

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE AUTOMAÇÃO E SISTEMAS**

Caio Merlini Giuliani

**ESTRATÉGIAS DE OTIMIZAÇÃO NÃO
DIFERENCIÁVEL APLICADAS À MAXIMIZAÇÃO DA
PRODUÇÃO DE CAMPOS DE PETRÓLEO**

Florianópolis

2013

Caio Merlini Giuliani

**ESTRATÉGIAS DE OTIMIZAÇÃO NÃO
DIFERENCIÁVEL APLICADAS À MAXIMIZAÇÃO DA
PRODUÇÃO DE CAMPOS DE PETRÓLEO**

Dissertação submetida ao Programa
de Pós-Graduação em Engenharia de
Automação e Sistemas para a obten-
ção do Grau de Mestre em Engenharia
de Automação e Sistemas.

Orientador: Prof. Eduardo Campo-
nogara, Dr.

Coorientador: Prof. Agostinho Pluce-
nio, Dr.

Florianópolis

2013

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Giuliani, Caio Merlini

Estratégias de otimização não diferenciável aplicadas à maximização da produção de campos de petróleo / Caio Merlini Giuliani ; orientador, Eduardo Camponogara ; co-orientador, Agostinho Plucenio. - Florianópolis, SC, 2013.
98 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Engenharia de Automação e Sistemas.

Inclui referências

1. Engenharia de Automação e Sistemas. 2. Otimização sem derivada. 3. Otimização da produção de petróleo. 4. Região de confiança. 5. Busca direta. I. Camponogara, Eduardo. II. Plucenio, Agostinho. III. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia de Automação e Sistemas. IV. Título.

Caio Merlini Giuliani

**ESTRATÉGIAS DE OTIMIZAÇÃO NÃO
DIFERENCIÁVEL APLICADAS À MAXIMIZAÇÃO DA
PRODUÇÃO DE CAMPOS DE PETRÓLEO**

Esta Dissertação foi julgada aprovada para a obtenção do Título de “Mestre em Engenharia de Automação e Sistemas”, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia de Automação e Sistemas.

Florianópolis, 10 de Setembro 2013.

Prof. Jomi Fred Hübner, Dr.
Coordenador do Curso

Prof. Eduardo Camponogara, Dr.
Orientador

Prof. Agostinho Plucenio, Dr.
Coorientador

Banca Examinadora:

Prof. Eduardo Camponogara, Dr.
Presidente

Prof. Alexandre Trofino Neto, Dr.

Eng. Alex Furtado Teixeira, M. Sc.

Eng. Bruno da Costa Flach, Dr.

Prof. Daniel Juan Pagano, Dr.

AGRADECIMENTOS

Inicialmente, agradeço à minha amada Família, meu suporte em toda a vida. Pai, Mãe, Emília, sou profundamente grato por todos vocês.

Agradeço à Elisa, pela companhia, apoio e paciência.

Ao meu orientador, Eduardo Camponogara, e coorientador, Agustinho Plucenio, agradeço pela orientação, por todas as oportunidades e pela confiança.

Agradeço aos colegas e amigos que me acompanharam durante o curso.

Agradeço aos membros da banca, pela atenção, pelas críticas e sugestões.

Agradeço ao Programa de Pós-graduação em Engenharia de Automação e Sistemas, pela oportunidade.

Agradeço ao Centro de Pesquisas da Petrobras (CENPES), pelo apoio financeiro com a bolsa.

Saruman believes it is only great power that can hold evil in check, but that is not what I have found. I found it is the small everyday deeds of ordinary folk that keep the darkness at bay. Small acts of kindness and love.

Gandalf

RESUMO

Este trabalho apresenta métodos de otimização não-diferenciável aplicados à produção de petróleo. Na indústria do petróleo e gás, a produção de reservatórios, poços e sistemas relacionados pode ser predita com a utilização de simuladores numéricos. Este trabalho estuda técnicas de otimização que não fazem uso de derivadas da função objetivo, sendo adequadas para a utilização direta de ferramentas de simulação. São apresentadas a “busca direta direcional” e “região de confiança não-diferenciável”. A primeira não faz qualquer uso de modelos, enquanto a segunda utiliza modelos que aproximam a função objetivo em uma região limitada. Ambas são estudadas em suas formas irrestritas e com restrições lineares nas variáveis. Foi feita uma análise computacional em que ambos os métodos foram utilizados para a alocação de gás de injeção a um campo de produção de petróleo, com as produções dos poços modeladas por funções suaves, que garantiam suas condições de convergência. Os dois métodos convergiram para os pontos ótimos, sendo que o de região de confiança apresentou maior eficiência. Uma segunda análise computacional, contemplando apenas o método de região de confiança foi realizada empregando um simulador fenomenológico de poços de petróleo. Ambos os algoritmos podem servir de base para a otimização também com restrições não-lineares. Para tanto, propomos a utilização do método de Lagrangiano aumentado, que substitui as restrições não-lineares por penalizações na função objetivo, transformando o problema não-linear em uma sequência de problemas com restrições lineares. É possível implementá-lo sem necessidade de informações sobre as derivadas. Apresentamos a teoria de como isto pode ser feito, porém sem uma análise numérica.

Palavras-chave: Otimização sem derivada. Otimização da produção de petróleo. Região de confiança. Busca direta.

ABSTRACT

This work presented methods of nondifferentiable optimization applied to the production of petroleum. In the petroleum industry, the production of reservoirs, wells and related systems can be accurately predicted using numerical computer simulators. This work presented techniques of optimization that do not use the derivative of the objective function, hence better suited to use directly those simulation tools. The methods of *directional direct-search* and *nondifferentiable trust-region* are presented. The former does not make use of any model, whereas the latter samples the objective function to build models in a limited region. Both are studied in their unconstrained form and with linear constraints on the variables. A computational analysis has been carried out, in which both methods were employed in order to optimize the lift-gas allocation in a field of petroleum wells, modeled by smooth functions, so that their convergence conditions were satisfied. Both methods converged to the optimum, being the trust-region the more effective. A second analysis has been conducted, using only the trust-region method and a phenomenological simulator of petroleum wells. Both algorithms can be used also in optimization with nonlinear constraints. To that end, we propose the method of *augmented Lagrangian*, in which the nonlinear constraints are substituted by penalizations on the objective function, rendering the solution of nonlinear problem a sequence of subproblems with linear constraints. This method can also be used without knowledge of derivatives. Part of the underlying theory is presented.

Keywords: Derivative free optimization. Petroleum production optimization. Trust region. Direct-search.

SUMÁRIO

1	Introdução	17
1.1	Organização do documento	18
2	Otimização não-diferenciável irrestrita	21
2.1	Busca direta direcional	21
2.2	Região de confiança	30
2.2.1	Definição do modelo	31
2.2.2	Polinômios de Lagrange	34
2.2.3	Cálculo do passo	37
2.2.4	Aceitação do passo e gerenciamento da região	38
2.2.5	Teste de criticidade	39
2.2.6	Algoritmo de região de confiança	39
2.3	Alternativas algorítmicas	42
2.4	Sumário	42
3	Otimização não-diferenciável com restrições lineares	45
3.1	Material básico sobre otimização restrita	45
3.2	Busca direta direcional com restrições lineares	47
3.2.1	Alternativas algorítmicas	52
3.3	Região de confiança	54
3.3.1	Passo generalizado de Cauchy	54
3.3.2	Algoritmo de região de confiança com restrições lineares	55
3.3.3	Alternativas algorítmicas	57
3.4	Sumário	57
4	Otimização não-diferenciável com restrições não-lineares	59
4.1	Lagrangiano aumentado	59
4.1.1	Lagrangiano aumentado com resolução aproximada dos sub-problemas	60
4.1.2	Lagrangiano aumentado com restrições de desigualdade	62
4.1.3	Eliminação parcial das restrições	64
4.2	Um algoritmo de Lagrangiano aumentado	65
4.3	Resolvendo os sub-problemas com busca direta direcional	69
4.4	Resolvendo os sub-problemas com região de confiança não-diferenciável	69
4.5	Sumário	70
5	Análise computacional	73

5.1	Otimização de função suave	73
5.1.1	Busca direta direcional	74
5.1.2	Região de confiança	80
5.2	Otimização baseada no simulador	83
5.3	Sumário	88
6	Conclusão	91
	Referências	93
A	Sobre a diferenciabilidade do Lagrangiano aumentado	97

1 INTRODUÇÃO

Este trabalho busca aplicar técnicas de otimização não diferenciável ao caso da produção de petróleo. Neste capítulo introduziremos este caso de uso, as técnicas de otimização e nossa motivação em estudar tal aplicação.

Técnicas de otimização matemática podem ser bem empregadas nas mais diversas áreas. A indústria do óleo e gás tem algumas particularidades que motivam ainda mais tal aplicação. O petróleo é a principal fonte de energia do mundo moderno e suas reservas, embora grandes, são finitas e irão exaurir [1]. Além disso, há uma demanda crescente por óleo [2], e seu preço, nos últimos 15 anos alcançou valores muito maiores que os do restante do século passado.

A otimização pode ser aplicada tanto no sentido de sugerir meios de se extrair de forma lucrativa a maior quantidade de óleo possível, quanto na própria eficiência dos meios de produção.

Alguns trabalhos procuram encontrar a produção ótima de campos de produção de petróleo utilizando procedimentos heurísticos [3] ou programação inteira-mista linear ou não-linear [4–6]. Para tanto, estes constroem modelos complexos, a fim de caracterizar a produção do campo de uma forma que pode ser utilizada eficientemente por algoritmos de otimização. Tais modelos, embora não sejam fenomenológicos, requerem bom conhecimento do problema para ser construídos. Além disso, nem todos os tipos de modelo são igualmente bem sucedidos na resolução dos problemas de otimização [1, 7].

Esses modelos de otimização são frequentemente obtidos por meio de dados de simulação numérica. Simuladores são amplamente utilizados em processos da produção de petróleo. Modelos de reservatório e ferramentas de simulação de processos de extração de petróleo são cada vez mais rápidos e exatos [2]. Além disso, por serem parte importante na previsão e gerenciamento da produção, modelos de simulação são frequentemente sintonizados de forma a reproduzir de maneira confiável o comportamento de reservatórios, poços e sistemas relacionados.

Também é possível tentar otimizar a produção fazendo uso direto dos simuladores, sem a construção dos modelos de otimização mencionados anteriormente. Uma possível dificuldade é que os simuladores podem fornecer previsões de produção com uma boa exatidão, mas tipicamente não fornecem derivadas, que são necessárias para se aplicar técnicas clássicas de otimização. Alguns algoritmos de otimização buscam justamente calcular as derivadas, seja por diferenças finitas ou métodos probabilísticos, para utilizar métodos clássicos. De qualquer

forma, é importante ter em vista que, embora os simuladores forneçam resultados com exatidão satisfatória em suas variáveis finais, os valores calculados frequentemente são o resultado de um processo iterativo, que é interrompido quando o erro esperado fica suficientemente pequeno. Esta é uma fonte de ruído, que pode acabar ampliado no cálculo de derivadas.

Nesta dissertação abordamos outra classe de métodos de otimização, os não-diferenciáveis. Estes são desenvolvidos exatamente para o caso em que não se dispõe das derivadas da função objetivo, assim podem fazer uso dos valores calculados pelas ferramentas de simulação diretamente. Alguns desses métodos são aplicados em [8, 9], também no contexto de produção de petróleo. Sob certas condições, específicas de cada método, é possível garantir que se encontre um ponto ótimo.

Nesta dissertação, apresentaremos duas classes de algoritmos de otimização sem derivadas: *busca direta direcional* e *região de confiança não-diferenciável*. Os algoritmos serão aplicados a um cenário de produção de petróleo, com poços operando por *gas-lift*, processo descrito a seguir.

Em casos onde a pressão do reservatório é suficiente para fazer os fluidos irem da formação, através do poço até a superfície, o poço é dito surgente. Quando isso não ocorre, mecanismos de elevação artificial são usados. Um importante método de elevação é o *gas-lift* contínuo, correspondendo a 70% da produção de petróleo no Brasil [10].

Neste método, gás a alta pressão é injetado na coluna de produção [11]. Por ser mais leve que o petróleo, tende a subir, mas também acaba gaseificando o conteúdo do *tubing*, que também se torna mais leve [12]. Deste modo, a pressão do reservatório, juntamente com a injeção de gás, impulsiona os fluidos até a superfície.

1.1 ORGANIZAÇÃO DO DOCUMENTO

Este documento foi organizado da seguinte maneira: no capítulo 2 introduzimos os algoritmos de busca direta direcional e região de confiança não-diferenciável. Ambos para otimização irrestrita.

No capítulo 3, os dois algoritmos são estendidos a fim de resolver problemas de otimização com restrições lineares nas variáveis. Em ambos os casos, os métodos buscam por descenso na função objetivo apenas por pontos viáveis, isto é, satisfazendo às restrições. Para a busca direta direcional, isto é feito gerando-se direções de busca que não saem do conjunto viável. Para a região de confiança, os sub-problemas de região de confiança são resolvidos na intersecção entre a região de confiança e

o conjunto viável.

No capítulo 4 propomos o uso do método de Lagrangiano aumentado, para a resolução de problemas de otimização com restrições não-lineares nas variáveis. Tal método substitui as restrições não-lineares por penalizações na função objetivo, gerando uma sequência de sub-problemas contendo apenas restrições lineares. Como podem ser implementados sem a necessidade de derivadas da função objetivo, os algoritmos do capítulo anterior podem ser usados a fim de resolver tais sub-problemas.

No capítulo 5 fazemos uma análise computacional dos métodos propostos no capítulo 3. Os métodos são aplicados para alocação de gás de *gas-lift* em um campo de poços produtores de petróleo. Num primeiro caso, ambos os métodos resolvem o problema da alocação de gás considerando as funções de produção dos poços modeladas por funções suaves. Posteriormente, o método da região de confiança é utilizado para resolver um problema de alocação de gás utilizando diretamente um simulador fenomenológico.

Finalmente, no capítulo 6 apresentamos nossas conclusões.

2 OTIMIZAÇÃO NÃO-DIFERENCIÁVEL IRRESTRITA

Na área de produção de petróleo e gás há uma grande variedade de ferramentas para a simulação de reservatórios, poços, dutos e dos vários sistemas que integram a produção de petróleo e gás. No entanto, as derivadas dos valores preditos podem não estar disponíveis. Embora os simuladores sejam capazes de calcular numericamente suas previsões, a exatidão com que o fazem pode não ser suficiente para que se proceda com o cálculo de derivadas. Ou então, o custo de avaliar a função objetivo f em um número de pontos suficiente para calcular numericamente sua derivada pode ser muito alto.

Este trabalho busca investigar a utilização de métodos de otimização que não necessitam das derivadas das funções objetivos, podendo usar diretamente os valores calculados pelos simuladores. Para isto, introduziremos neste capítulo métodos de otimização não-diferenciável.

Quando temos uma função $f : \mathbb{R}^n \rightarrow \mathbb{R}$ e desejamos resolver um problema do tipo

$$\min_{x \in \mathbb{R}^n} f(x)$$

e as derivadas ∇f e $\nabla^2 f$ estão disponíveis, podem ser empregadas técnicas clássicas, como o método de Newton, para encontrar um ótimo local. Dizemos que $x^* \in \mathbb{R}^n$ é ótimo local de primeira ordem (ou ponto estacionário de primeira ordem), se $\nabla f(x^*) = 0$.

Quando não dispomos da derivada primeira ∇f , podem ser utilizados métodos de otimização não-diferenciável. São métodos que, a partir de uma solução aproximada, buscam melhorar o valor assumido pela função objetivo. Sob certas condições, é possível garantir que alguns desses métodos encontram um ponto ótimo de primeira ordem.

2.1 MÉTODOS DE BUSCA DIRETA DIRECIONAL

Uma primeira classe de métodos que será apresentada é a *busca direta direcional*. É composta de algoritmos que amostram a função objetivo um número finito de vezes a cada iteração e não constroem qualquer modelo dela. As decisões de evolução do algoritmo são sempre tomadas com base nos próprios valores amostrados. Alguns destes podem ser usados mesmo em funções que não são numéricas.

São algoritmos simples, podem não ser muito eficientes, mas diante de um problema novo, pode ser mais rápido programar um algoritmo de busca direta direcional para chegar à solução do que construir um algoritmo novo, de convergência mais rápida, porém mais complexo de se implementar.

Há várias formas de se descrever os algoritmos dessa classe, cada uma com suas particularidades de terminologia ou pequenas diferenças algorítmicas, mas sempre preservando uma mesma estrutura, sem a construção de modelos. Neste trabalho, procuramos expor tais algoritmos seguindo a descrição e notação de [13].

Para descrever o funcionamento desses algoritmos, utilizaremos um problema de alocação de gás a 2 poços. Consideraremos que a produção de cada poço é função do gás injetado, conforme a expressão [14]:

$$q_p(q_{inj}) = c_1 + c_2 q_{inj} + c_3 (q_{inj})^2 + c_4 \ln(q_{inj} + 1).$$

Consideramos que a vazão total produzida q_p é composta de vazões de óleo q_o , gás q_g e água q_w . A proporção entre gás e óleo produzidos é chamada GOR (Gas-Oil Ratio), enquanto o *Water Cut* é a proporção de água na fase líquida produzida (composta de óleo e água).

Considerando os vários poços (neste caso $N = 2$), representamos as seguintes grandezas:

- q_{inj}^n é a taxa de injeção de gás no poço n .
- q_p^n é a vazão total produzida no poço n .
- q_o^n é a vazão de óleo produzido no poço n .
- q_w^n é a vazão de água produzida no poço n .
- q_g^n é a vazão de gás produzido no poço n .

Considerando os preços obtidos com a venda do óleo p_o , e do gás p_g , e os custos do tratamento da água p_w e da injeção de gás p_i , definimos a função de ganho econômico correspondente como

$$f(q_{inj}) = \sum_{n=1}^{N=2} (p_o q_o^n(q_{inj}^n) + p_g q_g^n(q_{inj}^n) - p_w q_w^n(q_{inj}^n) - p_i q_{inj}^n).$$

Desejamos maximizar uma função de ganho econômico, para $N = 2$ poços, da forma

$$\max_{q_{inj}} f(q_{inj}). \quad (2.1)$$

Para este caso, consideramos o poço 1 modelado por

- $q_p^1(q_{inj}^1) = -1080 - 0,26q_{inj}^1 + 398 \ln(q_{inj}^1 + 1)$
- $GOR = 0,286$

- $Water\ cut = 12,5\%$
- $q_o^1 = 0,7q_p^1$
- $q_g^1 = 0,2q_p^1$
- $q_w^1 = 0,1q_p^1$.

e, para o poço 2,

- $q_p^2(q_{inj}^2) = -1630 - 0,37q_{inj}^2 + 671 \ln(q_{inj}^2 + 1)$
- $GOR = 0,750$
- $Water\ cut = 42,9\%$
- $q_o^2 = 0,4q_p^2$
- $q_g^2 = 0,3q_p^2$
- $q_w^2 = 0,3q_p^2$.

A busca direta direcional inicia de uma estimativa de solução e, a cada iteração, avalia a função objetivo um número finito de vezes em torno da solução corrente. Na Figura 2.1 ilustramos o domínio da função f , com as curvas de nível traçadas. A estrela marca o ponto ótimo, enquanto a estimativa de solução é o maior dos pontos escuros. A cada iteração, a função objetivo é avaliada em quatro pontos, localizados há uma certa distância da solução corrente. Neste caso, tais pontos foram calculados utilizando a estimativa de solução inicial $(q_{inj}^1, q_{inj}^2)_0$ e somando-se, $+\Delta$ e $-\Delta$ em cada variável, a fim de se obter os quatro pontos de teste, onde a função objetivo é avaliada. Na figura, estes pontos ficam nas pontas dos traços pretos. O ponto que apresenta o melhor valor passa a ser a nova estimativa de solução (neste caso, o ponto acima).

Este procedimento é repetido, então, em torno da nova solução, conforme ilustrado na Figura 2.2, que é a continuação da anterior.

Na parte (a) da figura, entre todos os pontos considerados, aquele à direita apresentou o melhor valor da função objetivo, passando a ser considerado a nova estimativa de solução, conforme ilustrado em (b).

Em (b) ocorre que nenhum dos pontos de teste apresenta um valor para a função objetivo melhor que a solução corrente. Nesse caso, o iterando é mantido no mesmo ponto, e a busca por descenso continua em uma região mais próxima, reduzindo-se o comprimento do passo Δ . Essa redução é ilustrada no quadro seguinte.

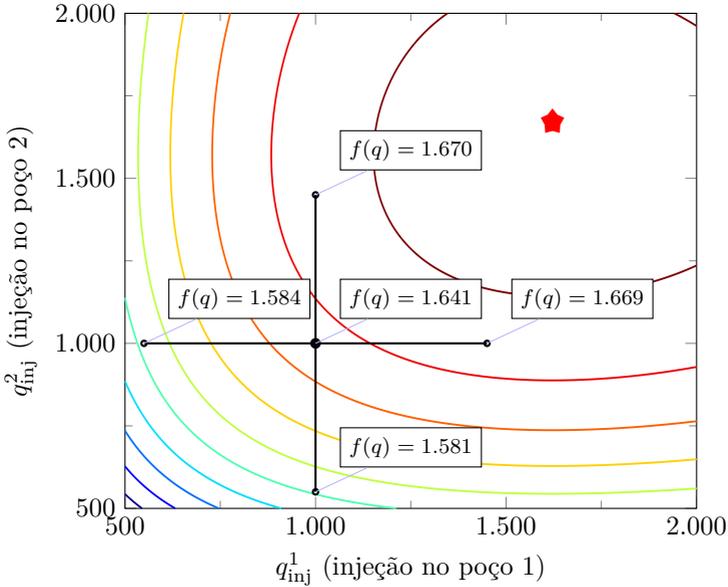


Figura 2.1: Busca direta direcional: a função objetivo é avaliada em quatro pontos em torno da estimativa de solução

Em (c), é encontrado um ponto melhor à direita. Em (d), o melhor ponto é acima. No quadro (e), novamente, nenhum dos pontos de teste apresenta um valor melhor que a solução corrente, e novamente o comprimento do passo é reduzido, enquanto mantém-se a mesma solução (f).

