

RESULTADOS NUMÉRICOS DO MÉTODO DE REESCALONAMENTO NÃO LINEAR BASEADO NA FUNÇÃO BARREIRA LOGARÍTMICA MODIFICADA

Iara da Cunha R. da Silva

Depto. de Engenharia de Sistemas, FEEC, UNICAMP
13083-852, Campinas, SP
iara@densis.fee.unicamp.br

Anésio dos Santos Júnior

Depto. de Engenharia de Sistemas, FEEC, UNICAMP
13083-852, Campinas, SP
anesio@densis.fee.unicamp.br

RESUMO

Neste artigo é apresentado um algoritmo para o método de reescalonamento não linear baseado na função barreira logarítmica modificada. O algoritmo foi implementado em Matlab e testado em problemas de otimização convexos e não convexos. Os experimentos numéricos apresentados visam analisar os efeitos, sobre a convergência, da variação do parâmetro associado à penalização da função logarítmica.

PALAVRAS-CHAVE. Reescalonamento não linear. Otimização. Função barreira logarítmica modificada. Área Principal: Programação Matemática

ABSTRACT

This paper presents an algorithm for nonlinear rescaling method based on the modified logarithmic barrier function. The algorithm was implemented in Matlab and tested on problems with convex and nonconvex optimization. The numerical experiments presented are intended to analyze the effects on the convergence of the parameter variation associated with the logarithmic penalty function

KEYWORDS. Nonlinear rescaling. Optimization. Modified logarithmic barrier function. Main area: Mathematical Programming.

1. Introdução

O método de reescalonamento baseado na função barreira logarítmica modificada apresentado neste artigo faz parte do grupo dos métodos da Lagrangiana aumentada não quadrática. Em Bertsekas (1982) o autor mostra que os termos de penalidades inseridos na função Lagrangiana de um problema de otimização pode afetar na convergência do método iterativo associado e conseqüentemente em seu desempenho. Uma outra consideração comentada em Polyak e Teboulle (1997) é a respeito da diferenciabilidade da Lagrangiana aumentada quadrática. Esta é apenas uma vez diferenciável, mesmo que a função objetivo e os funcionais das restrições apresentem diferenciabilidade de ordem superior a C^2 . Entretanto, a Lagrangiana modificada pelo reescalonamento não quadrático tem característica de diferenciabilidade garantida pela definição das funções de reescalonamento. A Lagrangiana resultante da aplicação do reescalonamento não linear, baseado na função logarítmica modificada, atende a essa característica que é bastante importante quando o algoritmo utilizado se fundamenta no método de Newton.

Os métodos de reescalonamento consistem em transformar o problema original reescalonando a função objetivo e/ou as restrições obtendo um outro equivalente, ou seja, que possui a mesma solução. A transformação não linear é dependente de um parâmetro positivo e baseia-se em uma função suave, chamada de **função de reescalonamento**. Os passos mais importantes dos métodos de reescalonamento são: minimização irrestrita da função Lagrangiana clássica do problema equivalente e atualizações das variáveis primais e duais. Essas últimas chamadas também de multiplicadores de Lagrange estão relacionadas às restrições de desigualdade e são atualizadas por uma fórmula explícita.

Neste artigo será apresentado um método de reescalonamento de Polyak, em especial o que possui como função de reescalonamento a barreira logarítmica modificada. Esse método foi implementado em Matlab e aplicado em problemas de otimização convexo e não convexo de duas variáveis. Para a verificação do desempenho do método foram realizados testes com vários valores de tolerância que controlam os passos das variáveis (primais e duais) e parâmetros de relaxação.

Este artigo está organizado da seguinte forma: na seção 2 descreve-se o algoritmo do método de reescalonamento não linear baseado na função barreira logarítmica modificada, na seção 3 apresenta-se os resultados obtidos dos testes realizados e na seção 4 finaliza-se o artigo com a conclusão.

2. Método de reescalonamento não linear baseado na função barreira logarítmica modificada

Essa seção tem como objetivo dar uma introdução teórica e apresentar o algoritmo para o método de reescalonamento não linear baseado na função barreira modificada.

