

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Testes computacionais de um algoritmo teórico para problemas irrestritos sem derivadas baseado em modelos quadráticos de interpolação com geometria autocorretiva

Ivan X. M. do Nascimento¹

Instituto de Matemática, Estatística e Computação Científica, Unicamp, Campinas, SP

Sandra Augusta Santos²

Instituto de Matemática, Estatística e Computação Científica, Unicamp, Campinas, SP

Resumo. Neste trabalho, consideramos o algoritmo teórico proposto por Scheinberg e Toint [*SIAM Journal on Optimization*, 20 (6) (2010), pp. 3512-3532] para a resolução de problemas de otimização irrestrita sem derivadas baseado na aproximação local da função objetivo por modelos quadráticos de interpolação (completa).

A convergência global desse algoritmo foi demonstrada por seus idealizadores a partir do estudo de uma curiosa propriedade de autocorreção da geometria do conjunto amostral de pontos, a qual se evidencia em iterações de insucesso. Dessa forma, o controle direto do posicionamento dos pontos amostrais ocorre apenas no estágio final do processo de otimização.

Nosso objetivo é apresentar resultados que comprovem o bom desempenho computacional de uma implementação própria desse algoritmo, comparando-o com outros dois *solvers* sem derivadas, seguindo a metodologia de Moré e Wild [*SIAM Journal on Optimization*, 20 (1) (2009), pp. 172-191], apropriada para essa área da otimização.

Palavras-chave. Otimização sem derivadas, Otimização irrestrita, Interpolação quadrática, *Poisedness*, Experimentos numéricos.

1 Introdução

Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função não linear e suponha que suas derivadas de primeira ordem existam e sejam Lipschitz contínuas. Nosso interesse reside na resolução do problema irrestrito

$$\underset{x \in \mathbb{R}^n}{\text{minimizar}} f(x) \quad (1)$$

em um contexto prático onde as derivadas da função não estão disponíveis ou não são precisas o suficiente para certa aplicação. Usualmente, em tais situações, o custo de avaliar a função objetivo é mais relevante do que os da álgebra linear computacional e do tempo de processamento do código envolvidos.

¹ivan@ime.unicamp.br, ivan.xmn@gmail.com

²sandra@ime.unicamp.br

Com o intuito de resolver o problema (1), adotaremos o algoritmo teórico proposto por Scheinberg e Toint [14], o qual chamaremos³ de DFO-LP, instanciado através de uma implementação própria em MATLAB, aqui denotada por DFO-LP. A fim de analisar seu desempenho computacional, compararemos nossa implementação com outros dois *solvers* sem derivadas: o `fminsearch` do MATLAB, versão do clássico método de Nelder-Mead [12], e o competitivo *Simplex Derivative in Pattern Search Method* (SID-PSM), de Custódio e Vicente [5]. Ou seja, teremos um representante da classe de métodos de busca direta e um da classe de métodos de busca padrão, respectivamente. Nesse sentido, a comparação com a abordagem de região de confiança do DFO-LP será enriquecida.

Para comparar efetivamente os *solvers* sem derivadas utilizaremos a metodologia de Moré e Wild [10], que lança mão dos chamados *data profile* e *performance profile*, aqui traduzidos livremente como *perfil informativo* e *perfil de desempenho*. Os perfis informativos foram propostos para auxiliar na comparação de *solvers* sem derivada e servem como uma complementação aos perfis de desempenho de Dolan e Moré [7], estes amplamente adotados em toda a área de otimização.

Para maiores detalhes sobre os procedimentos descritos neste trabalho, ver [11].

2 O surgimento do DFO-LP

Ao aplicarmos um método da classe de região de confiança sob a ausência de derivadas [4, Capítulo 10], cuidados especiais devem ser tomados com os modelos e pontos amostrais. Conn *et. al.* [3] demonstram a convergência global a pontos estacionários de primeira e segunda ordem desde que sejam incluídos passos especiais de melhora na geometria do conjunto amostral utilizado na construção dos modelos polinomiais.

Entretanto, Scheinberg e Toint [14] mostram que apesar do controle do posicionamento dos pontos amostrais ser essencial para a convergência global do método, é possível que tal controle ocorra de modo direto apenas no estágio final do algoritmo. Baseando-se nessas ideias e incorporando-as a um esquema algorítmico teórico, os autores investigam analiticamente uma curiosa propriedade de autocorreção da geometria dos pontos, a qual se evidencia nas iterações de insucesso. E, assim, surge o algoritmo central deste trabalho, consultável em [14, Algoritmo 2], como uma tentativa teoricamente exitosa de abrir mão das iterações especiais de controle da geometria dos pontos amostrais sem perder a garantia de convergência global a pontos estacionários de primeira ordem.

