Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, v. 9, n. 1, 2022.

Trabalho apresentado no XLI CNMAC, Unicamp - Campinas - SP, 2022.

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics
Preprint

O método ADMM para um problema de otimização polinomial não convexo

Orlando Sarmiento[1]
Campus UFRJ-Duque de Caxias Professor Geraldo Cidade, UFRJ, Duque de Caxias, RJ
Marcia Fampa[2]
PESC/COPPE - Programa de Engenharia de Sistemas e Computação, UFRJ, Rio de Janeiro, RJ

Resumo. Consideramos o problema NP-difícil de minimizar uma função polinomial de grau três sobre a fronteira da esfera. Devido à não convexidade do problema e às importantes aplicações em processamento de imagens, o desenvolvimento de algoritmos para encontrar boas soluções viáveis para o mesmo tem recebido considerável atenção. Neste trabalho, detalhamos o algoritmo ADMM para este problema, assim como particularizamos seus resultados de convergência. Comparamos numericamente o algoritmo com o solver fmincon, do Matlab, e analisamos o comportamento do algoritmo para diferentes inicializações do mesmo.

Palavras-chave. Problema de otimização cúbico, método ADMM, ponto de KKT, otimização polinomial, otimização global, ótimo local.

1 Introdução

Neste trabalho consideramos um problema de otimização não convexo, mais especificamente o problema de minimizar uma função polinomial de terceiro grau definida sobre a fronteira da esfera unitária, que pode ser formulado como:

$$\begin{array}{ll} \text{minimizar} & f(x) := \displaystyle\sum_{1\le i_1\le i_2\le i_3\le n} a_{i_1 i_2 i_3} x_{i_1} x_{i_2} x_{i_3} \\ \text{sujeito a} & \|x\| = 1, \end{array} \tag{POC}$$

onde $x \in \mathbb{R}^n$, para $n \geq 2$. Aplicações para o problema cúbico sobre a esfera podem ser encontradas, por exemplo em processamento de imagens e aproximações de posto baixo para tensores. Algumas dessas aplicações estão descritas em [12, 13] e suas referências [1, 2, 9, 11].

Algoritmos de aproximação de tempo polinomial para otimização polinomial podem ser encontrados na literatura, por exemplo em [4, 7, 10]. He et al. [4] apresenta resultados numéricos para o caso de minimizar funções homogêneas de quarto grau. Lasserre [7] mostra que o problema de minimizar funções polinomiais sobre um conjunto compacto definido por desigualdades polinomiais se reduz a resolver uma sequência de problemas convexos.

Nie [8] apresenta relaxações via soma de quadrados (SOS) para casos particulares de problemas de otimização com polinômios, que também podem ser convertidas em relaxações via programação semidefinida (SDP). No entanto, essas relaxações geralmente não são eficientes para problemas de grande porte, e suas aplicações estão restritas a problemas que os solvers possam resolver. Limites

[1] orlando@caxias.ufrj.br
[2] fampa@cos.ufrj.br

inferiores para o problema (POC) também são obtidos em [3], com o uso de decomposição Lagrangeana. O método resulta na resolução de uma série de problemas quadráticos parametrizados que podem ser resolvidos em tempo polinomial.

Em [3] o solver de programação não linear fmincon, do Matlab, foi utilizado para gerar limites superiores para o (POC) e validar a qualidade dos limites inferiores propostos. Neste trabalho, consideramos a aplicação do método *Alternating direction method of multipliers* (ADMM) para gerar estes limites e comparamos os resultados com os de fmincon. A principal estratégia empregada no ADMM é a reformulação do problema cúbico levando a função objetivo a ser definida por um tensor simétrico, assim como foi feito em [3]. O algoritmo ADMM para um problema mais geral de otimização polinomial foi analisado em [6], e através da reformulação da função objetivo do problema com o uso de tensores, os autores provaram a convergência do algoritmo para um ponto KKT do problema. Consideramos neste trabalho a particularização destes resultados para o (POC) e verificamos que para o (POC), os subproblemas resolvidos nas iterações do ADMM são extremamente simples, tendo uma fórmula fechada para a sua resolução. Apresentamos resultados computacionais para exemplos da literatura e para instâncias aleatórias do (POC), geradas como em [3], comparando o ADMM com o fmincon e analisando o comportamento do ADMM com diferentes metodologias de inicialização para o mesmo.