A teoria dos métodos de reescalonamento não linear foi desenvolvida por Polyak há mais de duas décadas. Esses métodos foram aplicados inicialmente em problemas de otimização com restrições de desigualdade como podem ser vistos em Polyak e Teboulle (1997), Polyak (2002), Polyak e Griva (2004), Griva e Polyak (2006). A base fundamental desses métodos é transformar um problema de otimização em um outro através de uma função de reescalonamento. Essa função pertence a uma classe de funções suaves Ψ que satisfazem as seguintes propriedades:

1. $\psi(0) = 0, \psi'(0) = 1,$
2. $\psi'(t) > 0,$

3. $\psi''(t) < 0$,
4. $\lim_{t \rightarrow +\infty} \psi'(t) = 0$,

onde ψ é uma função $\in C^2$ definida no intervalo $(a, +\infty)$, $-\infty \leq a < 0$, tal que $\psi(a) = -\infty$ e $\psi'(a) = +\infty$.

Para cada função ψ definida origina-se um tipo de método de reescalonamento não linear. Neste artigo o método de reescalonamento não linear que será apresentado utiliza a função barreira modificada de Polyak definida como: $\psi(t) = \ln(t + 1)$, onde $\psi : R \rightarrow R$.

Para aplicar esse método considere o seguinte problema:

$$\begin{aligned} &\text{Minimizar} && f(x) \\ &\text{sujeito a} && h_i(x) = 0, \quad i = 1, K, m \\ &&& g_j(x) \geq 0 \quad j = 1, K, r \end{aligned} \tag{1}$$

onde $f : R^n \rightarrow R$, $h_i : R^n \rightarrow R$, $g_j : R^n \rightarrow R$, e $f, h_i, g_j \in C^1, \forall i = 1, K, m, \forall j = 1, K, r$.

Para começar a construção do algoritmo é necessário obter o problema equivalente de (1). Primeiramente inserem-se as variáveis de excesso s_j nas restrições de desigualdade no problema (1) obtendo:

$$\begin{aligned} &\text{Minimizar} && f(x) \\ &\text{sujeito a} && h_i(x) = 0, \quad i = 1, K, m \\ &&& g_j(x) - s_j = 0, \quad j = 1, K, r \\ &&& s_j \geq 0 \quad j = 1, K, r \end{aligned} \tag{2}$$

Depois reescalona e parametriza, por um parâmetro de relaxação positivo c , as restrições de desigualdade do problema (2) resultando no seguinte problema equivalente:

$$\begin{aligned} &\text{Minimizar} && f(x) \\ &\text{sujeito a} && h_i(x) = 0, \quad i = 1, K, m \\ &&& g_j(x) - s_j = 0, \quad j = 1, K, r \\ &&& c^{-1} \ln(cs_j + 1) \geq 0 \quad j = 1, K, r \end{aligned} \tag{3}$$

Para comprovar que as restrições reescaladas não alteram a região factível do problema original considere o seguinte problema:

$$\begin{aligned} &\text{Minimizar} && f(x) \\ &\text{sujeito a} && g(x) - s = 0, \\ &&& s \geq 0. \end{aligned} \tag{4}$$

O problema equivalente de (4) é:

$$\begin{aligned} &\text{Minimizar} && f(x) \\ &\text{sujeito a} && g(x) - s = 0, \\ &&& c^{-1} \ln(cs + 1) \geq 0. \end{aligned} \tag{5}$$

Observe na figura 1 que a região factível da restrição $s \geq 0$ não se altera quando a mesma é reescalada. Para diferentes valores de c os pontos que satisfazem a restrição reescalada $c^{-1} \ln(cs + 1) \geq 0$ são os mesmos que satisfazem a restrição $s \geq 0$ do problema (4).