Neste capítulo procuraremos formalizar e generalizar esse método de solução e discutir sobre suas possibilidades de sucesso.

Para garantir que esses métodos cheguem a um ponto ótimo, os algoritmos cumprem as seguintes etapas:

1. Buscar uma direção de descenso.
2. Manter uma boa geometria das direções de busca (em um sentido formal).
3. Garantir que a região de busca diminua tanto quanto desejado. Isto é, exige-se que as regiões onde se busca por descenso decresçam, no limite.

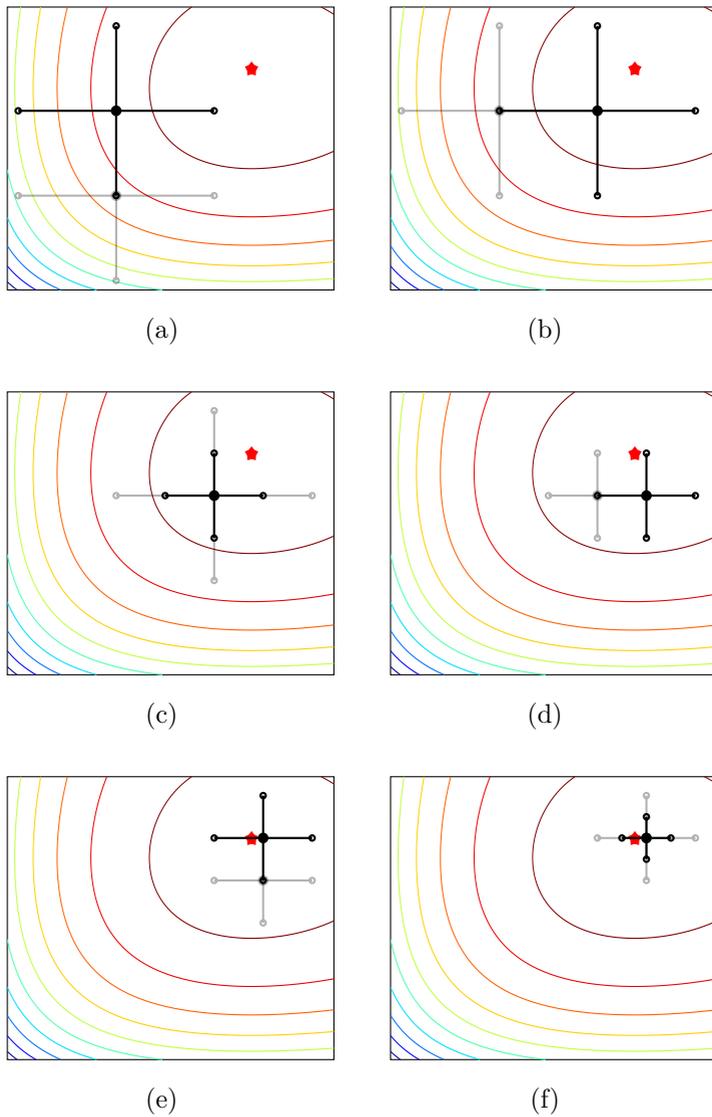


Figura 2.2: Busca direta direcional na otimização da produção em dois poços.

O primeiro ponto é encontrar uma direção de descenso, isto é, uma direção na qual a função objetivo f decresce, a partir da estimativa de solução corrente x . Métodos convencionais de otimização escolhem a direção baseados no gradiente $\nabla f(x)$. Por exemplo, o método do *máximo descenso* emprega $-\nabla f(x)$. No entanto, quando não se tem acesso à derivada, é necessário fazer outra escolha.

Suponha que a função f é continuamente diferenciável. Então, qualquer vetor $d \in \mathbb{R}^n$ que forma um ângulo agudo com $-\nabla f(x)$ é uma direção de descenso para f a partir de x . Isto é, se

$$-\nabla f(x)^T d > 0 \quad (2.2)$$

d é uma direção em que, a partir de x , f decresce ao menos por uma distância (possivelmente curta).

Métodos de busca direta direcional buscam em um conjunto de direções $\mathcal{D} = \{d_1, d_2, \dots\} \subset \mathbb{R}^n$, em que, para todo vetor $v \in \mathbb{R}^n$, $v \neq 0$, existe um $d_i \in \mathcal{D}$ de modo que

$$v^T d_i > 0. \quad (2.3)$$

Em particular, para $v = -\nabla f(x) \neq 0$, existe um d_i que satisfaz (2.2). Então, uma das direções de \mathcal{D} é direção de descenso, ainda que não se saiba, de antemão, qual.

Resta mostrar que é possível construir um conjunto \mathcal{D} conforme mencionado, com um número finito de elementos. Em [15], é demonstrado o seguinte teorema:

Teorema 1 *O conjunto $\mathcal{D} = \{d_1, d_2, \dots, d_r\}$ gera \mathbb{R}^n por meio de combinações lineares de coeficientes não-negativos se, e somente se, para todo vetor não-nulo $v \in \mathbb{R}^n$, existe um $d \in \mathcal{D}$ tal que*

$$v^T d > 0.$$

Pelo fato desses conjuntos gerarem \mathbb{R}^n por meio de combinações não-negativas (combinações cônicas), eles também podem ser chamados *geradores positivos* de \mathbb{R}^n . Por isso a classe de algoritmos apresentada nesta seção é chamada em [13] de “Generating Set Search” (busca [por meio de] conjuntos geradores [positivos]). Alguns exemplos de conjuntos geradores positivos estão na Figura 2.3.

Também de [15], o seguinte teorema é apresentado:

Teorema 2 *Suponha que $\mathcal{D} = \{d_1, d_2, \dots, d_r\}$, gera \mathbb{R}^n por meio de combinações cônicas, então o número de vetores $r \geq n + 1$, e é possível $r = n + 1$.*

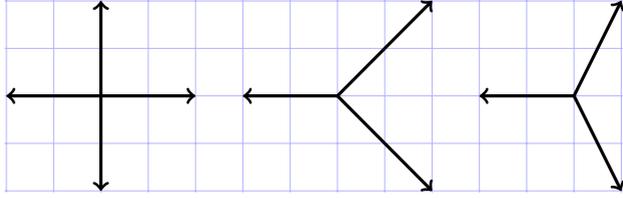


Figura 2.3: Três exemplos de conjuntos de vetores geradores positivos do \mathbb{R}^2 .

Para construir um tal conjunto, suponha que $\{d_1, d_2, \dots, d_n\}$ é uma base de \mathbb{R}^n . Então, para quaisquer $\lambda_i > 0$, definindo

$$d_{n+1} = -\lambda_1 d_1 - \lambda_2 d_2 - \dots - \lambda_n d_n \quad (2.4)$$

o conjunto $\{d_1, d_2, \dots, d_n, d_{n+1}\}$ é gerador positivo dos \mathbb{R}^n . Este conjunto também é chamado de base positiva mínima.

Outra possibilidade para construir \mathcal{D} a partir de uma base de \mathbb{R}^n para combinações lineares $\{d_1, \dots, d_n\}$ é unir tais vetores aos seus negativos:

$$\mathcal{D} = \{d_1, \dots, d_n\} \cup \{-d_1, \dots, -d_n\}. \quad (2.5)$$

Esta é chamada de base positiva máxima.

Em algoritmos de busca direta direcional, para cada iterando x_k com função objetivo $f(x_k)$, é construído um conjunto \mathcal{D}_k gerador positivo de \mathbb{R}^n , no qual se busca descenso para f .

Porém, não basta o conjunto \mathcal{D}_k ser gerador positivo de \mathbb{R}^n . É necessário garantir que todos os conjuntos usados tenham uma boa geometria, no sentido da seguinte definição:

Definição 1 Definimos a medida cosseno de um conjunto $\mathcal{D} \subseteq \mathbb{R}^n$ como

$$\kappa(\mathcal{D}) = \min_{\substack{v \in \mathbb{R}^n \\ v \neq 0}} \max_{d \in \mathcal{D}} \frac{v^T d}{\|v\| \|d\|}$$

Isto é, $\kappa(\mathcal{D})$ é o cosseno do maior ângulo interno formado entre um vetor não-nulo $v \in \mathbb{R}^n$ e o vetor $d \in \mathcal{D}$ mais próximo de v . Nos algoritmos de busca direta direcional, trata-se de uma medida de quão distante $-\nabla f(x)$ pode estar de \mathcal{D} .

Uma das formas de manter a boa geometria das direções de busca é impor uma medida cosseno mínima, um $\kappa_{\min} > 0$ de modo que:

$$\kappa(\mathcal{D}_k) \geq \kappa_{\min}.$$

Qualquer conjunto gerador positivo \mathcal{D} tem $\kappa(\mathcal{D}) > 0$, por consequência de (2.3). O conjunto formado pelas direções coordenadas e seus respectivos opostos

$$\{e_1, e_2, \dots, e_{n-1}, e_n\} \cup \{-e_1, -e_2, \dots, -e_{n-1}, -e_n\}$$

é um gerador positivo do \mathbb{R}^n , com medida cosseno $1/\sqrt{n}$.

Um exemplo de busca direta direcional é apresentado no Algoritmo 2.1, baseado em [13].

O algoritmo permite que se utilize, a cada iteração, um novo conjunto \mathcal{D}_k de direções de busca, o qual é formado por um conjunto \mathcal{G}_k , gerador positivo dos \mathbb{R}^n , e um outro conjunto \mathcal{H}_k . Este último contém direções de busca adicionais, permitindo-se a implementação de outros meios de busca por descenso durante as iterações. Porém, há limites nos comprimentos das direções:

$$\beta_{\min} \leq \|d\| \leq \beta_{\max} \quad \forall d \in \mathcal{G}_k, \quad (2.6)$$

$$\beta_{\min} \leq \|d\| \quad \forall d \in \mathcal{H}_k. \quad (2.7)$$

Os valores de $\beta_{\min} \leq \beta_{\max}$ são arbitrários, mas é importante que sejam fixos durante toda a execução do algoritmo, de modo que o comprimento final do passo seja determinado por Δ_k ($\Delta_k \beta_{\min} \leq \|\Delta_k d\| \leq \Delta_k \beta_{\max}$).

A iteração k é dita bem-sucedida ($k \in \mathcal{S}$) se existe uma direção $d \in \mathcal{D}_k$ tal que

$$f(x_k + \Delta_k d) < f(x_k) - \varrho(\Delta_k)$$

em que a *função forçante* $\varrho: \mathbb{R} \setminus \{0\} \rightarrow \mathbb{R} \setminus \{0\}$ é não-decrescente e

$$\lim_{t \rightarrow 0^+} \frac{\varrho(t)}{t} = 0. \quad (2.8)$$

Pode ser, por exemplo, qualquer função da forma $\varrho(t) = \gamma t^m$, em que $\gamma > 0$ e $m > 1$, constantes.

Caso não exista uma tal direção, a iteração k é dita mal-sucedida ($k \in \mathcal{U}$).

O tamanho do passo muda ao longo das iterações. Se não for encontrado descenso ele é reduzido, caso contrário, é aumentado, de acordo com as relações:

$$\Delta_{k+1} = \begin{cases} \phi_k \Delta_k, & \text{se a iteração } k \text{ foi bem-sucedida} \\ \theta_k \Delta_k, & \text{se a iteração } k \text{ foi mal-sucedida} \end{cases}$$

em que o parâmetro de contração do passo θ_k é tal que $0 < \theta_k \leq \theta_{\max} < 1$ e o de expansão do passo $\phi_k \geq 1$.

Tendo o algoritmo bem definido, podemos apresentar um primeiro resultado sobre sua convergência [13]:

Algoritmo 2.1: Um exemplo de busca direta direcional, de [13]

Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ dada;

Seja $x_0 \in \mathbb{R}^n$ um ponto inicial;

Defina

$\Delta_{tol} > 0$ a tolerância de convergência;

$\Delta_0 > \Delta_{tol}$ um tamanho de passo inicial;

$\phi_{\max} \geq 1$ um limite superior para o coeficiente de expansão

ϕ_k ;

$\phi_0 \in [1, \phi_{\max}]$ o valor inicial do coeficiente de expansão do passo;

$\theta_{\max} < 1$ um limite superior para o parâmetro de contração

θ_k ;

$\theta_0 \in (0, \theta_{\max}]$ o valor inicial do coeficiente de contração do passo;

$\beta_{\max} \geq \beta_{\min} > 0$ limites superior e inferior, respectivamente, para os comprimentos dos vetores de qualquer conjunto gerador positivo;

$\kappa_{\min} > 0$ um limitante superior para a medida cosseno dos conjuntos geradores positivos;

$\varrho : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ uma função não-decrescente satisfazendo a equação (2.8);

for $k = 0, 1, \dots$ **do**

Defina \mathcal{D}_k conjunto gerador positivo para \mathbb{R}^n satisfazendo

$\beta_{\min} \leq \|d\| \leq \beta_{\max}, \forall d \in \mathcal{D}_k$, com $\kappa(\mathcal{D}_k) \geq \kappa_{\min}$;

if *Existe* $d \in \mathcal{D}_k$ tal que $f(x_k + \Delta_k d) < f(x_k) - \varrho(\Delta_k)$ **then**

$x_{k+1} \leftarrow x_k + \Delta_k d$;

$\Delta_{k+1} \leftarrow \phi_k \Delta_k$;

else

$x_{k+1} \leftarrow x_k$;

$\Delta_{k+1} \leftarrow \theta_k \Delta_k$;

if $\Delta_{k+1} < \Delta_{tol}$ **then**

 Encerrar;

Determine $\theta_{k+1} \in (0, \theta_{\max}]$ e $\phi_{k+1} \in [1, \phi_{\max}]$;

Teorema 3 *Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$, com ∇f Lipschitziana com constante de Lipschitz M . O Algoritmo 2.1 produz iterandos tais que, para toda iteração mal-sucedida $k \in \mathcal{U}$*

$$\|\nabla f(x_k)\| \leq \frac{1}{\kappa(\mathcal{D}_k)} \left[M\Delta_k\beta_{\max} + \frac{\varrho(\Delta_k)}{\Delta_k\beta_{\min}} \right]. \quad (2.9)$$

Assim, obtemos uma cota superior (ainda que desconhecida) para $\|\nabla f(x_k)\|$, mesmo sem calcularmos tal valor em nenhum momento. Ainda, como κ_{\min} limita os valores de $\kappa(\mathcal{D}_k)$, o termo $1/\kappa(\mathcal{D}_k)$ é limitado superiormente ao longo das iterações. Além disso, como os valores de M , β_{\max} e β_{\min} são fixos, e considerando (2.8), se o tamanho do passo for reduzido indefinidamente, isto é, se

$$\lim_{\substack{k \rightarrow 0 \\ k \in \mathcal{U}}} \Delta_k = 0$$

então

$$\lim_{\substack{k \rightarrow 0 \\ k \in \mathcal{U}}} \|\nabla f(x_k)\| = 0.$$

Já foram abordados a maneira de encontrar uma direção de descenso (por meio de conjuntos geradores positivos) e o controle da geometria das direções de busca (por meio da medida cosseno). O ponto faltante é garantir que a direção de busca decresça. Fazendo-se $\lim_{k \rightarrow \infty} \Delta_k = 0$, o Teorema 3 garante convergência para um ponto estacionário de primeira ordem. Isto é, um ponto x^* em que

$$\nabla f(x^*) = 0.$$

Isso é enunciado no teorema seguinte:

Teorema 4 *Seja f limitada inferiormente. Suponha a função ϱ como dada em (2.8). Então, o Algoritmo 2.1 produz iterações de modo que*

$$\liminf_{k \rightarrow +\infty} \Delta_k = 0.$$

A demonstração pode ser vista em [13].

2.2 MÉTODOS DE REGIÃO DE CONFIANÇA

Métodos de região de confiança partem de uma aproximação para a solução e utilizam os valores já calculados da função objetivo para construir modelos. Estes buscam aproximar f localmente, em uma vizinhança do ponto atual chamada *região de confiança*.

O modelo construído, espera-se, é mais fácil de otimizar que a função original. Por um lado, porque se dispõe de técnicas clássicas de otimização com derivadas. Por outro, porque a avaliação do modelo pode ser muito mais rápida que a função f original.

Construído um bom modelo, ele é otimizado dentro da região de confiança, gerando um ponto x_k^+ , para então se calcular $f(x_k^+)$. Assim, é possível economizar no número de vezes que se avalia f . O que é uma vantagem com relação aos métodos de busca direta quando, por exemplo, cada cálculo de f é uma simulação que pode durar vários minutos. Por outro lado, métodos de região de confiança são mais complexos de se implementar.

Entre os vários métodos de região de confiança, há diferenças com relação ao tipo de modelo, forma da região de confiança, maneiras de manter ou melhorar o modelo, aceitação do passo e gerenciamento da região de confiança.

Nesta seção, apresentamos métodos de região de confiança não-diferenciáveis, em grande parte, conforme a descrição e nomenclatura de [16]. Neste livro há várias explicações abrangentes para o uso de modelos polinomiais de regressão, mínima norma ou modelos não polinomiais. Porém, nesta dissertação, vamos abordar apenas modelos de interpolação por polinômios de até segunda ordem, da forma

$$m_k(x_k + s) = m_k(x_k) + g_k^T s + \frac{1}{2} s^T H_k s$$

que aproximam a função objetivo f na região de confiança dada por uma bola de tamanho Δ_k em torno do iterando x_k :

$$\mathcal{B}(x_k, \Delta_k) = \{x : \|x - x_k\| \leq \Delta_k\}$$

onde a norma $\|\cdot\|$ pode se tratar da norma 2 ou norma infinito, por exemplo.

Para introduzir os métodos de região de confiança não-diferenciáveis, iniciamos com alguns conceitos básicos.

2.2.1 Definição do modelo

Entre os vários algoritmos de otimização que usam região de confiança, há uma grande diferença com relação ao tipo de modelo usado. Nesta seção, introduzimos o modelo que usamos no trabalho, de interpolação com polinômios de segunda ordem.

Para modelos de interpolação, definimos um conjunto de pontos $\mathcal{Y} = \{y^0, y^1, \dots, y^L\}$ onde o modelo construído coincide com a função

objetivo, isto é,

$$m(y^k) = f(y^k) \quad \forall y^k \in \mathcal{Y}. \quad (2.10)$$

É necessário também fixar uma base $\phi = \{\phi_1, \dots, \phi_L\}$ que represente o espaço dos polinômios de segunda ordem no \mathbb{R}^n . Então, as combinações lineares dos seus elementos ϕ_1, \dots, ϕ_L geram os polinômios, e o modelo m pode ser escrito como:

$$m(x) = \sum_{l=1}^L \alpha_l \phi_l(x). \quad (2.11)$$

em que $\phi_l \in \phi$.

Então, fixado um conjunto de pontos \mathcal{Y} , para se determinar o modelo polinomial de interpolação, a partir desta última equação, juntamente com as $L + 1$ equações (2.10), tem-se o sistema de equações lineares:

$$M(\phi, \mathcal{Y})\alpha_\phi = f(\mathcal{Y}) \quad (2.12)$$

em que

$$M(\phi, \mathcal{Y}) = \begin{bmatrix} \phi_0(y^0) & \phi_1(y^0) & \cdots & \phi_L(y^0) \\ \phi_0(y^1) & \phi_1(y^1) & \cdots & \phi_L(y^1) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(y^L) & \phi_1(y^L) & \cdots & \phi_L(y^L) \end{bmatrix}$$

e

$$f(\mathcal{Y}) = \begin{bmatrix} f(y^0) \\ f(y^1) \\ \vdots \\ f(y^L) \end{bmatrix}.$$

Para o modelo ser calculado, basta que a matriz $M(\phi, \mathcal{Y})$ seja invertível. Isto depende apenas do conjunto de pontos \mathcal{Y} , e não da base. Quando isso ocorre, dizemos que \mathcal{Y} , ou respectivamente o modelo m nele baseado é *bem posicionado* para interpolação. A seguir apresentaremos um algoritmo para o cálculo deste conjunto, baseado na eliminação Gaussiana.

Por outro lado, o bom condicionamento do sistema (2.12) depende de \mathcal{Y} e também da base. Este é um ponto que interfere tanto na exatidão com que se resolve o sistema, como na estimativa dos erros $\|m(x) - f(x)\|$ entre o modelo e a função objetivo.

Neste trabalho, fixamos o uso da base de monômios, que chamamos *base natural* e denotamos $\bar{\phi}$:

$$\begin{aligned}\phi &= \{\phi_0, \dots, \phi_L\} \\ &= \left\{ 1, x_1, x_2, \dots, x_n, \right. \\ &\quad \left. \frac{x_1^2}{2}, x_1x_2, \frac{x_2^2}{2}, x_1x_3, x_2x_3, \frac{x_3^2}{2}, \dots, \frac{x_n^2}{2} \right\}.\end{aligned}$$

Para proceder com o cálculo de um conjunto \mathcal{Y} de pontos bem posicionados para interpolação, apresentamos o Algoritmo 2.2, de [16]. O algoritmo procede de forma semelhante à eliminação gaussiana, aplicada à matriz $M(\phi, \mathcal{Y})$. Inicia com um conjunto de pontos \mathcal{Y} , possivelmente incompleto ou mal-posicionado, e com um conjunto de polinômios, chamados *polinômios pivô*. O laço mais externo consiste em encontrar, entre os pontos disponíveis, aquele em que o polinômio pivô corrente assume maior valor absoluto. Na eliminação gaussiana, equivale a encontrar para cada coluna de M , o maior valor de elemento pivô. A diferença é que, quando um pivô é zero, ou quando em \mathcal{Y} não há outros pontos, um novo ponto é calculado, substituindo o primeiro em \mathcal{Y} .

