivo no ponto x^{k+1} em relação ao ponto x^k .

Se o filtro for atualizado, definimos

$$F_{k+1} = \bar{F}_k$$

e

$$\mathcal{F}_{k+1} = \bar{\mathcal{F}}_k,$$

ao final da iteração e esta iteração é considerado do tipo “ h ”. Caso contrário será uma iteração do tipo “ f_0 ”.

Descrevemos a seguir alguns métodos de filtro encontrados na literatura.

3.3 Alguns algoritmos de filtro existentes

Na seção anterior apresentamos um algoritmo geral de filtro para resolver um problema de programação não linear.

Agora vamos apresentar alguns algoritmos existentes na literatura que serviram de base para o algoritmo proposto nesta dissertação, na Seção 3.4.

Em nosso trabalho vamos usar uma notação um pouco diferente da exibida nos artigos originais, com o objetivo de facilitar a discussão e a comparação com o algoritmo geral e os algoritmos entre si.

3.3.1 Algoritmo de PQS com filtro reto

O algoritmo clássico, conforme proposto por Fletcher e Leyffer [20], consiste num método de filtro com margem reta e que calcula o passo por PQS. Uma prova de convergência foi dada em [19].

Como vimos na Seção 2.2.1, um método de PQS resolve, a cada iteração, um subproblema quadrático do tipo

$$(PQ^{RC}) \quad \begin{cases} \text{minimizar} & \frac{1}{2}d^t W_k d + \nabla f_0(x^k)^t d \\ \text{sujeito a :} & f_{\mathcal{E}}(x^k) + A_{\mathcal{E}}(x^k)d = 0 \\ & f_{\mathcal{I}}(x^k) + A_{\mathcal{I}}(x^k)d \leq 0 \\ & \|d\| \leq \Delta. \end{cases}$$

Quando este problema é incompatível, o algoritmo proposto em [19, 20] chama uma fase de restauração, que consiste, essencialmente, em minimizar uma medida de inviabilidade h (3.1).

Caso seja compatível, o passo é dividido em duas componentes: uma normal e uma tangencial. O passo normal n^k deve satisfazer a restrição de região de confiança. Encontrado o passo normal, calcula-se o passo tangencial t^k , que deve fornecer uma redução suficiente na função objetivo.

A atualização do filtro se dá em dois momentos: quando o passo normal n^k está fora da fronteira da região de confiança (subproblema normal é incompatível) e quando o passo tentativo $x^k + d$ não pertence ao filtro temporário $\bar{\mathcal{F}}_k$, mas a redução dada no modelo da função objetivo (problema (PQ^{RC})) é negativa.

A seguir colocamos o algoritmo completo de PQS com filtro reto, inspirado em [19].

Algoritmo 3.3.1 PQS com filtro reto

Dados: $x^0 \in \mathbb{R}^n$, W_0 simétrica, $\Delta_0 > 0$, $F_0 = \emptyset$, $\mathcal{F}_0 = \emptyset$, $\alpha \in (0, 1)$,
 $\xi \in (0, 1)$, $c_p \in (0, 1)$, $\eta \in (0, 1)$.

$k = 0$

REPITA (enquanto x^k não for estacionário)

$$(\tilde{f}_0, \tilde{h}) = (f_0(x^k) - \alpha h(x^k), (1 - \alpha)h(x^k));$$

Defina os conjuntos $\bar{F}_k = F_k \cup \{(\tilde{f}_0, \tilde{h})\}$ e

$$\bar{\mathcal{F}}_k = \mathcal{F}_k \cup \{x \in \mathbb{R}^n \mid f_0(x) \geq \tilde{f}_0, h(x) \geq \tilde{h}\};$$

REPITA (enquanto o ponto x^{k+1} não for obtido)

SE o subproblema quadrático for incompatível

faça uma restauração, obtendo r^k , e escolha $\Delta_{k+1} > 0$ tal que

$x^k + r^k \notin \bar{\mathcal{F}}_k$ e o subproblema quadrático seja compatível;

$$x^{k+1} = x^k + r^k;$$

obtenha W_{k+1} simétrica;

$$F_{k+1} = \bar{F}_k, \quad \mathcal{F}_{k+1} = \bar{\mathcal{F}}_k; \quad (\text{atualiza o filtro})$$

$k = k + 1$; (sai do loop interno)

SENÃO,

calcule o passo tangencial t^k ;

$$d^k = n^k + t^k;$$

Calcule a redução predita pelo modelo, $\text{pred} = m_k(x^k) - m_k(x^k - d^k)$;

Calcule a redução verdadeira em f_0 , $\text{ared} = f_0(x^k) - f_0(x^k + d^k)$;

SE $\{x^k + d^k \in \bar{\mathcal{F}}_k\}$ ou $\{\text{pred} \geq c_p (h(x^k))^2\}$ e $\text{ared} < \eta \text{pred}\}$

$$\Delta_k = \frac{\Delta_k}{2}; \quad (\text{este é o caso em que continuamos no loop interno})$$

SENÃO

$$x^{k+1} = x^k + d^k;$$

obtenha W_{k+1} simétrica;

$$\Delta_{k+1} = \Delta_k;$$

SE $\text{pred} \geq c_p (h(x^k))^2$, então

$$F_{k+1} = F_k, \quad \mathcal{F}_{k+1} = \mathcal{F}_k \quad ; \quad (\text{iteração } f_0)$$

SENÃO,

$$F_{k+1} = \bar{F}_k, \quad \mathcal{F}_{k+1} = \bar{\mathcal{F}}_k \quad ; \quad (\text{iteração } h)$$

$k = k + 1$; (sai do loop interno).

Note que, como esse é um algoritmo com região de confiança, temos também a atualização do raio. Para o raio diminuir basta que o ponto tentativo pertença ao filtro temporário ou que a redução na função objetivo não seja suficiente. Nesse caso o algoritmo volta para o teste de compatibilidade. Se o problema quadrático permanecer compatível com a nova região, o algoritmo procede com o cálculo do passo tangencial. Caso contrário, faz uma restauração.

3.3.2 Algoritmo de PQS com filtro inclinado

Como vimos na Seção 3.1.1, Fletcher, Leyffer e Toint [22] fizeram uma modificação na definição do filtro e propuseram um filtro que chamamos de “inclinado”.

Conforme vimos em (3.3), dado $x^k \in \mathbb{R}^n$ e $\alpha \in (0, 1)$, o par (\tilde{f}_0, \tilde{h}) para o filtro inclinado é definido por

$$\begin{cases} \tilde{f}_0 = f_0(x^k) - \alpha(1 - \alpha)h(x^k) = f_0(x^k) - \alpha\tilde{h} \\ \tilde{h} = (1 - \alpha)h(x^k). \end{cases} \quad (3.4)$$

De acordo com a Definição 3.1.3, um par $(f_0(x), h(x))$ é dominado pelo par (\tilde{f}_0, \tilde{h}) se, e somente se

$$f_0(x) + \alpha h(x) \geq \tilde{f}_0 + \alpha\tilde{h}$$

e

$$h(x) \geq \tilde{h}.$$

Com esse critério podemos definir o conjunto dos pontos x proibidos pelo filtro temporário:

$$\bar{\mathcal{F}}_k = \mathcal{F}_k \cup \{x \in \mathbb{R}^n \mid f_0(x) + \alpha h(x) \geq \tilde{f}_0 + \alpha\tilde{h}, h(x) \geq \tilde{h}\}.$$

Os conjuntos \bar{F}_k e $\bar{\mathcal{F}}_k$ são temporários e podem se tornar permanentes segundo o critério de atualização do filtro que veremos no algoritmo.

O cálculo do passo tentativo é feito por um método de PQS com região de confiança.

Observamos que, no algoritmo proposto por Fletcher, Leyffer e Toint [22], o passo é obtido por PQS, no entanto não é decomposto, explicitamente, em componentes normal e tangencial.

Apresentamos a seguir o algoritmo baseado em PQS e com margem inclinada.

Algoritmo 3.3.2 PQS com filtro inclinado

Dados: $x^0 \in \mathbb{R}^n$, W_0 simétrica, $\Delta_0 > 0$, $F_0 = \emptyset$, $\mathcal{F}_0 = \emptyset$, $\alpha \in (0, 1)$,
 $\xi \in (0, 1)$, $c_p \in (0, 1)$, $\eta \in (0, 1)$.

