

Uma nova abordagem de Busca Padrão para otimização com restrições lineares

Deise G. Ferreira, Dra. Maria A. D. Ehrhardt

Depto de Matemática Aplicada, IMECC, UNICAMP,
13083-859, Cidade Universitária Zeferino Vaz, Campinas - SP
E-mail: ra070609@ime.unicamp.br, cheti@ime.unicamp.br

Resumo: Neste trabalho voltamos nossa atenção para métodos de otimização que não fazem uso de derivadas. Dentre esses, estamos interessadas em um método de busca padrão para minimização de funções com restrições lineares. Propomos um novo método baseado no algoritmo introduzido por Lewis e Torczon, ao qual incorporamos novas estratégias de busca e atualização do tamanho do passo, além de um novo Padrão de direções de busca. O algoritmo possui resultados de convergência global. Realizamos sua implementação computacional e testes numéricos, de modo a analisar o desempenho do método proposto com o pacote HOPSPACK.

Palavras-chave: Busca Padrão, Otimização sem derivadas, Otimização com restrições lineares

1 Métodos de Busca Padrão para problemas com restrições lineares

Após generalizar os Métodos de Busca Padrão para o problema irrestrito [7], Virginia Torczon e Robert Michael Lewis estendem o método para problemas com restrições lineares [3], definidos da seguinte forma:

$$(PRL) \begin{cases} \min & f(x) \\ \text{s.a} & l \leq Ax \leq u, \end{cases}$$

onde $f: \mathbb{R}^n \rightarrow \mathbb{R}$, $x \in \mathbb{R}^n$, $A \in \mathbb{Q}^{m \times n}$, $l, u \in \mathbb{R}^m$.

Da mesma forma como foi feito para o caso irrestrito, neste trabalho os autores generalizam as principais características do método e apresentam um resultado de convergência global. O método que propomos aqui se enquadra nas características gerais discutidas em [3] e portanto pode ser caracterizado como um Método de Busca Padrão aplicado ao problema com restrições lineares (PRL), possuindo portanto garantias teóricas de convergência.

A ideia geral do método segue os padrões definidos em [7] para o caso irrestrito; a grande diferença é que o ponto inicial deve ser factível, e a cada iteração a viabilidade não pode ser perdida. Assim, ao aceitar um novo iterando, além de exigir que o novo ponto possua o valor da função objetivo menor do que no iterando anterior, o ponto deve ser viável. Dessa forma, ao construir o padrão de direções de busca devemos levar em consideração a geometria da fronteira, de forma a ter a possibilidade de dar passos suficientemente longos sem deixar o conjunto factível.

Quando aplicamos o método ao problema irrestrito, a principal característica do padrão é que as direções de busca gerem positivamente o \mathbb{R}^n .

Para o problema com restrições lineares é preciso considerar informações específicas do problema, uma vez que, quando estamos próximos da fronteira da região viável, o padrão de pontos deve seguir sua geometria. A ideia geral neste caso é que o padrão de direções de busca deverá conter um conjunto de geradores positivos para o cone de direções viáveis. Como o padrão

deve seguir a fronteira da região viável, é natural que modifiquemos as direções de busca a cada iteração, o que é feito de acordo com as restrições ϵ -ativas no ponto corrente.

O método utiliza busca monótona a partir do ponto corrente, procurando um novo ponto que ofereça decréscimo no valor da função objetivo. Porém, para que os resultados de convergência sejam garantidos, existem certos critérios que devem ser satisfeitos ao realizarmos a busca, o que chamamos de hipóteses sobre os movimentos exploratórios.

Nenhuma condição de decréscimo suficiente é exigida, é necessário apenas que o passo aceito forneça um decréscimo simples no valor da função objetivo e seja da forma $s_i^k = \alpha_k c_i^k$, ou seja, o tamanho do passo é fixo na iteração corrente e as direções devem pertencer ao padrão P^k , e além disso o novo ponto deve permanecer viável. Antes que o fracasso seja declarado, todas as direções do padrão devem ser testadas e somente se nenhuma delas oferecer decréscimo à função objetivo o fracasso é declarado.