O trabalho está organizado como segue. Na Seção 2, apresentamos a notação utilizada ao longo do texto. Na Seção 3 apresentamos o método ADMM para o problema (POC). Na Seção 4, apresentamos nosso resultados computacionais. Na Seção 5, apresentamos nossas conclusões.

2 Notação

O conjunto de vetores reais de dimensão p é denotado por $\mathbb{R}^p$ e a norma euclidiana é denotada por $\|\cdot\|$, isto é, dado $x \in \mathbb{R}^p$, $\|x\| = \sqrt{\sum_{i=1}^{p} x_i^2}$. Dado um subconjunto $X \subseteq \mathbb{R}^p$ e uma função $f : X \to \mathbb{R}$ denotamos o dominio de f por $\operatorname{dom} f$ e definimos $\operatorname{argmin}_{x\in X} f := \{x \in X : f(x) = \min_X f\}$.

Visando a aplicação do método ADMM, faremos uso de tensores para modelar a função objetivo cúbica de (POC), seguindo as considerações realizadas por Jiang et al. em [6]. Denotamos por $S^d(\mathbb{R}^n)$, o espaço de tensores simétricos de dimensão $(\underbrace{n \times n \times \cdots \times n}_{d \text{ vezes}})$. Admitimos que um dado tensor $\mathcal{A}$ é simétrico se seus elementos $\mathcal{A}_{i_i i_2 \cdots i_d}$ são invariantes sob qualquer permutação dos índices $\{i_i, i_2, \cdots, i_d\}$. Considerando a função objetivo f definida em (POC), definimos o tensor $\mathcal{A} \in S^3(\mathbb{R}^n)$, com $\mathcal{A} := (\mathcal{A}_{i_i i_2 i_3})$ e

$$\mathcal{A}_{j_i j_2 j_3} := \frac{a_{i_1 i_2 i_3}}{|\pi(i_1 i_2 i_3)|}, \quad \forall (j_1 j_2 j_3) \in \pi(i_i i_2 i_3),$$

onde $\pi(i_1 i_2 i_3)$ é o conjunto de todas as permutações distintas dos índices $\{i_1, i_2, i_3\}$. Definimos a função $\mathcal{F} : \mathbb{R}^{n\times n\times n} \to \mathbb{R}^n$ com

$$\mathcal{F}(x^1, x^2, x^3) := \sum_{1\le i_1, i_2, i_3 \le n} \mathcal{A}_{i_i i_2 i_3} x^1_{i_1} x^2_{i_2} x^3_{i_3}.$$

Observamos que

$$\frac{\partial \mathcal{F}}{\partial x^1_j} = \sum_{1\le i_2, i_3 \le n} \mathcal{A}_{j i_2 i_3} x^2_{i_2} x^3_{i_3}, \quad \frac{\partial \mathcal{F}}{\partial x^2_j} = \sum_{1\le i_1, i_3 \le n} \mathcal{A}_{i_1 j i_3} x^1_{i_1} x^3_{i_3}, \quad \frac{\partial \mathcal{F}}{\partial x^3_j} = \sum_{1\le i_1, i_2 \le n} \mathcal{A}_{i_1 i_2 j} x^1_{i_1} x^2_{i_2},$$

para $j = 1, \ldots, n$. Definimos

$$\mathcal{F}(\bullet, \bar{x}^2, \bar{x}^3) := \nabla_{x^1} \mathcal{F}(\bar{x}^2, \bar{x}^3), \quad \mathcal{F}(\bar{x}^1, \bullet, \bar{x}^3) := \nabla_{x^2} \mathcal{F}(\bar{x}^1, \bar{x}^3), \quad \mathcal{F}(\bar{x}^1, \bar{x}^2, \bullet) := \nabla_{x^3} \mathcal{F}(\bar{x}^1, \bar{x}^2),$$

DOI: 10.5540/03.2022.009.01.0281

para todo $(\bar{x}^1, \bar{x}^2, \bar{x}^3) \in \mathbb{R}^{n\times n\times n}$. Por fim, observamos que para função objetivo f de (POC), temos $f(x) = \mathcal{F}(x, x, x)$, para todo $x \in \mathbb{R}^n$.

