Um Método de Otimização Irrestrita Baseado em Busca Sobre a Curva Trust-Region

Afonso H. Sampaio, Ricardo H. C. Takahashi
Departamento de Matemática - Universidade Federal de Minas Gerais
Av. Antônio Carlos, 6627, Pampulha, Belo Horizonte, MG CEP 31270-901, Brasil.
afonsohs@dcc.ufmg.br, taka@mat.ufmg.br

RESUMO

Métodos baseados em busca em linha (line search) e região de confiança (trust region) estão entre os mais utilizados na solução de problemas de otimização não linear irrestrita. Neste trabalho, desenvolvemos um método que combina estratégias de ambos os métodos e implementamos um algoritmo. O método proposto é comparado com dois métodos disponíveis em um pacote comercial (um método com busca em linha do tipo *quasi*-Newton e um método com região de confiança), sobre um conjunto de problemas-teste obtidos da literatura. Na comparação com o método *quasi*-Newton, o método proposto se mostra mais confiável, encontrando a solução dos problemas um número maior de vezes. Comparado ao método com região de confiança, o método proposto se mostra computacionalmente mais eficiente, requerendo um número menor de avaliações de gradiente (e, consequentemente, menor tempo de execução), para uma proporção similar de execuções bem sucedidas.

PALAVRAS CHAVE. Otimização Irrestrita, busca em linha, Região de Confiança

ABSTRACT

Line search and trust region methods are employed widely for solving unconstrained non-linear optimization problems. In this work, we present a new method that combines features of both methods in a single algorithm. The proposed method was compared with two methods available in a commercial package (a *quasi*-Newton line search and a trust region method) on a set of test problems borrowed from the literature. In the comparison with the *quasi*-Newton method, the proposed method performed more reliably, finding the problem solution in a larger number of runs. In the comparison with the trust region method, the proposed method presented an enhanced computational efficiency, requiring a smaller number of gradient evaluations (consequently a smaller run time), for a similar rate of successful runs.

KEYWORDS. Unconstrained Optimization, Line Search, Trust-Region

1. Introdução

Problemas de otimização não linear sem restrições na função objetivo surgem em diversas aplicações práticas e, além disso, técnicas para a resolução destes problemas formam a base de métodos mais sofisticados para a solução de problemas com função objetivo e restrições de diferentes estruturas. Alguns métodos para problemas de otimização com restrições na função objetivo, por exemplo, buscam por uma solução convertendo o problema em uma sequência de subproblemas irrestritos cujas soluções se aproximam da solução do problema original.

O problema de otimização não linear irrestrita consiste em minimizar uma função objetivo não linear $f : \mathbb{R}^n \to \mathbb{R}$ sobre todos os vetores $\mathbf{x} \in \mathbb{R}^n$. Em geral, a estrutura básica dos algoritmos que buscam por uma solução desse problema iterativamente consiste em, a partir de um ponto inicial, gerar uma sequência de pontos $\{\mathbf{x}_k\}_{k=0}$ até que o procedimento não possa mais continuar ou uma solução com precisão adequada para o problema seja encontrada. Os novos pontos são determinados a partir de informações locais sobre a função objetivo no ponto atual e/ou sobre pontos em iterações anteriores, de forma que $\lim_{k\to\infty} f(\mathbf{x}_k) = f(\mathbf{x}^*)$ onde $\mathbf{x}^*$ é um minimizador local da função f. Entre as técnicas para a solução deste problema, sendo a função objetivo suficientemente contínua, as estratégias de busca em linha e aquelas baseadas em região de confiança estão entre os métodos mais populares. Nenhum desses métodos apresenta um comportamento claramente superior ao outro, e ambos são utilizados em softwares comerciais de otimização (Dennis e Schnabel, 1989).

Esses métodos utilizam informações de um modelo quadrático da função objetivo em torno do ponto $\mathbf{x}_k$ para, na iteração k, definir o novo ponto $\mathbf{x}_{k+1}$. Enquanto os métodos com busca em linha definem, a priori, uma direção de decrescimento baseada nesse modelo, a tarefa principal consiste em determinar um passo $\alpha > 0$ ao longo dessa direção para definir o novo ponto. Por outro lado, os métodos baseados em região de confiança definem uma região dentro da qual o modelo possa ser considerado uma boa representação da função objetivo e buscam por um minimizador desse modelo dentro da região de confiança para definir o novo ponto. Propomos aqui um método que utiliza o modelo quadrático para parametrizar uma curva sobre a qual um método com região de confiança baseado no mesmo modelo determinaria os novos pontos $\mathbf{x}_{k+1}$ para diferentes tamanhos da região de confiança e, com uma técnica similar à de busca em linha, determinar o ponto $\mathbf{x}_{k+1}$ sobre essa curva. Dessa forma, a iteração de nosso método envolve os passos de: (i) parametrizar o caminho curvilíneo definido pelas soluções dos sub-problemas de região de confiança para uma variação contínua do valor do raio de confiança; (ii) determinar um minimizador da função objetivo sobre essa curva através de uma busca unidimensional sobre o parâmetro escalar da curva; (iii) atualizar a solução corrente; (iv) atualizar a aproximação quadrática da função objetivo.

O restante do trabalho está dividido na seguinte ordem. Na seção 2, apresentamos uma visão geral dos métodos baseados em busca em linha e região de confiança para que possamos abordar as estratégias utilizadas nesses métodos e que são utilizadas pelo método proposto. Na seção 3, ilustramos o algoritmo implementado para o método e apresentamos alguns resultados de sua convergência. Na seção 4, apresentamos alguns resultados numéricos alcançados com o método e, por fim, na seção 5, discutimos nossas conclusões e trabalhos futuros.