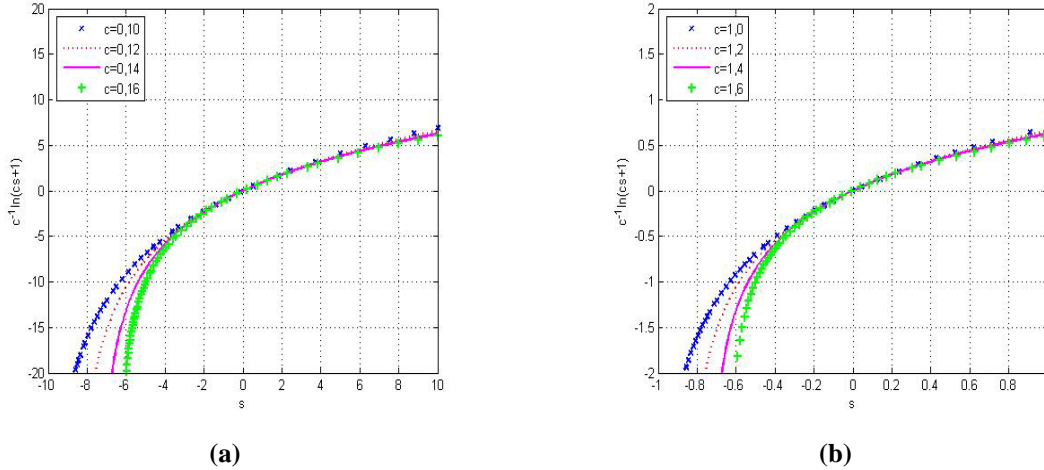


Figura 1 Valores da restrição reescalada para valores de c entre 0 e 1 (a) e para c maior ou igual a 1 (b).

O primeiro passo importante do algoritmo é minimizar problema (3) considerando μ fixo. Isso pode ser feito através das condições necessárias de primeira ordem, ou seja, resolver o seguinte sistema de equações não lineares:

$$\nabla L(x, s, \lambda, \pi, \mu) = 0 \tag{6}$$

onde $L(x, s, \lambda, \pi, \mu) = f(x) + \sum_{i=1}^m \lambda_i h_i(x) + \sum_{j=1}^r \pi_j (g_j(x) - s_j) - \sum_{j=1}^r \mu_j c^{-1} \ln(cs_j + 1)$.

O sistema de equações não lineares (6) pode ser resolvido utilizando algum método numérico, por exemplo, o método de Newton. Solucionando esse sistema obtêm-se as direções que atualizarão as variáveis x , s , λ e π . Já os multiplicadores de Lagrange, μ_j 's, são atualizados através da seguinte regra:

$$\mu_j^{k+1} = \mu_j^k / (c^k s_j^k + 1), \quad j = 1, K, r. \tag{7}$$

O algoritmo de reescalonamento não linear baseado na função logarítmica modificada é apresentado através de um fluxograma na figura 2.

Esse algoritmo é constituído por dois *loops*: externo e interno. O *loop* interno é constituído pelos seguintes passos: resolução do sistema (6), determinação do tamanho do passo e atualização das variáveis. O *loop* externo controla a convergência global do problema através das condições de KKT para o problema (2) e atualiza a variável dual μ^k .

Sejam os pontos $x^k, s^k, \lambda^k, \pi^k$ e μ^k determinados na k -ésima iteração do algoritmo apresentado na figura 2, as condições necessárias de KKT utilizadas como critério de parada do *loop* externo são:

- Estacionariedade:

$$\nabla f(x^k) + \sum_{i=1}^m \lambda_i^k \nabla h_i(x^k) + \sum_{j=1}^r \pi_j^k \nabla g_j(x^k) - \sum_{j=1}^r \mu_j^k s_j^k = 0$$