$k = 0$;

REPITA (enquanto x^k não for estacionário)

$$(\tilde{f}_0, \tilde{h}) = (f_0(x^k) - \alpha \tilde{h}, (1 - \alpha)h(x^k));$$

Defina os conjuntos $\bar{F}_k = F_k \cup \{(\tilde{f}_0, \tilde{h})\}$ e

$$\bar{\mathcal{F}}_k = \mathcal{F}_k \cup \{x \in \mathbb{R}^n \mid f_0(x) + \alpha h(x) \geq \tilde{f}_0 + \alpha \tilde{h}, h(x) \geq \tilde{h}\};$$

REPITA (enquanto o ponto x^{k+1} não for obtido)

SE o subproblema quadrático for incompatível

faça uma restauração, obtendo r^k , e escolha $\Delta_{k+1} > 0$ tal que

$x^k + r^k \notin \bar{\mathcal{F}}_k$ e o subproblema quadrático seja compatível;

$$x^{k+1} = x^k + r^k;$$

obtenha W_{k+1} simétrica;

$$F_{k+1} = \bar{F}_k, \quad \mathcal{F}_{k+1} = \bar{\mathcal{F}}_k; \quad (\text{atualiza o filtro})$$

$k = k + 1$; (sai do loop interno)

SENÃO,

resolva o subproblema quadrático obtendo d^k ;

Calcule a redução predita pelo modelo, $\text{pred} = m_k(x^k) - m_k(x^k - d^k)$;

Calcule a redução verdadeira em f_0 , $\text{ared} = f_0(x^k) - f_0(x^k + d^k)$;

SE $\{x^k + d^k \in \bar{\mathcal{F}}_k\}$ ou $\{\text{pred} \geq 0 \text{ e } \text{ared} < \eta \text{pred}\}$

$$\Delta_k = \frac{\Delta_k}{2}; \quad (\text{este é o caso em que continuamos no loop interno})$$

SENÃO

$$x^{k+1} = x^k + d^k;$$

obtenha W_{k+1} simétrica;

$$\Delta_{k+1} = \Delta_k;$$

SE $\text{pred} \geq 0$, então

$$F_{k+1} = F_k, \quad \mathcal{F}_{k+1} = \mathcal{F}_k \quad ; \quad (\text{iteração } f_0)$$

SENÃO,

$$F_{k+1} = \bar{F}_k, \quad \mathcal{F}_{k+1} = \bar{\mathcal{F}}_k \quad ; \quad (\text{iteração } h)$$

$k = k + 1$; (sai do loop interno).

3.3.3 Algoritmo de restauração inexata com filtro reto

O algoritmo proposto por Gonzaga, Karas e Vanti [28] é baseado no método de restauração inexata [41, 42] e utiliza o filtro reto como critério de aceitação do passo.

O conceito de dominância utilizado aqui é o dado em (3.2). O par (\tilde{f}_0, \tilde{h}) é o definido em (3.2)

$$\begin{cases} \tilde{f}_0 = f_0(x^k) - \alpha h(x^k) \\ \tilde{h} = (1 - \alpha)h(x^k). \end{cases} \quad (3.5)$$

Com isso temos o conjunto dos pontos proibidos:

$$\bar{\mathcal{F}}_k = \mathcal{F}_k \cup \{x \in \mathbb{R}^n \mid f_0(x) \geq \tilde{f}_0, h(x) \geq \tilde{h}\}.$$

O cálculo do passo é feito em duas fases: uma de viabilidade e uma de otimalidade. Na fase de viabilidade, procura-se reduzir a inviabilidade encontrando um ponto não proibido z^k . Na fase de otimalidade procura-se um ponto não proibido que reduza o valor da função objetivo em relação à z^k , mas sem perder muito da viabilidade conseguida na primeira fase. Para isso, o cálculo do passo x^{k+1} é feito na linearização das restrições no ponto z^k .

O filtro temporário passa a ser permanente, ou seja, o par (\tilde{f}_0, \tilde{h}) é incluído no filtro, caso a função objetivo não diminua de x^k para x^{k+1} . Chamamos essa iteração de uma iteração do tipo h .

O algoritmo pode parar em duas situações: quando um ponto estacionário é obtido ou quando o algoritmo da fase de viabilidade falha.

Apresentamos a seguir o algoritmo detalhado.

Algoritmo 3.3.3 *Restauração inexata com filtro reto*

Dados: $x^0 \in \mathbb{R}^n$, $F_0 = \emptyset$, $\mathcal{F}_0 = \emptyset$, $0 < \alpha < 1$.

$k = 0$;

REPITA

$$(\tilde{f}_0, \tilde{h}) = (f_0(x^k) - \alpha h(x^k), (1 - \alpha)h(x^k));$$

$$\bar{F}_k = F_k \cup \{(\tilde{f}_0, \tilde{h})\};$$

$$\bar{\mathcal{F}}_k = \mathcal{F}_k \cup \{x \in \mathbb{R}^n \mid f_0(x) \geq \tilde{f}_0, h(x) \geq \tilde{h}\};$$

FASE DE VIABILIDADE

$$SE \quad h(x^k) = 0$$

$$z^k = x^k;$$

SENÃO

Calcule $z^k \notin \bar{\mathcal{F}}_k$ tal que $h(z^k) < \tilde{h}$;

Se não for possível pare com insucesso;

FASE DE OTIMALIDADE

SE z^k é estacionário

Pare com sucesso;

SENÃO

Calcule $x^{k+1} \notin \bar{\mathcal{F}}_k$ tal que $x^{k+1} \in L(z^k)$ e $f_0(x^{k+1}) \leq f_0(z^k)$;

ATUALIZAÇÃO DO FILTRO

$$SE \quad f_0(x^{k+1}) < f_0(x^k) \quad (\text{iteração } f_0)$$

$$\mathcal{F}_{k+1} = \mathcal{F}_k; F_{k+1} = F_k;$$

SENÃO

(iteração h)

$$\mathcal{F}_{k+1} = \bar{\mathcal{F}}_k; F_{k+1} = \bar{F}_{k+1};$$

$$k = k + 1;$$

FIM

Em nosso trabalho vamos propor um algoritmo semelhante ao Algoritmo 3.3.3. A obtenção do passo é feita da mesma maneira, mas o conceito de dominância utilizado é o do filtro inclinado. Na seção seguinte descrevemos este algoritmo.

3.4 Proposta de algoritmo

Esta é a principal seção de nossa dissertação. Nela propomos um algoritmo de filtro globalmente convergente. Este algoritmo é baseado no Algoritmo 3.3.3 e usa como critério de aceitação do passo o filtro inclinado definido por Fletcher, Leyffer e Toint [22].

O algoritmo aqui proposto é uma versão modificada do algoritmo de Gonzaga, Karas e Vanti [28]. Em ambos os casos o passo é calculado por restauração inexata segundo Martínez e Pilotta [42]. Entretanto eles diferem na definição do

filtro. Em nossa proposta, utilizamos o filtro inclinado, enquanto que em [28], o filtro é reto.

3.4.1 Algoritmo

Vamos agora apresentar nosso algoritmo, discutindo em seguida sua convergência global.

Algoritmo 3.4.1 Algoritmo proposto

Dados: $x^0 \in \mathbb{R}^n, F_0 = \emptyset, \mathcal{F}_0 = \emptyset, 0 < \alpha < 1$.

$k = 0$;

REPITA

$$(\tilde{f}_0, \tilde{h}) = (f_0(x^k) - \alpha(1 - \alpha)h(x^k), (1 - \alpha)h(x^k));$$

$$\bar{F}_k = F_k \cup \{(\tilde{f}_0, \tilde{h})\};$$

$$\bar{\mathcal{F}}_k = \mathcal{F}_k \cup \{x \in \mathbb{R}^n \mid f_0(x) + \alpha h(x) \geq \tilde{f}_0 + \alpha \tilde{h}, h(x) \geq \tilde{h}\};$$

FASE DE VIABILIDADE

$$SE \quad h(x^k) = 0$$

$$z^k = x^k;$$

SENÃO

Calcule $z^k \notin \bar{\mathcal{F}}_k$ tal que $h(z^k) < \tilde{h}$;

Se não for possível pare com insucesso;

FASE DE OTIMALIDADE

SE z^k é estacionário

Pare com sucesso;

SENÃO

Calcule $x^{k+1} \notin \bar{\mathcal{F}}_k$ tal que $x^{k+1} \in L(z^k)$ e $f_0(x^{k+1}) \leq f_0(z^k)$;

ATUALIZAÇÃO DO FILTRO

$$SE \quad f_0(x^{k+1}) < f_0(x^k) \quad (\text{iteração } f_0)$$

$$\mathcal{F}_{k+1} = \mathcal{F}_k; \quad F_{k+1} = F_k;$$

SENÃO

(iteração h)

$$\mathcal{F}_{k+1} = \bar{\mathcal{F}}_k; \quad F_{k+1} = \bar{F}_{k+1};$$

$k = k + 1$;

FIM

Note que as únicas diferenças entre esse algoritmo e o Algoritmo 3.3.3 residem nas definições do par (\tilde{f}_0, \tilde{h}) e do conjunto $\bar{\mathcal{F}}_k$. Essas mudanças são uma consequência imediata do conceito de filtro utilizado.

Antes de estabelecermos a convergência global do Algoritmo 3.4.1, vamos fazer um comentário sobre as condições de otimalidade e o uso da expressão *ponto estacionário*.

Definimos a *direção de Cauchy projetada* ou *direção do gradiente projetado*, associada com cada $z \in \mathbb{R}^n$, como

$$d_C(z) = P_{L(z)}(z - \nabla f_0(z)) - z, \quad (3.6)$$

onde $P_{L(z)}(w)$ denota a projeção ortogonal de $w \in \mathbb{R}^n$ sobre o conjunto

$$L(z) = \{z + d \in \mathbb{R}^n \mid A_{\mathcal{E}}(z)d = 0, f_{\mathcal{I}}(z) + A_{\mathcal{I}}(z)d \leq \max\{0, f_{\mathcal{I}}(z)\}\},$$

definido em 2.15. A direção do gradiente projetado é bem conhecida (veja por exemplo Bertsekas [3]). Ela satisfaz $d_C(z) = 0$ se, e somente se, não existe direção de descida viável a partir de z em $L(z)$. Concluímos a partir dos fatos acima que, em um ponto viável z , as condições de KKT são equivalentes a $d_C(z) = 0$. Se $d_C(z) \neq 0$, então $\nabla f_0(z)^T d_C(z) < 0$.