2. Métodos de busca em linha e região de confiança

Denotamos $\nabla f(\mathbf{x}_k) = \nabla f_k$, $H_k \in \mathbb{R}^{n\times n}$ uma matriz simétrica definida positiva. De forma geral, um método com busca em linha consiste em identificar, a cada iteração k, uma direção de decrescimento $\mathbf{d}_k \in \mathbb{R}^n$ de f em $\mathbf{x}_k$, isto é $\mathbf{d}_k^T \nabla f_k < 0$, e um deslocamento a partir do ponto $\mathbf{x}_k$ na direção $\mathbf{d}_k$ para obter o novo ponto $\mathbf{x}_{k+1}$, isto é $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \mathbf{d}_k$ com $\alpha > 0$ e $f_{k+1} < f_k$. Os diferentes métodos de busca em linha utilizam estratégias diversas para avaliar o ponto $\mathbf{x}_{k+1}$, seja na forma como determinam a solução do (sub)problema unidimensional $\min_{\alpha} \varphi(\alpha) = f(\mathbf{x}_k + \alpha \mathbf{d}_k)$ ou a direção $\mathbf{d}_k$. O método mais simples consiste em tomar $\mathbf{d}_k = -\nabla f_k$, mas a convergência do algoritmo resultante geralmente é lenta. As estratégias onde $\mathbf{d}_k = -\nabla^2 f(\mathbf{x}_k)^{-1} \nabla f_k$ são comumente relacionadas à classe dos métodos de Newton. Nesses métodos, a direção $\mathbf{d}_k$ pode não ser uma direção de decrescimento - se a Hessiana não é definida positiva. Métodos de Newton práticos tratam esses casos modificando a matriz ou calculando aproximações para $\mathbf{d}_k$. Métodos *quasi*-Newton são bastante utilizados quando a matriz hessiana de f não está disponível ou seu cálculo é proibitivo. Nessas estratégias $\mathbf{d}_k = -H_k^{-1} \nabla f_k$, onde $\{H_k\}$ é uma sequência de aproximações da hessiana de f, $H_{k+1} = H_k + E$ e E é uma matriz de posto 1 ou 2. Os subproblemas unidimensionais geralmente são resolvidos de forma inexata, através de interpolações polinomiais de $\varphi(\alpha)$ ou através de técnicas de redução de incerteza.

Para obter uma solução do problema de otimização irrestrita, os métodos baseados na estratégia de região de confiança definem uma região em torno do ponto da atual iteração, dentro da qual um modelo quadrático possa ser uma boa aproximação para a função objetivo e determinam uma direção que minimize o modelo dentro dessa região. Isso é, cada iteração consiste em identificar a solução de subproblemas da forma:

$$\underset{\mathbf{d}=\mathbf{x}-\mathbf{x}_k}{\text{minimize}}\ m_k(\mathbf{d}) \stackrel{def}{=} f_k + \nabla f_k^T \mathbf{d} + 0.5 \mathbf{d}^T H_k \mathbf{d} \quad sujeito\ \text{à}\ \|\mathbf{d}\| \leq \Delta_k$$

onde $\Delta_k > 0$ é o raio de confiança na iteração k. A direção escolhida, $\bar{\mathbf{d}} = \bar{\mathbf{x}} - \mathbf{x}_k$ pode não ser satisfatória – $f_k < f(\bar{\mathbf{x}})$ – se o modelo quadrático não é uma boa representação para a função objetivo dentro da região. Nesses casos, o raio é reduzido e o subproblema resolvido novamente. Como nos métodos com busca em linha, diferentes escolhas para a matriz H_k podem ser feitas. Mas, diferentemente dos métodos com busca em linha, mesmo hessianas ou aproximações não definidas positivas podem ser utilizadas diretamente nos subproblemas. A solução dos subproblemas pode ser facilmente identificada para alguns casos mas, em geral, a solução exata requer algum esforço para ser calculada. Entretanto, os algoritmos baseados em região de confiança práticos obtêm boa convergência apenas buscando por soluções aproximadas para os subproblemas.

Métodos com busca em linha geralmente requerem mais iterações para encontrar um minimizador de f quando comparados com métodos de região de confiança, mas as iterações de uma busca em linha tendem a ser menos custosas (Dennis e Schnabel, 1989). De fato, nos métodos baseados em região de confiança as soluções dos subproblemas podem levar a pontos que são rejeitados durante uma iteração do método, sendo necessários a atualização do raio de confiança e o recálculo da solução do subproblema. (Dennis e Schnabel, 1989) fornecem um tratamento mais extensivo, incluindo algumas comparações entre os métodos. (Bazaraa et al., 2006) e (Nocedal e Stephen, 1999) apresentam vários algoritmos práticos para ambos os métodos.

3. Método proposto

(Peressini et al., 1988) mostram que, aplicando-se as condições de *Karush-Kuhn-Tucker* (KKT) ao subproblema do método com região de confiança na iteração k, existem um multiplicador $\lambda_k^* \geq 0$ e um minimizador $\mathbf{x}_k^*$ do modelo quadrático m_k que satisfazem o sistema

$$[H_k + \lambda_k^* \mathbb{I}_n](\mathbf{x}_k^* - \mathbf{x}_k) = -\nabla f_k.$$

tais que a cada λ_k^* corresponde um único $\mathbf{x}_k^*$ e, além disso:

$$\begin{aligned} &\text{se } \|H_k^{-1}\nabla f_k\| \leq \Delta_k \text{ então } \lambda_k = 0; \\ &\text{caso contrário, } \|\mathbf{x}_k^* - \mathbf{x}_k\| = \Delta_k. \end{aligned}$$

Note que, a partir de $\Delta_k = \|H_k^{-1}\nabla f_k\|$ e tomando-se valores cada vez menores de Δ_k, os valores λ_k^* crescem e a direção tomada pelo método se aproxima de $-\nabla f_k$ quando $\lambda_k^* \to \infty$, descrevendo uma curva que vai desde o passo de (*quasi*) Newton $\mathbf{x}_k - H_k^{-1}\nabla f_k$ até $\mathbf{x}_k$. Estratégias como o método de *Levenberg-Marquardt* ou o método proposto por (Dennis et al., 1991), conhecidos como métodos de busca curvilíneos, determinam pontos $\mathbf{x}_{k+1}$ resolvendo sistemas $\Gamma_k(\varepsilon_k)(\mathbf{x}_k^* - \mathbf{x}_k) = \nabla f_k$, $\Gamma_k(\varepsilon) = [H_k + \varepsilon \mathbb{I}_n]$, com o auxílio de uma decomposição para $\Gamma_k(\varepsilon_k)$. Note que o ponto obtido pode não satisfazer $f_{k+1} < f_k$ e, nesse caso, o valor de ε_{k+1} deve ser ajustado − por exemplo $\varepsilon_{k+1} = 4\varepsilon_k$ − e o sistema recalculado. De forma geral, os valores de ε são ajustados de forma análoga ao ajuste do raio de confiança de um método baseado em região de confiança clássico de acordo com o comportamento do modelo quadrático da função, com a diferença que no caso de pouca conformidade entre o modelo e a função a redução do raio corresponderia ao aumento do valor de ε e, no caso de maior conformidade, o aumento do raio corresponderia à diminuição de ε.

