Estudo de métodos para minimização irrestrita que utilizam direções aleatórias sem uso de derivadas

Alisson Lucas de Souza
Departamento Acadêmico de Matemática
Universidade Tecnológica Federal do Paraná
Cornélio Procópio, Brasil
alissonsouza@alunos.utfpr.edu.br

André Luís Machado Martinez
Departamento Acadêmico de Matemática
Universidade Tecnológica Federal do Paraná
Cornélio Procópio, Brasil
martinez@utfpr.edu.br

***Resumo*—Este trabalho apresenta uma análise teórica e numérica do Método das Direções Aleatórias. O método estudado é utilizado com objetivo de minimização irrestrita, com a ressalva de não utilizar cálculo de derivada em nenhum de seus passos, no qual o grande diferencial do método é gerar, a cada iteração, direções aleatórias, possibilitando utilizar inclusive direções de subida. Além desse estudo, propomos modificações no algoritmo original, utilizando o Método da Seção Áurea para busca linear e combinação convexa para a criação de direções diretivas, e realizamos testes numéricos no software MATLAB.**

***Palavras-chave*—otimização, derivadas, direções aleatórias, seção áurea, combinação convexa.**

I. Introdução

Neste trabalho apresentamos um estudo sobre o Método das Direções Aleatórias, que foi proposto por Diniz-Ehrhardt, Martínez e Raydan em 2008 [3]. Este é um método de otimização sem derivada que utiliza busca linear e considera o problema: $min\, f(x)$, onde $f : \mathbb{R}^n \to \mathbb{R}$, sujeito a $x \in \mathbb{R}^n$. Métodos de otimização sem derivada são aplicados, em geral, em problemas que têm como função objetivo um arquivo executável cujo código não está acessível ao usuário, conhecidos como caixas-pretas ([2], [3]).

O grande diferencial do método estudado é que o vetor diretor $d \in \mathbb{R}^n$ é gerado aleatoriamente a cada iteração, podendo permitir acréscimos na função objetivo e, eventualmente, utilizar direções de subida. Além da análise do algoritmo do Método das Direções Aleatórias, propomos duas modificações no algoritmo original: uma na busca linear realizada no algoritmo, na qual utilizamos Seção Áurea; e outra na criação das direções aleatórias, onde usamos combinações convexas, e, por fim, reproduzimos testes comparativos.

II. Método das Direções Aleatórias

O Método das Direções Aleatórias se caracteriza como um método globalmente convergente com estratégia de busca linear não monótona. Primeiramente, escolhe-se um parâmetro $M \in \mathbb{N}$. E em uma iteração k, definimos:

$$\overline{f}_k = max\{f(x_k), \ldots, f(x_{max\{k-M+1\}})\}.$$

Se a inequação

$$f(x_k + \alpha_k d_k) \leq \overline{f}_k + \eta_k - \alpha_k^2 \beta_k \qquad (1)$$

for satisfeita, tomamos $x_{k+1} = x_k + \alpha_k d_k$. O parâmetro η_k deve ser escolhido de modo que $\eta_k > 0, \forall\, k \in \mathbb{N}$ e $\sum_{k=0}^{\infty} \eta_k = \eta < \infty$. Já o parâmetro β_k deve ser tal que $\{\beta_k\}$ seja uma sequência limitada, isto é, exista $C \in \mathbb{R}$ tal que $\beta_k < C$, com $\beta_k > 0$, $\forall\, k \in \mathbb{N}$ e, para qualquer subconjunto infinito de índices $K \subset \mathbb{N}$, $\lim_{k \in K} \beta_k = 0 \Rightarrow \lim_{k \in K} \nabla f(x_k) = 0$.

A inequação (1) é a responsável por eventualmente serem admitidos pontos que representem acréscimo à função objetivo. Isso se deve às características do parâmetro η_k e ao fato de utilizarmos $\overline{f}_k$ ao invés de $f(x_k)$. A seguir está o Algoritmo proposto originalmente pelos autores do Método das Direções Aleatórias [2], [3].

Algoritmo 1. *Algoritmo do Método das Direções Aleatórias.*

Sejam $\{\beta_k\}$ e $\{\eta_k\}$ satisfazendo as relações apresentadas anteriormente. Consideremos também τ_{min} e τ_{max} tais que $0 < \tau_{min} < \tau_{max} < 1$, $M \in \mathbb{N}$, $0 < \Delta_{min} < \Delta_{max} < \infty$ e $c_{max} \in \mathbb{R}^+$. Os passos para determinar x_{k+1} a partir de x_k são os seguintes:

***Passo 1:** Calcule uma direção de descida aleatória $d \in \mathbb{R}^n$ tal que $\Delta_{min} \leq ||d|| \leq \Delta_{max}$.*