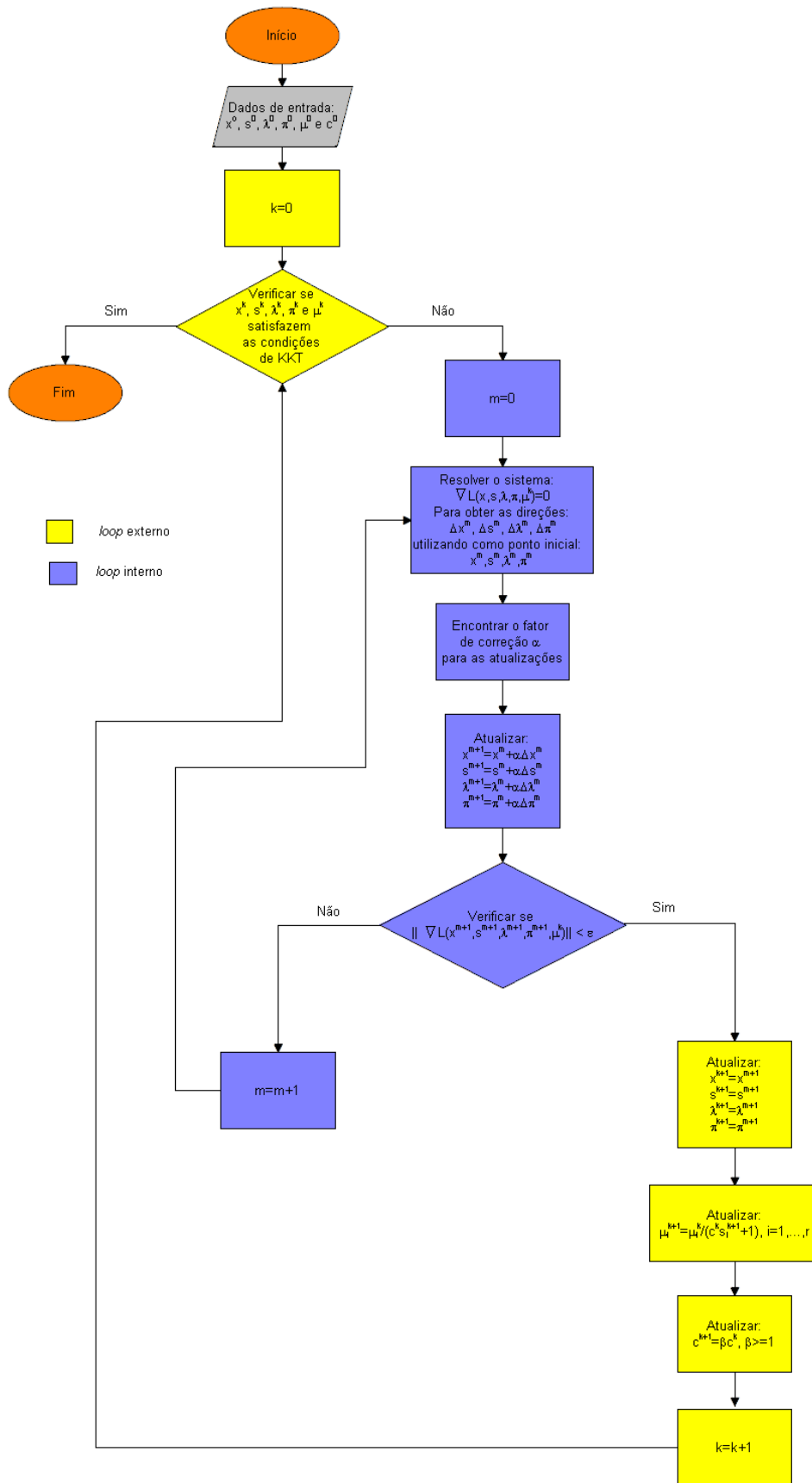


Figura 2. Fluxograma do método de rescalonamento não linear.

- Factibilidade primal:

$$\begin{aligned} h_i(x^k) &= 0, \quad i = 1, K, m \\ g_j(x^k) - s_j^k &= 0, \quad j = 1, K, r \\ s_j^k &\geq 0, \quad j = 1, K, r \end{aligned}$$

- Factibilidade dual:

$$\mu_j^k \geq 0, \quad j = 1, \dots, r$$

- Folgas Complementares:

$$\mu_j^k g_j(x^k) \geq 0, \quad j = 1, \dots, r$$

O algoritmo finaliza-se quando essas condições de KKT forem satisfeitas.

O *loop* interno é responsável em atualizar as variáveis x^m , s^m , λ^m e π^m através das direções Δx^m , Δs^m , $\Delta \lambda^m$ e $\Delta \pi^m$ encontradas pela resolução do sistema não linear apresentado em (6). A precisão desse passo implica nas atualizações das variáveis duais μ_j^k 's. Para resolver as equações não lineares, representadas vetorialmente por $\nabla L(x, s, \lambda, \pi, \mu^k) = 0$ utiliza-se o método de Newton:

$$\nabla^2 L(x, s, \lambda, \pi, \mu^k) \Delta d = -\nabla L(x, s, \lambda, \pi, \mu^k)$$

onde $\nabla^2 L$ é a jacobiana do funcional ∇L e $\Delta d = (\Delta x, \Delta s, \Delta \lambda, \Delta \pi)^t$.