Se ao invés de tomarmos qualquer uma das direções que ofereça decréscimo para o valor da função objetivo, exigirmos que a direção tomada ofereça o maior decréscimo possível entre as direções contidas no padrão, é possível obter um resultado de convergência mais forte. A principal mudança no ponto de vista prático é que, em toda iteração, devemos necessariamente testar todas as direções do padrão para procurar aquela que oferece o maior decréscimo, enquanto que pela hipótese fraca, assim que obtemos um ponto melhor, podemos encerrar a iteração e aceitar o passo.

Quanto ao tamanho do passo, este deve ser atualizado de acordo com o status da iteração, podendo aumentar ou ser mantido em caso de sucesso e devendo necessariamente diminuir em caso de fracasso. Os critérios de atualização, no entanto, não são livres e devem satisfazer certas condições que garantem convergência. Todas as hipóteses necessárias e os resultados de convergência podem ser obtidos em [3].

2 Apresentação do Método

Apresentamos um método que se encaixa nas hipóteses de convergência discutidas em [3]. Propomos novas estratégias de busca e de atualização do tamanho do passo, além de um novo padrão de direções. Nossas propostas se mostraram eficientes na prática proporcionando robustez ao método.

2.1 Estratégias de Busca

Adotamos uma estratégia que se mostrou eficiente para nossos objetivos, tanto no sentido de manter a viabilidade quanto em robustez. O que fazemos é testar a viabilidade do ponto antes de calcular o valor da função objetivo no mesmo. Sendo assim, somente passos viáveis são aceitos. Deste modo, a viabilidade sempre é preservada, além de evitar avaliações de função desnecessárias.

No caso de robustez observamos que muitas vezes, devido às características do problema, as direções de busca geradas a partir das restrições ϵ -ativas não funcionam bem. Porém, adicionando algumas direções ao padrão, o método volta a funcionar bem. Portanto, como estamos testando a viabilidade do passo, podemos adicionar essas direções sem correr o risco de dar passos ineficazes.

Nesta abordagem, em um primeiro momento não nos preocupamos com as restrições de caixa. Logo, ao gerar o padrão de direções, trabalhamos com um conjunto reduzido de restrições e somente ao testar a viabilidade do passo fiscalizamos se todas as restrições estão sendo satisfeitas, inclusive as de caixa.

2.2 O passo de busca

Realizamos uma variante da busca completa que consiste em procurar, entre as direções do padrão, aquela que oferece o maior decréscimo no valor da função objetivo. Isso quer dizer que,

ao encontrar uma direção que oferece decréscimo no valor da função objetivo, não finalizamos a busca. A variante que utilizamos consiste em continuar a busca, mas não a partir do ponto inicial considerado, mas a partir deste ponto que possui menor valor de função objetivo. Para garantir que estamos andando na direção que oferece o maior decréscimo, todas as direções do padrão devem ser testadas.

Observe que esta estratégia de busca satisfaz as hipóteses fortes sobre os movimentos exploratórios, e portanto a teoria de convergência garante que todo ponto limite da sequência convergente gerada pelo método é um ponto KKT do problema.

2.3 Atualização do tamanho do passo

Obedecendo aos critérios obrigatórios para garantir a convergência, propomos uma estratégia para atualizar o tamanho do passo que leva em consideração as iterações bem-sucedidas consecutivas.

Criamos a variável *suc* que é incrementada sempre que a iteração é bem-sucedida, e quando a iteração resulta em fracasso atribuímos zero a esta variável, que volta a ser incrementada somente quando o método obtiver sucesso novamente.

Então a atualização do tamanho do passo segue o algoritmo abaixo:

Algoritmo 2.1. Atualização do tamanho do Passo

Dados $\alpha_k > 0$, $\tau > 1$, $suc \geq 0$.

1. Se $suc > 0$ (sucesso) faça: $\alpha_{k+1} = \min\{\alpha_{max}, \tau^{suc}\alpha_k\}$.
2. Se $suc = 0$ (fracasso) faça: $\alpha_{k+1} = 0.5\alpha_k$.