3 O método ADMM para o (POC)

Considerando a notação introduzida na seção anterior, reformulamos (POC) como

$$\begin{array}{lll} \text{minimizar} & \mathcal{F}(x^1, x^2, x^3) & \\ \text{sujeito a:} & x^0 = x^i, & i = 1, 2, 3, \\ & \|x^i\|^2 = 1, & i = 0, 1, 2, 3. \end{array} \tag{P}$$

A função Lagrangeana aumentada para (P), associada apenas às restrições $x^0 = x^i$, $i = 1, 2, 3$, é

$$\mathcal{L}(x^0, x^1, x^2, x^3; \lambda^1, \lambda^2, \lambda^3) := \mathcal{F}(x^1, x^2, x^3) + \textstyle\sum_{i=1}^3 \langle \lambda^i, x^i - x^0\rangle + \frac{\beta}{2}\sum_{i=1}^3 \|x^i - x^0\|^2,$$

onde λ_i é o multiplicador de Lagrange associado à restrição $x^0 = x^i$, $\beta > 0$ é um parâmetro de penalização e dom $\mathcal{L} = (\mathbb{R}^n)^7$. Um método clássico para resolver problemas de programação não linear é o método Lagrangeano aumentado, o qual consiste em atualizar as variáveis primais e os multiplicadores de Lagrange iterativamente. Quando mais especificamente aplicado à (P), este método requer a cada iteração, a minimização da função Lagrangeana aumentada com respeito às variáveis x^i, sujeito a $\|x^i\|^2 = 1$, para todo $i = 0, 1, 2, 3$. Uma alternativa para a solução deste difícil problema de otimização é dada pelo método ADMM, onde o substituímos por uma de sequência de problemas onde minimizamos a função Lagrangeana aumentada com respeito a cada variável alternadamente. Desta forma, temos vários subproblemas em cada iteração e esses subproblemas são resolvidos eficientemente. Uma iteração do método ADMM para resolver (P) é dada por:

$$(x^0)^{(\ell+1)} = \text{argmin}_{\|x^0\|^2=1}\mathcal{L}(x^0, (x^1)^{(\ell)}, (x^2)^{(\ell)}, (x^3)^{(\ell)}; (\lambda^1)^{(\ell)}, (\lambda^2)^{(\ell)}, (\lambda^3)^{(k)}), \tag{1}$$

$$(x^1)^{(\ell+1)} = \text{argmin}_{\|x^1\|^2=1}\mathcal{L}((x^0)^{(\ell+1)}, x^1, (x^2)^{(\ell)}, (x^3)^{(\ell)}; (\lambda^1)^{(\ell)}, (\lambda^2)^{(\ell)}, (\lambda^3)^{(\ell)}), \tag{2}$$

$$(x^2)^{(\ell+1)} = \text{argmin}_{\|x^2\|^2=1}\mathcal{L}((x^0)^{(\ell+1)}, (x^1)^{(\ell+1)}, x^2, (x^3)^{(\ell)}; (\lambda^1)^{(\ell)}, (\lambda^2)^{(\ell)}, (\lambda^3)^{(\ell)}), \tag{3}$$

$$(x^3)^{(\ell+1)} = \text{argmin}_{\|x^3|^2=1}\mathcal{L}((x^0)^{(\ell+1)}, (x^1)^{(\ell+1)}, (x^2)^{(\ell+1)}, x^3; (\lambda^1)^{(\ell)}, (\lambda^2)^{(\ell)}, (\lambda^3)^{(\ell)}), \tag{4}$$

$$(\lambda^i)^{(\ell+1)} = (\lambda^i)^{(\ell)} + \beta_\ell((x^i)^{(\ell+1)} - (x^0)^{(\ell+1)}), \quad i = 1, 2, 3. \tag{5}$$

Note que com outras variáveis fixas, a função Lagrangeana aumentada $\mathcal{L}$ com respeito a x^i é uma simples função quadrática e cada subproblema transforma-se em minimizar $\|x^i - z\|^2$ para um dado z. Assim, cada subproblema (1)-(4) corresponde a uma projeção sobre a esfera unitária e tem solução dada por:

$$\text{normalizar}(z) := \frac{z}{\|z\|}.$$

Portanto, para $i = 1, 2, 3$ e $d = 3$, as iterações do algoritmo podem ser escritas mais especificamente como:

$$(x^0)^{(\ell+1)} := \text{normalizar}\left(\textstyle\sum_{i=1}^3 (x^i)^{(\ell)} + \frac{1}{\beta_\ell}(\lambda^i)^{(\ell)}\right), \tag{6}$$