Baseados nestas ideias, propomos um método no qual parametrizamos a curva trust-region sobre o intervalo $[0, 1]$ e procuramos por um minimizador aproximado de f sobre essa curva através de uma busca por seção áurea sobre o intervalo. A parametrização é obtida da seguinte maneira: para cada valor de Δ_k no intervalo $(0, \|H_k^{-1}\nabla f_k\|]$ o ponto $\mathbf{x}_k^*$ e o multiplicador λ_k^* fornecidos pelas pelas condições de (KKT) no subproblema trust-region na iteração k são tais que $\nabla m_k(\mathbf{x}_k^* - \mathbf{x}_k) + \lambda_k^* \nabla g(\mathbf{x}_k^* - \mathbf{x}_k) = 0$, onde a restrição do subproblema é escrita na forma $g(\mathbf{d}) = \|\mathbf{d}\|^2 \leq \Delta_k^2$. Para cada valor de $\lambda_k^* > 0$, tomamos um valor $\alpha \in (0, 1)$ com $\lambda_k^* = \alpha^{-1}(1 - \alpha)$. Note que $\alpha \to 0$, $\lambda_k^* \to \infty$ e $\alpha \to 1$, $\lambda_k^* \to 0$. Além disso

$$\begin{aligned} \alpha \nabla m_k(\mathbf{x}_k^* - \mathbf{x}_k) &= -(1 - \alpha)\nabla g(\mathbf{x}^* - \mathbf{x}_k) \\ \alpha[\nabla f_k + H_k(\mathbf{x}^* - \mathbf{x}_k)] &= -2(1 - \alpha)(\mathbf{x}_k^* - \mathbf{x}_k) \\ [\alpha H_k + 2(1 - \alpha)\mathbb{I}_n](\mathbf{x}_k^* - \mathbf{x}_k) &= -\alpha \nabla f_k \end{aligned}$$

de forma que, variando-se um parâmetro $\alpha \in [0, 1]$ percorremos uma curva trust-region que vai desde o ponto $\mathbf{x}_k$, para $\alpha = 0$, até o ponto $\mathbf{x}_k - H_k^{-1}\nabla f_k$, quando $\alpha = 1$. O ponto $\mathbf{x}_{k+1}$ é determinado através da minimização aproximada de f sobre essa curva com a aplicação do método da seção áurea sobre o intervalo $[0, 1]$ e a resolução dos sistemas $[\alpha H_k + 2(1 - \alpha)\mathbb{I}_n](\mathbf{x}^* - \mathbf{x}_k) = -\alpha \nabla f_k$ para cada valor de α gerado pelo método. Além disso, $A_k(\alpha) = [\alpha H_k + 2(1 - \alpha)\mathbb{I}_n]$ é definida positiva para

$0 \leq \alpha \leq 1$ e, portanto, $-\nabla f_k^T A_k(\alpha)^{-1} \nabla f_k < 0$ pelo que a direção $-A_k(\alpha)^{-1}\nabla f_k$ é uma direção de decrescimento de f em $\mathbf{x}_k$ se $\nabla f_k \neq 0$.

Métodos com busca em linha com direção *quasi*-Newton baseada em aproximações *BFGS* impõem que o ponto $\mathbf{x}_{k+1}$ escolhido satisfaça uma condição de decrescimento suficiente $f_{k+1} \leq f_k + \beta \nabla f_k(\mathbf{x}_{k+1} - \mathbf{x}_k)$ e uma condição de curvatura $s_k^T y_k > 0$, onde $s_k = (\mathbf{x}_{k+1} - \mathbf{x}_k)$ e $y_k = (\nabla f_{k+1} - \nabla f_k)$, de maneira que esta última condição garante que a atualização BFGS $H_{k+1} = H_k + E$ seja definida positiva, desde que H_k seja definida positiva. No método proposto, procuramos por pontos que satisfaçam a condição de *Armijo* mas não impomos a condição de curvatura positiva, uma vez que pontos que satisfaçam as duas condições eventualmente necessitariam ser escolhidos com parâmetros $\alpha > 1$ e, nesse caso, a matriz $A(\alpha)$ poderia não ser definida positiva. Para contornar essa questão, utilizamos atualizações *DampedBFGS* (Nocedal e Stephen, 1999) para $\{H_k\}$:

$$
\begin{aligned}
H_{k+1} &= H_k - \frac{H_k s_k s_k^T H_k}{s_k^T H_k s_k} + \frac{r_k^T r_k}{s_k^T r_k} \\
r_k &= \theta_k y_k + (1 - \theta_k) H_k s_k. \\
\theta_k &= \begin{cases} 1, & \text{se } s_k^T y_k \geq 0.2 s_k^T H_k s_k; \\ (0.8 s_k^T H_k s_k)(s_k^T H_k s_k - s_k^T y_k)^{-1}, & \text{se } s_k^T y_k < 0.2 s_k^T H_k s_k; \end{cases}
\end{aligned}
$$

Nos casos onde $s_k^T y_k$ é positivo mas não muito pequeno, a atualização se afasta um pouco da aproximação atual mas ainda se mantém definida positiva, de forma que a atualização é a mesma obtida pela fórmula BFGS padrão. Por outro lado, se $0.2 s_k^T H_k s_k > s_k^T y_k$ então

$$
\begin{aligned}
s_k^T r_k &= \theta_k s_k^T y_k + (1 - \theta_k s_k^T H_k s_k) \\
&= s_k^T H_k s_k + \theta_k (s_k^T y_k - s_k^T H_k s_k) \\
&= s_k^T H_k s_k + (0.8 s_k^T H_k s_k)(s_k^T H_k s_k - s_k^T y_k)^{-1}(s_k^T y_k - s_k^T H_k s_k) \\
&= 0.2 s_k^T H_k s_k > 0 \text{ já que } H_k \text{ é definida positiva.}
\end{aligned}
$$

portanto, a nova matriz H_{k+1} é definida positiva.