Adotamos essa estratégia pensando na possibilidade de dar passos mais ambiciosos quando o método está indo bem. É importante ressaltar também que estabelecemos um limite para o tamanho do passo, que não pode crescer indefinidamente, uma vez que a sequência $\{\alpha_k\}$ deve ir para zero e o tamanho de passo máximo aceitável pode ser definido de acordo com as características do problema.

A estratégia utilizada em caso de fracasso é a usual, ou seja, apenas dividimos o tamanho do passo por 2, porém nada impede de que adotemos uma estratégia que leve em consideração os fracassos consecutivos.

2.4 O Padrão de direções

A escolha do conjunto de direções de busca adequadas para o tipo de problema com o qual estamos trabalhando é a chave para o bom desempenho do método. Discutiremos brevemente diferentes formas de gerar o padrão de direções, considerando as características específicas das restrições que constituem o problema, ou mais especificamente o conjunto de trabalho na iteração corrente.

A construção do padrão P^k consiste basicamente em construir um conjunto gerador positivo para o cone $K^\circ(x, \epsilon)$, ilustrado na Figura 1. Maiores detalhes sobre a construção do cone K e do conjunto de restrições ϵ -ativas podem ser obtidos em [1, 3].

2.4.1 Minimização em Caixas

O problema de minimizar em uma caixa é um caso particular do problema de minimização com restrições lineares, onde a matriz de restrições A é igual a matriz identidade. Neste caso, é fácil ver que o cone de direções viáveis será sempre gerado por um subconjunto das direções coordenadas.

Portanto, é possível conhecer de antemão todos os possíveis geradores para o cone $K^\circ(x, \epsilon)$, independentemente do ponto $x \in \Omega$ e do valor de ϵ . Logo, como sabemos que o padrão P^k deve conter um conjunto de geradores positivos para o cone $K^\circ(x, \epsilon)$, podemos tomar $P^k = [I \ -I]$ e utilizar este padrão de direções em todas as iterações do método.

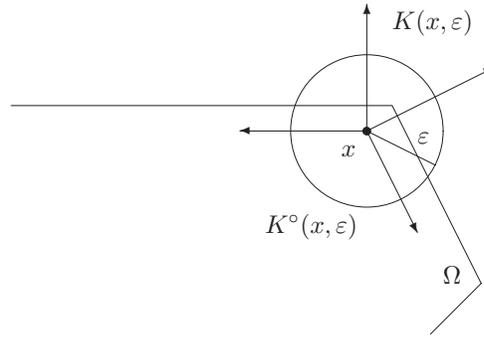


Figura 1: Geometria próximo à fronteira

2.4.2 Restrições Lineares de Igualdade

Ao aplicar o Método de Busca Padrão ao problema com restrições de igualdade, não temos nenhuma flexibilidade ao gerar o padrão, uma vez que pequenos passos podem fazer com que a viabilidade seja perdida.

Tendo em vista que a principal característica do método é manter a viabilidade em todas as iterações, é fácil ver que para manter a viabilidade é necessário andar por direções de modo que $A(x^k + s^k) = Ax^k$, ou seja, devemos ter $As^k = 0$.

Logo, as direções de busca devem necessariamente estar no núcleo da matriz de restrições A , $\mathcal{N}(A)$. Assim, temos que o padrão P^k deve conter um conjunto gerador positivo para o núcleo da matriz A .

Portanto, se todas as restrições do problema são de igualdade, construímos um padrão contendo uma base positiva para $\mathcal{N}(A)$ antes de iniciar o passo de busca, e o mesmo padrão é utilizado em todas as iterações, uma vez que nenhuma outra direção de busca é possível.

2.4.3 Conjunto de Trabalho Vazio

Estamos trabalhando com problemas restritos, porém, caso estejamos no interior do conjunto viável, ou seja, nenhuma das restrições está a uma distância menor do que ϵ do ponto corrente, então, pelo menos por uma certa distância, podemos andar livremente em qualquer direção do \mathbb{R}^n . Deste modo, recaímos momentaneamente no caso de minimizar sem restrições.