De fato, esta direção é a principal construção usada por Martínez e Svaiter [45] para definir uma condição de otimalidade que está entre KKT e Fritz-John em generalidade: um ponto viável \bar{x} satisfaz a condição de otimalidade Martínez-Svaiter se, e somente se,

$$\liminf_{x \rightarrow \bar{x}} \|d_C(x)\| = 0. \quad (3.7)$$

Esta condição de otimalidade é na verdade razoavelmente construtiva. O que iremos provar neste trabalho é que nossos algoritmos produzem pontos viáveis satisfazendo (3.7). Esses pontos serão chamados de *estacionários*.

3.4.2 Convergência global

Para estabelecer a convergência vamos supor que o algoritmo gera sequências infinitas $(x^k), (z^k) \in \mathbb{R}^n$ e que as seguintes hipóteses são satisfeitas:

(H1) As funções $f_i(\cdot)$ para $i = 0, 1, \dots, m$ são continuamente diferenciáveis com derivadas primeiras lipschitzianas, ou seja, existe uma constante $L > 0$ tal que

$$\|\nabla f_i(y) - \nabla f_i(x)\| \leq L \|y - x\|,$$

para todo $i = 0, 1, \dots, m$; $x, y \in \mathbb{R}^n$.

(H2) As sequências $(x^k), (z^k)$ permanecem em um conjunto convexo e compacto $X \subset \mathbb{R}^n$.

(H3) Todo ponto de acumulação viável $\bar{x} \in X$ de (x^k) satisfaz a condição de qualificação de Mangasarian-Fromovitz, a saber, os gradientes $\nabla f_i(\bar{x})$, $i \in \mathcal{E}$, são linearmente independentes e existe uma direção $d \in \mathbb{R}^n$ tal que $A_{\mathcal{E}}(\bar{x})d = 0$ e $A_{\bar{\mathcal{I}}}(\bar{x})d < 0$, onde $\bar{\mathcal{I}} = \{i \in \mathcal{I} \mid f_i(\bar{x}) = 0\}$.

(H4) Dado um ponto viável não estacionário $\bar{x} \in X$, existe $M > 0$ e uma vizinhança \bar{V} de \bar{x} tal que, se $x^k \in \bar{V}$ então

$$f_0(x^k) - f_0(x^{k+1}) \geq M\sqrt{H_k} \quad (3.8)$$

onde

$$H_k = \min\{1, \min\{\tilde{h}^j \mid (\tilde{f}_0^j, \tilde{h}^j) \in F_k, \tilde{f}_0^j \leq f_0(x^k)\}\}. \quad (3.9)$$

A hipótese (H4) define uma variável muito importante, que chamaremos de *folga do filtro* em x^k , e será essencial para a prova de convergência do algoritmo. Essa definição é dada pela relação (3.9) e está ilustrada na Figura 3.7.

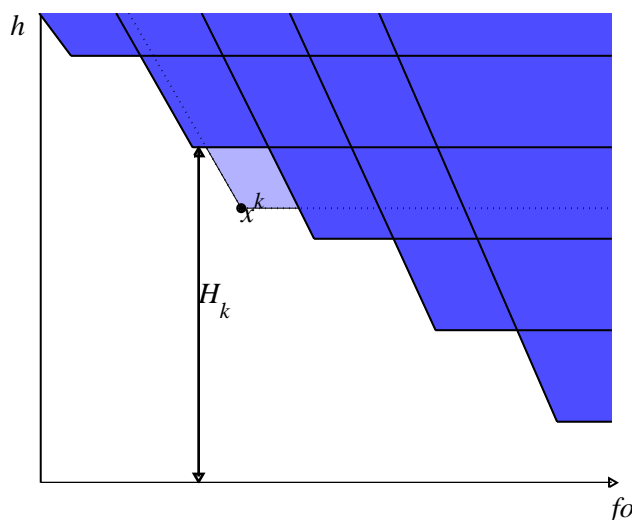


Figura 3.7: Folga do filtro.

As primeiras consequências das hipóteses (H1)-(H4) e da construção do algoritmo estão reunidas nos seguintes lemas.

Lema 3.4.2 *Suponha que sejam satisfeitas as Hipóteses (H1)-(H4). Dado $k \in \mathbb{N}$, são válidas as seguintes afirmações:*

(i) $x^{k+p} \notin \mathcal{F}_{k+1}$, para todo $p \geq 1$.

(ii) Pelo menos uma, das duas condições abaixo, é satisfeita:

1. $f_0(x^{k+1}) + \alpha h(x^{k+1}) < f_0(x^k)$,
2. $h(x^{k+1}) < (1 - \alpha)h(x^k)$.

(iii) $\tilde{h}^j > 0$ para todo $j \in \mathbb{N}$ tal que $(\tilde{f}_0^j, \tilde{h}^j) \in F_k$.

Demonstração. A primeira afirmação segue diretamente de $\mathcal{F}_{k+1} \subset \mathcal{F}_{k+p} \subset \bar{\mathcal{F}}_{k+p-1}$ e $x^{k+p} \notin \bar{\mathcal{F}}_{k+p-1}$. A segunda afirmação segue de $x^{k+1} \notin \bar{\mathcal{F}}_k$, assim temos, pela

definição de \bar{F}_k ,

$$f_0(x^{k+1}) + \alpha h(x^{k+1}) < \tilde{f}_0 + \alpha \tilde{h} = f_0(x^k) - \alpha \tilde{h} + \alpha \tilde{h} = f_0(x^k),$$

ou

$$h(x^{k+1}) < \tilde{h}(x) = (1 - \alpha)h(x^k).$$

Finalmente, note que se $\tilde{h} = 0$, por (ii) temos

$$f_0(x^{k+1}) \leq f_0(x^{k+1}) + \alpha h(x^{k+1}) < f^{(k)},$$

ou seja, a iteração k é do tipo f_0 , e portanto o par (\tilde{f}_0, \tilde{h}) não é incluído no filtro. Assim, se o par $(\tilde{f}_0, \tilde{h}) \in F_k$, $\tilde{h} > 0$, provando a terceira afirmação. \square

Lema 3.4.3 *Suponha que sejam satisfeitas as Hipóteses (H1)-(H4). Considere $\bar{x} \in X$ um ponto não estacionário. Então existe $\bar{k} \in \mathbb{N}$ e uma vizinhança \bar{V} de \bar{x} tal que se $k > \bar{k}$ e $x^k \in \bar{V}$, a iteração k é do tipo f_0 .*

Demonstração. Caso \bar{x} seja viável, a Hipótese (H4) garante que existe $M > 0$ e uma vizinhança \bar{V} de \bar{x} tal que se $x^k \in \bar{V}$, então

$$f_0(x^k) - f_0(x^{k+1}) \geq M\sqrt{H_k}. \quad (3.10)$$

Utilizando o Lema 3.4.2 (iii) e a definição de H_k em (3.9), concluímos que $H_k > 0$ e assim (3.10) implica que $f_0(x^{k+1}) < f_0(x^k)$, ou seja, temos uma iteração do tipo f_0 .

Consideremos agora o caso em que \bar{x} é inviável e suponhamos, por absurdo, que existe um conjunto infinito $\mathcal{K} \subset \mathbb{N}$ tal que $x^k \xrightarrow{\mathcal{K}} \bar{x}$ e todas as iterações em \mathcal{K} são do tipo h . Como f_0 e h são funções contínuas, temos

$$f_0(x^k) \xrightarrow{\mathcal{K}} f_0(\bar{x}) \quad \text{e} \quad h(x^k) \xrightarrow{\mathcal{K}} h(\bar{x}).$$

Consequentemente,

$$f_0(x^k) + \alpha h(x^k) - f_0(x^j) \xrightarrow{k,j \in \mathcal{K}} \alpha h(\bar{x}) > 0$$

e

$$h(x^k) - (1 - \alpha)h(x^j) \xrightarrow{k,j \in \mathcal{K}} \alpha h(\bar{x}) > 0.$$

Portanto existem $k, j \in \mathcal{K}$, $k > j$, tais que

$$f_0(x^k) + \alpha h(x^k) > f_0(x^j) \quad \text{e} \quad h(x^k) > (1 - \alpha)h(x^j),$$

resultando em $x^k \in \bar{\mathcal{F}}_j$. Além disso, como para $j \in \mathcal{K}$ a iteração é do tipo h , temos que $\mathcal{F}_{j+1} = \bar{\mathcal{F}}_j$. Assim $x^k \in \mathcal{F}_{j+1}$, contradizendo o Lema 3.4.2 (i). \square

Agora podemos apresentar o resultado principal desta seção, onde assumimos que a hipóteses (H1)-(H4) são satisfeitas, e provamos que a sequência gerada pelo algoritmo é globalmente convergente.

Teorema 3.4.4 *Suponha que sejam satisfeitas as Hipóteses (H1)-(H4). Então a sequência (x^k) tem um ponto de acumulação estacionário.*

Demonstração. Consideremos o conjunto

$$\mathcal{K}_1 = \{k \in \mathbb{N} \mid k \text{ é uma iteração do tipo } h\}.$$

Se \mathcal{K}_1 for infinito, pela Hipótese (H2) temos que $(x^k)_{k \in \mathcal{K}_1}$ é limitada e então existe uma subsequência convergente, ou seja, existe $\mathcal{K}_2 \subset \mathcal{K}_1$ e um $\bar{x} \in X$ tal que

$$x^k \xrightarrow{\mathcal{K}_2} \bar{x}.$$

Como as iterações são do tipo h temos, pelo Lema 3.4.3, que \bar{x} é estacionário.