O segundo passo do algoritmo consiste em realizar a eliminação propriamente dita, operando com os polinômios. Para cada um dos polinômios seguintes, se subtrai o polinômio pivô corrente, multiplicado pelo fator adequado. O resultado equivalente, na matriz dos polinômios pivô $M(u, \mathcal{Y})$, é que os elementos abaixo do pivô são zerados.

Ao final deste procedimento, tem-se um conjunto de polinômios pivô, e uma respectiva matriz $M(u, \mathcal{Y})$ não-singular, o que garante que o conjunto de pontos \mathcal{Y} é bem posicionado (note que, no primeiro passo, garantimos que todos os elementos-pivô da eliminação gaussiana fossem não-nulos). Para a construção do modelo, pode-se resolver o sistema triangular $M(u, \mathcal{Y}) = f(\mathcal{Y})$ com retro-substituição, gerando os coeficientes que multiplicam os polinômios-pivô para construir o modelo.

Porém, não basta que seja possível calcular um modelo de interpolação. É necessário que o sistema resultante $M(\phi, \mathcal{Y})$ seja bem-condicionado. A maneira de limitar o número de condicionamento de M é feita de maneira semelhante ao algoritmo anterior. No entanto, o critério para descartar (recalcular) um ponto durante a eliminação gaussiana em M é mais forte. Exige-se que todos os elementos-pivô tenham módulo maior que um valor ξ , fixado. O procedimento é ilustrado no Algoritmo 2.3, de [16].

Com este algoritmo, todos os pivôs da matriz M são, garantidamente, maiores que o limiar ξ escolhido. Se necessário, o respectivo ponto é trocado para que isso ocorra. Para tanto, a otimização que ocorre no

primeiro passo não é estritamente necessária. Basta que se encontre um ponto $y \in \mathcal{B}$ que satisfaz $|u_i(y)| \geq \xi$. A maneira de encontrá-lo será mostrada adiante.

Segundo [16], as estimativas de erros podem ser relacionadas com o número de condicionamento da matriz M , aplicada a um conjunto de pontos modificado para uma região de confiança dada por uma bola de raio 1 centrada na origem:

$$\begin{aligned}\hat{\mathcal{Y}} &= \{0, \hat{y}^1, \dots, \hat{y}^L\} \\ &= \{0, (\hat{y}^1 - \hat{y}^0)/\Delta(\mathcal{Y}), \dots, (\hat{y}^L - \hat{y}^0)/\Delta(\mathcal{Y})\} \subset \mathcal{B}(0, 1)\end{aligned}$$

em que $\Delta(\mathcal{Y})$ é o fator de escala que faz com que todos os elementos caibam (exatamente) em uma bola de raio 1:

$$\Delta(\mathcal{Y}) = \max_{1 \leq i \leq L} \|y^i - y^0\|.$$

de modo que o uso do conjunto $\hat{\mathcal{Y}}$ limita a norma

$$1 \leq \|M\| \leq (L + 1)^{\frac{3}{2}}.$$

O que motiva o uso do algoritmo neste conjunto $\hat{\mathcal{Y}}$. Por outro lado, a decomposição LU feita pelo algoritmo, provê uma cota superior para $\|M^{-1}\|$. Considere a decomposição dada, com ambas as matrizes L e U com números 1 na diagonal: $M = LDU$ ($M^{-1} = U^{-1}D^{-1}L^{-1}$). O limiar para os pivôs dá uma cota superior para a norma $\|D^{-1}\| \leq \sqrt{L+1}/\xi$, de modo que

$$\|M^{-1}\| = \|D^{-1}\| \|L^{-1}\| \|U^{-1}\| \leq \frac{\sqrt{L+1} \|L^{-1}\| \|U^{-1}\|}{\xi}.$$

O valor $\varepsilon_{\text{growth}} = \|L^{-1}\| \|U^{-1}\|$, chamado fator de crescimento da faturação pode ser muito grande, mas é limitado. A escolha do maior valor possível para os pivôs pode ajudar a diminuir esse termo.

Assim, o número de condicionamento de $\text{cond}(M(\phi, \hat{\mathcal{Y}}))$ é limitado.

2.2.2 Polinômios de Lagrange

Dado um conjunto de $L + 1$ pontos $\mathcal{Y} = \{y^0, y^1, \dots, y^L\}$, definimos os polinômios de Lagrange correspondentes como polinômios $l_j : \mathbb{R}^n \rightarrow \mathbb{R}$, em que $j = 1, \dots, n$, e

$$l_j(y^i) = \begin{cases} 1 & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases} \quad (2.13)$$

Algoritmo 2.2: Completando o conjunto de pontos de interpolação

Escolha uma aproximação inicial para os polinômios-pivô u_i , possivelmente usando a base: $u_i(x) = \phi_i(x)$. Considere o conjunto inicial $\mathcal{Y} = \{y^0, y^1, \dots, y^{p_{ini}}\}$, com $p_{ini} + 1$ pontos;

for $i = 0, 1, \dots$ **do**

Encontre $j_i = \operatorname{argmax}_{i \leq j \leq p_{ini}} |u_i(y^j)|$. Se $|u_i(y^{j_i})| > 0$ e $i \leq p_{ini}$, troque os pontos y^i e y^{j_i} no conjunto \mathcal{Y} . Senão, calcule (ou recalcule, caso $i \leq p_{ini}$) como

$$y^i \in \operatorname{argmax}_{x \in \mathcal{B}} |u_i(x)|$$

for $j = i + 1, \dots, L$ **do**

$$u_j(x) \leftarrow u_j(x) - \frac{u_j(y^i)}{u_i(y^i)} u_i(x)$$

Algoritmo 2.3: Melhorando o modelo e o conjunto de pontos

Escolha uma aproximação inicial para os polinômios-pivô u_i , possivelmente usando a base: $u_i(x) = \phi_i(x)$ para $i = 0, \dots, L$. Escolha um limiar $\xi > 0$ para a seleção de elementos-pivô. Considere o conjunto inicial \mathcal{Y} bem-posicionado com $L + 1$ pontos;

for $i = 0, 1, \dots, L$ **do**

Encontre se possível, $j_i \in \{i, \dots, L\}$ tal que $|u_i(y^{j_i})| \geq \xi$. Se tal j_i for encontrado, troque os pontos y^i e y^{j_i} de lugar em \mathcal{Y} . Caso contrário, recalcule y^i como

$$y^i \in \operatorname{argmax}_{x \in \mathcal{B}} |u_i(x)|.$$

interrompa caso $|u_i(y^i)| < \xi$ (O limiar ξ é muito grande).

for $j = i + 1, \dots, p$ **do**

$$u_j(x) \leftarrow u_j(x) - \frac{u_j(y^i)}{u_i(y^i)} u_i(x)$$

Isto é, para todo índice j , l_j é um polinômio que se anula em todos os pontos de \mathcal{Y} , exceto por y^j , com $l_j(y^j) = 1$. Além disso, tal polinômio é único, e independente de base, contanto que a matriz M de interpolação seja não-singular.

Assim, o modelo de interpolação para a função objetivo f é:

$$m(x) = \sum_{i=0}^L l_i(x) f(y^i) \quad (2.14)$$

Para que os $L + 1$ polinômios de Lagrange estejam bem definidos, é necessário que os pontos sejam bem posicionados; em caso contrário, não é possível determinar $L + 1$ polinômios satisfazendo (2.13).

Para chegar ao erro de interpolação, introduzimos a seguinte definição [16]:

Definição 2 *Seja $\Lambda > 0$ e um conjunto $\mathcal{B} \subset \mathbb{R}^n$ dado. Um conjunto bem posicionado $\mathcal{Y} = \{y^0, y^1, \dots, y^L\}$ é dito Λ -posicionado em \mathcal{B} se, e somente se, para a base de polinômios de Lagrange associados com \mathcal{Y} , tem-se que*

$$\Lambda \geq \max_{n=0, \dots, L} \max_{x \in \mathcal{B}} |l_n(x)|$$

Esta medida também estabelece cotas superiores para os erros de predição tanto do modelo $\|f - m\|$ como de sua derivada $\|\nabla f - \nabla m\|$ [17]:

Teorema 5 *Dada uma bola $\mathcal{B}(x, \Delta)$ e um conjunto de pontos de interpolação $\mathcal{Y} \subset \mathcal{B}(x, \Delta)$ bem posicionado, e seus polinômios de Lagrange correspondentes $\{l_n(y)\}_{n=0}^L$, existem constantes $\kappa_{ef} > 0$ e $\kappa_{eg} > 0$ tais que, para todo polinômio da forma (2.14) de grau maior que um e todo ponto $y \in \mathcal{B}(x, \Delta)$,*

$$\|f(y) - m(y)\| \leq \kappa_{ef} \sum_{n=0}^p \|y^n - y\|^2 |l_n(y)| \quad (2.15)$$

e

$$\|\nabla f(y) - \nabla m(y)\| \leq \kappa_{eg} \Lambda \Delta. \quad (2.16)$$

Dessa forma, mantendo-se Λ limitado, o erro de interpolação cai com o raio da região Δ . Estes resultados motivam a buscar modelos de interpolação polinomial construídos sobre pontos Λ -posicionados, para um Λ mantido constante.

Alguns algoritmos constroem modelos Λ -posicionados fazendo uso explícito do valor de Λ . Não é o caso do Algoritmo 2.3, que constrói modelos Λ -posicionados com um valor de Λ desconhecido, porém

limitado e que depende do parâmetro ξ . Optamos por este algoritmo por ser mais fácil de adaptar ao caso restrito.

2.2.3 Cálculo do passo

Para o cálculo do passo, uma possibilidade é tentar encontrar o ótimo global na região de confiança, resolvendo o problema

$$\min_{x \in \mathcal{B}(x_k, \Delta_k)} m_k(x)$$

Isto pode ser feito em casos particulares. Se a região de confiança for dada por uma bola na norma- ∞ do tipo $\mathcal{B}(x_k, \Delta_k) = \{x : \|x - x_k\|_\infty \leq \Delta_k\}$, trata-se de um problema de otimização com restrições lineares. Se o modelo m_k for linear, trata-se de um problema de programação linear:

$$\begin{aligned} \min m_k(x) &= g_k^T x \\ x - x_k &\leq \Delta_k \\ x - x_k &\geq -\Delta_k. \end{aligned}$$

O mesmo poderia ser dito de bolas na norma-1.

De modo semelhante, se o modelo for quadrático, e a bola for dada na norma-2 o problema de minimização

$$\begin{aligned} \min m_k(x + s) &= m_k(x_k) + g_k^T s + \frac{1}{2} s^T H_k s \\ \text{sujeito a } \|s\| &< \Delta_k \end{aligned}$$

pode ser resolvido eficientemente (em tempo polinomial), utilizando o algoritmo de Moré-Sorensen, o que torna o uso da norma-2 vantajoso. Tal algoritmo não será apresentado neste trabalho porque estamos interessados no caso de restrições lineares, que não podem ser tratadas, embora haja algumas heurísticas [18].

A otimização de um modelo quadrático em uma bola dada por restrições lineares torna-se um problema NP-difícil se a derivada segunda H tiver ao menos um auto-valor negativo [19]. E segundo [20], nesse caso é difícil até mesmo decidir se um determinado ponto é mínimo local.

De qualquer forma, não é necessário chegar ao ótimo global na região. Alguns algoritmos exigem que a minimização encontre ao menos uma fração do decréscimo obtido pela otimização global. Uma alternativa ainda menos exigente é por meio do passo de Cauchy. Defina:

$$t_k^C = \underset{\substack{t \geq 0 \\ x_k - t g_k \in \mathcal{B}(x_k, \Delta_k)}}{\operatorname{argmin}} m_k(x_k - t g_k)$$

o passo de Cauchy é dado por

$$s_k^C = -t_k^C g_k.$$

Então, o novo ponto é dado por $x_k^+ = x_k + s_k^C$. Os algoritmos que apresentaremos exigem que se encontre, a cada iteração, uma fração do passo de Cauchy.

De acordo com [20], verifica-se experimentalmente uma convergência linear do algoritmo quando, repetidamente, se encontra o passo de Cauchy a cada iteração. Em tal livro a otimização do modelo na região de confiança é tratada com grande riqueza de detalhes, seja para se encontrar um bom ponto de teste ou para encontrar o ótimo global.

2.2.4 Aceitação do passo e gerenciamento da região

Da otimização aproximada do problema na região de confiança, o valor do decréscimo encontrado com o modelo $m_k(x_k) - m_k(x_k^+)$ será sempre positivo. Algumas decisões do algoritmo são dadas com base no grau de concordância entre o decréscimo real obtido e o esperado:

$$\rho_k = \frac{f(x_k) - f(x_k^+)}{m_k(x_k) - m_k(x_k^+)}. \quad (2.17)$$

Se este valor for alto ($\rho_k \geq \eta_1$, $\eta_1 > 0$ fixo), o passo é aceito, o iterando é atualizado ($x_{k+1} \leftarrow x_k^+$), e a região de confiança é deslocada de acordo, ficando em torno de x_{k+1} .

Ainda, [16] sugere que se aceite o passo em casos mais modestos para o descenso encontrado. Então, o passo também é aceito se $\rho_k \geq \eta_0$, com η_0 fixado, contanto que o modelo que produziu tal passo seja suficientemente preciso (Λ -posicionado). Além disso, $\eta_1 \geq \eta_0 \geq 0$, e o uso de $\eta_0 = 0$ é encorajado por tais autores, possibilitando que se aceite descenso simples, na esperança de se aproveitar todas as avaliações da função objetivo, que podem ser custosas.

O outro uso para o coeficiente de concordância ρ_k é para o gerenciamento da região de confiança. Caso seja alto, $\rho_k \geq \eta_1$, o raio da região pode ser aumentado, por exemplo, $\Delta_{k+1} \leftarrow \min\{\gamma_{\text{inc}}\Delta_k, \Delta_{\text{max}}\}$, $\gamma_{\text{inc}} > 1$, sendo Δ_{max} o valor máximo para o raio da região.

Caso contrário, a falha em prever adequadamente o decréscimo da função pode se dever a dois motivos: ou o modelo não é bom, ou o modelo é bom, mas numa vizinhança menor do iterando, enquanto o raio da região de confiança está muito grande. Assim, caso o modelo já seja Λ -posicionado, o erro deve-se a um raio muito grande, então este é reduzido de um fator γ : $\Delta_{k+1} \leftarrow \gamma\Delta_k$.

Por outro lado, caso o modelo ainda não seja Λ -posicionado, não podemos concluir que o raio foi grande demais. Neste caso, o modelo deve ser melhorado (por exemplo, com o Algoritmo 2.3). O raio é mantido ($\Delta_{k+1} \leftarrow \Delta_k$), já que uma redução indevida afetaria, no mínimo, a taxa de convergência do algoritmo (e as provas de convergência também dependem deste ponto).

2.2.5 Teste de criticidade

Quando se detectar que os iterandos estão se aproximando de um ponto estacionário, o raio da região de confiança é reduzido, e novos modelos Λ -posicionados são calculados, de modo a se ter menores erros entre o modelo e a função objetivo.

Por outro lado, essa redução do raio não pode ser excessiva, de modo a não prejudicar o avanço do algoritmo. Para garantir a convergência, é necessário fazer o raio da região próximo de alguma medida de estacionariedade, como a norma da derivada do modelo $\|g_k\|$. Isso também motiva ter o raio da região de confiança como o critério de parada usual dos algoritmos de região de confiança: o raio da região converge para zero, ou fica limitado acima de zero junto com a medida ($\|g_k\|$), que está relacionada a $\|\nabla f(x_k)\|$.

Se, ao início de uma iteração, $\|g_k\|$ for pequeno o suficiente ($\|g_k\| < \epsilon_c$, para um valor fixo ϵ_c), executa-se o teste de criticidade, de modo a obter um modelo Λ -posicionado em uma região com um raio de, no máximo $\mu \|g_k\|$, para um valor fixo μ .

Isto é feito iterativamente, já que o valor de g_k depende do modelo m_k , que por sua vez depende da região de confiança, cujo raio será (no final) limitado por $\mu \|g_k\|$. Para tanto, o raio da região é repetidamente reduzido e novos modelos Λ -posicionados são criados, até que se encontre uma região com o raio $\tilde{\Delta}_k$ que não excede $\mu \|g_k\|$.

Também, para que esta redução do raio não seja excessiva, este valor $\tilde{\Delta}_k$ do raio é usado, contanto que seja maior que $\beta \|g_k\|$. Caso contrário, utiliza-se o valor $\min(\beta \|g_k\|, \Delta_k)$. O procedimento encontra-se formalizado no Algoritmo 2.4.

2.2.6 Algoritmo de região de confiança

O Algoritmo 2.5 é um exemplo de método de região de confiança segundo [16].

Agora apresentamos as condições suficientes para que o Algoritmo 2.5 convirja para um ponto ótimo de primeira ordem.

Algoritmo 2.4: Teste de criticidade

```

i = 0;
mk0 = mk;
for i = 0, 1, ... do
  | Incremente i em 1;
  |  $\tilde{\Delta}_k = \omega^{i-1} \Delta_k$ ;
  | Utilize o modelo anterior  $m_k^{(i-1)}$ , para calcular um novo
  | modelo  $m_k^{(i)}$ ,  $\Lambda$ -posicionado em  $\mathcal{B}(x_k, \tilde{\Delta}_k)$ ;
  | if  $\tilde{\Delta}_k \leq \mu \|g_k\|$  then
  |   | Retorne  $\tilde{\Delta}_k$  e  $\tilde{m}_k = m_k^{(i)}$ .

```

Vamos fazer algumas suposições sobre a função objetivo f . Como os iterandos do algoritmo sempre apresentam descendo com relação ao ponto anterior, todos os eles pertencem ao conjunto de nível (inferior) definido com relação ao primeiro iterando:

$$L(x_0) = \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\}$$

Para a construção dos modelos são usados pontos nas regiões de confiança, definidas por bolas em torno dos iterandos. Então, f não é avaliada apenas em $L(x_0)$, mas no conjunto maior

$$L_{enl}(x_0) = \bigcup_{x \in L(x_0)} \mathcal{B}(x, \Delta_{\max}).$$

Hipótese 1 *Sejam x_0 e Δ_{\max} dados. A função f é diferenciável e tem gradiente Lipschitz contínuo em um domínio aberto contendo $L_{enl}(x_0)$.*

Hipótese 2 *A função f é limitada inferiormente em $L(x_0)$, isto é, que existe uma constante f^* tal que para todo $x \in L(x_0)$, $f(x) \geq f^*$.*

E introduzimos, ainda, uma hipótese sobre o modelo

Hipótese 3 *Suponha que existe uma constante κ_{bhm} tal que para todo iterando x_k , a derivada segunda do modelo H_k satisfaz*

$$\|H_k\| \leq \kappa_{bhm}.$$

E uma hipótese sobre o passo escolhido durante o algoritmo:

Algoritmo 2.5: Um método de região de confiança

Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ dada;

Seja $x_0 \in \mathbb{R}^n$ um ponto inicial;

Seja $m_0 : \mathbb{R}^n \rightarrow \mathbb{R}^n$ um modelo inicial;

Seja $\mathcal{B}(x_0, \Delta_0)$ a região de confiança inicial;

Defina $0 \leq \eta_0 \leq \eta_1 < 1$ e $\eta_1 \neq 0$;

Defina os fatores de redução e aumento da região de confiança

$0 < \gamma < 1 < \gamma_{inc}$;

Defina o raio máximo Δ_{\max} da região de confiança;

for $k = 0, 1, \dots$ **do**

Construa o modelo $m_k(x_k)$ que interpola f em \mathcal{Y}_k . Calcule o gradiente $g_k = \nabla m_k(x_k)$;

if $\|g_k\| \leq \epsilon_g$ e \mathcal{Y}_k não é Λ -posicionado em $\mathcal{B}(x_k, \mu \|g_k\|)$ **then**

 Refine o modelo até que \mathcal{Y}_k seja Λ -posicionado em

$\mathcal{B}(x_k, \delta_k)$, $\delta_k \in (0, \mu \|g_k\|)$.

Calcule o ponto $x_k^+ = x_k + s_k$ que garanta descenso para o modelo m_k em \mathcal{B} ;

Calcule o grau de concordância entre a predição de m_k e o valor de f :

$$\rho_k = \frac{f(x_k) - f(x_k^+)}{m_k(x_k) - m_k(x_k^+)}; \quad (2.18)$$

Atualize o iterando:

$$x_{k+1} = \begin{cases} x_k^+ & \text{se } \rho_k \geq \eta_1 \text{ ou } \rho_k \geq \eta_0 \text{ e } \mathcal{Y}_k \text{ } \Lambda\text{-posicionado} \\ x_k & \text{caso contrário} \end{cases}$$

Atualize o tamanho da região de confiança:

$$\Delta_{k+1} = \begin{cases} \min(\gamma_{inc}\Delta_k, \Delta_{\max}), & \text{se } \rho_k \geq \eta_1 \\ \gamma\Delta_k, & \text{se } \rho_k < \eta_1 \text{ e } \mathcal{Y}_k \text{ } \Lambda\text{-posicionado} \\ \Delta_k, & \text{se } \rho_k < \eta_1 \text{ e } \mathcal{Y}_k \text{ não } \Lambda\text{-posicionado} \end{cases}$$

Atualize o modelo m_{k+1} ;

Hipótese 4 Para toda iteração k , o passo s_k escolhido satisfazendo ao menos uma fração κ_{fcd} do decréscimo referente ao passo de Cauchy:

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{fcd} [m_k(x_k) - m_k(x_k + s_k^C)]$$

para uma constante $\kappa_{fcd} \in (0, 1)$.

Com base nessas hipóteses, [16] apresenta o seguinte teorema:

Teorema 6 Suponha as hipóteses 1, 2, 3 e 4. Então

$$\lim_{k \rightarrow +\infty} \|\nabla f(x_k)\| = 0$$

2.3 ALTERNATIVAS ALGORÍTMICAS

Em [17] é apresentado um algoritmo para a manutenção do modelo de interpolação polinomial mais simples que aquele apresentado neste capítulo. Os resultados preliminares são interessantes, podendo-se economizar no número de avaliações da função objetivo.