Sendo assim, por simplicidade, realizamos uma busca coordenada. Isso quer dizer que utilizamos a matriz $P^k = [I \ -I]$ como padrão, sempre que nenhuma restrição for ϵ -ativa no ponto corrente.

2.4.4 Problemas com Restrições Lineares Gerais

Como vimos, em alguns casos o padrão de direções P^k pode ser obtido facilmente. No entanto, nem sempre esta tarefa é trivial, e muitas características do problema podem dificultar a escolha do padrão mais adequado.

Para começar, vamos definir o que entendemos por conjunto de trabalho degenerado e não-degenerado.

Definição 2.1. *Seja V a matriz cujas colunas geram o cone $K(x^k, \epsilon)$. Se as colunas de V formam um conjunto linearmente independente, diremos que o conjunto de trabalho em x^k , ou seja, na iteração corrente, é não-degenerado. Caso contrário, diremos que o conjunto de trabalho na iteração corrente é degenerado.*

Quando o conjunto de trabalho é não-degenerado podemos construir o padrão de um modo relativamente simples, uma vez que conseguimos provar analiticamente a existência de uma

base racional para o cone $K^\circ(x^k, \epsilon)$, satisfazendo as condições que garantem a convergência do método. Já para o caso degenerado, embora seja possível demonstrar a existência dessa base, determiná-la nem sempre é uma tarefa fácil.

2.4.5 Conjunto de Trabalho Não-degenerado

Para o caso não-degenerado, quando a matriz V tem posto completo, o resultado abaixo, cuja demonstração pode ser encontrada em [3], nos fornece uma forma simples de obter o conjunto gerador para o cone $K^\circ(x, \epsilon)$.

Proposição 2.1. *Suponha que para algum δ , $K(x, \delta)$ tem um conjunto linearmente independente de geradores racionais V . Seja N uma base racional positiva para o espaço nulo de V^T . Então para qualquer ϵ , $0 \leq \epsilon \leq \delta$, um conjunto racional de geradores para $K^\circ(x, \epsilon)$ pode ser obtido entre as colunas de N , $V(V^T V)^{-1}$ e $-V(V^T V)^{-1}$.*

Esse resultado é importante e nos fornece um modo simples de construir o padrão de direções, que deve conter um gerador positivo para o cone K° . Do ponto de vista prático, as colunas da matriz V , que geram o cone $K(x, \epsilon)$, é composta pelas linhas da matriz de restrições A cujas restrições estão à uma distância menor do que ϵ do ponto corrente.

Porém, como discutimos na Seção 2.4.2, quando o problema possui restrições de igualdade, só podemos nos mover por direções que estejam no núcleo de tais restrições. Deste modo, é fundamental para o bom funcionamento do método que essas restrições sejam corretamente consideradas no momento de gerar o padrão, de forma a evitar que todas as direções construídas sejam inviáveis para o problema. Perceba que teremos uma restrição de igualdade quando $l_i = u_i$. Desta forma, separaremos a matriz V em duas submatrizes V_d contendo as restrições de desigualdade ϵ -ativas no ponto corrente, e V_e contendo as restrições de igualdade, que serão ativas em todas as iterações.

Pode acontecer também de uma determinada restrição estar no conjunto de restrições de desigualdade ϵ -ativas, tanto pelo seu limitante inferior quanto pelo limitante superior. Neste caso, diremos que na iteração corrente a restrição é momentaneamente de igualdade e trabalharemos com ela juntamente com as restrições de igualdade do problema. Maiores detalhes desta construção podem ser obtidos em [1].

Adicionalmente, em alguns testes percebemos que ao alcançar a face ótima, o método acaba ficando “preso” nesta face. Isso acontece pois as direções do padrão não permitiam sair do ponto corrente por pontos viáveis diminuindo o valor da função objetivo, embora o ponto corrente não fosse estacionário.