$$(\nu^i)^{(\ell+1)} := \mathcal{F}((x^1)^{(\ell+1)}, \ldots, (x^{i-1})^{(\ell+1)}, \bullet, (x^{i+1})^{(\ell)}, \ldots, (x^d)^{(\ell)}), \tag{7}$$

$$(x^i)^{(\ell+1)} = \text{normalizar}\left((x^0)^{(\ell+1)} - \tfrac{1}{\beta_\ell}((\nu^i)^{(\ell+1)} + (\lambda^i)^{(\ell)}\right), \tag{8}$$

$$(\lambda^i)^{(\ell+1)} = (\lambda^i)^{(\ell)} + \beta_\ell((x^i)^{(\ell+1)} - (x^0)^{(\ell+1)}). \tag{9}$$

A seguir apresentamos a propriedade de convergência do algoritmo ADMM, que é uma particularização para o caso cúbico, do resultado mais geral demonstrado em [6, Teorema 3.2].

Teorema 3.1. *Seja* $z^\ell := \left([(x^i)^{(\ell)}]_{i=0}^3; [(\lambda^i)^{(\ell)}]_{i=1}^3\right)$ *a sequência gerada em (6)-(9). Assuma que*

$$\lim_{\ell \longrightarrow \infty} (z^{\ell+1} - z^\ell) = 0.$$

Então, qualquer ponto de acumulação de $\{z^\ell\}_{\ell=1}^\infty$ *é um ponto KKT de* (P). *Consequentemente, se a sequência* $\{(x^0)^{(\ell)}\}_{\ell=1}^\infty$ *converge, ela converge para um ponto KKT de* (POC).

Propomos então o algoritmo a seguir para obter um ponto de KKT para (POC).

Entrada: $\mathcal{A}$, β_0, ρ, ϵ, $\ell_{\max}$, $(x^0)^{(0)}, (x^1)^{(0)}, (x^2)^{(0)}, (x^3)^{(0)}, (\lambda^1)^{(0)}, (\lambda^2)^{(0)}, (\lambda^3)^{(0)}$.
Saída: $x^{0(*)}, x^{1(*)}, x^{2(*)}, x^{3(*)}, \lambda^{1(*)}, \lambda^{2(*)}, \lambda^{3(*)}$.

$z^0 := \left([(x^i)^{(0)}]_{i=0}^3; [(\lambda^i)^{(0)}]_{i=1}^3\right)$;
$\ell := 0$, $\Delta := +\infty$;
enquanto $\boldsymbol{\Delta > \epsilon}$ **faça**
  para $\boldsymbol{i = 1, 2, 3}$ **faça**
    $(x^0)^{(\ell+1)} :=$ normalizar $\left(\sum_{i=1}^3 (x^i)^{(\ell)} + \frac{1}{\beta_\ell}(\lambda^i)^{(\ell)}\right)$;
    $(\nu^i)^{(\ell+1)} := \mathcal{F}((x^1)^{(\ell+1)}, \ldots, (x^{i-1})^{(\ell+1)}, \bullet, (x^{i+1})^{(\ell)}, \ldots, (x^d)^{(\ell)})$;
    $(x^i)^{(\ell+1)} :=$ normalizar $\left((x^0)^{(\ell+1)} - \frac{1}{\beta}((\nu^i)^{(\ell+1)} + (\lambda^i)^{(\ell)}\right)$;
    $(\lambda^i)^{(\ell+1)} := (\lambda^i)^{(\ell)} + \beta_{(\ell)}((x^i)^{(\ell+1)} - (x^0)^{(\ell+1)})$;
  $\beta_{(\ell+1)} = \rho\beta_{(\ell)}$;
  $z^{\ell+1} := \left([(x^i)^{(\ell+1)}]_{i=0}^3; [(\lambda^i)^{(\ell+1)}]_{i=1}^3\right)$;
  $\Delta := \|z^{(\ell+1)} - z^{(\ell)}\|$;
  $\ell := \ell + 1$;

Algoritmo 1: ADMM para o problema (P)

4 Experimentos Numéricos

Nesta seção, comparamos o algoritmo ADMM apresentado na Seção 3 com o solver de otimização não linear fmincon, do Matlab. Visando analisar a capacidade de ambos os métodos em encontrar um ótimo global para as nossas instâncias, utilizamos também a ferramenta GloptiPoly 3 proposta por Henrion et al. [5], especializada em resolver problemas de otimização (não convexa) polinomial.