Tal abordagem procura utilizar ao máximo os pontos em que já se conhece a função objetivo. Não a apresentamos nesta dissertação porque não nos parece claro como estendê-la ao caso com restrições lineares, do qual trataremos no próximo capítulo.

2.4 SUMÁRIO

Neste capítulo introduzimos os métodos de otimização não-diferenciável. Trata-se de métodos de otimização que não necessitam de informação sobre as derivadas da função objetivo. Assim, podem ser usados em situações inacessíveis aos métodos clássicos de otimização, baseados em derivadas. Podem ser usados, por exemplo, na otimização de funções calculadas por simuladores ou por experimentos físicos.

Apresentamos dois métodos de otimização não-diferenciável: busca direta direcional e região de confiança não-diferenciável. O primeiro, amostra a função objetivo um número finito de vezes por iteração e utiliza tais valores para decidir sobre o andamento do algoritmo. Não constrói qualquer modelo da função objetivo.

O método da região de confiança, ao contrário, primeiro amostra a função objetivo a fim de construir um modelo que a aproxime localmente. Este modelo é utilizado para decidir onde se buscará por descenso.

Como estes métodos não fazem uso da informação das derivadas, é de se esperar que sua convergência não seja tão eficiente quanto a

dos métodos que as utilizam. Portanto, se derivadas estão disponíveis, é recomendado que se utilize um método que faça uso delas [16].

3 OTIMIZAÇÃO NÃO-DIFERENCIÁVEL COM RESTRIÇÕES LINEARES

No capítulo anterior apresentamos algoritmos de otimização não-diferenciável, porém não foram consideradas quaisquer restrições. Neste capítulo estenderemos os métodos de otimização já tratados para problemas sujeitos a restrições lineares nas variáveis, da forma seguinte:

$$\begin{aligned} \min \quad & f(x) \\ \text{sujeito a} \quad & x \in \Omega = \{x : Ax \leq b\}. \end{aligned}$$

Os métodos deste capítulo buscam por descenso apenas nos pontos do conjunto viável. Assim, todos os iterandos são sucessivas aproximações para o ponto ótimo.

3.1 MATERIAL BÁSICO SOBRE OTIMIZAÇÃO RESTRITA

O conjunto viável Ω contém os pontos que satisfazem às restrições do problema.

Definimos o *cone polar* de um conjunto $K \subseteq \mathbb{R}^n$, denotado K° como

$$K^\circ = \{w : w^T v \leq 0, \quad \forall v \in K\};$$

como o nome sugere, o cone polar é um cone, e é convexo. Alguns exemplos de cones polares estão na Figura 3.1.

Definimos o *cone normal* ao conjunto viável Ω no ponto x como o conjunto

$$\mathcal{N}_\Omega(x) = \{y \in \mathbb{R}^n : y^T(u - x) \leq 0, \quad \forall u \in \Omega\}$$

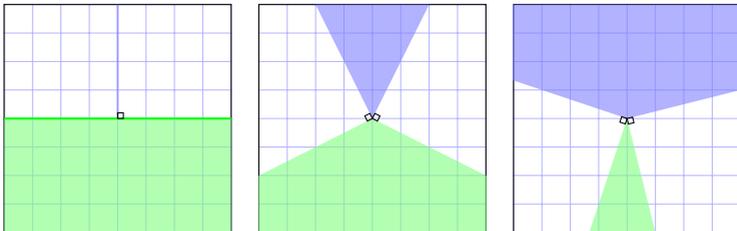


Figura 3.1: Conjuntos e seus polares. Para conjuntos cônicos convexos, o cone polar do cone polar é o próprio conjunto ($K = (K^\circ)^\circ$).

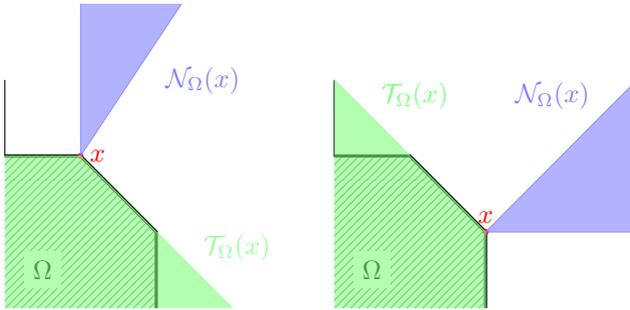


Figura 3.2: Cone normal e cone tangente a Ω em dois diferentes pontos da fronteira do conjunto viável.

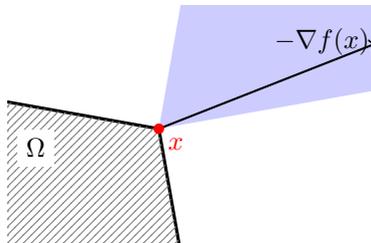


Figura 3.3: Ponto $x \in \Omega$ ótimo sob restrições: $-\nabla f(x) \in \mathcal{N}_\Omega(x)$

e o *cone tangente* a Ω em x como o polar do cone normal no mesmo ponto, isto é,

$$\mathcal{T}_\Omega(x) = \mathcal{N}_\Omega(x)^\circ = \{y \mid y^T x \leq 0, \quad \forall x \in \mathcal{N}_\Omega(x)\}.$$

Informalmente, o cone normal $\mathcal{N}_\Omega(x)$, contém as direções que “apontam para fora” de Ω . Enquanto o cone tangente $\mathcal{T}_\Omega(x)$ contém vetores v , em que, a partir de x , se pode andar na direção v alguma distância sem sair de Ω . Para um ponto interno a Ω , $\mathcal{T}_\Omega(x) = \mathbb{R}^n$ e $\mathcal{N}_\Omega(x) = \{0\}$. Uma ilustração se encontra na Figura 3.2.

Para definir ponto ótimo restrito, dizemos que x^* é ótimo de primeira ordem sob restrições se, e somente se

$$-\nabla f(x^*)^T d \leq 0, \quad \forall d \in \mathcal{T}_\Omega(x^*). \quad (3.1)$$

Uma ilustração de ponto ótimo restrito encontra-se na Figura 3.3.

Definimos a *projeção* de um vetor $x \in \mathbb{R}^n$ em um conjunto Ω , e

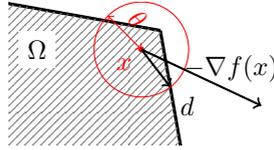


Figura 3.4: O vetor d que resolve a equação (3.3) na definição de $\chi(x, \theta)$.

denotamos $P_{\Omega}[x]$ como o mínimo para o problema

$$\min_{y \in \Omega} \|y - x\|_2$$

A chamada *decomposição de Moreau* do ponto $x \in \mathbb{R}^n$ relativa a $y \in \Omega$ é dada por

$$x = P_{\mathcal{T}_{\Omega}(y)}[x] + P_{\mathcal{N}_{\Omega}(y)}[x]. \quad (3.2)$$

Como no caso restrito não podemos usar, por exemplo, $\|\nabla f\|$ para detectar um ponto estacionário, definimos a medida

$$\chi(x, \theta) = \left| \min_{\substack{x+d \in \Omega \\ \|d\| \leq \theta}} \nabla f^T d \right|. \quad (3.3)$$

Esta medida é ilustrada na Figura 3.4: d é a direção mais próxima de $-\nabla f(x)$ que, a partir de x , se pode percorrer a distância θ sem sair do conjunto viável.

De acordo com [20], temos o seguinte teorema, que mostra que χ pode ser usada como medida de criticidade:

Teorema 7 *Suponha que f tem derivada segunda contínua e que o conjunto viável Ω é não-vazio, fechado e convexo. Então*

$$\chi(x) = \chi(x, 1), \quad x \in \Omega \quad (3.4)$$

é contínua, não-negativa e $\chi(x) = 0$ se, e somente se x é ponto ótimo.

Isto é, $\chi(x)$ pode ser vista como uma indicação da criticidade da função f associada. Se $\chi(x) \rightarrow 0$, conseqüentemente $-\nabla f(x)^T d \leq 0 \forall d \in K(x)$.

3.2 BUSCA DIRETA DIRECIONAL COM RESTRIÇÕES LINEARES

Uma ilustração das dificuldades introduzidas por restrições está na Figura 3.5. Pelo Teorema 1, sabemos que ao considerar um conjunto

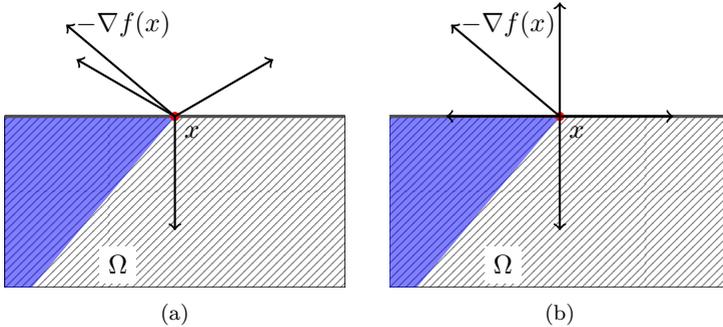


Figura 3.5: Conjunto viável Ω hachurado, abaixo da restrição (horizontal); direções de descendo viáveis sombreadas. Em um primeiro caso, a direção de busca próxima a $-\nabla f(x)$ não é viável. Usando-se direções de busca paralelas às faces, é possível encontrar descendo viável.

de direções de busca gerador positivo, é possível garantir que há uma das direções formando ângulo agudo com a direção de máximo descendo. No entanto, pode ocorrer que justamente tal direção se torne inviável (Fig. 3.5), levando para fora do conjunto viável, não importa quão pequeno seja o comprimento do passo. E no exemplo da figura, o iterando atual x_k não é ótimo restrito. Ao mesmo tempo, desejamos prosseguir com a busca sempre dentro do conjunto viável, de modo que todos os iterandos sejam sucessivas aproximações do ótimo do problema e a busca possa ser interrompida quando se obtiver uma exatidão suficiente na resposta. Dessa maneira, é necessário construir conjuntos de direções de busca mais abrangentes para o caso restrito. Para o caso da figura, alterando-se o conjunto das direções de busca de modo a incluir direções paralelas às faces do conjunto Ω já é possível encontrar descendo viável.

Ao utilizarmos a busca direta direcional, é necessário definir quais as faces do conjunto viável que estão excessivamente próximas do iterando x_k . E proceder a busca paralelamente a elas, evitando-se pontos inviáveis.

Considere que o conjunto viável é dado por restrições lineares, da forma $\Omega = \{x : Ax \leq b\}$.

Cada restrição linear, respectivamente cada linha da matriz A , define uma face do conjunto viável, da forma

$$\{x : a_i^T x = b_i\},$$

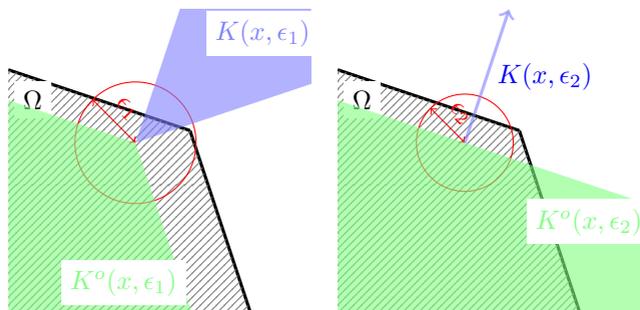


Figura 3.6: Os cones $K(x, \epsilon)$ e seus polares. A partir de x_k é possível andar ϵ_i em qualquer direção de $K^o(x_k, \epsilon_i)$.

em que a_i^T é a linha i da matriz A e b_i é a componente i do vetor b . Além disso, a_i^T é o vetor ortogonal à respectiva face.

Para $x \in \Omega$, definimos o conjunto das *restrições ativas* por

$$I(x) = \{i \in \{1, \dots, m\} : a_i^T x = b_i\}$$

e, dado $\epsilon > 0$, o conjunto das *restrições ϵ -ativas* por

$$I(x, \epsilon) = \{i \in \{1, \dots, m\} : a_i^T x - b_i \geq -\epsilon\}.$$

Assim, $I(x, \epsilon)$ contém justamente os índices das restrições que estão a uma distância ϵ de serem violadas. Isto é, todos os i tais que $x + \epsilon a_i \notin \Omega$.

Definimos também o cone

$$K(x, \epsilon) = \left\{ d \in \mathbb{R}^n \mid d = \sum_{i \in I(x, \epsilon)} \lambda_i a_i, \quad \lambda_i > 0 \right\}.$$

Em termos gerais, trata-se do cone gerado pelas direções a_i que apontam para fora do conjunto viável. O polar correspondente, $K^o(x, \epsilon)$, aproxima localmente a geometria do conjunto viável Ω . Assim, é possível partir de x e percorrer ao menos uma distância ϵ em qualquer direção de $K^o(x, \epsilon)$, permanecendo-se em Ω , conforme ilustrado na Figura 3.6 e formalizado na proposição seguinte:

Proposição 1 *Se $x \in \Omega$ e $w \in K^o(x, \epsilon)$, $\|w\| \leq \epsilon$, então $x + w \in \Omega$.*

Dessa maneira, o tratamento de restrições que apresentaremos utiliza conjuntos geradores positivos que satisfazem uma das seguintes condições:

Condição 1 Para $\epsilon_k > 0$, \mathcal{D}_k gera positivamente $K^o(x_k, \epsilon_k)$.

Condição 2 \mathcal{D}_k inclui geradores positivos para todos os cones $K^o(x, \epsilon)$, $0 \leq \epsilon \leq \epsilon^*$ para algum $\epsilon^* > 0$ independente da iteração k .

O teorema seguinte aponta uma maneira de se gerar as direções de busca para satisfazer tais condições [21, 22].

Teorema 8 *Suponha que um cone $K(x, \epsilon_0) \subseteq \mathbb{R}^n$ é gerado positivamente pelos vetores que compõem as colunas de uma matriz V . Suponha que tal matriz tem posto-coluna completo (os vetores são linearmente independentes).*

- *Sejam u_1, \dots, u_r os vetores que geram positivamente o núcleo (espaço-nulo) de V^T .*
- *Sejam w_1, \dots, w_s os vetores das colunas da matriz $V(V^T V)^{-1}$.*

Então,

1. *O conjunto de direções $\mathcal{D} = \{u_i\}_{1 \leq i \leq r} \cup \{-w_i\}_{1 \leq i \leq s}$ satisfaz a condição 1 (isto é, gera o cone polar $\bar{K}^o(x, \epsilon_0)$).*
2. *O conjunto de direções $\mathcal{D} = \{u_i\}_{1 \leq i \leq r} \cup \{-w_i, w_i\}_{1 \leq i \leq s}$ satisfaz a condição 2, (gera o cone polar $K^o(x, \epsilon)$ para todo $0 < \epsilon < \epsilon_0$).*

Outra informação dada pelo teorema é a quantidade de direções necessárias para conduzir a busca no \mathbb{R}^n . Note que o número s das colunas de $V(V^T V)^{-1}$ é o posto de V . Se V tem posto completo ($s = n$), o número de direções necessárias para atender à condição 2 é $2n$. Caso $s < n$, o núcleo de V^T tem dimensão $n - s$, podendo ser gerado positivamente por $r = n - s + 1$ vetores. Nesse caso, o número de vetores do conjunto \mathcal{D} das direções de busca é $(n - s + 1) + s + 2 < 2n$.

Lembrando que $\chi(x)$, definida conforme o Teorema 7, é uma medida de estacionariedade, a seguinte proposição ilustra como um ϵ suficientemente pequeno induz uma cota superior para essa medida.

Proposição 2 *Considere $\gamma > 0$. Existe $r > 0$ dependendo apenas de γ e da matriz de restrições A , tal que vale o seguinte: suponha que $x \in \Omega$, $\|\nabla f(x)\| < \gamma$ e $\chi(x) > 0$. Dado $\epsilon \geq 0$, seja \mathcal{D} o conjunto de geradores de $K^o(x, \epsilon)$. Então existe um número $\kappa_{restr}(\mathcal{D}) > 0$, dependendo apenas de \mathcal{D} tal que se $\epsilon < r$, para algum vetor $d \in \mathcal{D}$,*

$$\kappa_{restr}(\mathcal{D})\chi(x) \|d\| \leq -\nabla f(x)^T d. \quad (3.5)$$

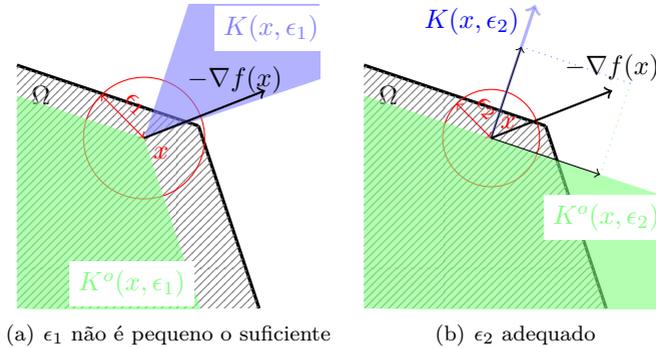
(a) ϵ_1 não é pequeno o suficiente(b) ϵ_2 adequado

Figura 3.7: Para se encontrar uma direção de descenso viável em $K^o(x, \epsilon)$, ϵ precisa ser pequeno.

A condição $\epsilon < r$ do teorema é ilustrada com a Figura 3.7. Para um valor ϵ_1 muito grande, a direção de máximo descenso $-\nabla f(x)$ não tem projeção em $K^o(x, \epsilon_1)$, enquanto para um ϵ_2 suficientemente pequeno, sim.

Isto é, κ_{restr} tem um papel análogo ao feito pela medida cosseno no caso irrestrito. De modo semelhante, também é necessário garantir uma boa geometria das direções de busca. Para tanto, introduzimos a seguinte condição

Condição 3 *Existe um $\kappa_{\min} > 0$ tal que para todo ϵ e todo conjunto \mathcal{D}_k , que gera o cone $K^o(x_k, \epsilon)$, a relação (3.5) é satisfeita com*

$$\kappa_{restr} > \kappa_{\min}. \quad (3.6)$$

De acordo com [13], tal condição só precisa ser levada em consideração explicitamente quando $K^o(x_k, \epsilon)$ contém um sub-espaço. Ao mesmo tempo, conjuntos de direções dados pelo Teorema 8 ainda admitem liberdade na escolha dos vetores que geram positivamente o núcleo de V^T .

Uma opção é partir de uma base ortogonal e completá-la com os negativos de todas as suas direções, ou conforme a equação (2.4).

Alguns teoremas de [13] mostram a convergência do método apresentado:

Teorema 9 *Suponha que ∇f é lipchitziana com constante M . Suponha que os iterandos $\{x_k\}$ do Algoritmo 3.1 pertencem a um conjunto limitado, e que $\|\nabla f(x)\| \leq \gamma$ em tal conjunto. Então, existem constantes c*

e r independentes da iteração k , tais que

$$\chi(x_k) \leq c \left[M\Delta_k\beta_{\max} + \frac{\rho(\Delta_k)}{\Delta_k\beta_{\min}} \right]$$

se a condição 2 for utilizada e

$$\chi(x_k) \leq c \left[M\Delta_k\beta_{\max} + \frac{\rho(\Delta_k)}{\Delta_k\beta_{\min}} + r \frac{\gamma}{\beta_{\min}} \epsilon_k \right]$$

se a condição 1 for utilizada.

Concluimos os resultados de convergência do método com o teorema de [13]:

Teorema 10 *Se a condição 1 for utilizada,*

$$\liminf_{k \rightarrow +\infty} \chi(x_k) = 0,$$

se a condição 2 for utilizada,

$$\lim_{k \rightarrow +\infty} \chi(x_k) = 0.$$

3.2.1 Alternativas algorítmicas

Um algoritmo semelhante à busca direta direcional, com fortes características de convergência e capaz de tratar restrições ainda mais gerais, bastando que o conjunto seja conexo, está em [23]. É um algoritmo capaz de tratar até mesmo restrições do tipo caixa-preta, isto é, que não se conhece de antemão. Por outro lado, tal algoritmo não faz uso das restrições conhecidas, como o que apresentamos nesta dissertação.

Algoritmo 3.1: Um algoritmo de busca direta direcional para restrições lineares

Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ dada;
 Seja $x_0 \in \Omega$ um ponto inicial viável;
 Seja $\Delta_{tol} > 0$ a tolerância de convergência;
 Seja $\Delta_0 > \Delta_{tol}$ um tamanho de passo inicial;
 Seja $\phi_{\max} \geq 1$ um limite superior para o coeficiente de expansão ϕ_k ;
 Seja $\theta_{\max} < 1$ um limite superior para o parâmetro de contração θ_k ;
 Sejam $\beta_{\max} \geq \beta_{\min} > 0$ limites superior e inferior, respectivamente, para os comprimentos dos vetores de qualquer conjunto gerador positivo;
 Seja $\kappa_{\min} > 0$ um limitante superior para a medida cosseno dos conjuntos geradores positivos;
 Seja $\rho : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ uma função não-decrescente satisfazendo a equação (2.8);
 Se for utilizada a condição 1, seja ϵ_0 a tolerância inicial para as restrições localmente ativas e $0 < \theta_\epsilon < 1$ o fator de redução de ϵ_k ;
 Se for utilizada a condição 2, defina $\epsilon > 0$ para a tolerância das restrições localmente ativas;

for $k = 1, 2, \dots$ **do**

Defina \mathcal{D}_k conjunto gerador positivo satisfazendo a condição 3 e 2 ou 1 e $\beta_{\min} \leq \|d\| \leq \beta_{\max}$, $\forall d \in \mathcal{D}_k$, com $\kappa(\mathcal{D}_k) \geq \kappa_{\min}$;

for $d \in \mathcal{D}_k$ **do**

Defina $\alpha_k(d)$ como o maior valor tal que

$(x_k + \alpha_k(d)d) \in \Omega$;

Defina $\bar{\Delta}_k = \min(\Delta_k, \alpha_k(d))$;

if *Existe* $d \in \mathcal{D}_k$ tal que $f(x_k + \bar{\Delta}_k d) < f(x_k) - \rho(\Delta_k)$

then

$x_{k+1} \leftarrow x_k + \bar{\Delta}_k d$;

$\Delta_{k+1} \leftarrow \phi_k \Delta_k$, em que $1 \leq \phi_k \leq \phi_{\max}$;

else

$x_{k+1} \leftarrow x_k$;

$\Delta_{k+1} \leftarrow \theta_k \Delta_k$, $0 < \theta_k \leq \theta_{\max}$;

Se está sendo usada a condição 1, $\epsilon_{k+1} \leftarrow \theta_k \epsilon_k$;

if $\Delta_{k+1} < \Delta_{tol}$ **then**

Encerrar;

3.3 MÉTODOS DE REGIÃO DE CONFIANÇA COM RESTRIÇÕES LINEARES

Métodos de região de confiança podem ser estendidos a problemas com restrições lineares de forma mais intuitiva que os de busca direta direcional. Em princípio, pode-se pensar em restringir os algoritmos apresentados ao conjunto viável. No entanto, se f for amostrada somente em pontos viáveis, pode não ser possível obter bons modelos (isto é, modelos Λ -posicionados), que são essenciais nas provas de convergência daqueles algoritmos.