Por esse motivo, quando estamos em uma das faces, ou seja, uma ou mais restrições de desigualdade atingiram um de seus limitantes, adicionamos ao padrão direções que estejam no núcleo destas restrições. Assim permitimos que o método possa se mover na face. A inclusão dessas direções é o grande diferencial do nosso método, uma vez que os resultados melhoraram drasticamente após incluirmos a busca na face.

É importante ressaltar que essas direções são consideradas separadamente, uma vez que, se não estamos em uma face ótima, deve ser possível sair desta face. Sendo assim considere V_e e V_d , como descrito acima e também V_{e2} a matriz contendo as informações das faces atingidas.

Dessa forma, dadas as matrizes V_e , V_d e V_{e2} , o padrão é obtido da seguinte forma:

- determine uma base racional T para o núcleo da matriz V_e ;
- determine uma base racional N para o núcleo da matriz V_d ;
- determine uma base racional T_2 para o núcleo da matriz V_{e2} ;
- determine a matriz $F = V_d(V_d^T V_d)^{-1}$, a qual é obtida pela resolução de vários sistemas lineares utilizando a mesma matriz de coeficientes, sendo assim, utilizamos a fatoração de Cholesky da matriz $(V_d^T V_d)$;

- o padrão P^k fica da seguinte forma:

$$[P^k] = [-T \quad T \quad -N \quad N \quad F \quad -F \quad T_2 \quad -T_2].$$

Observe que o padrão proposto possui um número amplo de direções de busca, mas cumpre a hipótese de convergência de conter um conjunto gerador positivo para o cone $K^\circ(x^k, \epsilon)$, o que nos fornece a teoria de convergência para o método.

2.4.6 Conjunto de trabalho degenerado

Como vimos, o modo como o padrão é construído impede o cálculo da componente F do padrão. Isso acontece porque se o conjunto de trabalho é degenerado, a matriz $(V_d^T V_d)$ será singular, não sendo possível obter a solução do sistema linear, o qual determina o termo principal do padrão.

Porém, ainda que o sistema não tenha solução exata, a solução de Quadrados Mínimos sempre pode ser obtida. Logo, o que propomos é utilizar a solução de Quadrados Mínimos, obtida através da fatoração QR, no lugar da solução exata do sistema linear.

Utilizando esta estratégia o padrão para o caso degenerado pode ser obtido praticamente com o mesmo custo do caso não degenerado, exceto pelo custo de calcular a fatoração QR, que é mais cara que a fatoração de Cholesky. No entanto, se essas direções forem satisfatórias, ainda que a convergência seja mais lenta, por se tratar de um conjunto degenerado, o problema pode ser resolvido sem que haja a necessidade de utilizarmos uma rotina externa para construir o conjunto de busca.

3 Experimentos Numéricos

A implementação computacional do método proposto foi realizada em Matlab versão 7.10.0, e fizemos usos dos pacotes disponíveis para a construção do padrão de direções. O conjunto de testes é composto por 35 problemas com restrições lineares e de caixa extraídos de [2] e [5].

Testamos nosso algoritmo com o parâmetro $\tau = 2$ e fixamos o número máximo de avaliações de função em 10^5 , para podermos analisar a robustez do método. Finalizamos as iterações se $\alpha_k \leq 10^{-8}$ ou ao atingirmos o número máximo de avaliações de função.

Para analisar o desempenho do método proposto vamos compará-lo com o pacote HOPSPACK [6], um pacote sem derivadas para problemas de otimização com restrições gerais, implementado em C++. Utilizamos este pacote por fornecer uma robusta implementação para o GSS [4], um dos métodos sem derivadas mais conhecidos para resolver problemas com restrições lineares, e que utiliza estratégias semelhantes às que utilizamos neste trabalho.

A Figura 2 apresenta o gráfico de performance profile comparativo entre os dois métodos. Na construção do gráfico, consideramos que um problema foi resolvido pelo método se o valor ótimo obtido está a uma distância menor do que 10^{-6} da solução ótima reportada nas referências.

Perceba que os resultados obtidos foram excelentes. Fica claro pelo perfil de desempenho apresentado que o método proposto é significativamente mais robusto que o pacote HOPSPACK e embora tenha se mostrado menos eficiente inicialmente, ele rapidamente supera o HOPSPACK.