***Passo 2:** Se $f(x_k + d) \leq f(x_k) + \eta_k - \beta_k$, faça $\alpha_k = 1$, $d_k = d$ e vá para o Passo 5. Se $f(x_k - d) \leq f(x_k) + \eta_k - \beta_k$, faça $\alpha_k = 1$, $d_k = -d$ e vá para o Passo 5.*

***Passo 3:** Interpolação Quadrática. Calcule $\tilde{\alpha}$, o minimizador da parábola que interpola os pontos $(-1, f(x_k - d))$, $(0, f(x_k))$ e $(1, f(x_k + d))$.*

1) *Se $\tilde{\alpha}$ existe e pertence a $[\tau_{min}, \tau_{max}]$, faça $d_k = d$ e vá para o Passo 4.*
2) *Se $\tilde{\alpha}$ existe e pertence a $[-\tau_{min}, -\tau_{max}]$, faça $d_k = -d$ e vá para o Passo 4.*
3) *Se $f(x_k + d) \leq f(x_k - d)$, faça $d_k = d$, $\tilde{\alpha} = \frac{1}{2}$ e vá para o Passo 4.*
4) *Se $f(x_k + d) > f(x_k - d)$, faça $d_k = -d$, $\tilde{\alpha} = \frac{1}{2}$ e vá para o Passo 4.*

***Passo 4:** Backtracking. Faça $\alpha = \tilde{\alpha}$. Se* (1) *for satisfeita, faça $\alpha_k = \alpha$, $x_{k+1} = x_k + \alpha_k d_k$ e termine a iteração. Caso contrário, calcule $\alpha_{new} \in [\tau_{min}\alpha, \tau_{max}\alpha]$ usando Interpolação Quadrática com salvaguarda, faça $\alpha = \alpha_{new}$ e repita o teste em* (1)*.*

***Passo 5:** Extrapolação.*

1) *Faça* $c = 1$.
2) *Se* $2c > c_{max}$*, faça* $x_{k+1} = x_k + c\alpha_k d_k$ *e termine a iteração.*
3) *Se* $f(x_k + 2c\alpha_k d_k) > f(x_k + c\alpha_k d_k)$*, faça* $x_{k+1} = x_k + c\alpha_k d_k$ *e termine a iteração.*
4) *Faça* $c = 2c$ *e vá para o Passo 2 da Extrapolação.*

Este método tem como principal característica o uso de direções aleatórias, então mantemos os parâmetros iniciais do algoritmo, bem como os Passos 1, 2 e 5. Nossa modificação está relacionada aos Passos 3 e 4, nos quais os autores propõem o uso do Método da Interpolação Quadrática para encontrar o tamanho do passo. Optamos por utilizar o Método da Seção Áurea ([1], [4], [5]), como especificado a seguir:

Passo 3: *Seção Áurea*. Utilizando o método da Seção Áurea, calcule $\tilde{\alpha} \in [-1, 1]$ o mínimo aproximado de $f(x_k + \tilde{\alpha} d_k)$. Se $\tilde{\alpha} < 0$, faça $d_k = -d_k$, $\tilde{\alpha} = |\tilde{\alpha}|$ e vá ao Passo 4.

Passo 4: *Backtracking*. Faça $\alpha = \tilde{\alpha}$. Se (1) for satisfeita, faça $\alpha = \tilde{\alpha}$, $x_{k+1} = x_k + \alpha_k d_k$ e termine a iteração. Caso contrário, calcule $\alpha_{new} \in [\tau_{min}\alpha, \tau_{max}\alpha]$, usando Seção Áurea com salvaguarda. Faça $\alpha = \alpha_{new}$ e repita o teste (1).

Ao aplicar Seção Áurea, utilizamos precisão $\varepsilon = 10^{-2}$. Assim, durante os testes, observamos que o algoritmo satisfazia a inequação (1) mais rapidamente que ao usar Interpolação Quadrática. A estratégia de minimizar o intervalo $[x-d, x+d]$ se apresentou como alternativa razoável para decidir a direção de descida (d_k ou $-d_k$).