3 Implementação Computacional

A fim de comparar o desempenho do DFO-LP, implementamos uma instância desse algoritmo em MATLAB. Como é possível observar em [14, Algoritmo 2], a versão teórica do DFO-LP deixa em aberto diversas decisões práticas a serem tomadas no momento de sua implementação. Dessa forma, o DFO-LP considerado neste trabalho é apenas uma implementação possível para o original e se caracteriza a partir dessas decisões de caráter prático.

³Em alusão ao termo *Derivative-Free Optimization based on Lagrange Polynomials*, em inglês.

Como é característica do MATLAB, a representação dos objetos matemáticos, em geral, é modelada sob a forma de matrizes (*arrays*). Portanto, os conjuntos de pontos, os (coeficientes dos) polinômios de Lagrange e as demais estruturas de dados envolvidas na implementação foram programadas como *arrays* de dimensão conveniente.

Sobre o Passo 0 de inicialização do DFO-LP, optou-se pelo procedimento descrito por Powell [13, Seção 5], o qual trata de um mecanismo perspicaz e conveniente de construção dos conjuntos \mathcal{Y}_0 (posicionado) e L_0 , além do modelo aproximador inicial, a partir de um único ponto y^1 dado, e que resulta em estruturas iniciais consideravelmente esparsas. A partir da primeira iteração, a atualização dos coeficientes dos polinômios de Lagrange é feita de maneira eficiente seguindo as orientações de Powell [13, Seção 2].

Originalmente presente no Algoritmo DFO-LP, o parâmetro ϵ_0 determina o momento em que, pela primeira vez na execução do método proposto, será exigida a construção de um modelo Λ -posicionado (cf. [4, Seção 3.3]). O valor desse parâmetro utilizado em nossa implementação foi obtido experimentalmente em [11, Capítulo 8] e corresponde a 25% da norma do gradiente do modelo inicial.

Para construir os modelos Λ -posicionados do Passo 1 do DFO-LP optou-se pela utilização de um subalgoritmo, originalmente proposto por Conn, Scheinberg e Vicente [4, Algoritmo 6.2], que faz uso dos polinômios de Lagrange disponíveis (e atualizados) a cada iteração.

As decisões de descarte de pontos no Passo 4 (itens *a* e *b*) foram implementadas seguindo as ideias de Gratton, Toint e Tröltzsch [9, Seção 3.5].

O subproblema de região de confiança foi resolvido com a função `trust.m`, nativa do MATLAB e recomendável apenas para problemas de pequeno porte, como os tratados aqui. Por fim, optou-se por atualizar o raio da região de confiança dinamicamente, de acordo com a sugestão de Conn, Gould e Toint [1, p. 782].

4 Metodologia de comparação de *solvers* sem derivada

A metodologia de comparação neste trabalho foi a de Moré e Wild [10]. O princípio por trás da proposta é o de fornecer subsídios de decisão para usuários com diferentes “orçamentos” disponíveis. Além disso, os autores argumentam que o usuário de *solvers* de otimização sem derivadas usualmente está interessado no maior decréscimo possível do valor da função objetivo, e não em soluções de alta precisão, já que não é realista esperar por soluções acuradas, nesse cenário.

Dessa forma, adotamos o seguinte critério de convergência. Sejam \mathcal{S} , o conjunto de *solvers* a serem comparados, e \mathcal{P} , um conjunto de problemas-teste a serem submetidos a cada um dos elementos \mathcal{S} . Diremos que um método $s \in \mathcal{S}$ resolveu um problema $p \in \mathcal{P}$ se, para algum iterando x ,

$$f(x^0) - f(x) \geq (1 - \tau)(f(x^0) - f_L^p), \quad (2)$$

sendo que x^0 é o ponto inicial para o problema, $\tau \in [0, 1)$ é uma tolerância e f_L^p é o menor valor de função objetivo dentre os obtidos pelos *solvers* de \mathcal{S} para o problema p com um orçamento μ_f de avaliações de função disponíveis.