Se \mathcal{K}_1 for finito então existe $k_0 \in \mathbb{N}$ tal que toda iteração $k \geq k_0$ é do tipo f_0 . Assim $(f_0(x^k))_{k \geq k_0}$ é decrescente e limitada, portanto convergente, e temos que

$$f_0(x^k) - f_0(x^{k+1}) \rightarrow 0. \quad (3.11)$$

Como a partir de $k \geq k_0$ as iterações são todas do tipo f_0 , temos que $F_k = F_{k_0}$, para todo $k \geq k_0$. Assim, pelo Lema 3.4.2 (iii) e pela definição de H_k em (3.9) temos que

$$H_k \geq H_{k_0} > 0. \quad (3.12)$$

Se $\mathcal{K}_3 = \{k \in \mathbb{N} \mid \alpha h(x^{k+1}) < f_0(x^k) - f_0(x^{k+1})\}$ é infinito então (3.11) garante que $h(x^{k+1}) \xrightarrow{\mathcal{K}_3} 0$. Seja $\tilde{\mathcal{K}}_3 = \{k+1 \mid k \in \mathcal{K}_3\}$ então

$$h(x^k) \xrightarrow{\tilde{\mathcal{K}}_3} 0.$$

Se \mathcal{K}_3 é finito então a partir de um certo índice $k \geq k_1, k \in \mathbb{N}$, o Lema 3.4.2 (ii-2) deve ser satisfeito, ou seja, $h(x^{k+1}) < (1 - \alpha)h(x^k)$ e temos

$$h(x^k) \rightarrow 0.$$

Assim, em qualquer caso existe um $\mathbb{N}' \subset \mathbb{N}$ tal que

$$h(x^k) \xrightarrow{\mathbb{N}'} 0. \quad (3.13)$$

Como $(x^k)_{k \in \mathbb{N}'}$ é limitada existe um ponto de acumulação \bar{x} , ou seja,

$$(x^k) \xrightarrow{\mathbb{N}''} \bar{x} \quad (\mathbb{N}'' \subset \mathbb{N}').$$

Pela continuidade de h temos $h(x^k) \xrightarrow{\mathbb{N}''} h(\bar{x})$. Mas por (3.13) temos

$$h(x^k) \xrightarrow{\mathbb{N}''} 0.$$

Pela unicidade do limite, temos

$$h(\bar{x}) = 0,$$

ou seja, \bar{x} é um ponto de acumulação viável de (x^k) .

Afirmamos que \bar{x} é estacionário. De fato, se \bar{x} não fosse estacionário, concluiríamos, usando a Hipótese (H4) que existem $M > 0$ e $k_2 \in \mathbb{N}$ tais que

$$f_0(x^k) - f_0(x^{k+1}) \geq M\sqrt{H_k},$$

para todo $k \in \mathbb{N}'$, $k \geq k_2$. Assim,

$$f_0(x^k) - f_0(x^{k+1}) \geq M\sqrt{H_k} \geq M\sqrt{H_{k_0}} > 0,$$

o que contradiz (3.11). □

Como vimos acima a hipótese (H4) é essencial na prova de convergência do algoritmo. Essa hipótese foi introduzida por Gonzaga, Karas e Vanti [28] e é a hipótese mais forte feita aqui. Essa hipótese refere-se a cada iteração completa e sua interpretação é simples: perto de um ponto não estacionário, o passo de otimalidade domina, e a redução de f_0 é grande.

Discussão sobre a hipótese (H4). Vamos agora mostrar como (H4) pode ser substituída por hipóteses mais simples, feitas para os passos de viabilidade e otimalidade separadamente.

Condição do passo de viabilidade

(H5) Existe uma constante $C_1 \geq 0$ tal que, para todo $k \in \mathbb{N}$

$$h(x^k) - h(z^k) \geq C_1 \|z^k - x^k\|. \quad (3.14)$$

Podemos reescrever essa expressão, pois $h(z^k) \geq 0$. Assim, existe $C_2 \geq 0$ tal que

$$\|z^k - x^k\| \leq C_2 h(x^k). \quad (3.15)$$

Como $\nabla f_0(\cdot)$ é limitado em X , pelo teorema do valor médio, existe uma constante $C_3 \geq 0$ tal que, para todo $k \in \mathbb{N}$,

$$|f_0(z^k) - f_0(x^k)| \leq C_3 \|z^k - x^k\|. \quad (3.16)$$

Usando isso e (3.15), temos que

$$|f_0(z^k) - f_0(x^k)| \leq C_4 h(x^k), \quad (3.17)$$

onde $C_4 = C_2 C_3$.

Condição do passo de otimalidade

(H6) Dado um ponto viável não estacionário $\bar{x} \in X$, existe uma vizinhança V de \bar{x} e uma constante $N \geq 0$ tal que, para qualquer iterada $x^k \in V$,

$$f_0(z^k) - f_0(x^{k+1}) \geq N \sqrt{H_k}. \quad (3.18)$$

A hipótese (H5) é uma condição global e significa que o passo de viabilidade deve ter uma redução suficiente na medida de inviabilidade h , ou seja, a direção $z^k - x^k$ deve ser uma boa direção de descida. Martínez [41] discute essa hipótese em seu trabalho e mostra que, sob condições razoáveis, ela é satisfeita.

A hipótese (H6) isola o passo tangencial e está associada a cada ponto viável não estacionário. Sua interpretação é a mesma que a da hipótese (H4), mas feita para o passo tangencial. No entanto, essa hipótese não é totalmente independente do passo de viabilidade, pois ela usa H_k , que está associada ao ponto x^k . Além disso, a condição é expressa para $x^k \in V$ e não $z^k \in V$, mas, por (3.15), se x^k está perto de \bar{x} então z^k também está.

Agora vamos demonstrar dois lemas. O primeiro será utilizado na demonstração do segundo, que prova que (H5) e (H6) implicam (H4).

Lema 3.4.5 *Considere \bar{x} um ponto viável. Sejam (a_k) e (b_k) sequências reais e as constantes $c_1 \geq 0$ e $c_2 \geq 0$, satisfazendo $|a_k| \leq c_1 h(x^k)$ e $b_k \geq c_2 \sqrt{H_k}$. Dado $c > 0$, existe uma vizinhança V_1 de (\bar{x}) tal que, se $x^k \in V_1$, então*

$$|a_k| \leq c b_k. \quad (3.19)$$

Demonstração. Seja $c > 0$ uma constante. Como $h(\bar{x}) = 0$, podemos considerar uma vizinhança V_1 de \bar{x} tal que, se $x^k \in V_1$, então

$$h(x^k) = \min \left\{ 1, \left(\frac{c_2}{c_1} \right)^2 \right\}.$$

Pela definição de H_k , temos $h(x^k) \leq H_k$. Usando a hipótese do lema, temos

$$|a_k| \leq c_1 h(x^k) = c_1 \sqrt{h(x^k)} \sqrt{h(x^k)} \leq c_1 \sqrt{h(x^k)} \sqrt{H_k}.$$

Conseqüentemente, usando a hipótese temos,

$$|a_k| \leq \frac{c_1}{c_2} \sqrt{h(x^k)} b_k \leq c b_k.$$

□

Lema 3.4.6 *Considere que (H1)-(H3) valem. Então as hipóteses (H5) e (H6) implicam (H4).*

Demonstração. Sejam \bar{x} um ponto viável não estacionário e V a vizinhança definida pela Hipótese (H6).

Considere V_1 a vizinhança dada pelo Lema (3.4.5) para $c = \frac{1}{2}$. Tome $\bar{V} = V_1 \cap V_2$. Por (3.17) e (3.18) podemos aplicar o Lema (3.4.5) e concluir que, para $x^k \in \bar{V}$, vale

$$-\frac{1}{2} (f_0(z^k) - f_0(x^{k+1})) \leq f_0(x^k) - f_0(z^k) \leq \frac{1}{2} (f_0(z^k) - f_0(x^{k+1})).$$

Assim, usando (H6) e definindo $M = \frac{1}{2}N$, obtemos

$$\begin{aligned} f_0(x^k) - f_0(x^{k+1}) &= f_0(x^k) - f_0(z^k) + f_0(z^k) - f_0(x^{k+1}) \\ &\geq \frac{1}{2} [f_0(z^k) - f_0(x^{k+1})] \\ &\geq \frac{1}{2} N \sqrt{H_k} \\ &= M \sqrt{H_k}, \end{aligned}$$

concluindo a demonstração. □

3.4.3 Discussão sobre os algoritmos internos

Discutiremos agora os passos internos usados em cada iteração do Algoritmo 3.4.1. Vamos supor que esse algoritmo gerou seqüências (x^k) e (z^k) e que as hipóteses (H1)-(H3) são satisfeitas.