Pontos $\mathbf{x}_{k+1}$ que satisfaçam a condição de Armijo sempre podem ser encontrados sobre a curva trust-region, desde que $\|\mathbf{x}_{k+1} - \mathbf{x}_k\|$ seja suficientemente pequeno, como mostrado em (Dennis et al., 1991). Note que, na nossa parametrização, para que isso ocorra basta que α seja suficientemente próximo de 0.

O algoritmo 1 ilustra as estratégias adotadas para, a partir de um ponto $\mathbf{x}_k$ e percorrendo-se a curva trust-region parametrizada através de um método de busca unidimensional com seção áurea, obter um novo ponto $\mathbf{x}_{k+1}$, com $f_{k+1} < f_k$, utilizando uma matriz H_k aproximada pela atualização *DampedBFGS* para a hessiana de f.

Uma técnica que alguns algoritmos de busca em linha com direção *quasi*-Newton utilizam para obter rápida taxa de convergência é avaliar primeiro o ponto $\mathbf{x}_k - H_k^{-1}\nabla f_k$ antes de executar a busca unidimensional. Se esse ponto satisfizer as condições de decrescimento suficiente e de curvatura então é aceito como o novo ponto $\mathbf{x}_{k+1}$ (Nocedal e Stephen, 1999). No nosso algoritmo, antes da busca por seção áurea sobre o intervalo $[0, 1]$, também avaliamos primeiro o passo $\alpha = 1$ e o aceitamos se ele satisfizer apenas a condição de Armijo.

Para funções f que não sejam *quasi*-convexas, os valores ω_1 e μ_1, obtidos pelo método da seção áurea, podem gerar pontos $\mathbf{x}_{\omega_1}$ e $\mathbf{x}_{\mu_1}$ para os quais $f(\mathbf{x}_{\omega_1}) > f_k$ e $f(\mathbf{x}_{\mu_1}) > f_k$ e, portanto, não há garantias de que o intervalo de incerteza gerado pelo método no passo seguinte contenha um ponto com valor de função menor que o ponto $\mathbf{x}_k$. Nesses casos, o algoritmo reduz o intervalo inicial $[0, 1]$ para $[0, b_1]$, $b_1 = \omega_1$, e refaz o algoritmo da seção áurea sobre esse intervalo. Caso a situação de não decrescimento de f continue para ambos os novos pontos obtidos, o intervalo é novamente reduzido até que algum dos pontos $\mathbf{x}_{\omega_1}$ ou $\mathbf{x}_{\mu_1}$ tenha menor valor de função do que o ponto $\mathbf{x}_k$.

Algorithm 1: A partir do ponto $\mathbf{x}_k$, $\mathbf{x}_{k+1}$ é uma nova aproximação para o minimizador de f obtido ao longo da curva trust-region. $\Phi = \frac{\sqrt{5}-1}{2}$.

Dados $\mathbf{x}_k$, ∇f_k, H_k;
se $f(\mathbf{x}_k - H_k^{-1}\nabla f_k) \leq f_k + \beta \nabla f_k^T[-H_k^{-1}\nabla f_k]$ **então**
  $\bar{\mathbf{x}} = \mathbf{x}_k - H_k^{-1}\nabla f_k$;
senão
  Intervalo inicial $[a_1 = 0, b_1]$, $\omega_1 = (1-\Phi)b_1$, $\mu_1 = \Phi b_1$, $i = 1$;
  Resolva os sistemas $A_k(\omega_i)^{-1}[\mathbf{x}_{\omega_i} - \mathbf{x}_k] = -\omega_i \nabla f_k$,
  $A_k(\mu_i)^{-1}[\mathbf{x}_{\mu_i} - \mathbf{x}_k] = -\mu_i \nabla f_k$;
  $\bar{\mathbf{x}} = \underset{\mathbf{x}_{\omega_i}, \mathbf{x}_{\mu_i}}{\operatorname{argmin}} f(\mathbf{x})$;
  enquanto $f(\bar{\mathbf{x}}) > f_k + \beta \nabla f_k^T[\bar{\mathbf{x}} - \mathbf{x_k}]$ **faça**
    se $f(\mathbf{x}_{\omega_i}) > f(\mathbf{x}_{\mu_i})$ **então**
      $\mathbf{x}_{\omega_{i+1}} = \mathbf{x}_{\mu_i}$;
      $a_{i+1} = \omega_i$ $b_{i+1} = b_i$, $\omega_{i+1} = \mu_i$, $\mu_{i+1} = a_{i+1} + \Phi(b_{i+1} - a_{i+1})$;
      Resolva $A_k(\mu_{i+1})^{-1}[\mathbf{x}_{\mu_{i+1}} - \mathbf{x}_k] = -\mu_{i+1}\nabla f_k$.
    senão
      $\mathbf{x}_{\mu_{i+1}} = \mathbf{x}_{\omega_i}$;
      $a_{i+1} = a_i$, $b_{i+1} = \mu_i$, $\mu_{i+1} = \omega_i$, $\omega_{i+1} = a_{i+1} + (1-\Phi)(b_{i+1} - a_{i+1})$;
      Resolva $A_k(\omega_{i+1})^{-1}[\mathbf{x}_{\omega_{i+1}} - \mathbf{x}_k] = -\omega_{i+1}\nabla f_k$.
    $\bar{\mathbf{x}} = \underset{\mathbf{x}_{\omega_{i+1}}, \mathbf{x}_{\mu_{i+1}}}{\operatorname{argmin}} f(\mathbf{x})$;
    $i = i + 1$;
$\mathbf{x}_{k+1} = \bar{\mathbf{x}}$;