Para obter o valor f_L^p , é necessário submeter o problema p a todos os *solvers* de \mathcal{S} de maneira que o único critério de parada adotado seja o do número máximo de avaliações de função. Ou seja, a determinação da convergência é feita *a posteriori* e é totalmente dependente do desempenho de \mathcal{S} como um todo. Observa-se que a Equação (2) exige que a redução $f(x^0) - f(x)$ obtida com o ponto x seja de, pelo menos, $(1 - \tau)$ vezes a melhor redução que se pode obter quando se dispõe dos *solvers* de \mathcal{S} .

5 Experimentos Computacionais

Os experimentos descritos nesta seção foram realizados em uma máquina com sistema operacional **LINUX** 64 bits e processador **Intel® Core i7 - 2600K**, de 3.40 GHz de frequência e com memória *cache* L3 de 8 Mb. A versão do **MATLAB** utilizada foi a 7.10.0.499 (*release R2010a*) para computadores de 64 bits.

Sob as hipóteses do contexto de otimização sem derivadas, os resultados são apresentados em termos da unidade de medida **número de avaliações de função**. No conjunto \mathcal{S} de *solvers*, consideramos as versões **SID-PSM1** e **SID-PSM2** do **SID-PSM**, caracterizadas em [2], e uma versão do **Nelder-Mead** [12], totalizando quatro métodos a serem confrontados. Os testes consistiram em submeter 57 dos 60 problemas-teste⁴ considerados em [8] a cada um dos *solvers* de \mathcal{S} , armazenando convenientemente os dados envolvidos no processo.

Com o interesse canalizado no comportamento dos *solvers* a curto prazo, disponibilizou-se para cada par $(p, s) \in \mathcal{P} \times \mathcal{S}$ um orçamento de avaliações de função que garantisse pelo menos o equivalente à construção de 150 gradientes simplex (cf. [4, Seção 2.3]). Como a maior dimensão presente no conjunto \mathcal{P} é $n = 15$, adotou-se $\mu_f = 2400$.

6 Comparação dos *solvers* de \mathcal{S}

Nesta seção, apresentamos os resultados computacionais obtidos com o **DF0-LP**, instância do **DF0-LP**, e realizamos a comparação dos *solvers* segundo a metodologia estabelecida anteriormente.

Através dos perfis informativos exibidos na Figura 1, é possível observar que, de um modo geral, o **DF0-LP** é capaz de resolver mais problemas do conjunto \mathcal{P} do que seus rivais, principalmente com o aumento da exigência na acurácia, isto é, com a diminuição do valor de τ . Ainda, fixado τ , observa-se que quanto menor o número de gradientes simplex disponível, mais equiparáveis são os desempenhos dos *solvers*, ainda que nitidamente o **fminsearch** não tenha superado nenhum dos demais *solvers* nesses casos. Naturalmente, quanto menor a exigência na acurácia do valor ótimo da função, mais problemas são resolvidos por um mesmo *solver*.

Os perfis de desempenho para os *solvers* de \mathcal{S} são apresentados na Figura 2, em escala semilogarítmica. Novamente, confirmando as constatações dos perfis informativos, em ambos os níveis de acurácia, a rotina **DF0-LP** foi mais robusta e mais eficiente do que as

⁴ Dois deles já não se encontram disponíveis no ambiente **CUTEr** [6] (**SINGULAR** e **NASTY**) e o terceiro possui variáveis limitadas (**CHEBYQAD**).