Algoritmo do passo de viabilidade. O objetivo da fase de viabilidade é encontrar um ponto z^k que não seja proibido pelo filtro inclinado e que se aproxime do conjunto viável Ω . Mais precisamente, queremos um ponto $z^k \notin \bar{\mathcal{F}}_k$ e que satisfaça

$$h(z^k) < (1 - \alpha)h(x^k). \quad (3.20)$$

A princípio, qualquer algoritmo iterativo que diminua h pode ser usado nesta fase. Nós não iremos descrever nenhum algoritmo específico para esta fase, pois o passo de viabilidade estudado por Martínez [41] satisfaz a hipótese (H5) e pode ser aplicado ao nosso caso.

Algoritmo do passo de otimalidade.

Nesta fase o algoritmo deve encontrar um ponto $x^{k+1} \notin \bar{\mathcal{F}}_k$, que esteja no conjunto linearizado $L(z^k)$ e satisfaça $f_0(x^{k+1}) \leq f_0(x^k)$.

Lembrando a definição do conjunto $L(z^k)$ dada em (2.15), temos

$$L(z^k) = \{z^k + d \in \mathbb{R}^n \mid A_{\mathcal{E}}(z^k)d = 0, f_{\mathcal{I}}(z^k) + A_{\mathcal{I}}d \leq f_{\mathcal{I}}^+(z^k)\}. \quad (3.21)$$

O algoritmo que vamos descrever para o passo de otimalidade é o mesmo que o de [28], e é um método de região de confiança bastante geral. Faremos uma breve análise deste algoritmo e, em seguida, vamos mostrar que o passo resultante satisfaz a hipótese (H6).

Assim como no método de região de confiança descrito no Capítulo 1, este algoritmo também minimiza uma aproximação quadrática da função objetivo. A grande diferença é que, aqui, além da restrição da região de confiança temos que encontrar um ponto em um conjunto linearizado.

Lembrando que estamos na fase de otimalidade, já temos um ponto z^k obtido na fase de viabilidade. Assim, o modelo quadrático da função objetivo associado ao ponto z^k é

$$m_k(d) = f_0(z^k) + \nabla f_0(z^k)^t d + \frac{1}{2} d^t \mathcal{B}_k d, \quad (3.22)$$

onde $d = x - z^k$ e \mathcal{B}_k é uma matriz simétrica $n \times n$, que pode ser uma aproximação de $\nabla^2 f_0(z^k)$, ou qualquer outra matriz simétrica, desde que a hipótese (H7), dada abaixo, seja verificada.

(H7) Existe $\beta > 0$ tal que o modelo quadrático (3.22) satisfaz $\|\mathcal{B}_k\| \leq \beta$ para todo $k \in \mathbb{N}$.

Como vimos no Capítulo 1, um algoritmo de região de confiança usa um raio $\Delta > 0$ para limitar o tamanho do passo $d(z^k, \Delta) \in \mathbb{R}^n$. Com isso podemos aplicar as mesmas definições que as dadas em (1.5) e (1.4) para a redução predita pelo modelo no ponto z^k :

$$pred(x^k, \Delta) = m_k(z^k) - m_k(z^k + d(z^k, \Delta)), \quad (3.23)$$

e a redução verdadeira

$$\text{ared}(z^k, \Delta) = f_0(x^k) - f_0(z^k + d(z^k, \Delta)). \quad (3.24)$$

Antes de descrevermos o algoritmo do passo de otimalidade, vamos discutir mais alguns detalhes deste algoritmo.

Um passo $d(z^k, \Delta)$ somente é aceito se ele satisfaz uma condição de decréscimo suficiente, que é a mesma que a utilizada no algoritmo 1.2.1:

$$\text{ared}(z^k, \Delta) > \eta \text{pred}(z^k, \Delta),$$

para um dado $\eta \in (0, 1)$.

O passo de região de confiança pode ser uma solução aproximada do sub-problema:

$$\begin{cases} \text{minimizar} & m_k(d) \\ \text{sujeito a} & : z^k + d \in L(z^k) \\ & \|d\| \leq \Delta, \end{cases} \quad (3.25)$$

onde $\|\cdot\|$ é qualquer norma em \mathbb{R}^n .

Agora explicaremos o que significa “resolver aproximadamente”. Dado $z \in X$ e o conjunto $L(z)$, a direção do gradiente projetado é definida por

$$d_C(z) = P_{L(z)}(z - \nabla f_0(z)) - z. \quad (3.26)$$

Defina

$$\varphi(z) = -\nabla f_0(z)^T \frac{d_C(z)}{\|d_C(z)\|}.$$

Então φ é a taxa de decréscimo de f_0 ao longo de d_C . Como usual, denotamos $d_C^k = d_C(z^k)$, $\varphi^k = \varphi(z^k)$. Como discutido em [28], $\varphi(z) > 0$ sempre que z é um ponto viável não estacionário.

Agora usamos os resultados conhecidos sobre minimização de $m_k(\cdot)$ ao longo de uma direção (veja a discussão do ponto de Cauchy no Capítulo 1). Definimos o ponto de Cauchy generalizado como o minimizador de $m_k(\cdot)$ ao longo de d_C , na região de confiança $\{x \in \mathbb{R}^n \mid \|x - z^k\| \leq \Delta\}$,

$$x_C = \operatorname{argmin} \{m_k(x) \mid \|x - z^k\| \leq \Delta, x = z^k + \lambda d_C^k, \lambda \in [0, 1]\}.$$

Sabemos que

$$m_k(z^k) - m_k(x_C) \geq \frac{\xi \varphi^k}{2} \min \left\{ \frac{\varphi^k}{\|\mathcal{B}_k\|}, \|d_C^k\|, \Delta \right\},$$

onde ξ depende da norma usada. Usando a Hipótese (H7), isto pode ser reescrito como

$$m_k(z^k) - m_k(x_C) \geq \frac{\xi \varphi^k}{2} \min \left\{ \frac{\varphi^k}{\beta}, \|d_C^k\|, \Delta \right\}. \quad (3.27)$$

Aceitamos como uma solução aproximada de (3.25) qualquer solução viável para este problema satisfazendo a desigualdade de (3.27).

Na Figura 3.8, as curvas elípticas representam as curvas de nível da função objetivo. A área delimitada por uma poligonal é o conjunto linearizado $L(z^k)$. A área hachurada contida no conjunto linearizado é o conjunto dos pontos melhores que o ponto de Cauchy, ou seja, o conjunto dos pontos que consideramos como solução aproximada de (3.25).

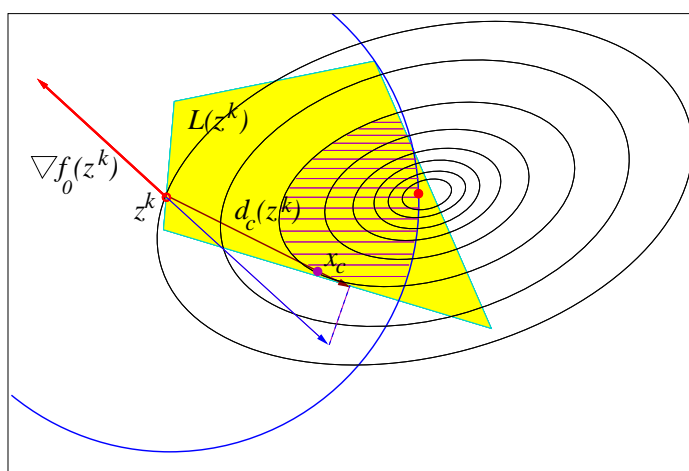


Figura 3.8: Ponto de Cauchy e melhores que ele.

Apresentamos a seguir a descrição do cálculo do passo de otimalidade.

Algoritmo 3.4.7 Dado: $\eta \in (0, 1)$, $\Delta_{\min} > 0$, $z^k \notin \bar{\mathcal{F}}_k$, $\Delta = \Delta^0 \geq \Delta_{\min}$.

REPITA

Calcule $d = d(z^k, \Delta)$ resolvendo aproximadamente o subproblema (3.25) ;

$ared(z^k, \Delta) = f_0(x^k) - f_0(z^k + d(z^k, \Delta))$;

$pred(z^k, \Delta) = m_k(z^k) - m_k(z^k + d(z^k, \Delta))$;

SE $z^k + d(z^k, \Delta) \notin \bar{\mathcal{F}}_k$ e $ared(z^k, \Delta) > \eta pred(z^k, \Delta)$

$x^{k+1} = z^k + d$;

$\Delta_k = \Delta$;

SENÃO

$\Delta = \frac{\Delta}{2}$;

FIM

Vamos agora mostrar que este algoritmo satisfaz a hipótese (H6), feita para o passo de otimalidade.

Note que o Algoritmo (3.4.7) difere daquele apresentado em [28] para o passo de otimalidade somente na definição do filtro. Assim, os dois próximos lemas, que independem do filtro, permanecem válidos. Apresentamos seus enunciados. Suas demonstrações são estabelecidas em [28].

Lema 3.4.8 *Para todo $z \in X$, $d \in \mathbb{R}^n$ tal que $(z + d) \in L(z)$, existe uma constante $C_5 > 0$ tal que*

$$|h(z + d) - h(z)| \leq C_5 \|d\|^2.$$

Demonstração. [28, Lema 3.3].

Agora vamos estudar o passo de otimalidade perto de um ponto $\bar{x} \in X$ viável não estacionário. O próximo lema diz que se ignorarmos o filtro, então o passo de região de confiança será grande perto de \bar{x} .