Para medir o quão próximo $(x^k, s^k, \lambda^k, \pi^k, \mu^k)^t$ está da solução ótima utiliza-se a norma do vetor ∇L , se $\|\nabla L(x^k, s^k, \lambda^k, \pi^k, \mu^k)\|$ está perto do zero encerra-se o *loop* interno. Como computacionalmente não é possível chegar ao zero estipula-se este através de uma tolerância com valor muito pequeno. Desse modo, todos os valores menores que essa tolerância dada representa o zero.

Diferentemente do método da barreira clássica apresentada em Fiacco (1968), onde os pontos gerados são factíveis, o método apresentado na figura 2 permite gerar pontos infactíveis. Mas esses pontos respeitam a relaxação estipulada pelo método que é $s_j^k \geq -1/c^k$, $j = 1, \dots, r$.

O controle de passo, citado no algoritmo, visa encontrar α tal que s_j^{m+1} continue na região factível relaxada, ou seja, $s_j^{m+1} \geq -1/c^k$. Impondo essa condição para a atualização $s_j^{m+1} = s_j^m + \alpha \Delta s_j^m$, obtém-se:

$$\alpha \geq -\frac{1}{c^k \Delta s_j^m} - \frac{s_j^m}{\Delta s_j^m}, \quad j = 1, \dots, r.$$

Definindo um limite inferior a α como sendo $-1/c^k \Delta s_j^m - s_j^m / \Delta s_j^m$, obtém-se o controle de passo para as variáveis de folga s_j^m 's:

$$\alpha = \rho \left(-\frac{1}{c \Delta s_j^m} - \frac{s_j^m}{\Delta s_j^m} \right), \quad \rho > 1 \tag{7}$$

Para as demais variáveis: x^m , λ^m e π^m o valor de α é igual a um.

Na próxima seção será apresentado os resultados numéricos do algoritmo da figura 2 aplicado a problemas de otimização convexa e não convexa.

3. Resultados Numéricos

O algoritmo descrito na seção anterior foi implementado em Matlab e testado em quatro problemas de otimização que podem ser encontrados em Schittkowski (1987):

Problema 1:

$$\begin{aligned} &\text{Minimizar } f(x) = -x_2 \\ &\text{sujeito a } h_1(x) = x_1^2 + x_2^2 - 1 = 0 \\ &g_1(x) = 1 + x_1 - 2x_2 \geq 0 \\ &g_2(x) = x_1 \geq 0 \end{aligned}$$

Problema 2:

$$\begin{aligned} &\text{Minimizar } f(x) = -x_2 \\ &\text{sujeito a } g_3(x) = -x_1^2 - x_2^2 + 1 \geq 0 \\ &g_1(x) = 1 + x_1 - 2x_2 \geq 0 \\ &g_2(x) = x_1 \geq 0 \end{aligned}$$

Problema 3:

$$\begin{aligned} &\text{Minimizar } f(x) = -x_2 \\ &\text{sujeito a } \bar{h}_1(x) = 2x_1 + x_2 - 2 = 0 \\ &g_1(x) = 1 + x_1 - 2x_2 \geq 0 \\ &g_2(x) = x_1 \geq 0 \end{aligned}$$

Problema 2:

$$\begin{aligned} &\text{Minimizar } f(x) = -x_2 \\ &\text{sujeito a } \bar{g}_3(x) = -2x_1 - x_2 + 2 \geq 0 \\ &g_1(x) = 1 + x_1 - 2x_2 \geq 0 \\ &g_2(x) = x_1 \geq 0 \end{aligned}$$

As figuras 3, 4, 5 e 6 representam graficamente esses problemas.

O problema 1 é bem parecido com o 2, o que difere é a não convexidade inserida pela restrição de igualdade. Os problemas 3 e 4 são lineares e convexos.