Vale destacar também que a precisão obtida nos resultados foi muito boa. Obtemos um método robusto e competitivo aos pacotes disponíveis atualmente.

4 Conclusões

O Método de Busca Padrão proposto possui um algoritmo bem simples e com grande apelo geométrico, podendo ser aplicado quase que imediatamente em grande parte das aplicações, uma vez que basta que sejamos capazes de comparar o valor da função objetivo nos pontos visitados pelo algoritmo.

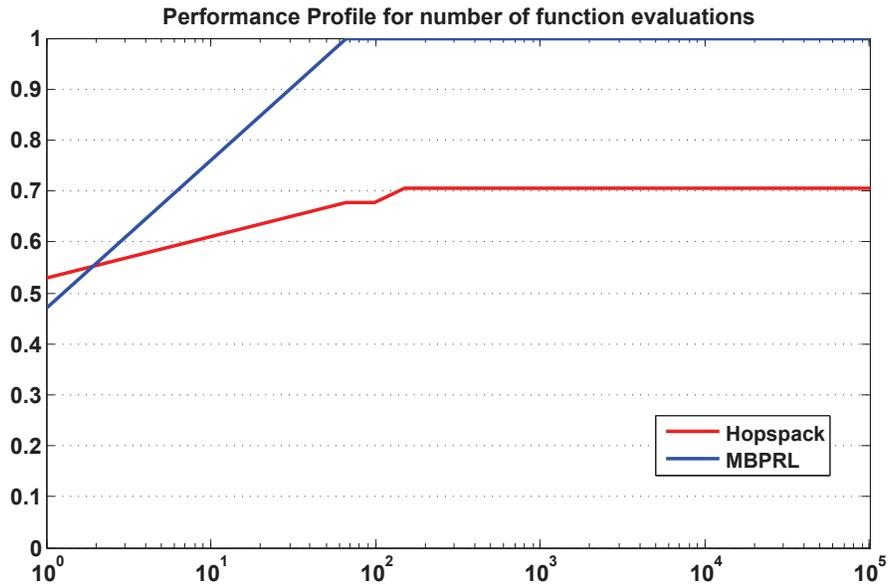


Figura 2: Perfil de desempenho - Número de avaliações de função

Associado a isso as estratégias propostas se mostraram bastante eficientes na prática, e o método proposto é bastante robusto, além de possuir garantias teóricas de convergência. Levando em conta a importância da resolução de problemas com restrições lineares em otimização, podemos concluir que os resultados obtidos foram satisfatórios e com grande potencial para aplicações e trabalhos futuros baseados no método que apresentamos.

Referências

- [1] D. G. Ferreira. Sobre Métodos de Busca Padrão para Minimização de Funções com Restrições Lineares. Dissertação de Mestrado em Matemática Aplicada, IMECC, Unicamp, março 2013.
- [2] W. Hock, K. Schittkowski. “Test examples for nonlinear programming code”, Springer, Berlin, (1981).
- [3] R. M. Lewis, V. Torczon. Pattern search algorithms for linearly constrained minimization. *SIAM Journal on Optimization* 10, pp. 917-941, (2000).
- [4] R. M. Lewis, A. Shepherd, V. Torczon. Implementing generating set search methods for linearly constrained minimization. *SIAM Journal on Scientific Computing* 29, 6, pp. 2507-2530, (2007).
- [5] D. Orban, N.I.M. Gould, P.L. Toint. CUTER: Constrained and Unconstrained Testing Environment, revisited. Disponível em: <http://www.cuter.rl.ac.uk>, (2011).
- [6] T. D. Plantenga HOPSPACK 2.0 User Manual, Sandia National Laboratories, Report SAND2009-6265, disponível em: http://www.sandia.gov/hopspack/HopspackUserManual_2_0_2.pdf, (2009).
- [7] V. Torczon. On the convergence of pattern search algorithms. *SIAM Journal on Optimization* 7, pp. 1-25, (1997).