Nossas implementações foram realizadas em MATLAB R2017b e nossos experimentos computacionais foram realizados num notebook Intel core i5-8250U 1.60 GHz com 8GB de RAM, rodando sob o sistema operacional Ubuntu 18.04.

Em todos os experimentos, consideramos os pontos iniciais para os algoritmos como pontos viáveis aleatórios seguindo uma distribuição normal em $[-1, 1]$, e os parâmetros iniciais do Algoritmo 1 como: $\beta_0 = 1$, $\rho = 0.95$, $\epsilon = 10^{-6}$ e número máximo de iterações $\ell_{max} = 10^3$.

Para uma melhor compreensão dos resultados, apresentamos a seguir uma lista das abreviações mencionadas na Tabela 2 .

fmincon	:	solver fmincon com um ponto inicial aleatório $\bar{x}$;
ADMM1	:	Algoritmo 1 com ponto inicial $(x^0)^{(0)} = (x^1)^{(0)} = (x^2)^{(0)} = (x^3)^{(0)} = \bar{x}$;
ADMM2	:	Algoritmo 1 com ponto inicial $(x^0)^{(0)} = \bar{x}$ e $(x^i)^{(0)}$, $i = 1, 2, 3$, aleatórios;
val.	:	valor da função objetivo na solução obtida pelo método correspondente;
GLP	:	solver GloptiPoly3.
st.	:	1 significa que GloptiPoly3 encontra o ótimo global do problema.

Os três exemplos considerados a seguir foram obtidos da literatura e têm valores ótimos conhecidos (ver [9, 12]). Pela facilidade em resolver estes exemplos pequenos, aplicamos apenas o algoritmo ADMM1 a eles e apresentamos o resultado obtido pelo algoritmo, o tempo computacional (em segundos) para aplicá-lo e a solução ótima dos problemas na Tabela 1.

Exemplo 4.1 (Exemplo 4.2 em [12]). $\mathcal{A} \in S^3(\mathbb{R}^3)$ *é definido por*

$$\begin{aligned}&\mathcal{A}_{111} = -0.1281, \mathcal{A}_{112} = 0.0516, \mathcal{A}_{113} = -0.0954,\\ &\mathcal{A}_{122} = -0.1958, \mathcal{A}_{123} = -0.1790, \mathcal{A}_{133} = -0.2679,\\ &\mathcal{A}_{222} = 0.3251, \mathcal{A}_{223} = 0.2513, \mathcal{A}_{233} = 0.1773, \mathcal{A}_{333} = 0.0338.\end{aligned}$$

Exemplo 4.2 (Exemplo 3.3 em [9]). $\mathcal{A} \in S^3(\mathbb{R}^3)$ *definido por*

$$\begin{aligned}&\mathcal{A}_{111} = 0.0517, \mathcal{A}_{112} = 0.3579, \mathcal{A}_{113} = 0.5298, \mathcal{A}_{122} = 0.7544, \mathcal{A}_{123} = 0.2156,\\ &\mathcal{A}_{133} = 0.3612, \mathcal{A}_{222} = 0.3943, \mathcal{A}_{223} = 0.0146, \mathcal{A}_{233} = 0.6718, \mathcal{A}_{333} = 0.9723.\end{aligned}$$

Exemplo 4.3 (Exemplo 3.5 em [9]). $\mathcal{A} \in S^3(\mathbb{R}^5)$ *definido por*

$$\mathcal{A}_{i_1,i_2,i_3} = \frac{(-1)^{i_1}}{i_1} + \frac{(-1)^{i_2}}{i_2} + \frac{(-1)^{i_3}}{i_3}.$$

Tabela 1: Comparação com exemplos da literatura

Problema	Solução ADMM	tempo	Solução ótima conhecida
Exemplo 4.1	-0.8730	0.012	-0.8730
Exemplo 4.2	-2.1110	0.010	-2.1110
Exemplo 4.3	-9.9779	0.018	-9.9779

Podemos notar que o Algoritmo 1 funcionou bem para essas instâncias pequenas, pois atinge os valores ótimos conhecidos na literatura em um tempo computacional menor que 0.02 segundos.