Nesta seção mostraremos que, com pequenas alterações, o Algoritmo 2.5 já apresentado funciona para o caso com restrições lineares nas variáveis. Utilizaremos a abordagem que [20] apresenta ao abordar o caso diferenciável com restrições convexas.

Idealmente, a função objetivo seria avaliada apenas em pontos viáveis, candidatos à solução do problema. Mas o formato do conjunto viável Ω pode não permitir que se consiga um modelo de interpolação adequado. Para manter um modelo Λ -posicionado não faremos tal restrição. Os pontos usados na construção do modelo na região $\mathcal{B}(x_k, \Delta_k)$ serão tomados sem levar em consideração se são viáveis ou não.

Para tanto, propomos utilizar os Algoritmos 2.2 e 2.3 sem qualquer alteração. Resta comentar sobre o cálculo do passo.

3.3.1 Passo generalizado de Cauchy

Como no capítulo anterior, a cada iteração precisamos encontrar um ponto candidato à solução que satisfaça uma condição mínima de descenso para o modelo.

Como no caso irrestrito, se o modelo usado for linear, em uma bola na norma $\|\cdot\|_\infty$, o acréscimo de (outras) restrições lineares não introduz maiores dificuldades. Ainda é possível chegar ao ótimo global utilizando programação linear. No entanto, para modelos quadráticos, encontrar o ótimo global pode ser um problema NP-difícil se o problema for não-convexo. Nesta seção introduziremos o passo generalizado de Cauchy conforme [20], que define um descenso suficiente para garantir a convergência do algoritmo e pode ser encontrado em todas as iterações.

Como no caso anterior, o passo de Cauchy envolve a busca por descenso utilizando a parte linear do modelo, isto é, a direção g_k , partindo do ponto x_k , mas agora sujeito a $x_k + s \in \Omega$.

Para tanto, defina o *caminho do gradiente projetado* (*projected gradient path* de [20]) como

$$p(t, x) = P_\Omega [x - t\nabla f(x)], \quad t \geq 0.$$

Isto é, a projeção em Ω de um ponto da reta que passa por x e tem direção $-\nabla f(x)$. Uma generalização do passo de Cauchy é encontrar o mínimo do modelo sobre o caminho do gradiente projetado.

3.3.2 Algoritmo de região de confiança com restrições lineares

Agora estamos em condições de definir o método de região de confiança para o caso com restrições lineares (Algoritmo 3.2). Difere, com relação ao anterior, apenas no fato de manter todos os seus iterandos (isto é, os pontos x_k onde são centradas as regiões), viáveis. A construção e manutenção de modelos é feita da mesma forma.

Parte-se de um ponto viável $x_0 \in \Omega$. E para o cálculo do passo $x_k^+ = x_k + s_k$, se busca por descenso para m_k em $\mathcal{B}(x_k, \Delta_k) \cap \Omega$, de forma a não sair do conjunto viável.

Algoritmo 3.2: Um método de região de confiança

Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ dada;

Seja $x_0 \in \mathbb{R}^n$ um ponto inicial;

Seja $m_0 : \mathbb{R}^n \rightarrow \mathbb{R}^n$ um modelo inicial;

Seja $\mathcal{B}(x_0, \Delta_0)$ a região de confiança inicial;

Defina $0 \leq \eta_0 \leq \eta_1 < 1$ e $\eta_1 \neq 0$;

Defina os fatores de redução e aumento da região de confiança

$0 < \gamma < 1 < \gamma_{inc}$;

Defina o raio máximo Δ_k da região de confiança;

for $k = 0, 1, \dots$ **do**

Construa o modelo $m_k(x_k)$ que interpola f em \mathcal{Y}_k . Calcule o gradiente $g_k = \nabla m_k(x_k)$;

if $\|g_k\| \leq \epsilon_g$ e \mathcal{Y}_k não é Λ -posicionado em $\mathcal{B}(x_k, \mu \|g_k\|)$ **then**

 Refine o modelo até que \mathcal{Y}_k seja Λ -posicionado em
 $\mathcal{B}(x_k, \delta_k)$, $\delta_k \in (0, \mu \|g_k\|)$.

Calcule o ponto $x_k^+ = x_k + s_k$ que garanta descenso para o modelo m_k em $\mathcal{B} \cap \Omega$;

Calcule o grau de concordância entre a predição de $m_k(x_k^+)$ e o valor de $f(x_k^+)$:

$$\rho_k = \frac{f(x_k) - f(x_k^+)}{m_k(x_k) - m_k(x_k^+)}; \quad (3.7)$$

Atualize o iterando:

$$x_{k+1} = \begin{cases} x_k^+ & \text{se } \rho_k \geq \eta_1 \text{ ou } \rho_k \geq \eta_0 \text{ e } \mathcal{Y}_k \text{ } \Lambda\text{-posicionado} \\ x_k & \text{caso contrário} \end{cases}$$

Atualize o tamanho da região de confiança:

$$\Delta_{k+1} = \begin{cases} \min(\gamma_{inc} \Delta_k, \Delta_{\max}), & \text{se } \rho_k \geq \eta_1 \\ \gamma \Delta_k, & \text{se } \rho_k < \eta_1 \text{ e } \mathcal{Y}_k \text{ é } \Lambda\text{-posicionado} \\ \Delta_k, & \text{se } \rho_k < \eta_1 \text{ e } \mathcal{Y}_k \text{ não é } \Lambda\text{-posicionado} \end{cases}$$

Atualize o modelo m_{k+1} ;

3.3.3 Alternativas algorítmicas

Em [18], é apresentado um algoritmo eficiente para o caso de limites superiores e inferiores nas variáveis, estendendo o apresentado em [17]. Quando uma variável chega a um limite, ela é fixada e o problema segue, otimizando-se apenas as variáveis restantes. Dessa maneira, os modelos são construídos apenas com variáveis irrestritas, evitando o problema descrito no parágrafo acima. Uma grande vantagem dessa abordagem é que os modelos ficam mais simples conforme as variáveis chegam aos seus limites, diminuindo o esforço de manutenção do modelo. O algoritmo procura economizar nos passos de melhora de modelo e nas avaliações da função objetivo f .

Por outro lado, tal algoritmo não é extensível para outras restrições além de limites superiores e inferiores. A autora avalia a possibilidade de tratar demais restrições usando métodos como SQP e Lagrangiano aumentado. A abordagem apresentada neste capítulo, trata diretamente problemas que não são resolvidos por [18]. Ao custo de um maior esforço na manutenção dos modelos e provável maior número de avaliações da função objetivo f , por incluir pontos inviáveis na construção do modelo m .

3.4 SUMÁRIO

Neste capítulo apresentamos alguns conceitos de otimização restrita e estendemos os algoritmos de busca direta direcional e de região de confiança. Em ambos os casos, os algoritmos buscam por descenso sem deixar o conjunto viável.

A busca direta direcional pode ser estendida ao caso com restrições lineares por meio de conjuntos de direções de busca que reproduzam, de certa maneira, a geometria local do conjunto viável, possibilitando a busca por descenso em direções viáveis.

A extensão do algoritmo de região de confiança é mais simples. Basta resolver o sub-problema da região de confiança respeitando também as restrições lineares. É importante lembrar, porém, que para a construção do modelo pode ser necessário incluir-se também pontos inviáveis.

O próximo capítulo apresenta um algoritmo de Lagrangiano aumentado para a resolução de problemas também com restrições não-lineares. Tal abordagem acaba gerando sub-problemas com restrições lineares, que podem ser resolvidos com os algoritmos do presente capítulo.

4 OTIMIZAÇÃO NÃO-DIFERENCIÁVEL COM RESTRIÇÕES NÃO-LINEARES

Nos capítulos anteriores apresentamos algoritmos capazes de resolver o problema de otimização com restrições lineares. Na produção de petróleo, no entanto, se forem considerados itens como limites da produção de gás ou de algum contaminante, os problemas resultantes terão restrições não-lineares. Assim, não podem ser resolvidos com os métodos vistos. A abordagem utilizada neste trabalho para resolver tais problemas foi a de usar um método de Lagrangiano aumentado.

É um método que consiste em substituir as restrições do problema de otimização por uma sequência de sub-problemas de otimização. Estes são irrestritos, ou com restrições mais simples. As restrições não-lineares originais são substituídas por termos de penalização na função objetivo. Assim, os métodos dos capítulos anteriores podem ser empregados na resolução dos sub-problemas.

4.1 LAGRANGIANO AUMENTADO

A fim de resolver o problema na função objetivo $f : \mathbb{R}^n \rightarrow \mathbb{R}$, com restrições não-lineares $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$:

$$\begin{aligned} \min \quad & f(x) \\ \text{sujeito a} \quad & h(x) = 0. \end{aligned}$$

O método consiste em resolver uma sequência de sub-problemas, derivados deste, porém com as restrições de igualdade h transformadas em penalizações, incluídas na função objetivo, em vez de serem tratadas explicitamente. Ao contrário do capítulo anterior, admitimos que a busca ocorra também por pontos inviáveis.

Para tanto, introduzimos a função *Lagrangiano aumentado* associada a f , dada por

$$\begin{aligned} \mathcal{L}(x, \lambda, c) &= f(x) + \sum_{p=1}^m \lambda_p h_p(x) + \frac{1}{2} \sum_{p=1}^m c h_p^2(x) \\ &= f(x) + \lambda^T h(x) + \frac{c}{2} \|h(x)\|^2. \end{aligned}$$

Para pontos viáveis ($x \in \Omega$), $h(x) = 0$ e $\mathcal{L}(x, \lambda, c) = f(x)$. O *parâmetro de penalização* c determina o quanto as violações das restrições h são penalizadas, enquanto λ consiste de estimativas para os multiplicadores de Lagrange do problema original.

Os problemas a serem resolvidos, então, são dados por

$$\min_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda, c)$$

Quanto maior o valor de c , maior a penalização imposta aos pontos inviáveis $x \notin \Omega$. Assim, se c for alto, a minimização de $\mathcal{L}(x, \lambda, c)$ tenderá a produzir pontos onde $h(x) \approx 0$. Nesse caso, $\mathcal{L}(x, \lambda, c) \approx f(x)$, já que o ponto x quase é viável, assim a minimização de $\mathcal{L}(x, \lambda, c)$ aproxima a de $f(x)$. [24].

Se λ é uma boa estimativa para os multiplicadores de Lagrange exatos λ^* (correspondentes ao ponto ótimo x^*), a minimização de $\mathcal{L}(x, \lambda, c)$ provê um ponto x próximo de x^* , contanto que c seja suficientemente alto para que se possa realizar tal minimização [24].

4.1.1 Lagrangiano aumentado com resolução aproximada dos sub-problemas

Porém, não é necessário encontrar com exatidão o ponto ótimo em cada um dos sub-problemas. Estamos particularmente interessados em métodos iterativos nos quais, em vez de buscar por um ponto x^k em que

$$\|\nabla_x \mathcal{L}(x, \lambda, c)\| = 0,$$

o usual é que o método pare quando satisfeito um critério da forma

$$\|\nabla_x \mathcal{L}(x, \lambda, c)\| \leq \epsilon^k,$$

isto é, com a norma do gradiente (ou critério semelhante) baixa, ainda que não-nula.

A convergência de um algoritmo feito pela resolução de tais sub-problemas é esclarecida, em parte, com a seguinte proposição, de [24]:

Proposição 3 *Sejam f e h continuamente diferenciáveis. Para a iteração $k = 0, 1, \dots$, considere que x^k satisfaz*

$$\|\nabla \mathcal{L}(x^k, \lambda^k, c^k)\| \leq \epsilon^k,$$

em que a sequência $\{\lambda^k\}$ é limitada, bem como $\{\epsilon^k\}$ e $\{c^k\}$ satisfazem

$$0 < c^k < c^{k+1}, \quad \forall k, \quad c^k \rightarrow \infty,$$

$$0 \leq \epsilon^k, \quad \forall k, \quad \epsilon^k \rightarrow 0$$

Considere que a subsequência $\{x^k\}_K$ converge para um vetor x^* tal que $\nabla h(x^*)$ tem posto m . Então

$$\{\lambda^k + c^k h(x^k)\}_K \rightarrow \lambda^*,$$

em que λ^* satisfaz, juntamente com x^* , as condições de otimalidade de primeira ordem para o problema original (com restrição de igualdade):

$$\nabla f(x^*) + \nabla h(x^*)\lambda^* = 0, \quad h(x^*) = 0.$$

As hipóteses da proposição podem não ser satisfeitas, como comentários a seguir [24].

Em primeiro lugar, pode não ser possível encontrar um $\{x^k\}$ satisfazendo $\|\nabla \mathcal{L}(x^k, \lambda^k, c^k)\| \leq \epsilon^k$. A causa disso, normalmente, é que o Lagrangiano aumentado $\mathcal{L}(\cdot, \lambda^k, c^k)$ não é limitado inferiormente.

Uma segunda possibilidade é que, ou a sequência $\{x^k\}$ não converge, ou converge para um ponto x^* em que $\nabla h(x^*)$ tenha colunas linearmente dependentes. Isto normalmente ocorre quando o Lagrangiano aumentado é limitado inferiormente, porém o problema original não tem uma solução viável. Neste caso, quando se aumenta a penalização c , o termo quadrático $\frac{c}{2}\|h(x)\|^2$ domina e a solução do sub-problema é justamente um ponto estacionário desta função, de modo que

$$\frac{c}{2}\nabla\|h(x^*)\|^2 = 0.$$

Um terceiro caso, que raramente ocorre na prática, é que $\{x^k\}$ converge para um ponto $\{x^*\}$ ao qual não há multiplicadores de Lagrange associados. Neste caso, a sequência $\{\lambda^k + c^k h(x^k)\}$ também diverge.

Apontados esses casos, patológicos, [24] comenta que o normal é que o algoritmo chegue a um par (x^*, λ^*) satisfazendo as condições de otimalidade. Ainda, a grande experiência prática consolida que o método, de modo geral, é confiável e normalmente converge para um ponto que é, pelo menos, mínimo local do problema original. As falhas, usualmente se devem ao fato de que minimizar $\mathcal{L}(\cdot, \lambda^k, c^k)$ torna-se mais difícil conforme $c^k \rightarrow \infty$.

Existe, ainda, outro motivo que torna ineficiente a busca pela solução exata dos sub-problemas: os multiplicadores de Lagrange não são exatos. A solução do problema interno pode se afastar da solução do problema original justamente por isso. Assim, alguns métodos chegam a iterar de maneira ainda mais intensa entre a resolução de sub-problemas e a atualização dos multiplicadores de Lagrange [24].

4.1.2 Lagrangiano aumentado com restrições de desigualdade

Tendo apresentado o caso do método do Lagrangiano sujeito a restrições de igualdade, seguimos com o problema com desigualdades

$$\begin{aligned} \min \quad & f \\ \text{sujeito a} \quad & h(x) = 0 \\ & g(x) \leq 0. \end{aligned}$$

em que $g : \mathbb{R}^n \rightarrow \mathbb{R}^r$, e como antes, $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$.

Este pode ser facilmente colocado na forma anterior, introduzindo-se as variáveis de folga z_1, \dots, z_r [25]:

$$\begin{aligned} \min \quad & f(x) \\ \text{sujeito a} \quad & h(x) = 0 \\ & g(x) + z^2 = 0, \end{aligned}$$

em que z^2 representa o vetor (z_1^2, \dots, z_r^2) .

Então, podemos usar o mesmo método, usando nos sub-problemas a função Lagrangiano aumentado, a seguir reescrita com as variáveis adicionais:

$$\begin{aligned} \mathcal{L}(x, z, \lambda, \mu, c) = f(x) + \lambda^T h(x) + \frac{c}{2} \|h(x)\|^2 + \\ \sum_{l=1}^r \left\{ \mu_l (g_l(x) + z_l^2) + \frac{c}{2} |g_l(x) + z_l^2|^2 \right\}, \end{aligned}$$

em que $g_l(x)$ é a componente l de $g(x)$. Os sub-problemas consistem em

$$\min_{x, z} \mathcal{L}(x, z, \lambda, \mu, c). \quad (4.1)$$

A fim de encontrar x e z que resolvam (4.1), é possível, para cada x , minimizar em z o Lagrangiano aumentado $\mathcal{L}(x, \cdot, \lambda, \mu, c)$. Para tanto, basta minimizar cada termo dependente de z , problema este reescrito na variável $u_l = z_l^2$:

$$\min_{u_l \geq 0} \left\{ \mu_l [g_l(x) + u_l] + \frac{1}{2} c |g_l(x) + u_l|^2 \right\}. \quad (4.2)$$

Nos pontos de mínimo irrestrito \hat{u}_l de cada uma destas funções, a derivada vale zero:

$$\mu_l + c[g_l(x) + \hat{u}_l] = 0,$$

consequentemente,

$$\hat{u}_l = -[(\mu_l/c) + g_l(x)].$$

Se, por outro lado, este valor \hat{u}_l for negativo, o mínimo restrito de (4.2) é $u_l^* = 0$ (A expressão de (4.2) é convexa em u_l).

Então:

$$u_l^* = \max\{0, -[(\mu_l/c) + g_l(x)]\},$$

e

$$g_l(x) + u_l^* = \max\{g_l(x), -(\mu_l/c)\}.$$

Assim, denotando

$$g^+(x, \mu, c) = \begin{bmatrix} \max\{g_1(x), -(\mu_1/c)\} \\ \vdots \\ \max\{g_r(x), -(\mu_r/c)\} \end{bmatrix},$$

a função Lagrangiano aumentado, já minimizada em z é:

$$\begin{aligned} \min_z \mathcal{L}(x, z, \lambda, \mu, c) &= f(x) + \lambda^T h(x) + \frac{1}{2} c \|h(x)\|^2 + \\ &+ \mu^T g^+(x, \mu, c) + \frac{1}{2} c \|g^+(x, \mu, c)\|^2. \end{aligned}$$

Isto motiva a definição do Lagrangiano aumentado para problemas com restrições de desigualdades

$$\begin{aligned} \mathcal{L}(x, \lambda, \mu, c) &= f(x) + \lambda^T h(x) + \mu^T g^+(x, \mu, c) + \\ &+ \frac{1}{2} c \{ \|h(x)\|^2 + \|g^+(x, \mu, c)\|^2 \} \quad (4.3) \end{aligned}$$

que pode ser reescrito [25] como:

$$\begin{aligned} \mathcal{L}(x, \lambda, \mu, c) &= f(x) + \lambda^T h(x) + \frac{1}{2} c \|h(x)\|^2 + \\ &+ \frac{1}{2c} \sum_{l=1}^r \{ \max[0, \mu_l + c g_l(x)]^2 - \mu_l^2 \}. \quad (4.4) \end{aligned}$$

A equivalência entre (4.3) e (4.4) pode ser verificada coordenada por coordenada. Note que se $g_l(x) \geq -(\mu_l/c)$, então $g_l^+(x, \mu, c) = g_l(x)$, mas por outro lado,

$$\begin{aligned} \frac{1}{2c} \{ \max[0, \mu_l + c g_l(x)]^2 - \mu_l^2 \} &= \mu_l g_l(x) + \frac{1}{2} c g_l(x)^2 \\ &= \mu_l g_l^+(x, \mu, c) + \frac{1}{2} c g_l^+(x, \mu, c)^2. \end{aligned}$$

Caso contrário, $g_l(x) \leq -(\mu_l/c)$, então $g_l^+(x, \mu, c) = -(\mu_l/c)$, enquanto

$$\begin{aligned} \frac{1}{2c} \{ \max[0, \mu_l + c g_l(x)]^2 - \mu_l^2 \} &= \frac{1}{2c} (-\mu_l^2) \\ &= \mu_l \left(-\frac{\mu_l}{c} \right) + \frac{1}{2} c \left(\frac{\mu_l}{c} \right)^2 \\ &= \mu_l g_l^+(x, \mu, c) + \frac{1}{2} c g_l^+(x, \mu, c)^2. \end{aligned}$$

Então, os sub-problemas devem ser resolvidos utilizando o Lagrangiano aumentado dado por (4.3) ou (4.4), sem a necessidade de usar nenhuma das variáveis adicionais z_l . O problema continua com n variáveis.

No Apêndice A mostramos ainda, que se as funções f , h e g são continuamente diferenciáveis, então o Lagrangiano (4.4) também o é.