A figura 1 ilustra a iteração k do algoritmo para uma função com duas variáveis de otimização. O passo de Newton a partir do ponto $\mathbf{x}_k$ não fornece redução suficiente na função objetivo e o algoritmo procede para a busca unidimensional sobre a curva trust-region, dividindo o intervalo $[0, 1]$ em intervalos de confiança $[a_i, b_i]$ para os quais existe um valor $\alpha \in [a_i, b_i]$ com $f(\mathbf{x}_k - \alpha A_k(\alpha)^{-1}\nabla f_k) < f_k$. Quando a redução na função objetivo satisfaz a condição de Armijo, encontramos o ponto $\mathbf{x}_{k+1}$.

4. Resultados numéricos

Nesta seção, discutimos o comportamento prático do algoritmo proposto para alguns problemas de otimização irrestrita listados abaixo. As funções $f(\mathbf{x}) = (x_1, ..., x_n)$ foram extraídas de (Moré et al., 1981) e (Conn et al., 1986). Aproximamos os gradientes dessas funções f através do método de diferenças finitas. Utilizamos um parâmetro $\beta = 10^{-4}$ para o critério de decrescimento suficiente e $H_0 = \mathbb{I}_n$ como estimativa inicial da hessiana. Implementamos o algoritmo sobre a plataforma **Matlab** e listamos os

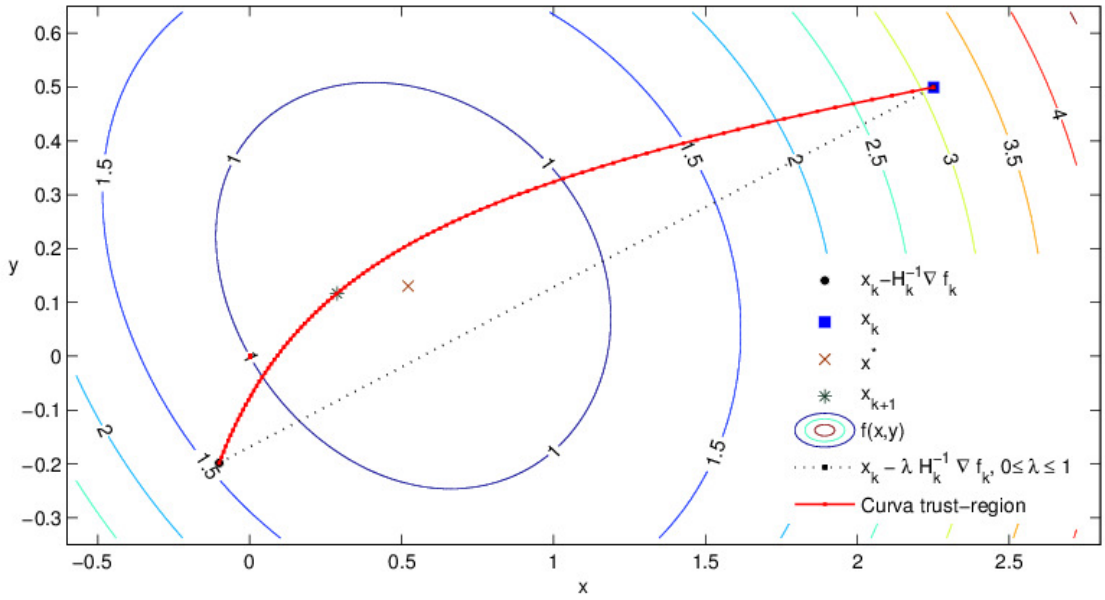


Figura 1: Iteração k do algoritmo para a função $f(x,y) = e^{-x-y} + 0.5(x^2 + 4y^2)$ com solução em $\mathbf{x}^* = (0.5212, 0.1303)$ e $f(\mathbf{x}^*) = 0.6910$. O passo de Newton a partir do ponto $\mathbf{x}_k$ não fornece redução suficiente na função objetivo e o ponto $\mathbf{x}_{k+1}$ é obtido através da busca unidimensional sobre a curva trut-region parametrizada (linha sólida).

resultados obtidos. Como referência para comparação, também listamos os resultados obtidos com as implementações de métodos com busca em linha e baseados em região de confiança fornecidos pela plataforma Matlab. O algoritmo de busca em linha utilizado é um método *quasi*-Newton com aproximações BFGS e interpolação cúbica. O segundo algoritmo é um método de região de confiança com subespaços baseado no método de Newton reflexivo com gradiente conjugado (MathWorks, 2012). Em todos os algoritmos, adotamos como critério de convergência $\|f_{k+1} - f_k\|_\infty \leq 10^{-6}$, $\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_\infty \leq 10^{-6}$. Além disso, para que pudéssemos fazer algumas comparações entre os algoritmos, também adotamos como critério de parada $\|f^* - f_k\|_\infty \leq 10^{-4}$, onde f^* é o valor ótimo para a função f. Dessa forma, se na iteração k, $\mathbf{x}_k$ for tal que $\|f_k - f^*\|_\infty \leq 10^{-4}$, consideramos que o algoritmo solucionou o problema e reportamos as métricas − avaliações de função e gradiente − até esse ponto . Por outro lado, se o algoritmo faz pouco progresso numa iteração de acordo com um dos outros critérios então o algoritmo termina. Assim, as soluções obtidas com cada método tem aproximadamente a mesma qualidade ($\|f^* - f_k\|_\infty \approx 10^{-4}, 10^{-6}$) e comparamos os métodos de acordo com o número de avaliações executadas da função objetivo e de seu gradiente.