Lema 3.4.9 *Considere $\bar{x} \in X$ um ponto viável não estacionário satisfazendo uma condição M-F. Então existem uma vizinhança \tilde{V} de \bar{x} , $\tilde{\Delta} \in (0, \Delta_{min})$ e uma constante $\tilde{c} > 0$ tais que, para todo $z^k \in \tilde{V}$,*

- (i) *para todo $\Delta > 0$, $pred(z^k, \Delta) \geq \tilde{c} \min\{\Delta, \tilde{\Delta}\}$,*
- (ii) *para todo $\Delta \in (0, \tilde{\Delta})$, $ared(z^k, \Delta) > \eta \ pred(z^k, \Delta) \geq \eta \tilde{c} \Delta$.*

Demonstração. [28, Lema 3.4].

O próximo resultado depende da definição do filtro e será usado na demonstração do principal teorema dessa dissertação. Assim, os dois próximos resultados fazem parte da contribuição original deste trabalho.

Lema 3.4.10 *Seja \bar{x} um ponto viável não estacionário, $\tilde{\Delta} > 0$, \tilde{V} a vizinhança e \tilde{c} a constante dadas no Lema 3.4.9. Considere C_5 a constante do Lema 3.4.8, C_4 de (3.17) e $\bar{\Delta} = \min \left\{ \tilde{\Delta}, \frac{\eta \tilde{c}}{8\alpha C_5}, \frac{\alpha \eta \tilde{c}}{4C_2 C_5}, \frac{\eta \tilde{c}}{C_5(4 - 3\alpha)} \right\}$, $\Delta \in \left(\frac{\bar{\Delta}}{2}, \bar{\Delta} \right)$. Existe uma vizinhança $V_2 \subset \tilde{V}$ de \bar{x} tal que, se $x^k, z^k \in V_2$, então*

$$f_0(x^k) - f_0(z^k + d(z^k, \Delta)) \geq \alpha h(z^k + d(z^k, \Delta)). \quad (3.28)$$

Além disso, se $z^k + d(z^k, \Delta)$ foi recusado pelo filtro, então

$$h(z^k + d(z^k, \Delta)) \geq H_k. \quad (3.29)$$

Demonstração. Considere a vizinhança \tilde{V} dada no Lema 3.4.9. Se $x^k, z^k \in \tilde{V}$, então

como $\Delta < \bar{\Delta} < \tilde{\Delta}$, temos

$$\begin{aligned} \text{ared}(z^k, \Delta) &= f_0(z^k) - f_0(z^k + d(z^k, \Delta)) \\ &> \eta \text{pred}(z^k, \Delta) \\ &\geq \eta \tilde{c} \min\{\Delta, \tilde{\Delta}\} \\ &\geq \eta \tilde{c} \frac{\bar{\Delta}}{2}. \end{aligned}$$

Por (3.17) temos $|f_0(x^k) - f_0(z^k)| \leq C_4 h(x^k)$. Seja $V_1 \subset \tilde{V}$ uma vizinhança de \bar{x} tal que $C_4 h(x^k) \leq \frac{1}{4} \eta \tilde{c} \bar{\Delta}$, para todo $x^k \in V_1$. Assim, se $x^k \in V_1$,

$$\begin{aligned} \text{ared}(x^k, \Delta) &= f_0(x^k) - f_0(z^k + d(z^k, \Delta)) \\ &= f_0(x^k) - f_0(z^k) + f_0(z^k) - f_0(z^k + d(z^k, \Delta)) \\ &\geq \frac{1}{4} \eta \tilde{c} \bar{\Delta}. \end{aligned} \tag{3.30}$$

Por outro lado, pelo Lema 3.4.8, existe $C_5 > 0$ tal que

$$|h(z^k + d(z^k, \Delta)) - h(z^k)| \leq C_5 \|d(z^k, \Delta)\|^2 \leq C_5 \Delta^2.$$

Como $\Delta < \bar{\Delta}$ e $h(z^k) < (1 - \alpha)h(x^k)$, temos

$$\begin{aligned} h(z^k + d(z^k, \Delta)) &\leq h(z^k) + C_5 \bar{\Delta}^2 \\ &\leq (1 - \alpha)h(x^k) + C_5 \bar{\Delta}^2. \end{aligned}$$

Considerando $V_2 \subset V_1$ uma vizinhança de \bar{x} tal que $h(x^k) < \frac{\eta \tilde{c} \bar{\Delta}}{8\alpha(1 - \alpha)}$ para todo $x^k \in V_2$, e a definição de Δ e $\bar{\Delta}$, temos que

$$\begin{aligned} \alpha h(z^k + d(z^k, \Delta)) &\leq \alpha(1 - \alpha)h(x^k) + \alpha C_5 \bar{\Delta}^2 \\ &\leq \frac{\eta \tilde{c} \bar{\Delta}}{8} + \frac{\eta \tilde{c} \bar{\Delta}}{8} \\ &= \frac{\eta \tilde{c} \bar{\Delta}}{4}. \end{aligned}$$

Usando isso e (3.30), temos

$$f_0(x^k) - f_0(z^k + d(z^k, \Delta)) \geq \frac{\eta \tilde{c} \bar{\Delta}}{4} \geq \alpha h(z^k + d(z^k, \Delta)),$$

que demonstra (3.28). Para finalizar a demonstração, suponha que, $z^k + d(z^k, \Delta)$ foi recusado pelo filtro. Assim pela definição do filtro e de H_k , temos

$$h(z^k + d(z^k, \Delta)) \geq H_k,$$

completando a demonstração. □

Vamos apresentar agora o principal resultado desta seção, que garante que a hipótese (H6) é satisfeita pelo Algoritmo 3.4.7.

Teorema 3.4.11 *Considere $\bar{x} \in X$ um ponto viável não estacionário e que as hipóteses (H1)-(H3) e (H5) valem. Então existe uma vizinhança V de \bar{x} e uma constante $N > 0$ tal que, para $x^k \in V$*

$$f_0(z^k) - f_0(x^{k+1}) \geq N\sqrt{H_k}, \quad (3.31)$$

onde $x^{k+1} = z^k + d(z^k, \Delta)$ é calculado pelo Algoritmo 3.4.7.

Demonstração. Seja Δ^k o Δ obtido no cálculo do ponto x^{k+1} . Pelo Algoritmo 3.4.7 temos que

$$f_0(z^k) - f_0(x^{k+1}) = \text{ared}(z^k, \Delta^k) \geq \eta \text{pred}(z^k, \Delta^k). \quad (3.32)$$

Pelo Lema 3.4.8, existe uma constante $C_5 > 0$ tal que

$$h(z^k + d(z^k, \Delta)) - h(z^k) \leq C_5 \|d(z^k, \Delta)\|^2 \leq C_5 \Delta^2, \quad (3.33)$$

para todo $\Delta > 0$.

Considere V_1 uma vizinhança de \bar{x} tal que $h(x) < 1$, para todo $x \in V_1$.

Por (3.20) e pela definição de H_k , temos que, para $x^k \in V_1$,

$$h(z^k) < (1 - \alpha)h(x^k) < (1 - \alpha)H_k. \quad (3.34)$$

Considere a vizinhança V_2 e a constante $\bar{\Delta} > 0$ dadas no Lema 3.4.10, e as constantes $\tilde{\Delta}$ e \tilde{c} dadas no Lema 3.4.9. Considere $V = V_1 \cup V_2$ e $x^k \in V$.

Vamos considerar dois casos:

(i) $\Delta^k \geq \frac{\bar{\Delta}}{2}$

Pelo Lema 3.4.9(i) temos que

$$\text{pred}(z^k, \Delta^k) \geq \tilde{c} \min \left\{ \Delta^k, \tilde{\Delta} \right\} \geq \tilde{c} \frac{\bar{\Delta}}{2}.$$

Utilizando (3.32) e a definição de H_k temos

$$f_0(z^k) - f_0(x^{k+1}) \geq \eta \tilde{c} \frac{\bar{\Delta}}{2} \geq \eta \tilde{c} \frac{\bar{\Delta}}{2} \sqrt{H_k}.$$

Tomando $N = \eta \tilde{c} \frac{\bar{\Delta}}{2}$, temos o resultado.

(ii) $\Delta^k < \frac{\bar{\Delta}}{2}$

O algoritmo começa com $\Delta > \Delta_{\min}$ e vai reduzindo o raio à metade caso o passo $z^k + d(z^k, \Delta)$ seja recusado.

Pelo Lema 3.4.10 temos que, para todo $\Delta \in \left(\frac{\bar{\Delta}}{2}, \bar{\Delta}\right)$,

$$h(z^k + d(z^k, \Delta)) \geq H_k. \quad (3.35)$$

Vamos analisar duas situações.

(ii – 1) Suponha que (3.35) vale também para todo $\Delta \in \left[\Delta^k, \frac{\bar{\Delta}}{2}\right]$.

Usando (3.34), temos que

$$h(z^k + d(z^k, \Delta^k)) - h(z^k) \geq H_k - (1 - \alpha)H_k = \alpha H_k. \quad (3.36)$$

Por outro lado, usando (3.33)

$$h(z^k + d(z^k, \Delta^k)) - h(z^k) \leq C_5 \Delta^{k^2},$$

consequentemente usando (3.36)

$$\Delta^k \geq \sqrt{\frac{\alpha}{C_5}} \sqrt{H_k}. \quad (3.37)$$

Por (3.32), Lema 3.4.9 (i) e (3.37), temos que

$$f_0(z^k) - f_0(x^{k+1}) \geq \eta \tilde{c} \min\{\Delta^k, \tilde{\Delta}\} = \eta \tilde{c} \Delta^k \geq \eta \tilde{c} \sqrt{\frac{\alpha}{C_5}} \sqrt{H_k}.$$

Tomando $N = \eta \tilde{c} \sqrt{\frac{\alpha}{C_5}}$, obtemos o resultado.