A atualização dos multiplicadores de Lagrange pode ser feita de forma semelhante ao caso irrestrito. Considerando que a minimização de $\mathcal{L}(\cdot, \lambda^k, \mu^k, c)$ resulta em um ponto x^k ,

$$\lambda^{k+1} = \lambda^k + c^k h(x^k) \quad (4.5a)$$

$$\mu^{k+1} = \mu^k + c^k g^+(x^k, \lambda^k, \mu^k). \quad (4.5b)$$

Ainda, (4.5b) pode ser reescrito, coordenada por coordenada:

$$\mu_l^{k+1} = \max\{0, \mu_l^k + c^k g_l(x)\}. \quad (4.6)$$

4.1.3 Eliminação parcial das restrições

Nos casos anteriores, todas as restrições eram transformadas em penalizações da função objetivo, e os sub-problemas consistiam em um problema de minimização irrestrita do Lagrangiano aumentado. Porém, não é necessário fazer esta substituição em todas as restrições. É possível incluir no Lagrangiano aumentado apenas algumas, mantendo as restantes nos sub-problemas, que então as tratam de maneira explícita.

Em problemas com restrições lineares e não-lineares, é possível por exemplo, incluir as não-lineares como penalizações, mas seguir tratando as lineares explicitamente. Isto permitirá utilizar os métodos do capítulo anterior na resolução dos sub-problemas.

Para o problema

$$\begin{aligned} \min \quad & f(x) \\ \text{sujeito a} \quad & h(x) = 0 \\ & g(x) \leq 0, \end{aligned}$$

se incluirmos no Lagrangiano aumentado apenas as restrições referentes à igualdade $h(x) = 0$, chegamos ao sub-problema seguinte:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \mathcal{L}(x, \lambda, c) = f(x) + \lambda^T h(x) + \frac{c}{2} \|h(x)\|^2 \\ \text{sujeito a} \quad & g(x) \leq 0. \end{aligned}$$

Isto não interfere na estratégia de atualização dos multiplicadores de Lagrange, que pode ser mantida a mesma.

De modo geral, as restrições não incluídas no Lagrangiano aumentado são mantidas nos sub-problemas. Não há necessidade de manter as restrições de desigualdades, incluindo no Lagrangiano as de igualdade. Qualquer mistura das duas pode ser penalizada ou mantida [24].

Neste trabalho, é de interesse o caso em que as restrições mantidas são dadas por inequações lineares, da forma $Ax - b = g(x) \leq 0$.

4.2 UM ALGORITMO DE LAGRANGIANO AUMENTADO

Havendo exposto a teoria sobre o método de lagrangiano aumentado, apresentaremos a seguir um algoritmo completo (Algoritmo 4.1), baseado em [26].

O problema que queremos resolver contém restrições não-lineares (a serem incluídas no Lagrangiano aumentado) e restrições lineares (mantidas explicitamente). O problema é dado por:

$$\begin{aligned} \min \quad & f \\ \text{sujeito a} \quad & g(x) \leq 0 \\ & Ax \leq b. \end{aligned}$$

em que $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $A \in \mathbb{R}^{p \times n}$, $b \in \mathbb{R}^p$.

Este algoritmo permite utilizar mais de um parâmetro de penalização c , permitindo que diferentes restrições tenham penalizações distintas. Para tanto, as m restrições não-lineares são divididas em q grupos, denotados \mathcal{Q}_j , possivelmente segundo o tipo de não-linearidade. A cada grupo \mathcal{Q}_j , é associada a penalização c_j .

A função Lagrangiano aumentado correspondente é semelhante à (4.4), com as restrições divididas nos grupos \mathcal{Q} , cada um com um parâmetro de penalização próprio:

$$\mathcal{L}(x, \mu, c) = f(x) + \sum_{j=1}^q \frac{1}{2c_j} \sum_{i \in \mathcal{Q}_j} (\{\max[0, \mu_i + c_j g_i(x)]^2 - \mu_i^2\}).$$

O sub-problema a ser resolvido envolve as restrições lineares, que não foram incluídas no Lagrangiano aumentado:

$$\min_{x \in \mathbb{R}^n} \mathcal{L}(x, \mu, c) \quad (4.7a)$$

$$\text{sujeito a } Ax \leq b. \quad (4.7b)$$

A cada iteração k , o iterando x^k é calculado, resolvendo-se o sub-problema aproximadamente, utilizando-se como critério de parada a norma da projeção do gradiente da função no cone tangente ao espaço viável

$$\|P_{K^o(x, \omega^k)}[-\nabla \mathcal{L}(x, \mu, c)]\| \leq \omega^k, \quad (4.8)$$

em que ω^k determina a exatidão com que se resolve o problema.

Os multiplicadores de Lagrange e os parâmetros de penalização são atualizados de forma alternada. Caso as restrições incluídas no Lagrangiano aumentado estejam suficientemente satisfeitas, isto é

$$\|\max\{g(x)_{[Q_j]}, 0\}\| \leq \eta^k, \quad (4.9)$$

em que $\eta^k > 0$ e o sub-índice $[Q_j]$ denota uma partição do vetor associada ao grupo de restrições $[Q_j]$, o parâmetro de penalização do grupo correspondente é mantido o mesmo,

$$c_j^{k+1} = c_j^k,$$

e os multiplicadores de Lagrange são atualizados, de forma semelhante à (4.6):

$$\mu_{[Q_j]}^{k+1} = \max\{0, \mu_{[Q_j]}^k + c_j^k g_{[Q_j]}(x^k)\}, \quad j = 1, \dots, q.$$

Caso as restrições não sejam satisfeitas conforme (4.9), os multiplicadores de Lagrange são mantidos

$$\mu_{[Q_j]}^{k+1} = \mu_{[Q_j]}^k$$

e os parâmetros de penalização correspondentes são atualizados conforme a fórmula

$$c_j^{k+1} = \tau_j^k c_j^k$$

em que

$$\tau_j^k = \begin{cases} \tau & \text{se } c_j^k = \alpha^k \\ \max(\tau, \alpha^k) & \text{caso contrário} \end{cases}$$

e α^k é o menor dos parâmetros de penalização ao início da iteração:

$$\alpha^k = \min(c_1^k, \dots, c_q^k).$$

Dessa maneira, caso os sub-problemas não cheguem a satisfazer suficientemente as restrições, os parâmetros de penalização c correspondentes são progressivamente aumentados. No limite, $c \rightarrow \infty$, que é uma das formas de garantir a convergência do método.

Ainda, resta garantir que os ω^k e η^k convirjam para zero, de modo que, no limite, as restrições sejam satisfeitas e o ponto encontrado seja ótimo para a função original f .

Algoritmo 4.1: Método de Lagrangiano Aumentado, baseado em [26]

Inicialização: defina as constantes positivas: $\omega^* \ll 1$, $\eta^* \ll 1$, $\tau > 1$, $\alpha_\eta < 1$ e $\beta_\eta < 1$. Seja $k = 0$, $\alpha^0 = \min_{j=1, \dots, q} c_j^0$, $\omega^0 = \alpha^0$, $\eta_0 = (\alpha^k)^{\alpha_\eta}$;

for $k = 1, 2, \dots$ **do**

Resolva o sub-problema (4.7) de modo a encontrar um ponto x^k satisfazendo

$$\|P_{K^o(x, \omega^k)}[-\nabla \mathcal{L}(x, \mu, c)]\| \leq \omega^k$$

Caso $\|P_{K^o(x, \omega^k)}[-\nabla \mathcal{L}(x, \mu, c)]\| \leq \omega^*$ e

$\|\max\{g(x)_{[\mathcal{Q}_j]}, 0\}\| \leq \eta^*$, encerre.

for $j = 1, \dots, q$ **do**

if $\|\max\{g(x)_{[\mathcal{Q}_j]}, 0\}\| \leq \eta^k$ **then**

Atualize os multiplicadores de Lagrange

$$c_j^{k+1} = c_j^k,$$

$$\mu_{[\mathcal{Q}_j]}^{k+1} = \max\{0, \mu_{[\mathcal{Q}_j]}^k + c_j^k g_{[\mathcal{Q}_j]}(x^k)\}, \quad j = 1, \dots, q.$$

else

Aumente as penalizações

$$\mu_{[\mathcal{Q}_j]}^{k+1} = \mu_{[\mathcal{Q}_j]}^k$$

$$c_j^{k+1} = \tau_j^k c_j^k$$

$$\alpha^{k+1} = \min_{j=1, \dots, q} c_j^{k+1}$$

if $\alpha^{k+1} > \alpha^k$ **then**

$$\omega^{k+1} = \frac{1}{\alpha^{k+1}},$$

$$\eta^{k+1} = \frac{1}{(\alpha^{k+1})^{\alpha_\eta}}.$$

else

$$\omega^{k+1} = \omega^k \frac{1}{\alpha^{k+1}},$$

$$\eta^{k+1} = \eta^k \frac{1}{(\alpha^{k+1})^{\beta_\eta}}.$$

4.3 RESOLVENDO OS SUB-PROBLEMAS COM BUSCA DIRETA DIRECIONAL

O algoritmo apresentado na seção anterior é bastante prático, no sentido de exigir, na resolução dos sub-problemas, critérios de parada que são usuais em algoritmos de otimização. Isto permitiu a [27] adaptá-lo para utilizar a busca direta direcional na resolução dos sub-problemas.

Em primeiro lugar, é necessário resolver os sub-problemas com a exatidão desejada. Um resultado análogo ao Teorema 9 fornece uma cota superior para $\|P_{K^o(x^k, \epsilon^k)}[-\nabla f(x^k)]\|$ em função do tamanho do passo Δ^k . Dessa maneira, como critério de parada para os sub-problemas, podemos utilizar

$$\Delta^k \leq \delta^k$$

em que δ^k faz o papel de ω^k .

O mesmo pode ser feito com o critério de parada do algoritmo de Lagrangiano aumentado que passa a ser:

$$\delta^k \leq \delta^* \quad \text{e} \quad \|\max(g(x), 0)\| \leq \eta^*.$$

Resta definir a forma de atualizar as tolerâncias δ^k dos sub-problemas que seja equivalente à do algoritmo da seção anterior.

Para tanto, considere $\theta_{tol} \gg 1$, e defina a função

$$\theta(\lambda, c) = \max \left\{ 1, \left(1 + \|\lambda\| + \sum_{p=1}^q c_p \right) / \theta_{tol} \right\}.$$

Em que β_{\max} limita superiormente os comprimentos das direções de busca $\|d_k\| \leq \beta_{\max}$. Considere que a tolerância ϵ_0 utilizada para a detecção de restrições quase-ativas, e consequentemente $K(x, \epsilon)$ e $K^o(x, \epsilon)$, é dada por $\epsilon_0 = \min\{\epsilon_{\max}, \beta_{\max}\Delta\}$.

Então a tolerância para os sub-problemas é atualizada conforme a regra [27]:

$$\delta_{k+1} = \omega^{k+1} / (\beta_{\max} \theta(\lambda^{k+1}, c^{k+1})).$$

4.4 RESOLVENDO OS SUB-PROBLEMAS COM REGIÃO DE CONFIANÇA NÃO-DIFERENCIÁVEL

Para o uso do algoritmo de Lagrangiano aumentado proposto, o Algoritmo 3.2, de região de confiança com restrições lineares pode ser utilizado sem grandes alterações.

O critério de parada usual do algoritmo é o raio da região de confiança, mas vamos propor uma alternativa próxima ao proposto na

seção 4.2. Considere $\hat{\omega}^k \leq \omega^k$. Vamos mostrar que, sob certas condições, podemos considerar um critério de parada baseado no modelo m :

$$\|P_{K^o(x, \omega^k)}[-\nabla m(x)]\| \leq \hat{\omega}^k.$$

Resta mostrar que este novo critério provê a exatidão necessária para a convergência do algoritmo. Se o modelo for Λ -posicionado, de (2.16), temos:

$$\kappa_{eg}\Lambda\Delta \geq \|\nabla\mathcal{L}(x, \mu, c) - \nabla m(x)\| \geq \quad (4.10)$$

$$\geq \|P_{K^o(x, \omega^k)}[\nabla\mathcal{L}(x, \mu, c)] - P_{K^o(x, \omega^k)}[\nabla m(x)]\| \geq \quad (4.11)$$

$$\geq \|P_{K^o(x, \omega^k)}[\nabla\mathcal{L}(x, \mu, c)]\| - \|P_{K^o(x, \omega^k)}[\nabla m(x)]\| \geq \quad (4.12)$$

$$\geq \|P_{K^o(x, \omega^k)}[\nabla\mathcal{L}(x, \mu, c)]\| - \omega^k \quad (4.13)$$

em que utilizamos o fato de que projeções em conjuntos convexos encurtam distâncias [28] e uma desigualdade triangular. Então, o modelo m pode ser utilizado no critério de parada, já que:

$$\|P_{K^o(x, \omega^k)}[\nabla\mathcal{L}(x, \mu, c)]\| \leq \hat{\omega}^k + \kappa_{eg}\Lambda\Delta.$$

Então, se for escolhido $\hat{\omega}^k$ de modo que

$$\omega^k \geq \hat{\omega}^k + \kappa_{eg}\Lambda\Delta \quad (4.14)$$

a cota 4.8 é satisfeita. O valor de Λ pode não ser conhecido, mas é finito. Já o valor de κ_{eg} depende do condicionamento da função $\mathcal{L}(\cdot, \mu, c)$, que depende da penalização c . Se os multiplicadores λ estiverem convergindo para seus valores corretos, os parâmetros de penalização c permanecerão limitados. Nesse caso, com um Δ suficientemente pequeno (ou decrescente, a fim de que $\kappa_{eg}\Lambda\Delta \rightarrow 0$) este critério de parada alternativo é válido.

Caso isto não ocorra, e c crescer indefinidamente, este critério não pode ser usado. Nesta situação, porém, a dificuldade está em fazer que os modelos m sejam boas aproximações da função Lagrangiano aumentado $\mathcal{L}(\cdot, \lambda, c)$. Então, o algoritmo todo terá dificuldade em convergir e mesmo o critério de parada usual (baseado apenas no raio da região) pode apresentar problemas.

4.5 SUMÁRIO

Neste capítulo apresentamos o método do Lagrangiano aumentado para a resolução de problemas de otimização com restrições não-lineares. O método substitui restrições do problema original em penalizações para a função objetivo.

A função lagrangiano aumentado inclui penalizações para as violações de restrições e estimativas para os multiplicadores de Lagrange do problema original. O método consiste em resolver uma sequência de subproblemas de otimização do Lagrangiano aumentado, atualizando-se as estimativas dos multiplicadores de Lagrange e, eventualmente aumentando as penalizações associadas às restrições.

O método, inicialmente para tratar problemas com restrições de igualdades, já foi estendido para o tratamento de desigualdades, neste capítulo apresentamos o caso unilateral, mas também é possível tratar desigualdades como $l \leq Ax \leq u$ sem a necessidade de introdução de mais multiplicadores.

Por fim, apresentamos um algoritmo desta classe em que as restrições lineares não são incluídas como penalizações, podendo ser tratadas explicitamente na otimização do Lagrangiano aumentado. Mostramos como os sub-problemas podem ser resolvidos utilizando os algoritmos não-diferenciáveis do capítulo anterior.

5 ANÁLISE COMPUTACIONAL

Neste capítulo fazemos uma análise do desempenho dos algoritmos de busca direta direcional e região de confiança não-diferenciável para a resolução de problemas com restrições lineares nas variáveis. Especificamente, problemas envolvendo a alocação de gás, sujeitos a uma disponibilidade limitada de gás para *gas-lift*.

Primeiramente, apresentamos um conjunto de problemas que foi resolvido empregando-se os dois métodos, com os dados de [29]. Em seguida, apresentamos a resolução de um problema em que o método de região de confiança empregou diretamente o simulador para a avaliação da função objetivo.

5.1 OTIMIZAÇÃO DE FUNÇÃO SUAVE

O problema resolvido na presente seção consiste em alocar gás de *gas-lift* para um conjunto de poços, com restrições lineares

$$\begin{aligned} \max \quad & f(q_{\text{inj}}) \\ \text{sujeito a} \quad & Aq_{\text{inj}} \leq b \end{aligned}$$

em que $q_{\text{inj}} = (q_{\text{inj}}^1, \dots, q_{\text{inj}}^N)$ é o vetor de vazões alocadas de gás de elevação,

$$A = \begin{bmatrix} I_N \\ -I_N \\ e_{1 \times N} \end{bmatrix}$$

$b = (u_1, \dots, u_N, -l_1, \dots, -l_N, q_{\text{max}})$, I_N é a matriz identidade de ordem N , e $e_{1 \times N}$ é um vetor linha com todos os elementos iguais a 1. Os valores das restrições foram $u_1 = \dots = u_N = 4000$ Mscf/d e $l_1 = \dots = l_N = 80$ Mscf/d. Foram resolvidos 10 casos diferentes do problema, com a disponibilidade total de gás do campo variada logaritmicamente de um valor muito restrito $q_{\text{max}} = 2.800$ Mscf/d, até $q_{\text{max}} = 28.000$ Mscf/d, em que o ótimo irrestrito se torna viável. Desta maneira obtivemos dez diferentes versões do problema. Ainda, para cada versão, resolvemos o problema partindo de 10 diferentes pontos iniciais.

A função f , como já apresentado, é da forma

$$f = \sum_{n=1}^N (p_o q_o^n(q_{\text{inj}}^n) + p_g q_g^n(q_{\text{inj}}^n) - p_w q_w^n(q_{\text{inj}}^n)) - \sum_{n=1}^N p_{\text{inj}} q_{\text{inj}}^n$$

neste caso, porém, utilizamos a aproximação de [14] para modelar a produção total de cada poço:

$$q_p^n(q_{\text{inj}}^n) = c_1^n + c_2^n q_{\text{inj}}^n + c_3^n (q_{\text{inj}}^n)^2 + c_4^n \ln(q_{\text{inj}}^n + 1).$$

Os parâmetros c_1, \dots, c_4 , foram identificados de um problema [3], adaptado em [2], utilizando mínimos quadrados lineares, e estão listados na Tabela 5.1. Os valores das vazões de óleo q_o^n , gás q_g^n e água q_w^n produzidas foram calculados, para cada poço, utilizando os valores de GOR (relação gás-óleo) e *Water cut* (proporção de água na fase líquida) fixos, conforme a Tabela 5.2.

Para cada um dos cenários propostos, a disponibilidade total de gás para *gas-lift* e o valor ótimo da função objetivo são dados na Tabela 5.3. O ponto ótimo foi calculado numericamente, utilizando-se um algoritmo baseado em derivadas.

Desta maneira, podemos testar o desempenho dos algoritmos para funções suaves (em que há garantia teórica de convergência) mantendo a estrutura do problema de interesse.

5.1.1 Busca direta direcional

No algoritmo de busca direta direcional, na iteração k , o conjunto de direções de busca \mathcal{D}_k gera $K^o(x, \epsilon)$ para $0 < \epsilon < \epsilon_c$, com $\epsilon_0 = 0,5$. Para tanto, utilizamos o procedimento do Teorema 8, colocando os geradores dos cones $K(x, \epsilon)$ como as colunas de uma matriz V . Para o cálculo dos vetores u_i mencionados no Teorema utilizamos uma base ortogonal que gera o espaço nulo de V^T por combinações lineares e a completamos com os negativos de todas as direções, como em (2.5), de modo a ter uma base positiva. A estes vetores acrescentamos os w_i , colunas de $V(V^T V)^{-1}$, com seus negativos $-w_i$, de modo a ter um conjunto \mathcal{D}_k satisfazendo a Condição 2 (p. 50).

Como função forçante, que determina o descenso mínimo para a aceitação do passo, foi utilizada $\varrho(t) = \frac{1}{4}t^2$. O parâmetro de controle do tamanho do passo foi iniciado em $\Delta_0 = 1$. O critério de parada da busca direta direcional é dado pelo tamanho deste parâmetro, enquanto que o critério usado pelo método de região de confiança sem derivadas é o raio da região. Para obtermos critérios de parada equivalentes, ambos os métodos foram aplicados ao problema com um critério de parada exigente, de modo a satisfazer com folga à exatidão desejada, de erro menor que 1 em cada coordenada. A partir daí, foram derivados os critérios de parada de um e de outro algoritmo. Para o de busca direta direcional,

$$\Delta_k < \Delta_{tol} = 0,0038.$$

Na Tabela 5.4 apresentamos um sumário das resoluções dos problemas, considerando, para cada cenário, as médias dos tempos e dos números de avaliações das funções objetivos. Na segunda parte da tabela, tendo em vista que o valor ótimo q_{inj}^* é conhecido (Tabela 5.3),

Tabela 5.1: Parâmetros do problema resolvido. $q_p^n(q_{\text{inj}}^n) = c_1^n + c_2^n q_{\text{inj}}^n + c_3^n (q_{\text{inj}}^n)^2 + c_4^n \log(q_{\text{inj}}^n + 1)$.