As funções $f : \mathbb{R}^n \to \mathbb{R}$ avaliadas são:

1. $f(\mathbf{x}) = (2x_1 + x_2 + ... + x_n - n - 1)^2 + (x_1 + 2x_2 + ... + x_n - n - 1)^2 + ... + (x_1 + x_2 + ...2x_{n-1})^2 + (x_1 x_2 ... x_n - 1)^2$. $\mathbf{x}^* = (1, ..., 1)$ e $f(\mathbf{x}^*) = 0$. $\mathbf{x}_1 = (x_1, ..., x_n)$, $-\frac{-e}{2} \leq x_i \leq \frac{-e}{2}$.

2. $f(\mathbf{x}) = \sum_{i=1}^{n-1}[(x_i^2)^{(x_{i+1}^2+1)} + (x_{i+1}^2)^{(x_i{}^2+1)}]$. $\mathbf{x}^* = (0, 0, ..., 0)$ e $f(\mathbf{x}^*) = 0$. $\mathbf{x}_1 = (x_1, ..., x_n)$, $-3 \leq x_i \leq 3$.

3. $f(\mathbf{x}) = 1 + \sum_{i\in J}[(100(x_{i+1} - x_i^2)^2) + (1 - x_i)^2 + 90(x_{i+3} - x_{i+2}^2) + (1 - x_{i+2})^2 + 10(x_{i+1} + x_{i+3} - 2)^2]$, onde n é múltiplo de 4 e J={1,5,9,...}. $\mathbf{x}^* = (1, 1, 1, 1, ...)$ e $f(\mathbf{x}^*) = 1$. $\mathbf{x}_1 = (x_1, ..., x_n)$, $-3 \leq x_i \leq 3$.

4. $f(\mathbf{x}) = \sum_{i\in J}[(x_i - 10x_{i+1})^2) + 5(x_{i+2} - x_{i+3})^2 + (x_{i+1}^2) - 2x_{i+2})^4 + 10(x_i - x_{i+3})^4]$, onde n é múltiplo de 4 e J={1,5,9,...}. $\mathbf{x}^* = (0, 0, 0, 0, ...)$ e $f(\mathbf{x}^*) = 0$. $\mathbf{x}_1 = (x_1, ..., x_n)$, $-3 \leq x_i \leq 3$.

5. $f(\mathbf{x}) = \sum_{i\in J}[(e^{x_i} - x_{i+1})^4 + 100(x_{i+1} - x_{i+2})^6 + tan^4(x_{i+2} - x_{i+3}) + x_i^8 + (x_{i+3} - 1)^2]$ onde n é múltiplo de 4 e J={1,5,9,...}. $\mathbf{x}^* = (0, 1, 1, 1, ..., 0, 1, 1, 1)$ e $f(\mathbf{x}^*) = 0$. $\mathbf{x}_1 = (x_1, ..., x_n)$, $-1.5 \leq x_i \leq 1.5$.

Para cada problema, listamos nas tabelas seguintes o número total de variáveis de otimização, n, o número médio de avaliações da função objetivo e de seu gradiente (f/g), o número de vezes que o algoritmo encontrou solução com a qualidade desejada (sol.) e o tempo gasto (t) pelo algoritmo, em segundos. Cada linha nas tabelas representa os resultados da execução dos algoritmos num conjunto de 50 pontos $\mathbf{x}_0 \in \mathbb{R}^n$. As referências para as funções f fornecem apenas uma escolha para $\mathbf{x}_0$, pelo que geramos os 50 diferentes pontos $\mathbf{x}_0 = (x_1, ..., x_n)$ para os nossos testes atribuindo a cada x_i um valor extraído uniformemente de um intervalo que contenha o valor sugerido para a variável x_i. Listamos o tempo apenas para ilustrar que, embora o algoritmo com região de confiança consiga resolver os problemas para um maior número de pontos iniciais, o tempo gasto é muito maior do que nos outros algoritmos. Isso não se deve a um número muito maior de avaliações de função ou gradiente e sim à complexidade do método.

	proposto			**line search**			**trust-region**		
n	f/g	sol.	t(s)	f/g	sol.	t(s)	f/g	sol.	t(s)
5	37/16	44	0.5	21/21	49	0.9	18/106	46	3.5
10	26/11	48	0.4	28/28	47	1.5	13/144	39	6.3
15	42/15	44	0.8	28/28	50	1.9	16/255	44	14.3
20	85/24	33	1.7	29/29	48	2.5	18/373	45	26.1
25	30/10	50	0.9	25/25	50	2.5	20/511	35	43.0
30	30/10	50	1.0	22/22	50	2.6	17/527	45	52.0
35	83/19	43	2.6	25/25	50	3.3	21/770	40	87.5
40	32/10	50	1.4	22/22	50	3.2	20/828	40	106.8
45	33/11	50	1.7	22/22	50	3.5	20/928	41	134.0
50	32/10	50	1.7	22/22	50	3.9	21/1075	41	171.8

Tabela 1: Resultados obtidos para o problema **1**

Observe que no método proposto o número de avaliações do gradiente da função objetivo é menor do que nos demais métodos. De fato, o algoritmo apenas avalia o gradiente para os pontos $\mathbf{x}_k$ determinados em cada iteração do algoritmo, ao passo que no algoritmo com busca em linha os gradientes de outros pontos são avaliados na interpolação da função unidimensional $\varphi(\alpha) = f(\mathbf{x}_k + \alpha\mathbf{d}_k)$. No algoritmo com região de confiança, embora o número de avaliações de função seja bem menor em todos os testes, um número bem maior de avaliações de gradiente é necessário. O algoritmo avaliado utiliza um método de gradientes conjugados para determinar um subespaço

	proposto			line search			trust-region		
n	f/g	sol.	t(s)	f/g	sol.	t(s)	f/g	sol.	t(s)
5	62/21	46	0.4	35/35	35	1.2	15/92	50	2.6
10	101/34	44	1.0	47/47	23	1.9	22/237	50	8.3
15	124/41	40	1.8	61/61	13	3.2	25/403	50	18.5
20	153/52	42	3.1	62/62	9	4.2	27/577	50	34.5
25	173/57	41	4.4	65/65	9	5.5	29/762	50	57.7
30	183/60	41	5.8	64/64	7	6.6	30/932	50	87.5
35	197/64	39	8.7	67/67	5	8.4	31/1103	50	125.7
40	212/69	35	11.1	68/68	2	10.1	32/1311	50	177.8
45	229/75	37	14.1	70/70	1	12.0	33/1503	50	239.5
50	238/76	35	16.6	70/70	1	13.7	34/1740	50	321.8