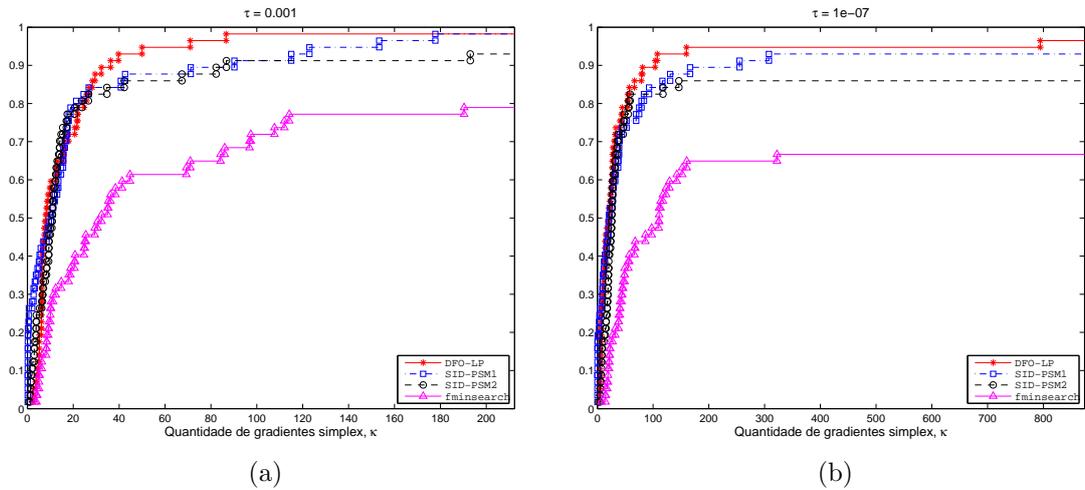


Figura 1: Perfis informativos dos *solvers* de \mathcal{S} para dois níveis de acurácia τ , considerando o teste de convergência (2). Eixo horizontal em unidade equivalente ao número de avaliações de função necessário para o cálculo de um gradiente simplex.

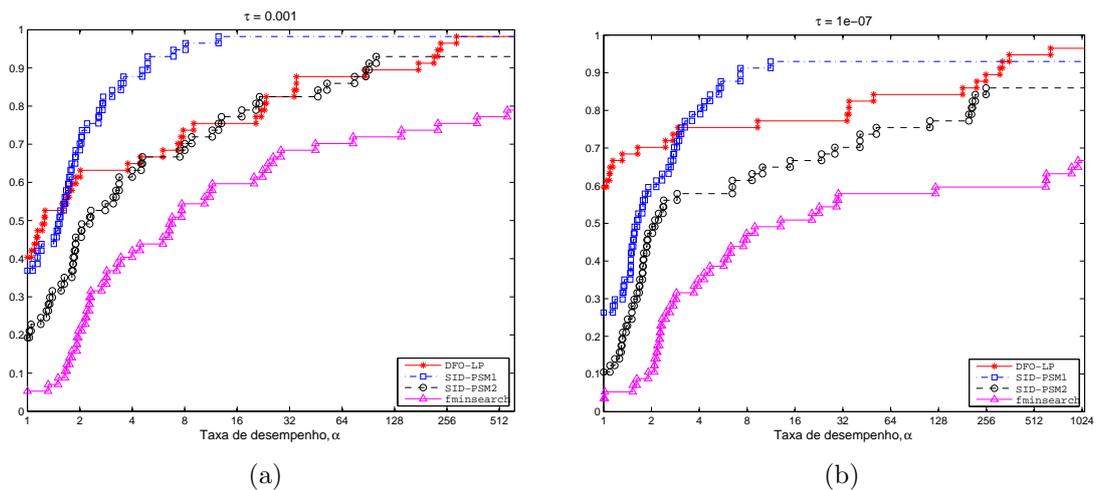


Figura 2: Perfis de desempenho dos *solvers* de \mathcal{S} para dois níveis de acurácia τ , considerando o teste de convergência (2). O eixo horizontal representa a taxa de desempenho relacionada à medida “número de avaliações de função”.

demais. Considerando o cenário aparentemente mais rigoroso com os *solvers*, isto é, o correspondente ao Gráfico 2b, os números sobre o desempenho do DFO-LP são notáveis:

- ele resolveu com menos avaliações de função que os demais *solvers* uma fatia de aproximadamente 60% dos problemas de \mathcal{P} com um nível $\tau = 10^{-7}$ de acurácia, respeitado o limite $\mu_f = 2400$ no número máximo de avaliações de função;
- dentro do mesmo orçamento μ_f , foi capaz de resolver aproximadamente 95% dos

problemas de \mathcal{P} – isto representa em torno de 5% a mais do que o segundo *solver* mais robusto e 10% a mais do que o terceiro, nesse mesmo cenário.

Outras informações podem ser extraídas dos perfis de desempenho. Por exemplo, no Gráfico 2b, o DFO-LP atinge o valor aproximado de 77% quando a taxa de desempenho está em torno de 8, enquanto que para esta mesma taxa, o SID-PSM1 alcança um patamar próximo aos 90%. Das definições de taxa e perfil de desempenho é possível concluir que o SID-PSM1 resolveu em torno de 13% a mais de problemas de \mathcal{P} consumindo menos do que 8 vezes o gasto mínimo possível para esses problemas com os *solvers* de \mathcal{S} , se comparado ao DFO-LP.