Os parâmetros modificados para a realização dos testes foram:

- tolerância externa (**tol_ex**): parâmetro utilizado no critério de parada do *loop* externo,
- tolerância interna (**tol_in**): parâmetro utilizado no critério de parada do *loop* interno,
- c^k : parâmetro de relaxação,
- β : parâmetro que controla a taxa de crescimento de c^k .

Realizaram-se vários testes onde foram selecionados três casos para ilustrar o comportamento do algoritmo. No primeiro caso utilizou-se **tol_ex** = **tol_in** = 10^{-4} , para o

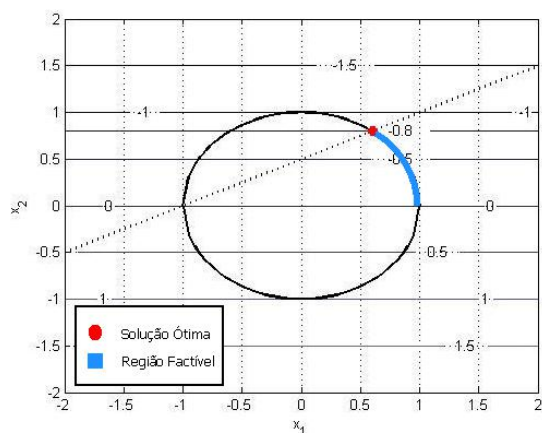


Figura 3. Representação gráfica do problema 1.

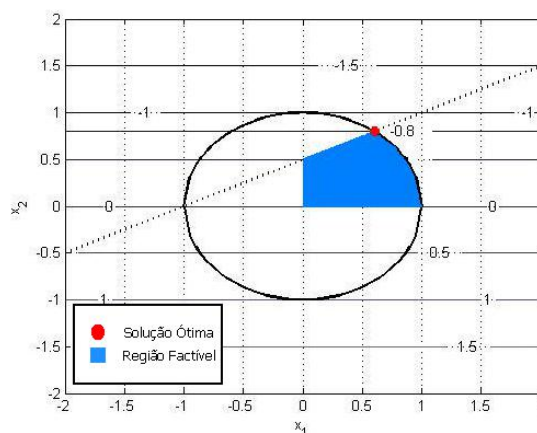


Figura 4. Representação gráfica do problema 2.

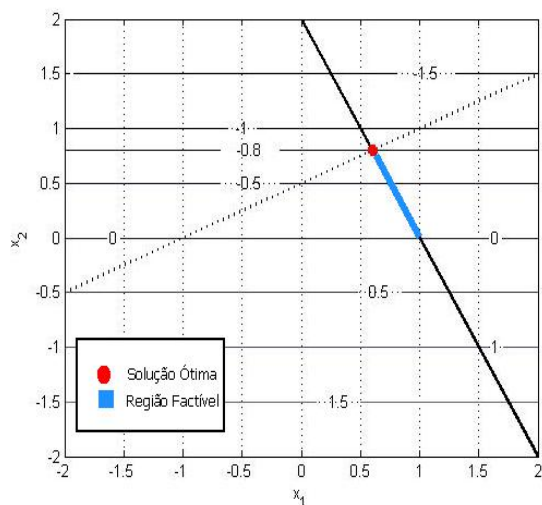


Figura 5. Representação gráfica do problema 3.

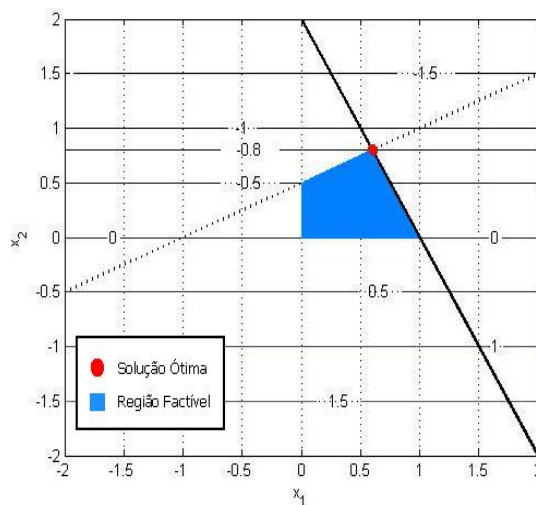


Figura 6. Representação gráfica do problema 4.

segundo $tol_ex = tol_in = 10^{-6}$ e para o terceiro caso $tol_ex = tol_in = 10^{-8}$. Cada caso foi testado para diferentes valores de c^k e β . Na tabela 1 encontram-se todos os valores empregados nos testes:

	Teste 1	Teste 2	Teste 3
c^0	1	0,01	0,01
β	1	1,5	3

Tabela 1. Parâmetros utilizados nos testes.