Poço	c_1	c_2	c_3	c_4
1	-1081	-0,2559	$1,400 \times 10^{-5}$	398,6
2	-1132	-0,2092	$-8,202 \times 10^{-6}$	471,5
3	-1357	-0,3079	$6,750 \times 10^{-7}$	559,5
4	-1191	-0,2580	$2,236 \times 10^{-6}$	476,7
5	-1297	-0,3074	$1,687 \times 10^{-5}$	478,4
6	-1357	-0,2505	$-9,927 \times 10^{-6}$	565,5
7	-1629	-0,3698	$9,045 \times 10^{-7}$	671,5
8	-1282	-0,2908	$6,551 \times 10^{-6}$	504,7
9	-1791	-0,4068	$9,881 \times 10^{-7}$	738,5
10	-594,0	-0,1285	$9,991 \times 10^{-7}$	238,1
11	-1084	-0,2482	$6,276 \times 10^{-6}$	425,4
12	-1433	-0,3254	$7,900 \times 10^{-7}$	590,8
13	-1151	-0,2798	$1,952 \times 10^{-5}$	413,3
14	-1058	-0,0749	$-3,558 \times 10^{-5}$	443,7
15	-1059	-0,1566	$-1,931 \times 10^{-5}$	494,3
16	-1266	-0,2569	$2,363 \times 10^{-6}$	489,8
17	-1293	-0,3010	$1,745 \times 10^{-5}$	474,9
18	-1121	-0,1566	$-2,101 \times 10^{-5}$	517,7
19	-1182	-0,1893	$-2,440 \times 10^{-5}$	534,5
20	-1019	-0,1601	$-1,402 \times 10^{-5}$	445,2
21	-1725	-0,3515	$-1,706 \times 10^{-5}$	724,5
22	-478,7	-0,0871	$-3,015 \times 10^{-6}$	214,3
23	-843,3	-0,1463	$-1,170 \times 10^{-5}$	377,5
24	-1072	-0,1564	$-2,598 \times 10^{-5}$	517,6
25	-1121	-0,1566	$-2,101 \times 10^{-5}$	517,7
26	-1058	-0,0749	$-3,558 \times 10^{-5}$	443,7
27	-1059	-0,1566	$-1,931 \times 10^{-5}$	494,3
28	-1266	-0,2569	$2,363 \times 10^{-6}$	489,8
29	-1191	-0,2580	$2,236 \times 10^{-6}$	476,7
30	-1297	-0,3074	$1,687 \times 10^{-5}$	478,4
31	-1357	-0,2505	$-9,927 \times 10^{-6}$	565,5
32	-1629	-0,3698	$9,045 \times 10^{-7}$	671,5

Tabela 5.2: Valores de GOR e *Water cut* dos poços considerados

Poço	GOR	<i>Water cut</i> (%)
1	0.286	12,5
2	0.227	9,6
3	0.385	13,3
4	0.308	18,7
5	0.500	14,3
6	0.150	9,1
7	0.462	7,1
8	0.212	3,6
9	0.417	20,0
10	0.329	9,1
11	0.385	13,3
12	0.267	6,3
13	0.282	11,9
14	0.208	8,6
15	0.232	4,5
16	0.235	12,3
17	0.269	5,6
18	0.317	6,4
19	0.278	12,7
20	0.333	6,1
21	0.161	10,2
22	0.137	11,8
23	0.194	15,0
24	0.262	5,6
25	0.431	9,7
26	0.276	31,0
27	0.281	22,0
28	0.088	1,1
29	1.429	30,0
30	1.000	33,3
31	0.167	33,3
32	0.750	42,9

Tabela 5.3: Cenários propostos, com valores das soluções.

Cenário	q_{\max}	$f(q_{\text{inj}}^*)$
1	$2,80 \times 10^3$	$4,36 \times 10^5$
2	$3,62 \times 10^3$	$4,87 \times 10^5$
3	$4,67 \times 10^3$	$5,35 \times 10^5$
4	$6,03 \times 10^3$	$5,79 \times 10^5$
5	$7,79 \times 10^3$	$6,21 \times 10^5$
6	$1,01 \times 10^4$	$6,57 \times 10^5$
7	$1,30 \times 10^4$	$6,89 \times 10^5$
8	$1,68 \times 10^4$	$7,13 \times 10^5$
9	$2,17 \times 10^4$	$7,28 \times 10^5$
10	$2,80 \times 10^4$	$7,31 \times 10^5$

apresentamos o erro médio entre os valores produzidos ao final do algoritmo com este, tanto no valor da função objetivo ($f(q_{\text{inj}}^*) - f(q_{\text{inj}})$), quanto na distância entre as soluções ($\|q_{\text{inj}}^* - q_{\text{inj}}\|_{\infty}$). Os erros são baixos, tendo-se em consideração os valores ótimos conhecidos na Tabela 5.3.

De modo geral, o caso mais restrito teve uma convergência mais rápida, necessitou de menos avaliações da função objetivo, e induziu erros menores. Provavelmente devido à proximidade entre o ponto inicial e o ponto ótimo.

Na Figura 5.1 mostramos o valor assumido pela função objetivo no iterando corrente durante uma das execuções do algoritmo em três dos cenários apresentados. Os valores da função objetivo foram normalizados, considerando-se 0 o ponto inicial e 1 o ponto ótimo. Como o ponto inicial é variável, a normalização é diferente para cada curva. Percebemos que a evolução do valor é irregular, e fica mais lenta nas proximidades do ponto ótimo.

Para um dos casos, apresentamos a distância Euclidiana ao ponto ótimo, na Figura 5.2, em que verificamos o mesmo comportamento, mais lento próximo ao ponto ótimo.

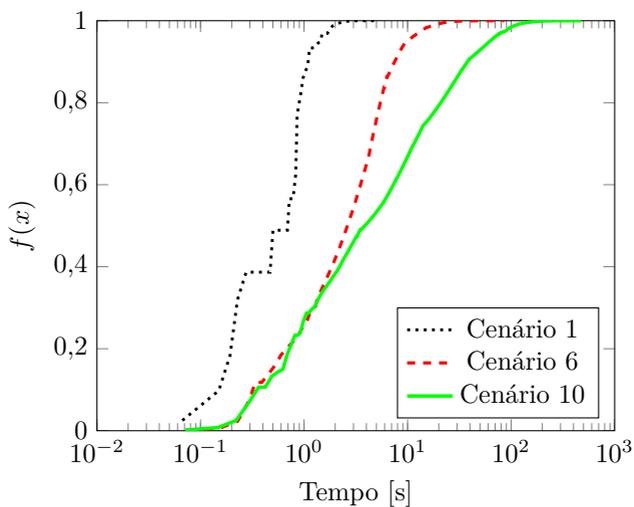


Figura 5.1: Valor da função objetivo durante a busca direta direcional, normalizado de 0 a 1.

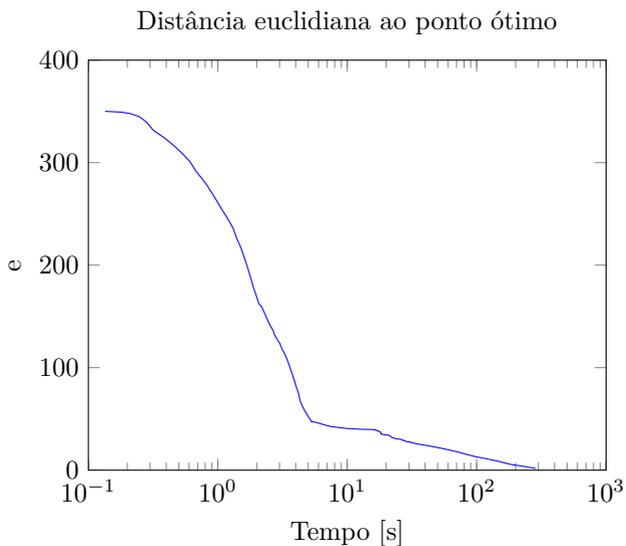


Figura 5.2: Distância Euclidiana entre o ponto ótimo e o iterando corrente da busca direta direcional.

Tabela 5.4: Resolução de todos os cenários usando busca direta direcional. Média dos casos em cada cenário.

Cenário	Tempo	Avaliações f	$f(q_{\text{inj}}^*) - f(q_{\text{inj}})$	$\ q_{\text{inj}}^* - q_{\text{inj}}\ _{\infty}$
1	7s	8.183	4.58×10^{-5}	4.40×10^{-3}
2	14s	22.087	8.66×10^{-5}	6.36×10^{-3}
3	24s	40.629	8.93×10^{-5}	8.77×10^{-3}
4	42s	70.664	1.43×10^{-4}	1.18×10^{-2}
5	67s	116.602	2.79×10^{-4}	2.27×10^{-2}
6	114s	196.331	3.18×10^{-4}	4.37×10^{-2}
7	199s	339.503	5.04×10^{-4}	9.27×10^{-2}
8	306s	517.005	1.05×10^{-3}	1.36×10^{-1}
9	484s	819.350	3.08×10^{-3}	2.98×10^{-1}
10	598s	1.100.179	4.30×10^{-3}	1.88×10^{-1}

5.1.2 Região de confiança

Nesta aplicação, foram consideradas regiões de confiança dadas por bolas na norma ℓ_∞ em torno do iterando:

$$\mathcal{B}_\infty(x_k, \Delta_k) = \{x : \|x - x_k\|_\infty \leq \Delta_k\}.$$

Os modelos usados foram polinomiais de segunda ordem, com matriz Hessiana diagonal H_k :

$$m(x_k + s) = m(x_k) + \frac{1}{2}s^T H_k s + g_k^T s, \quad s \in \mathcal{B}_\infty(x_k, \Delta_k).$$

O raio inicial foi $\Delta_0 = 1$ e para a aceitação do passo foram usados os coeficientes $\eta_1 = 0,4$ e $\eta_0 = 0$, de modo que, mesmo descenso simples é aceito, contanto que o modelo seja Λ -posicionado.

O raio foi aumentado e diminuído utilizando-se os parâmetros $\gamma_{\text{inc}} = 2$ e $\gamma = 0,5$, respectivamente, que foram mantidos constantes ao longo do algoritmo. O teste de criticidade foi executado com $\hat{q}_k < \epsilon_0 = 32$, usando $\mu = 10$ e $\beta = 9$. Os passos de teste foram calculados resolvendo-se o sub-problema da região de confiança utilizando a parte linear dos modelos.

Como no caso anterior, o critério de parada foi calculado para equivaler em exatidão ao resultado alcançado pela busca direta direcional, na seção anterior. Chegamos ao critério

$$\Delta_k < \Delta_{\text{tol}} = 0,0156.$$

Apresentamos um sumário da resolução dos problemas na Tabela 5.5. Para cada cenário é apresentada a média entre todos os pontos iniciais. Tanto o tempo como o número de avaliações da função objetivo foram pouco sensíveis à restrição de disponibilidade de gás para *gas-lift*.

Na Figura 5.3 apresentamos a evolução do valor corrente da função objetivo, normalizado de 0 a 1. É mostrado apenas o caso mediano de três dos cenários. Dos três, o caso mais restrito foi o mais rápido de ser resolvido, seguido do menos restrito. O caso intermediário acabou levando mais tempo. Nesta figura, ainda, para o Cenário 1, é possível notar um período em que a função objetivo fica num mesmo valor. Podemos especular que em tal momento foram necessárias várias execuções do algoritmo de melhora de modelo, ou do passo de criticidade, que podem ser custosos e, em si, não trazem melhora na função objetivo.

A distância euclidiana ao ponto ótimo durante as iterações de um dos casos é mostrada na Figura 5.4, em um gráfico linear. O algoritmo chega rapidamente na proximidade do ponto ótimo, onde é gasta a maior parte do tempo.

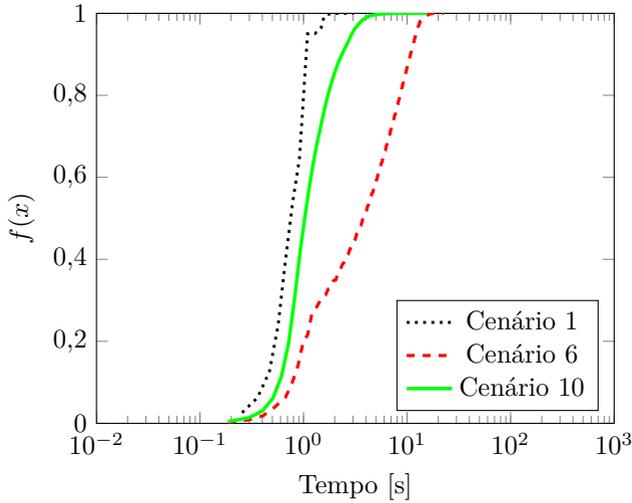


Figura 5.3: Valor da função objetivo durante a execução do algoritmo de região de confiança, normalizado de 0 a 1.

Tabela 5.5: Resolução de todos os cenários usando região de confiança. Média dos casos em cada cenário.

Cenário	Tempo	Avaliações f	$f(q_{inj}^*) - f(q_{inj})$	$\ q_{inj}^* - q_{inj}\ _{\infty}$
1	4s	755	2.33×10^{-3}	3.37×10^{-2}
2	17s	4.602	2.10×10^{-2}	2.73×10^{-1}
3	19s	5.327	9.39×10^{-3}	2.12×10^{-1}
4	27s	7.712	7.66×10^{-3}	2.58×10^{-1}
5	28s	7.725	4.67×10^{-3}	2.68×10^{-1}
6	23s	6.447	2.20×10^{-3}	2.31×10^{-1}
7	19s	5.287	9.62×10^{-4}	1.93×10^{-1}
8	18s	4.995	6.25×10^{-4}	2.19×10^{-1}
9	17s	4.636	3.34×10^{-4}	1.95×10^{-1}
10	16s	4.567	3.79×10^{-4}	2.40×10^{-1}

Comparando com a busca direta direcional, há uma grande redução no tempo de execução e, principalmente, no número de avaliações da função objetivo. Essa redução ficará mais relevante quando tais funções forem obtidas como resultado de um cálculo de simulador.

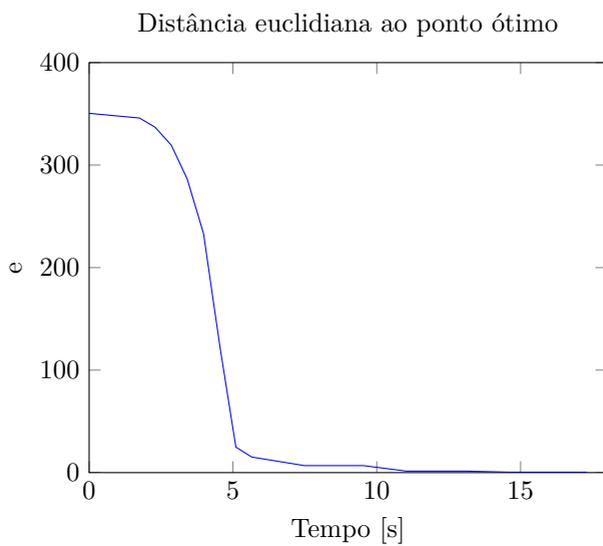


Figura 5.4: Distância Euclidiana entre o valor ótimo encontrado e o valor do iterando corrente durante a execução do algoritmo de região de confiança.

5.2 OTIMIZAÇÃO BASEADA NO SIMULADOR

Na seção anterior aplicamos os métodos de otimização estudados para a otimização em um campo de petróleo em que a produção era modelada por funções suaves. Naquele caso, a teoria relacionada aos métodos era capaz de garantir a convergência.

Na presente seção faremos a otimização da alocação de gás em um campo de produção de petróleo utilizando diretamente um simulador para os poços e sistemas relacionados. Não apenas o modelo é mais completo, mas também há presença de ruído, proveniente do cálculo das simulações, realizado iterativamente. Como a utilização do simulador torna a resolução mais lenta, optamos por realizar os experimentos apenas com o método de região de confiança.

Consideramos o cenário de [7], com a diferença que sua abordagem envolvia sofisticados modelos linearizados por partes, enquanto nossa será com um simulador numérico.

O campo de produção consiste de 16 poços (Figura 5.5), com a produção direcionada para dois *manifolds*, e cada *manifold* direciona sua produção a um separador. Os poços numerados de 1 a 8 localizam-se a 1 km do *manifold* 1, e 10 km do *manifold* 2, enquanto os poços de 9 a 16, estão a 1 km do *manifold* 2 e 10 km do *manifold* 1.

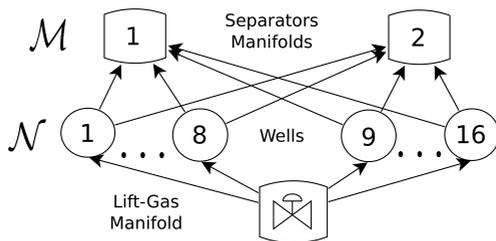


Figura 5.5: O gás para *gas-lift* deve ser alocado a 16 poços, cuja produção é distribuída entre 2 *manifolds* [7].

Como nosso estudo não é capaz de otimizar o roteamento entre poços e *manifolds*, mantivemos este constante, conforme a Tabela 5.6.

A tubulação ligando o *manifold* 1 ao seu separador tem 100 m, enquanto a tubulação entre o *manifold* 2 e seu separador tem 50 m, com diâmetro interno de 4,5 in e rugosidade absoluta de $R = 0,001$ in.

Os poços têm tubos de produção (*tubings*) de diâmetro interno de 3 in, comprimento total de perfuração de 3,7 km, profundidade de 2,7 km e ponto de injeção a 2,7 km.

Tabela 5.6: Alinhamento entre poços e *manifolds*

Poço	Manifold
1	1
2	2
3	1
4	1
5	2
6	1
7	1
8	2
9	2
10	2
11	2
12	2
13	1
14	1
15	2
16	2

Apresentamos na Tabela 5.7 algumas características dos poços produtores. Também estão indicados a pressão estática do reservatório p_r e o índice de produtividade p_i (*productivity index*). O modelo deste campo foi simulado utilizando o software PIPESIM.

O modelo deste campo foi simulado utilizando um simulador fenomenológico. Sempre que a função objetivo precisava ser avaliada, era feita uma simulação numérica da produção do campo.

O problema resolvido foi o seguinte:

$$\begin{aligned} \max \quad & f = \sum_{n=1}^N (p_o q_o^n + p_g q_g^n - p_w q_w) - \sum_{n=1}^N p_{inj} q_{inj}^n \\ \text{sujeito a} \quad & l \leq q_{inj}^n \leq u \\ & \sum_{n=1}^N q_{inj}^n \leq q_{inj}^{\max}, \end{aligned}$$

em que as vazões de óleo q_o , gás q_g e água q_w de cada poço são calculadas pelo simulador, e a vazão de gás para gas-lift q_{inj} está limitada, em cada poço de $l = 0$ até $u = 226.534 \text{ Nm}^3/\text{d}$ e $N = 16$. A disponibilidade total de gás do campo utilizada foi $q_{inj}^{\max} = 8 \times u = 1.812.278 \text{ Nm}^3/\text{d}$. Utilizamos os preços de óleo $p_o = 20$, e gás $p_g = 2$. O custo do tratamento da

Tabela 5.7: Características dos poços

Poço	GOR (Nm ³ /Nm ³)	Water cut (%)	p_r (psi a)	p_i (STB/d/psi)
1	200	0	2100	15
2	200	20	2300	2
3	300	10	1950	12
4	300	40	2050	15
5	400	0	1750	4
6	400	20	1700	9
7	500	10	1700	11
8	500	40	2100	10
9	200	10	1900	5
10	200	40	2200	9
11	300	0	1850	11
12	300	20	2300	6
13	400	10	1825	14
14	400	40	2200	7
15	500	0	1600	8
16	500	20	1800	5

água foi $p_w = 1$ e o custo da injeção de gás foi $p_{inj} = 5$.

Para a resolução, implementamos o algoritmo de região de confiança em Matlab, que foi usado para gerar código binário, que faz uso do simulador.

Como na seção anterior, utilizamos uma região de confiança dada por uma bola na norma ℓ_∞ em torno do iterando:

$$\mathcal{B}_\infty(x_k, \Delta_k) = \{x : \|x - x_k\|_\infty \leq \Delta_k\}.$$

Os modelos usados foram de interpolação, polinomiais lineares:

$$m(x_k + s) = m(x_k) + g_k^T s, \quad s \in \mathcal{B}_\infty(x_k, \Delta_k).$$

O raio inicial foi $\Delta_0 = 1$, o raio máximo foi $\Delta_{\max} = 16.384$ e o critério de parada, $\Delta < 10^{-3}$.

Para a aceitação do passo usamos os coeficientes $\eta_1 = 0,1$ e $\eta_0 = 0$. Os parâmetros de aumento e contração do raio foram os mesmos em todas as iterações: $\gamma_{\text{inc}} = 2$ e $\gamma = 0,5$. O limiar para a substituição de pivôs durante o algoritmo de melhora de modelo foi $\xi = \frac{1}{16}$.

Desta vez, para o cálculo do passo, resolvemos o problema de otimização dentro da região de confiança, utilizando a função *linprog*, do Matlab, com o algoritmo *active-set*.

Tabela 5.8: Sumário da resolução do problema: A maior parte do tempo foi gasta nas simulações

Pto. inicial	n.º simulações	Tempo simulação	Tempo algoritmo
1	1383	2 h 18 min	22 s
2	1294	2 h 22 min	22 s
3	1166	2 h 5 min	21 s
4	1190	2 h 6 min	22 s
5	1242	2 h 12 min	21 s
6	1253	2 h 15 min	21 s
7	1418	2 h 38 min	16 s
8	1132	1 h 60 min	15 s
9	1110	2 h 4 min	16 s
10	1075	1 h 57 min	18 s
11	1002	1 h 48 min	13 s
12	1080	1 h 58 min	15 s

Resolvemos o problema a partir de 15 diferentes pontos iniciais, escolhidos aleatoriamente. Em 12 dos casos, o algoritmo convergiu, aproximadamente, para um mesmo ponto, sendo \bar{x} a média deles, com valor $f(\bar{x}) = 2.136.131$ para a função objetivo. Com relação à média \bar{x} , o desvio padrão da distância dos pontos finais a ela $\|x - \bar{x}\|_2$ foi $19,2 \text{ Nm}^3/\text{d}$, enquanto que $\|\bar{x}\|_2 = 1.258 \text{ Nm}^3/\text{d}$. Considerando todos os valores encontrados para a função objetivo, o desvio padrão foi de 0,69.

Considerando estes 12 pontos iniciais, apresentamos alguns detalhes da resolução na Tabela 5.8: número de simulações feitas, tempo gasto com simulações e tempo gasto no restante do algoritmo. A maior parte do tempo foi gasta com as simulações, que foram executadas sequencialmente. O tempo gasto pelo restante do algoritmo, em si, foi de 22s, no pior caso.

Na tabela 5.9 fazemos um sumário da exatidão das soluções encontradas. Para tanto, consideramos \hat{x} o melhor ponto encontrado entre todos os cenários. Na tabela estão os desvios absolutos, com relação a tal ponto, e relativos, comparando-se com a norma deste ponto ou com a distância entre o melhor ponto \hat{x} e o ponto inicial de cada caso.

Na Figura 5.6 mostramos o valor corrente da função objetivo durante a execução do algoritmo, para dois pontos iniciais diferentes, em função do número de simulações feitas.

No entanto, em 3 dos 15 pontos, o algoritmo não foi capaz de se afastar muito do ponto original. As previsões dos modelos não foram

Tabela 5.9: Sumário da resolução do problema: exatidão das soluções encontradas.