Para os experimentos aleatórios, geramos instâncias com tensores simétricos $\mathcal{A}$ com entradas seguindo uma distribuição normal em $[-1, 1]$. Na Tabela 2, mostramos resultados para 10 instâncias aleatórias, para cada dimensão $n = 5, 10, 15$. Destacamos com asteriscos, as instâncias em que o método ADMM1 ou ADMM2 atinge o valor ótimo global, concordando com o resultado obtido ao aplicar o método GloptiPoly 3. Ressaltamos em negrito, o melhor resultado entre fmincon, ADMM1 e ADMM2. Quando há empate entre os métodos, nenhum resultado é colocado em negrito. Notando que o fmincon melhora o resultado do ADMM em três instâncias apenas. Vale ressaltar que o método ADMM requer menos esforço computacional em comparação com os outros dois métodos.

5 Considerações Finais

Concluímos que o método ADMM é uma ótima alternativa para a obtenção de ótimos locais para o problema de otimização cúbica restrito à fronteira da esfera de raio unitário. Em nossos experimentos, aplicamos o método duas vezes para cada instância, considerando os pontos de inicializão aleatórios para as diferentes cópias das variáveis consideradas na reformulação do problema, primeiramente todos iguais e depois todos distintos. A primeira abordagem obteve melhores resultados para a maior parte das instâncias, mas a aplicação das duas abordagens, com o retorno da melhor solução obtida, ainda é bem mais rápida que a aplicação do solver fmincon, do Matlab, e leva uma melhor solução que o mesmo em 90% das instâncias testadas. Em 50% destas instâncias, o ADMM levou ao ótimo global do problema. Como trabalho futuro, pretendemos aplicar o ADMM em um algoritmo de otimização global, juntamente com a metodologia introduzida em [3] para obtenção de limites inferiores.

DOI: 10.5540/03.2022.009.01.0281

Tabela 2: Comparação do Algoritmo 1, fmincon e GloptiPoly 3 para instâncias aleatórias

I	fmincon		ADMM1		ADMM2		GLP		
	val.	tempo	val.	tempo	val.	tempo	val.	tempo	st.
				Dimensão n=5					
1*	-1.6709	0.158	-1.6709	0.005	-1.6709	0.110	-1.6709	0.637	1
2*	-2.0983	0.274	-2.0983	0.005	-2.0983	0.007	-2.0983	0.282	1
3*	-2.3251	0.158	-2.3251	0.004	-2.3251	0.008	-2.3251	0.217	1
4	-0.9111	0.148	**-1.4948**	0.022	-1.4768	0.013	-1.4954	0.290	1
5	-1.2800	0.132	-1.2800	0.003	**-1.6808**	0.012	-1.8658	0.226	1
6*	-1.3465	0.147	**-1.7997**	0.168	-1.7975	0.363	-1.7997	0.210	1
7*	-1.1852	0.193	-1.1851	0.008	**-1.8743**	0.008	-1.8743	0.217	1
8	-0.6906	0.103	-0.9114	0.225	**-1.0118**	0.219	-1.0998	0.303	1
9*	-2.0754	0.148	-2.0754	0.003	-2.0754	0.014	-2.0754	0.211	1
10*	-1.4243	0.127	-1.4243	0.009	-1.4236	0.010	-1.4243	0.228	1
				Dimensão n=10					
1*	-3.0775	0.562	-3.0775	0.032	-3.0775	0.024	-3.0775	4.965	1
2*	-2.3441	0.374	-2.3432	0.006	**-3.2206**	0.023	-3.2206	3.728	1
3*	-2.3521	0.391	**-2.8480**	0.058	-2.5976	0.149	-2.8480	3.713	1
4*	-2.1209	0.414	**-2.3637**	0.035	-2.3637	0.023	-2.3637	5.068	1
5	-1.8636	0.286	**-2.5643**	0.036	-1.7490	0.023	-2.8186	3.975	1
6	-2.2803	0.390	**-2.3020**	0.009	-2.3018	0.010	-2.5910	3.932	1
7	**-2.7049**	0.473	-2.6290	0.324	-2.6281	0.330	-2.7177	4.483	1
8*	-2.0583	0.274	**-2.9269**	0.032	-2.9269	0.042	-2.9269	4.250	1
9	-1.9066	0.252	-1.9066	0.007	-1.9066	0.011	-2.7668	4.270	1
10*	-2.9934	0.300	-2.9934	0.007	-2.9934	0.010	-2.9934	3.973	1
				Dimensão n=15					
1	**-3.2863**	0.447	-3.2862	0.009	-3.1883	0.016	-3.4059	314.95	1
2*	-3.3136	0.481	-3.1628	0.060	-3.3136	0.054	-3.3136	254.60	1
3*	-3.0960	0.455	**-3.1495**	0.433	-3.1392	0.073	-3.1495	273.92	1
4	-3.2560	0.610	-3.2468	0.442	**-3.2946**	0.385	-3.4196	233.11	1
5	**-3.0320**	0.455	-2.9617	0.236	-2.9617	0.077	-3.1132	233.14	1
6	-3.6454	0.471	**-3.6456**	0.057	-2.5465	0.013	-3.7346	214.47	1
7	-3.2442	0.427	-3.2442	0.013	-3.2442	0.052	-3.3709	249.22	1
8	-2.5596	0.396	-2.2615	0.436	**-2.9480**	0.339	-3.1447	232.97	1
9	-3.3545	0.433	-3.3545	0.010	-3.2083	0.125	-3.7399	250.07	1
10	-2.9150	0.501	-2.9150	0.011	**-3.0413**	0.161	-3.1265	215.40	1