Tabela 2: Resultados obtidos para o problema **2**

	proposto			line search			trust-region		
n	f/g	sol.	t(s)	f,g	sol.	t(s)	f,g	sol.	t(s)
4	143/38	50	0.8	56,56	50	1.6	102,509	50	13.4
8	303/69	50	1.9	110,110	50	3.7	43,383	50	11.4
12	453/94	50	3.3	163,163	49	6.6	51,657	50	22.5
16	631/126	50	5.5	204,204	49	9.8	58,991	50	40.3
20	789/148	50	7.7	247,247	49	13.7	63,1316	50	62.4
24	952/171	50	10.5	290,290	49	18.4	69,1724	50	94.6
28	1128/191	50	13.5	321,321	46	23.0	71,2049	50	126.6
32	1283/205	50	16.6	361,361	48	28.6	70,2320	49	161.0
36	1451/228	50	23.8	394,394	49	34.7	76,2819	50	216.7
40	1627/244	50	28.5	430,430	49	41.4	77,3164	50	268.2

Tabela 3: Resultados obtidos para o problema **3**

definido pelo gradiente da função e por uma direção de (*quasi-*)Newton em cada subproblema. Dessa forma, o método é robusto (conhecido como um algoritmo *Large Scale*) mas baseia-se na capacidade de executar um grande número de avaliações do gradiente da função objetivo.

O algoritmo com busca em linha somente conseguiu obter solução para um maior número de pontos iniciais no problema 1, embora os outros algoritmos também consigam encontrar solução para a maior parte dos pontos testados nesse problema. Nos demais problemas, os algoritmos proposto e com região de confiança obtêm solução para um maior número de pontos iniciais do que o algoritmo com busca em linha − o algoritmo começa a executar passos muito pequenos ($\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_\infty \leq 10^{-6}$) ou reduz a função objetivo muito lentamente ($\|f_{k+1} - f_k\|_\infty \leq 10^{-6}$) antes de atingir uma solução com a qualidade desejada. No que diz respeito ao número de avaliações de função e gradiente, mesmo considerando a soma dessas duas métricas o algoritmo proposto faz menos chamadas a essas duas funções do que o algoritmo com região de confiança em todos os problemas de teste, embora não consiga obter solução para todos os pontos iniciais avaliados nos problemas 1, 2 e 4. O algoritmo baseado em região de confiança consegue obter solução para todos os pontos iniciais nos proble-

	proposto			line search			trust-region		
n	f/g	sol.	t(s)	f/g	sol.	t(s)	f/g	sol.	t(s)
4	105/25	50	0.5	52/52	19	1.5	167/837	50	21.3
8	230/39	49	1.2	107/107	3	3.7	37/332	50	10.3
12	346/52	49	2.1	158/158	1	6.7	27/350	50	12.9
16	454/61	49	3.1	198/198	0	10.3	27/464	50	20.6
20	553/71	49	4.5	235/235	0	14.2	29/610	50	31.9
24	644/83	49	6.0	260/260	0	18.2	31/764	50	46.6
28	723/93	49	7.8	283/283	0	22.7	34/984	50	71.6
32	811/105	47	10.1	309/309	0	28.1	33/1100	50	91.1
36	895/111	47	13.9	326/326	0	33.1	34/1260	50	117.5
40	968/121	48	17.0	338/338	0	38.3	36/1469	50	152.8

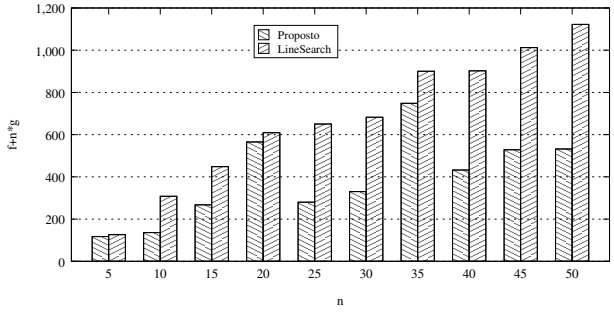
Tabela 4: Resultados obtidos para o problema **4**

	proposto			line search			trust-region		
n	f/g	sol.	t(s)	f/g	sol.	t(s)	f/g	sol.	t(s)
4	60/22	34	0/4	47/47	38	1/4	17/87	40	2/5
8	115/43	27	1/2	99/99	23	3/7	20/182	24	6/1
12	153/62	20	2/3	141/141	15	6/6	25/319	18	12/8
16	188/76	17	3/7	173/173	8	9/9	51/860	13	50/1
20	223/91	15	5/4	200/200	8	13/8	34/722	8	43/6
24	263/108	11	7/8	221/221	4	18/1	37/927	6	66/4
28	291/116	9	9/9	231/231	4	21/9	40/1159	2	97/1
32	318/129	7	12/8	249/249	3	26/9	47/1535	2	148/6
36	356/147	6	19/0	256/256	2	31/5	53/1948	1	215/7
40	375/148	3	21/7	262/262	1	36/0	57/2323	1	290/8

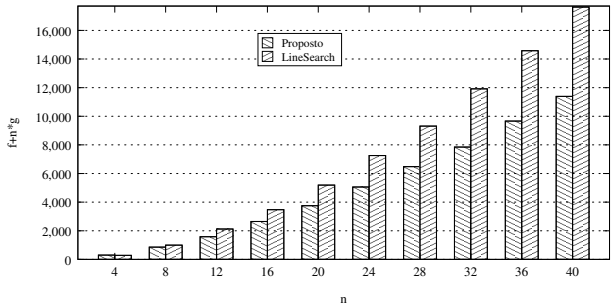
Tabela 5: Resultados obtidos para o problema **5**

mas 2 e 4, mas o número de avaliações do gradiente é bastante superior ao número de avaliações feitas no algoritmo proposto, que ainda assim obtém solução na maior parte dos pontos iniciais nesses problemas − no problema 3 obtém solução em todos os pontos, no problema 5 consegue obter mais soluções do que os outros dois algoritmos. Os resultados indicam que, para as funções que consideramos nos testes, o algoritmo proposto tem convergência local próxima ao algoritmo com região de confiança avaliado. Considerando as avaliações de função e gradiente juntas, o algoritmo proposto executa menos chamadas a essas funções até alcançar soluções de mesma qualidade que os algoritmos avaliados.