Foram analisados os seguintes itens: total de iterações externa e média das iterações interna. Os resultados obtidos pelo algoritmo podem ser vistos nas tabelas 2 e 3:

Casos:	Teste 1			Teste 2			Teste 3		
	1	2	3	1	2	3	1	2	3
Problema 1	17	27	37	17	19	21	8	9	10
Problema 2	18	28	38	18	20	21	8	9	10
Problema 3	18	27	37	17	19	21	8	9	10
Problema 4	19	28	38	18	19	21	8	9	10

Tabela 2. Total de iterações externas.

Casos:	Teste 1			Teste 2			Teste 3		
	1	2	3	1	2	3	1	2	3
Problema 1	3	3	3	4	4	5	5	4	6
Problema 2	3	3	3	4	4	5	5	5	6
Problema 3	3	3	3	3	4	5	5	5	6
Problema 4	3	3	3	3	4	5	4	5	6

Tabela 3. Média das iterações internas de cada iteração externa do algoritmo.

4. Conclusão

Os testes realizados com o algoritmo de reescalonamento não linear baseado na função barreira logarítmica modificada foram satisfatórios. Os resultados convergiram para as soluções ótimas dos problemas. O mais interessante das análises foi o bom comportamento desse algoritmo no problema não convexo utilizado neste artigo. Notou-se que em relação ao seu problema similar, mas convexo, a convergência é mais rápida. Com relação ao número de iterações externas observou-se que para um dado c^0 há uma diminuição quando aumenta o parâmetro β . É possível verificar que para cada problema existe um β ótimo e caso queira-se aumentar mais esse valor o algoritmo deixa de convergir. Esse artigo visou experimentar os parâmetros do algoritmo implementado para que se possa aplicar em problemas de sistemas elétricos, como por exemplo, o problema de fluxo de potência ótimo que é um problema não convexo.

Referências

- Bertsekas, D.**, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, NY, 1982.
- Dussault, J. P.** (2004), Augmented non-quadratic penalty algorithms, *Mathematical Programming*, 99, 467-486.
- Fiacco, A. V. e McCormick, G. P.**, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley & Sons, New York, 1968.
- Kort, B. e Bertsekas, D. P.** (1973), Multiplier Methods for Convex Programming, in *Proceeding of the IEEE Decision and Control Conference*, San Diego, CA, 260-264.
- Polyak, R.**, *Controlled processes in extremal and equilibrium problems*, VINITI, deposited manuscript, Moscow (1986) (in Russian).
- Polyak, R.** (1992), Modified barrier functions (theory and methods), *Mathematical Programming*, 54, 177-222.
- Polyak, R. e Teboulle, M.** (1997), Nonlinear rescaling and proximal-like methods in convex optimization, *Mathematical Programming*, 76, 265-284.
- Polyak, R.** (2002), Nonlinear rescaling vs. smoothing technique in convex optimization, *Mathematical Programming*, 92, 197-235.
- Polyak, R. e Griva, I.** (2004), Primal-Dual Nonlinear Rescaling Method for Convex Optimization, *Journal of Optimization Theory and Applications*, 122, 111-156.
- Griva, I. e Polyak, R. A.** (2006), Primal-dual nonlinear rescaling method with dynamic scaling parameter update, *Mathematical Programming*, 106, 237-259.
- Schittkowski, K.**, *More Test Examples for Nonlinear Programming Codes*, Lecture Notes in Economics and Mathematical Systems, vol. 282, Springer, NY, 1987.
- Teboulle, M.** (1992), Entropic Proximal Mappings with Applications to Nonlinear Programming, *Mathematics of Operations Research*, 17, 670-690.