7 Considerações finais

Neste trabalho, foram apresentados os resultados de uma implementação do método de região de confiança direcionado a problemas não lineares de otimização irrestrita sem derivadas que se favorece do mecanismo de autocorreção da geometria dos pontos amostrais, proposto em [14], onde também podemos encontrar a demonstração de sua convergência global. Com a utilização de modelos de interpolação quadráticos completos, os experimentos numéricos relatados aqui sugerem tanto a eficiência como a robustez do método DFO-LP, dentro da representatividade dos conjuntos \mathcal{S} e \mathcal{P} .

Ainda, enfatizou-se o uso de uma metodologia relativamente nova de comparação de *solvers*, com o uso dos perfis informativos [10], mais focada na aplicação prática dos métodos e no processo de decisão a que um usuário de otimização sem derivadas tipicamente é submetido. Quanto à escolha do valor ótimo de referência adotado no teste de convergência (2), está implícita a suposição de que cada problema pode ser razoavelmente bem resolvido por algum dos *solvers* do conjunto \mathcal{S} ; caso contrário, as conclusões sobre eficiência e robustez podem ser ilusórias ou extremamente limitadas. Por outro lado, sob um aspecto mais conceitual, não se pode esperar que o desempenho de *solvers* com e sem derivadas sejam sempre equiparáveis, o que advoga a favor desse tipo de referenciação.

Um possível prolongamento deste trabalho está na melhoria das subrotinas implementadas e na consideração de um conjunto maior de problemas e de *solvers* a serem comparados.

Agradecimentos

Agradecemos à CAPES, ao CNPq (304032/2010-7) e à FAPESP (2013/05475-7 e 2013/07375-0) pelo apoio financeiro com o qual este trabalho foi realizado.

Referências

- [1] A. R. Conn, N. I. M. Gould and Ph. L. Toint, Trust Region Methods, MPS-SIAM Series on Optimization, SIAM, (2000).
- [2] B. H. Cervelin, Sobre um Método de Minimização Irrestrita Baseado em Derivadas Simplex, Dissertação de Mestrado em Matemática Aplicada, Unicamp, (2013).
- [3] A. R. Conn, K. Scheinberg and L. N. Vicente, Global convergence of general derivative-free trust-region algorithms to first-and second-order critical points, SIOPT, vol. 20, 387–415, (2009).
- [4] A. R. Conn, K. Scheinberg, and L. N. Vicente, Introduction to derivative-free optimization, MPS-SIAM Ser. Optim. vol. 8, SIAM, (2009).
- [5] A. L. Custódio and L. N. Vicente, Using sampling and simplex derivatives in pattern search methods, SIOPT, vol. 18, 537–555, (2007).
- [6] N. I. M. Gould, D. Orban and Ph. L. Toint, CUTER: a Constrained and Unconstrained Testing Environment, revisited, www.cuter.rl.ac.uk/ (Acessado em 12/04/2015).
- [7] E. D. Dolan and J. J. Moré, Benchmarking optimization software with performance profiles, Mathematical Programming, vol. 91, 201–213, (2002).
- [8] G. Fasano, J. L. Morales and J. Nocedal, On the geometry phase in model-based algorithms for derivative-free optimization, Optimization Methods & Software, vol. 24, 145–154, (2009).
- [9] S. Gratton, Ph. L. Toint and A. Tröltzsch, An active-set trust-region method for derivative-free nonlinear bound-constrained optimization, Optimization Methods & Software, vol. 26, 873–894, (2011).
- [10] J. J. Moré and S. M. Wild, Benchmarking derivative-free optimization algorithms, SIOPT, vol. 20, 172–191, (2009).
- [11] I. X. M. do Nascimento, Otimização sem Derivadas: sobre a construção e a qualidade de modelos quadráticos na solução de problemas irrestritos, Dissertação de Mestrado em Matemática Aplicada, Unicamp, (2014).
- [12] J. A. Nelder and R. Mead, A simplex method for function minimization, The Computer Journal, vol. 7, 308–313, (1965).
- [13] M. J. D. Powell, On the Lagrange functions of quadratic models that are defined by interpolation, Optimization Methods & Software, vol. 16, 289–309, (2001).
- [14] K. Scheinberg and Ph.L. Toint, Self-correcting geometry in model-based algorithms for derivative-free unconstrained optimization, SIOPT, vol. 20, 3512–353, (2010).