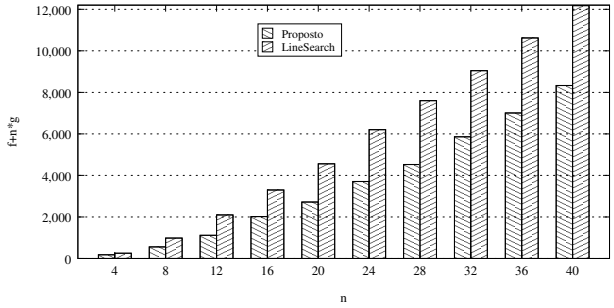
Utilizamos um método de diferenças finitas para aproximar os gradientes das funções. Dessa forma, considerando que são necessárias n avaliações da função objetivo para a aproximação de seu gradiente em um ponto, o algoritmo proposto mostra melhor desempenho que o método com busca em linha no que diz respeito ao número total de avaliações de função. A figura 2 ilustra o número total de avaliações feitas por cada método nos problemas 1, 3 e 5 − nesses problemas, ambos os algoritmos encontram um número próximo de soluções.



(a) Problema 1



(b) Problema 3



(c) Problema 5

Figura 2: Total de avaliações da função objetivo, considerando n avaliações para cada chamada ao gradiente por diferenças finitas, nos algoritmos proposto e com busca em linha.

5. Conclusões

Neste trabalho, propomos um novo método para otimização irrestrita no qual procuramos por minimizadores da função sobre uma curva trust-region parametrizada sobre o intervalo $[0, 1]$. No lugar de resolver o subproblema no método de região de confiança e ajustar um raio de confiança, nosso método utiliza uma estratégia de busca unidimensional para percorrer a curva parametrizada e visitar pontos que seriam obtidos por um método padrão para raios de confiança $\Delta_k \leq \|H_k^{-1}\nabla f_k\|$. Utilizamos uma atualização BFGS adaptada para contornar situações onde os pontos escolhidos, apesar de produzirem decrescimento suficiente em f, não satisfazem a condição de curvatura positiva. Os resultados obtidos com algumas funções teste, embora não determinem a superioridade do método, mostram que o algoritmo consegue resolver problemas para os quais um algoritmo com busca em linha comercial não obtém boa convergência ou um algoritmo comercial baseado em região de confiança, devido à sua maior complexidade, é muito mais custoso.

Deve-se notar que adotamos aqui uma implementação onde os sistemas $A_k(\alpha)(\mathbf{x}^* - \mathbf{x}_k) = -\alpha\nabla f_k$ são solucionados sem o auxílio de uma fatoração de $A_k(\alpha)$. Por exemplo, (Dennis et al., 1991) utilizam uma decomposição $H_k = Q_k T_k Q_k$ e os sistemas $[H_k + \varepsilon\mathbb{I}_n](\mathbf{x}^* - \mathbf{x}_k) = -\nabla f_k$ podem ser solucionados de forma menos custosa. Em trabalhos futuros, pretendemos investigar o efeito da utilização de decomposições desse tipo no desempenho do algoritmo proposto.

Referências

Bazaraa, M. S., Sherali, H. D. e Shetty, C. M. (2006), *Nonlinear Programming: theory and algorithms*, 3 edn, Wiley Interscience.

Conn, A. R., Gould, N. I. M. e Toint, P. L. (1986), 'Testing a Class of Methods for Solving Minimization Problems with Simple Bounds on the Variables'.

Dennis, J. E., Echebest, N., Guardarucci, M. T., Martinez, J. M., Scolnik, H. D. e Vaccino, C. (1991), 'A Curvilinear Search Using Tridiagonal Secant Updates for Unconstrained Optimization', *SIAM Journal on Optimization* **1**(3), 333–357.

Dennis, Jr., J. E. e Schnabel, R. B. (1989), Optimization, Elsevier North-Holland, Inc., New York, NY, USA, chapter A view of unconstrained optimization, pp. 1–72.

MathWorks (2012), 'Matlab optimization toolbox documentation for fminunc', `http://www.mathworks.com/help/toolbox/optim/ug/fminunc.html`.

Michael Gertz, E. (2004), 'A quasi-Newton trust-region method', *Math. Program.* **100**(3), 447–470.

Moré, J. J., Garbow, B. S. e Hillstrom, K. E. (1981), 'Testing Unconstrained Optimization Software', *ACM Trans. Math. Softw.* **7**(1), 17–41.

Nocedal, J. e Stephen, J. W. (1999), *Numerical optimization*, 1 edn, Springer-Verlag.

Peressini, A. L., Sullivan, F. E. e Uhl, J. J. (1988), *The Mathematics of Nonlinear Programming*, Undergraduate Texts in Mathematics, Springer.

Powell, M. J. D. (2010), 'On the convergence of a wide range of trust region methods for unconstrained optimization', *IMA Journal of Numerical Analysis* **30**, 289–301.

Yuan, G., Lu, S. e Wei, Z. (2010), 'A Line Search Algorithm for Unconstrained Optimization', *J. Software Engineering and Applications* (3), 503–509.

Zhu, D. (2001), 'Curvilinear Paths and trust Region Methods With Nonmonotonic Back Tracking Techinique for Unconstrained Optimization', *Journal of Computational Mathematics* **19**(3), 241–258.