Pto. inicial	$\ x - \hat{x}\ $	$\frac{\ x - \hat{x}\ }{\ \hat{x}\ }$	$\frac{\ x - \hat{x}\ }{\ \hat{x} - x^0\ }$	$f(\hat{x}) - f(x)$	$\frac{f(\hat{x}) - f(x)}{f(\hat{x})}$
1	0	0	0	0	0
2	3.87	0.304×10^{-2}	0.171×10^{-4}	0.182	0.085×10^{-6}
3	3.87	0.304×10^{-2}	0.462×10^{-4}	0.182	0.085×10^{-6}
4	3.87	0.304×10^{-2}	0.249×10^{-4}	0.182	0.085×10^{-6}
5	3.87	0.304×10^{-2}	0.259×10^{-4}	0.182	0.085×10^{-6}
6	3.87	0.304×10^{-2}	0.206×10^{-4}	0.182	0.085×10^{-6}
7	21.77	1.711×10^{-2}	1.179×10^{-4}	0.720	0.337×10^{-6}
8	31.12	2.445×10^{-2}	1.626×10^{-4}	1.016	0.476×10^{-6}
9	32.05	2.519×10^{-2}	1.639×10^{-4}	1.017	0.476×10^{-6}
10	26.48	2.081×10^{-2}	1.480×10^{-4}	1.272	0.596×10^{-6}
11	33.67	2.646×10^{-2}	1.607×10^{-4}	1.276	0.597×10^{-6}
12	51.71	4.064×10^{-2}	4.262×10^{-4}	2.315	1.084×10^{-6}

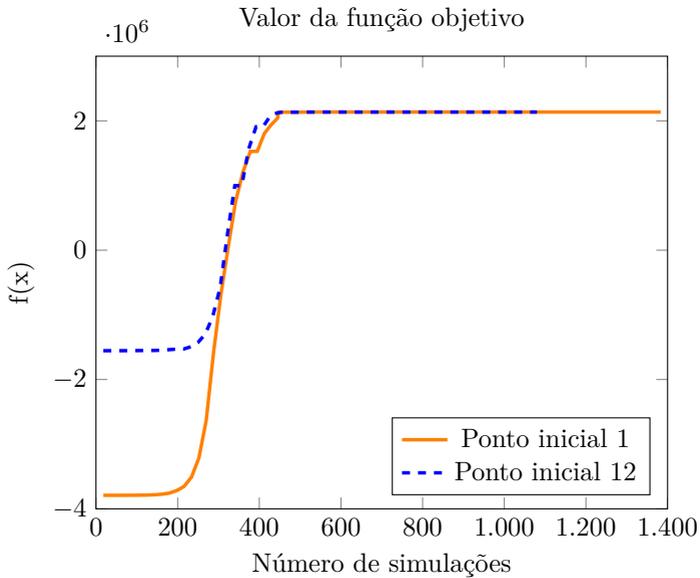


Figura 5.6: Valor da função objetivo ao longo da execução da otimização

boas o suficiente para o raio da região crescer muito e foi feito pouco progresso.

Notamos que o custo atribuído à injeção de gás acabou sendo muito elevado. Isto fez com que vários dos pontos iniciais acabassem, ao longo da resolução, a quase zerar a injeção de gás para depois voltar a aumentá-la.

Também comparamos a solução obtida com a de [7], que utiliza um modelo linear por partes do simulador. Esta última encontrou um valor de 2.234.150 para a função objetivo. Ao aplicar a solução no simulador, o valor realmente obtido foi 2.155.338. No entanto, esta abordagem também encontrou um alinhamento ótimo entre poços e *manifolds*, que foi diferente daquele que utilizamos durante nossa resolução. Se aplicarmos o resultado que encontramos (de injeção de gás) com o alinhamento de [7], nosso valor para o ganho econômico é um pouco maior: 2.203.409. O tempo de resolução também foi muito diferente: a resolução de [7] durou apenas 55 s, porém com o modelo de otimização construído previamente.

Estas diferenças entre as duas abordagens mostram a relevância do tratamento para os alinhamentos entre poço e *manifold*. Além disso, a estratégia que utiliza modelos lineares por partes pode apresentar imprecisões, seja em função das escolhas as variáveis modeladas, seja no número de pontos utilizados para a construção do modelo.

Uma possibilidade de corrigir essas distâncias entre o resultado obtido pela programação inteira e o resultado obtido com o simulador é utilizar métodos sem derivada como meio de melhorar a solução do resultado da programação inteira.

5.3 SUMÁRIO

Neste capítulo mostramos aplicações dos algoritmos de busca direta direcional e o de região de confiança não-diferenciável. Em um primeiro estudo de caso, ambos foram aplicados na maximização de um ganho econômico, no problema da alocação de gás em um campo produtor de petróleo.

As curvas de produção dos poços eram modeladas por funções suaves e ambos os algoritmos foram capazes de encontrar o valor ótimo para as injeções, sendo que a busca direta direcional necessitou de muito mais avaliações da função objetivo.

No segundo caso, analisamos a otimização diretamente com um simulador de poços de petróleo e sistemas relacionados. Essa possibilidade de usar diretamente os simuladores é um grande motivador para o

uso de métodos não-diferenciáveis. Neste caso, as funções de produção dos poços não eram suaves, tendo em vista o ruído causado pelo simulador numérico. Além disso, há interação entre as produções de diversos poços, tendo em vista que produzem em *manifolds* comuns. Para este caso, foi aplicado apenas o algoritmo de região de confiança. Das 15 tentativas de solução, 12 foram bem-sucedidas e convergiram para a vizinhança de um mesmo ponto.

Comparamos a solução encontrada com a obtida por terceiros com outra abordagem, baseada em programação inteira, que também otimiza o alinhamento entre poços e *manifolds*. A abordagem com programação inteira faz uso de um modelo específico para otimização, construído previamente, tendo um tempo para solução bastante inferior à nossa, que utiliza diretamente o simulador.

Os métodos que estudamos não resolvem a questão do roteamento da produção, enquanto que a abordagem que utiliza modelos lineares por partes pode ter erros de modelação. Uma possibilidade é utilizar os métodos não-diferenciáveis como uma alternativa para melhorar o resultado obtido com programação inteira.

6 CONCLUSÃO

Na indústria do petróleo há várias ferramentas de simulação numérica dos processos de produção. No entanto, os simuladores normalmente não fornecem derivadas das variáveis calculadas, o que impede seu uso para otimização com algoritmos clássicos. Neste trabalho apresentamos dois métodos de otimização que não fazem uso de derivadas: busca direta direcional e região de confiança não-diferenciável. Eles foram expostos em sua forma irrestrita e com restrições lineares. Ambos foram utilizados em um estudo numérico em que se otimizou a produção de um campo que produz petróleo por *gas-lift*. Para o tratamento de restrições não-lineares, propomos o método do Lagrangiano aumentado, que pode fazer uso dos métodos anteriores para resolver a sequência de sub-problemas resultante. Neste método, as restrições não-lineares são substituídas por penalizações na função objetivo, restando apenas as restrições lineares.

Os algoritmos de busca direta direcional são de implementação mais simples, enquanto os de região de confiança demandam mais trabalho. Ambos têm garantia de convergência, dependendo da suavidade da função que se está otimizando. Os algoritmos também podem fazer uso de ferramentas de simulação numérica diretamente, embora neste caso não se seja possível verificar a suavidade da função.

Nos estudos numéricos que realizamos, o algoritmo de otimização por região de confiança não-diferenciável foi mais eficiente que o de busca direta direcional, chegando ao ponto ótimo em menos tempo e com menos avaliações da função objetivo. Também fizemos um estudo com o algoritmo de região de confiança e um simulador de redes de petróleo. A maioria do tempo de solução foi gasta no cálculo da função objetivo (simulações), e não no algoritmo em si.

Em nossos estudos, para ambos os algoritmos as avaliações de função objetivo foram feitas sequencialmente. O desempenho pode ser melhorado se várias avaliações forem feitas em paralelo. Isto é particularmente marcante para o algoritmo de busca direta direcional, que utiliza várias avaliações por iteração. Em particular, a abordagem de busca direta direcional tem um algoritmo que pode ser utilizado de forma distribuída [30], permitindo a execução paralela e com certo balanço de carga.

No caso de alocação de gás para *gas-lift*, se o usuário dispuser de uma boa estimativa para a solução, proveniente seja de conhecimentos de operação, seja do resultado de uma otimização aproximada prévia, o uso de algoritmos não-diferenciáveis pode dar suporte para melhorar tal estimativa.

Os algoritmos apresentados convergem para ótimos locais. Se a otimalidade global for necessária, devem ser utilizados outros métodos. Para o caso em que as restrições são limites superiores e inferiores, existem os algoritmos DIRECT (DIviding RECTangles) [31] e MCS (Multilevel Coordinate Search) [32]. Outra alternativa é recorrer a métodos auxiliares que indiquem um bom ponto inicial para a solução.

Os algoritmos apresentados não resolvem problemas com variáveis inteiras. No contexto de campos de produção de petróleo, estas ocorrem, inevitavelmente, nas decisões sobre o roteamento da produção. Além disso, o problema da alocação de gás utilizando outras estratégias como programação inteira mista pode ser mais eficiente em encontrar o ótimo. Se o número de otimizações a ser feito for alto, ou se o custo de manutenção de um modelo de otimização preciso for baixo, é a melhor abordagem.

O presente estudo apresentou métodos que buscassem fazer uso do conhecimento das restrições lineares. Há métodos semelhantes que tratam apenas limites superiores e inferiores. Neste caso, outras restrições devem ser tratadas com métodos de penalização, como Lagrangiano aumentado. Um estudo comparando tal alternativa com nossa escolha (tratamento explícito de restrições lineares), ainda pode ser feito.

Em trabalhos futuros faremos um estudo numérico do método do Lagrangiano aumentado para o tratamento das restrições não-lineares. No caso específico de alocação de gás para *gas-lift*, tais restrições ocorrem nas produções admissíveis de gás ou contaminantes, mas também na modelação da capacidade de compressão de gás do campo.

Ainda em trabalhos futuros, podemos utilizar os métodos abordados para problemas que não estão sendo resolvidos com programação inteira-mista. Entre esses, incluem-se otimização em reservatórios de petróleo e escoamentos com mudança de fase (*flashing*).

REFERÊNCIAS

- [1] SILVA, T. L. *Formulações inteiras mistas para modelos lineares por partes multidimensionais*. Dissertação (Mestrado) — Programa de Pós-graduação em Engenharia de Automação e Sistemas, Universidade Federal de Santa Catarina, Florianópolis, 2012.
- [2] CODAS, A. *Otimização da produção de poços de petróleo com injeção contínua de gás e alinhamento poço-separador: modelos lineares por partes e algoritmos*. Dissertação (Mestrado) — Programa de Pós-graduação em Engenharia de Automação e Sistemas, Universidade Federal de Santa Catarina, Florianópolis, 2012.
- [3] BUITRAGO, S.; RODRÍGUEZ, E.; ESPIN, D. Global optimization techniques in gas allocation for continuous flow gas lift systems. In: *Proceedings of the Gas Technology Conference*. [S.l.]: Society of Petroleum Engineers, 1996.
- [4] CODAS, A. et al. Integrated production optimization of oil fields with pressure and routing constraints: the Urucu field. *Computers & Chemical Engineering*, v. 46, p. 178–189, 2012. ISSN 0098-1354.
- [5] KOSMIDIS, V. D.; PERKINS, J. D.; PISTIKOPOULOS, E. N. Optimization of well oil rate allocations in petroleum fields. *Industrial & Engineering Chemistry Research*, v. 43, n. 14, p. 3513–3527, 2004.
- [6] MISENER, R.; GOUNARIS, C. E.; FLOUDAS, C. A. Global optimization of gas lifting operations: A comparative study of piecewise linear formulations. *Industrial & Engineering Chemistry Research*, v. 48, n. 13, p. 6098–6104, 2009.
- [7] SILVA, T. L.; CODAS, A.; CAMPONOGARA, E. A computational analysis of convex combination models for multidimensional piecewise-linear approximation in oil production optimization. In: *Proceedings of the 2012 IFAC Workshop on Automatic Control in Offshore Oil and Gas Production*. Trondheim: [s.n.], 2012. v. 1, p. 292–298.
- [8] GUNNERUD, V.; CONN, A.; FOSS, B. Embedding structural information in simulation-based optimization. *Computers & Chemical Engineering*, v. 53, p. 35 – 43, 2013. ISSN 0098-1354.
- [9] CIAURRI, D. E.; ISEBOR, O. J.; DURLOFSKY, L. J. Application of derivative-free methodologies to generally constrained oil production optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimisation*, v. 2, n. 2, p. 134–161, 2011.

- [10] GANZAROLI, C. A. *Modelagem, simulação e controle da dinâmica de poços operando com gas-lift contínuo*. Dissertação (Mestrado) — Programa de Pós-graduação em Engenharia de Automação e Sistemas, Universidade Federal de Santa Catarina, Florianópolis, 2011.
- [11] THOMAS, J. E. (Ed.). *Fundamentos de engenharia de petróleo*. 2. ed. Rio de Janeiro, RJ: Editora Interciência, 2001.
- [12] PLUCENIO, A. *Automação da produção de poços de petróleo operando com elevação artificial por injeção contínua de gás*. Dissertação (Mestrado) — Programa de Pós-graduação em engenharia elétrica, Universidade Federal de Santa Catarina, Florianópolis, 2003.
- [13] KOLDA, T. G.; LEWIS, R. M.; TORCZON, V. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Review*, v. 45, n. 3, p. 385–482, August 2003.
- [14] ALARCÓN, G. A.; TORRES, C. F.; GÓMEZ, L. E. Global optimization of gas allocation to a group of wells in artificial lift using nonlinear constrained programming. *ASME Journal of Energy Resources Technology*, v. 124, n. 4, p. 262–268, 2002.
- [15] DAVIS, C. Theory of positive linear dependence. *American Journal of Mathematics*, The Johns Hopkins University Press, v. 76, n. 4, p. 733–746, 1954. ISSN 00029327.
- [16] CONN, A. R.; SCHEINBERG, K.; VICENTE, L. N. *Introduction to Derivative-Free Optimization*. 1. ed. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2009.
- [17] SCHEINBERG, K.; TOINT, P. L. Self-correcting geometry in model-based algorithms for derivative-free unconstrained optimization. *SIAM Journal on Optimization*, SIAM Publications, Philadelphia, PA, v. 20, n. 6, p. 3512–3532, 2010. ISSN 1052-6234.
- [18] TRÖLTZSCH, A. *Active sets in bound-constrained optimization without derivatives*. Saarbrücken: Lambert Academic Publishing, 2012.
- [19] PARDALOS, P. M.; VAVASIS, S. A. Quadratic programming with one negative eigenvalue is NP-Hard. *Journal of Global Optimization*, Kluwer Academic Publishers, v. 1, p. 15–22, 1991. ISSN 0925-5001.
- [20] CONN, A. R.; GOULD, N. I. M.; TOINT, P. L. *Trust Region Methods*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2000.

- [21] LEWIS, R. M.; TORCZON, V. Pattern search methods for linearly constrained minimization. *SIAM Journal on Optimization*, v. 10, n. 3, p. 917–941, 2000.
- [22] LUCIDI, S.; SCIANDRONE, M.; TSENG, P. Objective-derivative-free methods for constrained optimization. *Mathematical Programming*, Springer Berlin / Heidelberg, v. 92, p. 37–59, 2002. ISSN 0025-5610.
- [23] ABRAMSON, M. A. et al. OrthoMADS: A deterministic MADS instance with orthogonal directions. *SIAM Journal on Optimization*, v. 20, n. 2, p. 948–966, 2009. ISSN 10526234.
- [24] BERTSEKAS, D. P. *Nonlinear Programming*. 2. ed. Belmont, MA: Athena Scientific, 1999.
- [25] BERTSEKAS, D. P. *Constrained optimization and Lagrange multiplier methods*. Belmont, MA: Athena Scientific, 1996.
- [26] CONN, A. R. et al. Convergence properties of an augmented Lagrangian algorithm for optimization with a combination of general equality and linear constraints. *SIAM Journal on Optimization*, v. 6, n. 3, p. 674–703, August 1996.
- [27] KOLDA, T. G.; LEWIS, R. M.; TORCZON, V. *A generating set direct search augmented Lagrangian algorithm for optimization with a combination of general and linear constraints*. Albuquerque, NM, 2006. Disponível em: <<http://www.prod.sandia.gov/cgi-bin/techlib/access-control.pl/2006/065315.pdf>>.
- [28] PHELPS, R. R. Convex sets and nearest points. *Proceedings of the American Mathematical Society*, American Mathematical Society, v. 8, n. 4, p. pp. 790–797, 1957. ISSN 00029939.
- [29] GIULIANI, C. M.; CAMPONOGARA, E.; PLUCENIO, A. A computational analysis of nondifferentiable optimization: Applications to production maximization in gas-lifted oil fields. In: *Proceedings of the 9th IEEE International Conference on Automation Science and Engineering (CASE)*. Madison, WI: [s.n.], 2013. p. 292–297.
- [30] HOUGH, P. D.; KOLDA, T. G.; TORCZON, V. J. Asynchronous parallel pattern search for nonlinear optimization. *SIAM Journal on Scientific Computing*, v. 23, n. 1, p. 134–156, June 2001.

-
- [31] JONES, D. R.; PERTTUNEN, C. D.; STUCKMAN, B. E. Lipschitzian optimization without the lipschitz constant. *Journal of Optimization Theory and Applications*, v. 79, n. 1, p. 157–181, 1993. ISSN 00223239.
- [32] HUYER, W.; NEUMAIER, A. Global optimization by multilevel coordinate search. *Journal of Global Optimization*, v. 14, n. 4, p. 331–355, 1999. ISSN 09255001.
- [33] SPIVAK, M. Significance of the derivative. In: _____. *Calculus*. 3. ed. Houston, TX: Publish or Perish, 1994. cap. 11.

A SOBRE A DIFERENCIABILIDADE DO LAGRANGIANO AUMENTADO

Na seção 4.1.2, apresentamos uma maneira de tratar restrições de desigualdades no problema do Lagrangiano aumentado sem a necessidade de variáveis adicionais.

Para tanto, os sub-problemas resultantes envolviam uma função Lagrangiano aumentado já otimizada nas variáveis de folga, da seguinte maneira:

$$\mathcal{L}(x, \lambda, \mu, c) = f(x) + \lambda^T h(x) + \frac{1}{2} c \|h(x)\|^2 + \frac{1}{2c} \sum_{l=1}^r \{ \max[0, \mu_l + c g_l(x)]^2 - \mu_l^2 \}. \quad (4.4)$$

Na discussão subsequente, tal expressão foi presumida suave. Neste apêndice vamos provar que, de fato, (4.4) é continuamente diferenciável.

Para tanto, a suposição que fazemos é que as funções f , h e g são continuamente diferenciáveis. Diante dessa hipótese, é imediato que os primeiros termos de $\mathcal{L}(\cdot, \lambda, \mu, c)$ são continuamente diferenciáveis. Trataremos de modo separado a parcela que envolve a função “max”.

Para provar sua diferenciabilidade, introduzimos um teorema de [33]:

Teorema 11 *Suponha que a função $f : \mathbb{R} \rightarrow \mathbb{R}$ é contínua no ponto a , e que $f'(x)$ existe para todo x em um intervalo contendo a , exceto, possivelmente para $x = a$. Suponha, ainda, que $\lim_{x \rightarrow a} f'(x)$ existe. Então $f'(a)$ existe, e*

$$f'(a) = \lim_{x \rightarrow a} f'(x).$$

Vamos usar o teorema para mostrar a suavidade da função $p(x) = \max[0, q(x)]^2$, com $q : \mathbb{R} \rightarrow \mathbb{R}$ continuamente diferenciável. Em (4.4) ocorre um caso particular em que $q(x) = \mu_l + c g_l(x)$.

A função p também pode ser escrita como

$$p(x) = \begin{cases} 0, & \text{se } q(x) \leq 0 \\ q(x)^2, & \text{se } q(x) \geq 0. \end{cases}$$

Claramente p é continuamente diferenciável em todos os pontos em que $q(x) < 0$ ou $q(x) > 0$.

Agora suponha que existe um ponto $a \in \mathbb{R}$ tal que $q(a) = 0$. Se, na vizinhança de a não ocorre troca de sinal, por exemplo, $q(x) \geq 0$,

também é imediato que p é continuamente diferenciável. Resta a dúvida com relação àqueles pontos em que há troca de sinal.

Suponha que $q(a) = 0$ e $q(x) > 0$ para $x > a$ e $q(x) < 0$ se $x < a$, na vizinhança de a . Então, o limite, por um lado é

$$\begin{aligned} \lim_{x \rightarrow a^+} p'(x) &= \lim_{x \rightarrow a^+} q(x)^2 = \\ &= \lim_{x \rightarrow a^+} 2q(x)q'(x) = \\ &= 2 \left(\lim_{x \rightarrow a^+} q(x) \right) \left(\lim_{x \rightarrow a^+} q'(x) \right) = \\ &= 2 \times 0 \times q'(a) = \\ &= 0. \end{aligned}$$

Enquanto, pelo outro lado, o limite é

$$\begin{aligned} \lim_{x \rightarrow a^-} p'(x) &= \lim_{x \rightarrow a^-} 0 = \\ &= 0. \end{aligned}$$

Consequentemente,

$$\lim_{x \rightarrow a} p'(x) = 0. \tag{A.1}$$

Como p é contínua, e sua derivada tem limite no ponto a (A.1), pelo Teorema 11, p é continuamente diferenciável em a , com $p'(a) = 0$. O caso recíproco ($q(x) > 0$ se $x > a$ e $q(x) < 0$ se $x < a$) pode ser resolvido de forma análoga.

Assim, o Lagrangiano aumentado proposto (4.4) é composto por uma soma na qual todas as parcelas são continuamente diferenciáveis, e portanto, continuamente diferenciável.