Agradecimentos

M. Fampa foi apoiada em parte pelo CNPq (projetos 305444/2019-0 e 434683/2018-3).

Referências

[1] P. J. Basser e D. K. Jones. "Diffusion-tensor MRI: theory, experimental design and data analysis-a technical review". Em: **NMR Biomed.** 15 (2002), pp. 456–467. DOI: 10.1002/nbm.783.

[2] P. J. Basser, J. Mattiello e D LeBihan. "Estimation of the effective seldiffusion tensor from the NMR spin echo". Em: **Journal of Magnetic Resonance** 8 (1994), pp. 247–254. DOI: 10.1006/jmrb.1994.1037.

[3] C. Buccheim, M. Fampa e O. Sarmiento. "Lower Bounds for Cubic Optimization over the Sphere". Em: **Journal of Optimization Theory and Applications** 3 (2021), pp. 823–846. DOI: 10.1007/s10957-021-01809-y.

[4] S. He, Z. Li e S Zhang. "Approximation algorithms for homogeneous polynomial optimization with quadratic constraints". Em: **Mathematical Programming** 2 (2010), pp. 353–383. DOI: 10.1007/s10107-010-0409-z.

[5] D. Henrion, J. B. Lasserre e J. Loefberg. "GloptiPoly3: moments, optimization and semidefinite programming." Em: **Optim. Meth. Softw.** 24 (2009), pp. 761–779. DOI: 10.1080/10556780802699201.

[6] B. Jiang, S. Ma e S. Zhang. "Alternating direction method of multipliers for real and complex polynomial optimization models". Em: **Optimization: A Journal of Mathematical Programming and Operations Research** 6 (2014), pp. 883–898. DOI: 10.1080/02331934.2014.895901.

[7] J. B. Laserre. "Global optimization with polynomials and the problem of moments". Em: **SIAM Journal on Optimization** 3 (2001), pp. 796–817. DOI: 10.1137/S1052623400366802.

[8] J. Nie. "Sum of squares methods for minimizing polynomial forms over spheres and hypersurfaces". Em: **Frontiers of Mathematics in China** 2 (2012), pp. 321–346. DOI: 10.1007/s11464-012-0187-4.

[9] J. Nie e L. Wang. "Semidefinte relaxations for best rank-1 tensor approximations". Em: **SIAM Journal on Matrix Analysis and Applications** 35 (2014), pp. 1155–1179. DOI: 10.1137/130935112.

[10] A. M. So. "Deterministic approximation algorithms for sphere constrained homogeneous polynomial optimization problems". Em: **Mathematical Programming** 2 (2011), pp. 357–382. DOI: 10.1007/s10107-011-0464-0.

[11] R. J. Stern e Wolkowicz H. "Indefinite trust region subproblems and nonsymmetric eigenvalue perturbations". Em: **SIAM Journal on Optimization** 5 (1995), pp. 286–313. DOI: 10.1137/0805016.

[12] X. Zhang, C. Ling e L. Qi. "The best rank-1 approximation of a symmetric tensor and related spherical optimization problems". Em: **SIAM J. Matrix Anal. Appl.** 33 (2012), pp. 806–821. DOI: 10.1137/110835335.

[13] X. Zhang, L. Qi e Y. Ye. "The cubic spherical optimization problems". Em: **Mathematics of Computation** 81 (2012), pp. 1513–1525. DOI: 10.1090/S0025-5718-2012-02577-4.