III. Resultados e Discussões

Dos vinte problemas que os autores propõem para os testes numéricos, o algoritmo original consegue resolver dois deles com total sucesso e outros seis com a solução próxima do que é apresentado em [6]. Implementamos o algoritmo modificado no software MATLAB e repetimos os testes em [6]. A Tabela I a seguir mostra os resultados obtidos pelos algoritmos original e modificado:

Tabela I
Resultado dos Testes Numéricos

Problema		Original			Modificado		
Prob	Dim	Conv	It	f	Conv	It	f
1	2	Não	430	4.91D-5	Próx	360	3.27D-6
2	2	Próx	5000	4.91D+1	Sim	118	4.898D+1
3	2	Não	23	1.352D-1	Não	61	7.591D-1
4	2	Não	341	10.0D+11	Não	5000	9.89D+15
5	2	Não	849	2.55D-3	Próx	191	7.33D-8
6	2	Sim	3769	1.24D+2	Sim	138	1.243D+2
7	3	Não	134	NaN	Não	848	1.83D-1
8	3	Sim	1453	8.52D-3	Sim	396	8.214D-3
9	3	Próx	4	1.01D-6	Não	1732	1.08D-1
10	3	Não	2	7.13D+8	Não	649	1.89D+9
11	3	Não	1010	6.67	Não	550	7.04D-8
12	3	Não	682	1.28D-4	Próx	392	6.3D-9
13	4	Não	487	3.52D-5	Próx	1242	3.97D-6
14	4	Não	5000	1.78	Próx	2442	8.77D-4
15	4	Não	22	2.92D-3	Sim	4091	3.075D-4
16	4	Próx	5000	6.35D+6	Sim	1119	8.582D+8
17	4	Não	356	6.57D-2	Não	5000	6.61D-2
18	6	Próx	1556	3.44D-2	Não	5000	1.43D-6
19	11	Próx	407	7.65	Não	5000	4.77D-3
20	6	Próx	1587	2.0D-2	Próx	5000	2.46D-3

Vale ressaltar que as direções geradas pelo algoritmo são sempre feitas aleatoriamentes. Com isso em mente, comparando com as soluções dadas em [6], pode-se ver que cinco dos vinte problemas foram resolvidos com sucesso (Problemas 2, 6, 8, 15 e 16) e para seis dos problemas (Problemas 1, 5, 12, 13, 14 e 20) o valor da função objetivo foi próximo do que é apresentado em [6].

A outra modificação no algoritmo do Método das Direções Aleatórias é a criação de várias direções aleatórias e usar uma combinação convexa para criar a direção a ser utilizada em cada iteração. Ressalta-se que a forma de criar as direções auxiliares é a mesma da direção aleatória pelo algoritmo original. A combinação convexa é feita da seguinte forma:

Tendo criada as direções aleatórias $d_1, d_2, \cdots, d_n$, verifica-se se d_i ou $-d_i$ é direção de descida, para cada $i = 1, 2, \cdots, n$. Usaremos, então, somente as direções de descida. Dessa forma, dado $\varepsilon = 10^{-2}$ cria-se, para $i = 1, 2, \cdots, n$, os seguintes parâmetros:

$$\lambda_i = \frac{f(x_k) - f(x_k + \varepsilon d_k)}{f(x_k)}.$$

A direção aleatória usada na iteração é dada por:

$$d_k = \sum_{i=1}^{n} \lambda_i d_i \Bigg/ \sum_{i=1}^{n} \lambda_i \ .$$

Repetimos os testes numéricos, implementando o algoritmo novamente modificado no software MATLAB, e utilizamos 2, 3, 5 e 10 direções auxiliares. Essa implementação foi feita tanto no algoritmo original quanto no algoritmo modificado anteriormente. Os testes mostraram que essa estratégia não surtiu nenhum efeito positivo na performance em ambos os algoritmos testados.

IV. Conclusões

A primeira modificação proposta se mostrou promissora, o que nos leva a testar novas alterações no algoritmo, como perspectivas futuras, que possam melhorar sua performance, tentando manter os principais aspectos teóricos e validar as alterações com mais testes numéricos. Em relação a segunda modificação, apesar de não ter mostrado nenhuma melhora na performance dos algoritmos, iremos estudar os possíveis motivos de não ter funcionado.

Referências

[1] S. C. Chapra and R. P. Canale. *Métodos Numéricos para a Engenharia*. 5ª ed. MCGRAW-HILL, 2008.

[2] M. A. Diniz-Ehrhardt, V. L. da R. Lopes e L. G. Pedroso. *Métodos sem Derivadas para Minimização Irrestrita*. Notas em Matemática Aplicada, SBMAC, vol. 49, 2010.

[3] M. A. Diniz-Ehrhardt, J. M. Martínez and M. Raydan. A derivative-free nonmonotone line search technique for unconstrained optimization, *Journal of Computational and Aplied Mathematics*, 219:383–397, 2008.

[4] A. Izmailov e M. Solodov. *Otimização: Condições de Otimalidade, Elementos de Análise Convexa e de Dualidade*. IMPA, vol. 1, 2005.

[5] D. G. Luenberger and Y. Ye. *Linear and Nonlinear Programming*. Springer, Stanford, 2008.

[6] J. J. Moré, B. S. Garbow and K. E. Hillstrom. Testing unconstrained optimization software, *ACM Transactions on Mathematical Software*, 7:17–41, 1981.