(ii – 2) Suponha agora que para algum $\Delta < \frac{\bar{\Delta}}{2}$,

$$h(z^k + d(z^k, \Delta)) < H_k. \quad (3.38)$$

Defina $\hat{\Delta}$ como o primeiro $\Delta < \frac{\bar{\Delta}}{2}$ que satisfaz (3.38).

O raio $2\hat{\Delta}$ foi recusado pelo algoritmo, portanto, pela definição de $\hat{\Delta}$ e por (3.35), temos que

$$h(z^k + d(z^k, 2\hat{\Delta})) \geq H_k. \quad (3.39)$$

Usando isso, (3.33) e (3.34) temos que

$$\alpha H_k \leq h(z^k + d(z^k, 2\hat{\Delta})) - h(z^k) \leq 4C_5 \hat{\Delta}^2.$$

Portanto,

$$\hat{\Delta} \geq \sqrt{\frac{\alpha}{4C_5}} \sqrt{H_k}. \quad (3.40)$$

Pelo Lema 3.4.9 (ii),

$$f_0(z^k) - f_0(z^k + d(z^k, \hat{\Delta})) > \eta_{pred}(z^k, \hat{\Delta}) \geq \eta \tilde{c} \hat{\Delta} \geq N \sqrt{H_k}, \quad (3.41)$$

$$\text{com } N = \eta \tilde{c} \sqrt{\frac{\alpha}{4C_5}}.$$

Para finalizar a demonstração provaremos que $x^{k+1} = z^k + d(z^k, \hat{\Delta})$, ou seja, $\Delta^k = \hat{\Delta}$. Como $ared(z^k, \hat{\Delta}) > \eta_{pred}(z^k, \hat{\Delta})$ e $\hat{\Delta}$ satisfaz (3.38), basta provar que

$$f_0(z^k + d(z^k, \hat{\Delta})) + \alpha h(z^k + d(z^k, \hat{\Delta})) < f_0(x^k).$$

Usando (3.33) e (3.34) temos que

$$h(z^k + d(z^k, \hat{\Delta})) \leq (1 - \alpha)h(x^k) + C_5 \hat{\Delta}^2.$$

Usando (3.40) e o fato que $h(x^k) < H_k$ temos

$$\alpha h(z^k + d(z^k, \hat{\Delta})) \leq \alpha \left((1 - \alpha) \frac{4}{\alpha} + 1 \right) C_5 \hat{\Delta}^2.$$

Como $\hat{\Delta} < \frac{\bar{\Delta}}{2} < \frac{\eta \tilde{c}}{2C_5(4 - 3\alpha)}$, temos

$$\alpha h(z^k + d(z^k, \hat{\Delta})) \leq \frac{\eta \tilde{c}}{2} \hat{\Delta}. \quad (3.42)$$

Por outro lado, por (3.17) e (3.40) temos

$$|f_0(z^k) - f_0(x^k)| \leq C_4 h(x^k) \leq C_4 H_k \leq \frac{4C_4 C_5}{\alpha} \hat{\Delta}^2.$$

Como $\hat{\Delta} < \frac{\bar{\Delta}}{2} < \frac{\alpha \eta \tilde{c}}{8C_5 C_4}$, segue

$$|f_0(z^k) - f_0(x^k)| \leq \frac{\eta \tilde{c}}{2} \hat{\Delta}.$$

Usando isso e (3.41) temos

$$f_0(x^k) - f_0(z^k + d(z^k, \hat{\Delta})) = f_0(x^k) - f_0(z^k) + f_0(z^k) - f_0(z^k + d(z^k, \hat{\Delta})) \geq \frac{\eta \tilde{c}}{2} \hat{\Delta}.$$

Comparando com (3.42) temos

$$\alpha h(z^k + d(z^k, \hat{\Delta})) \leq \frac{\eta_{\tilde{c}}}{2} \hat{\Delta} \leq f_0(x^k) - f_0(z^k + d(z^k, \hat{\Delta})),$$

o que completa a demonstração. □

Conclusões

Estudamos nesse trabalho métodos de filtro para programação não linear. Esses métodos, relativamente recentes, são uma alternativa ao uso das funções de mérito. A idéia é usar o método de filtro para induzir a convergência, interferindo o mínimo possível nas iterações. Propomos um algoritmo de filtro baseado em restauração inexata, segundo Martínez [41] e Martínez e Pilotta [42], e que usa o filtro inclinado definido por Fletcher, Leyffer e Toint [22] como critério de aceitação do passo. Esse algoritmo é, na verdade, uma versão modificada do algoritmo de filtro proposto por Gonzaga, Kara e Vanti [28]. Essa modificação consiste na definição de filtro utilizada, em [28] é usado o filtro reto definido por Fletcher e Leyffer [20] e no algoritmo que propomos é usado o filtro inclinado definido por Fletcher, Leyffer e Toint [22].

Cada iteração é composta de uma fase de restauração, que procura reduzir a inviabilidade do problema e uma fase de otimalidade, onde a intenção é reduzir o valor da função objetivo, mas sem perder muito da viabilidade conseguida na primeira fase. Essas duas fases são totalmente independentes e o único acoplamento entre elas é dado pelo filtro. O método tem grande liberdade na escolha dos algoritmos internos, desde que esses algoritmos satisfaçam algumas hipóteses razoáveis sobre sua eficiência.

Fizemos a análise de convergência do algoritmo proposto, provando que a sequência gerada tem pelo menos um ponto de acumulação estacionário.

Perspectivas para outros trabalhos. O algoritmo de filtro proposto por Gonzaga, Karas e Vanti [28] tem uma propriedade mais forte sobre convergência, garantindo que todo ponto de acumulação da sequência gerada pelo algoritmo é estacionário. Para tanto é feita uma pequena modificação no critério de atualização do filtro, de modo que mais pontos possam ser adicionados no filtro, tornando o algoritmo mais restritivo.

Acreditamos que o algoritmo que propomos nesse trabalho também tenha essa propriedade, mas sem ser necessária qualquer modificação na atualização do filtro.

Referências Bibliográficas

- [1] J. Abadie and J. Carpentier. Generalization of the Wolfe reduced-gradient method to the case of nonlinear constraints. In R. Fletcher, editor, *Optimization*, pages 37–47. Academic Press, New York, 1968.
- [2] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear Programming Theory and Algorithms*. John Wiley, New York, 2nd edition, 1993.
- [3] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, Massachusetts, 1995.
- [4] P. T. Boggs and J. W. Tolle. Sequential quadratic programming. *Acta Numer.*, 4:1–51, 1996.
- [5] J. F. Bonnans, J. C. Gilbert, C. Lemaréchal, and C.A. Sagastizábal. *Numerical Optimization: Theoretical and Practical Aspects*. Springer Verlag, Berlin, 2002.
- [6] R. H. Byrd. Robust trust region methods for constrained optimization. Third SIAM Conference on Optimization, 1987.
- [7] R. H. Byrd, J. C. Gilbert, and J. Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Mathematical Programming*, 89(1):149–185, 2000.
- [8] R. H. Byrd, M. E. Hribar, and J. Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900, 1999.
- [9] R. H. Byrd, R. B. Schnabel, and G. A. Schultz. A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties. *SIAM Journal on Numerical Analysis*, 22:47–67, 1985.
- [10] R. H. Byrd, R. B. Schnabel, and G. A. Schultz. Aproximate solution of the trust regions problem by minimization over two-dimensional subspaces. *Mathematical Programming*, 40:247–263, 1988.

- [11] C. M. Chin. *A new trust region based on SLP-filter algorithm which uses EQP active set strategy*. PhD thesis, Department of Mathematics, University of Dundee, Scotland, 2001.
- [12] C. M. Chin and R. Fletcher. On the global convergence of an SLP-filter algorithm that takes EQP steps. *Mathematical Programming*, 96(1):161–177, 2003.
- [13] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust-Region Methods*. MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2000.
- [14] D. den Hertog. *Interior Point Approach to Linear, Quadratic and Convex Programming*. Mathematics and its Applications 277. Kluwer Academic Publishers, Dordrecht, 1992.
- [15] J. E. Dennis and R. B. Schnabel. *A view of unconstrained optimization*. Optimization, vol.1 of Handbooks in Operations Research and Management. Elsevier Science Publishers, Amsterdam, the Netherlands, 1989.
- [16] A. S. El-Bakry, R. A. Tapia, T. Tsuchya, and Y. Zhang. On the formulation and theory of the Newton interior-point method for nonlinear programming. *Journal of Optimization Theory and Applications*, 89:507–541, 1996.
- [17] A. V. Fiacco and G. P. McCormick. *NonLinear Programming: Sequential Unconstrained Minimization Techniques*. John Wiley & Sons, New York, 1968.
- [18] R. Fletcher. *Practical Methods of Optimization*. John Wiley, New York, 2nd edition, 1990.
- [19] R. Fletcher, N. Gould, S. Leyffer, P. Toint, and A. Wächter. Global convergence of trust-region and SQP-filter algorithms for general nonlinear programming. *SIAM J. Optimization*, 13(3):635–659, 2002.
- [20] R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. *Mathematical Programming - Ser. A*, 91(2):239–269, 2002.
- [21] R. Fletcher, S. Leyffer, and P. L. Toint. On the global convergence of an SLP-filter algorithm. Technical Report NA/183, Dundee University, Dept. of Mathematics, 1998.
- [22] R. Fletcher, S. Leyffer, and P. L. Toint. On the global convergence of a filter-SQP algorithm. *SIAM J. Optimization*, 13(1):44–59, 2002.
- [23] U. M . Garcia-Palomares and O. L. Mangasarian. Superlinearly convergent quasi-newton methods for nonlinearly constrained optimization problems. *Mathematical Programming*, 11:1–13, 1976.

- [24] D. M. Gay, M. L. Overton, and M. H. Wright. A primal-dual interior method for nonconvex nonlinear programming. In Yuan Y, editor, *Advances in Nonlinear Programming*, pages 31–56. Kluwer Academic Publishers, Dordrecht, 1998.
- [25] F. A. M. Gomes. A sequential quadratic programming algorithm that combines merit function and filter ideas. Technical report, University of Campinas, IMECC, Department of Applied Mathematics, Brazil, 2004.
- [26] F. A. M. Gomes, M. C. Maciel, and J. M. Martínez. Nonlinear programming algorithms using trust regions and augmented Lagrangians with nonmonotone penalty parameters. *Mathematical Programming*, 84(1):161–200, January 1999.
- [27] C. C. Gonzaga. Um curso de programação não linear. Notas de aula, 2004.
- [28] C. C. Gonzaga, E. W. Karas, and M. Vanti. A globally convergent filter method for nonlinear programming. *SIAM J. Optimization*, 14(3):646–669, 2003.
- [29] N. I. M. Gould, S. Leyffer, and P. L. Toint. A multidimensional filter algorithm for nonlinear equations and nonlinear least-squares. *SIAM Journal on Optimization*, 15(1):17–38, 2005.
- [30] N. I. M. Gould, C. Sainvitu, and P. L. Toint. A filter-trust-region method for unconstrained optimization. *SIAM Journal on Optimization*, 2005. To appear.
- [31] N. I. M. Gould and P. L. Toint. SQP methods for large-scale nonlinear programming. Technical report, CCLRC Rutherford Appleton Laboratory, England, August 1999.
- [32] S. P. Han. Superlinearly convergent variable metric algorithms for general nonlinear programming problems. *Mathematical Programming*, 11:263–282, 1976.
- [33] S. P. Han. A globally convergent method for nonlinear programming. *Journal of Optimization Theory and Applications*, 22:297–309, 1977.
- [34] B. Jansen. *Interior Point Techniques in Optimization*. Applied Optimization 6. Kluwer Academic Publishers, Dordrecht, 1997.
- [35] E. W. Karas, A. A. Ribeiro, C. Sagastizábal, and M. Solodov. A bundle-filter method for nonsmooth convex constrained optimization. *Mathematical Programming*, submitted, 2005.
- [36] N. Karmakar. A new polynomial-time algorithm for linear programming. *Combinatorics*, 4, pages 373–395, 1984.

- [37] M. Kojima, S. Mizuno, and A. Yoshise. A primal-dual interior point algorithm for linear programming. in n. megiddo, editor. *Progress in Mathematical Programming: Interior Point and Related Methods*, pages 29–47, Springer Verlag, New York, 1989.
- [38] K. Levenberg. A method for the solution of certain nonlinear problems in least squares. *Quart. Appl. Math.*, 2:164–168, 1944.
- [39] D. G. Luenberger. *Linear and Nonlinear Programming*. Addison - Wesley Publishing Company, New York, 1986.
- [40] D. W. Marquardt. An algorithm for least squares estimation of nonlinear parameters. *SIAM J. Appl. Math.*, 11:431–441, 1963.
- [41] J. M. Martínez. Inexact-restoration method with Lagrangian tangent decrease and a new merit function for nonlinear programming. *Journal of Optimization Theory and Applications*, 111:39–58, 2001.
- [42] J. M. Martínez and E. A. Pilotta. Inexact restoration algorithms for constrained optimization. *Journal of Optimization Theory and Applications*, 104:135–163, 2000.
- [43] J. M. Martínez and S. A. Santos. Métodos computacionais de otimização. 20.^o Colóquio Brasileiro de Matemática - IMPA, 1995.
- [44] J. M. Martínez and S. A. Santos. Métodos computacionais de otimização. 20.^o Colóquio Brasileiro de Matemática - IMPA, July 1995. In Portuguese.
- [45] J. M. Martínez and B. F. Svaiter. A practical optimality condition without constraint qualifications for nonlinear programming. Technical report, Institute of Mathematics, University of Campinas, Brazil, 2001.
- [46] A. Miele, V. Levy, and E. Cragg. Modifications and extensions of the conjugate-gradient restoration algorithm for mathematical programming problems. *Journal of Optimization Theory and Applications*, 7:450–472, 1971.
- [47] A. Miele, E. Sims, and V. Basapur. Sequential gradient-restoration algorithm for mathematical programming problems with inequality constraints, part 1, theory. Technical report, Report No. 168, Rice University, Aero-Astronautics, 1983.
- [48] A. Miele, H. Y. Yuang, and J. C. Heideman. Sequential gradient-restoration algorithm for the minimization of constrained functions, ordinary and conjugate gradient version. *Journal of Optimization Theory and Applications*, 4:213–246, 1969.

- [49] J. J. Moré. Recent developments in algorithms and software for trust region methods. In A. Bachem, M. Grötschel, and B. Korte, editors, *Mathematical Programming, The State of the Art*. Springer Verlag, 1983.
- [50] J. J. Moré and D. C. Sorensen. Computing a trust region step. *SIAM Journal on Sci. Statist. Comput.*, 4:553–572, 1983.
- [51] A. M. Mota. Convergência de algoritmos para programação não linear. Master’s thesis, Universidade Federal do Paraná, 2005.
- [52] H. Mukai and E. Polak. On the use of approximations in algorithm for optimization problems with equality and inequality constraints. *SIAM Journal on Numerical Analysis*, 15:674–693, 1978.
- [53] Y. E. Nesterov and A. S. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*, volume 13 of *SIAM Studies in Applied Mathematics 13*. SIAM, Philadelphia, 1994.
- [54] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer-Verlag, 1999.
- [55] E. Omojokun. *Trust Region Algorithms for Optimization with Nonlinear Equality and Inequality Constraints*. PhD thesis, Dept. of Computer Science, University of Colorado, 1991.
- [56] M. J. D. Powell. A hybrid method for nonlinear equations. In P. Rabinowitz, editor, *Numerical Methods for Nonlinear Algebraic Equations*. Gordon and Breach, London, 1970.
- [57] M. J. D. Powell. Convergence properties of a class of minimization algorithms. In O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, editors, *Nonlinear Programming 2*, pages 1–27. Academic Press, New York, 1975.
- [58] M. J. D. Powell. Algorithms for nonlinear constraints that use lagrangian functions. *Mathematical Programming*, 14:224–248, 1978.
- [59] M. J. D. Powell. The convergence of variable metric methods for nonlinearly constrained optimization calculations. *Nonlinear Programming 3*, pages 27–63, 1978.
- [60] M. J. D. Powell. A fast algorithm for nonnonlinearly constrained optimization calculations. *Numerical Analysis Dundee 1977*, pages 144–157, 1978.
- [61] A. A. Ribeiro. *Convergência global dos métodos de filtro para programação não linear*. PhD thesis, Universidade Federal do Paraná, 2005.

- [62] M. Rom and M. Avriel. Properties of the sequential gradient-restoration algorithm (sgra), part 2: Convergence analysis. *Journal of Optimization Theory Applications*, 62:99–126, 1989.
- [63] C. Roos, T. Terlaky, and J. P. Vial. *Theory and Algorithms for Linear Optimization: An Interior Point Approach*. John Wiley & Sons, Chichester, 1997.
- [64] J. B. Rosen. The gradient projection method for nonlinear programming, part 1, linear constraints. *SIAM Journal on Applied Mathematics*, 8:181–217, 1960.
- [65] J. B. Rosen. The gradient projection method for nonlinear programming, part 2, nonlinear constraints. *SIAM Journal on Applied Mathematics*, 9:514–532, 1961.
- [66] C. Sagastizábal and M. Solodov. An infeasible bundle method for nonsmooth convex constrained optimization without a penalty function or a filter. *SIAM J. Optimization*, submitted, 2004. Available at <http://www.impa.br/~optim/solodov.html>.
- [67] M. Ulbrich, S. Ulbrich, and L. N. Vicente. A globally convergent primal-dual interior-point filter method for nonlinear programming. *Mathematical Programming, Ser. A*, 100(2):379–410, 2004.
- [68] R. J. Vanderbei and D. F. Shanno. An interior-point algorithm for nonconvex nonlinear programming. *Computational Optimization and Applications*, 13:231–252, 1999.
- [69] R. B. Wilson. *A Simplicial Algorithm for Convex Programming*. PhD thesis, Harvard University, Graduate School of Business Administration, 1963.
- [70] S. J. Wright. *Primal-Dual Interior-Point Methods*. SIAM Publication, Philadelphia, 1997.