

## Métodos computacionales en optimización polinomial: una revisión de la literatura

Orlando Sarmiento<sup>1</sup> y Liliana Jurado<sup>2</sup>

<sup>1</sup> Universidade Federal do Rio de Janeiro - Campus UFRJ Duque de Caxias Prof. Geraldo Cidade, Rio de Janeiro, Brasil

<sup>2</sup> Universidade do Estado do Rio de Janeiro - UERJ, Rio de Janeiro, Brasil.

Recibido el 10 de diciembre del 2024. Aceptado el 10 de enero del 2024. Publicado en 2 de febrero del 2025

### Resumen

En un problema de optimización polinomial se busca minimizar funciones polinomiales sujetos a restricciones definidas por funciones polinomiales. En el caso general, el problema es NP-difícil y generaliza varios problemas de optimización como problemas de programación lineal, cuadrático, cúbico, etc. En los últimos años, el estudio sobre optimización polinomial se ha intensificado por las diversas aplicaciones que se encuentran en las áreas de Ingeniería y Ciencias Exactas, como por ejemplo, en procesamiento de señales, problemas de imagen de resonancia magnética, teoría de aproximación de matrices de bajo rango, entre otros. Debido a la dificultad de encontrar soluciones óptimas exactas para este problema, diversas investigaciones surgen en la búsqueda de desarrollar eficientes algoritmos y encontrar las mejores aproximaciones de una solución óptima. En este trabajo de investigación mostramos una revisión de literatura de los principales métodos computacionales que buscan resolver eficientemente problemas de optimización polinomial. Mostramos las formulaciones matemáticas de estos métodos, así como también mostramos experimentos numéricos que fueron realizados al implementar algunos de los algoritmos citados en el artículo.

**Descriptor:** *optimización polinomial, problema NP-difícil, métodos computacionales*

### Abstract

In a polynomial optimization problem we seek to minimize polynomial functions subject to constraints defined by polynomial functions. In the general case, the problem is NP-hard and generalizes various optimization problems such as linear, quadratic or cubic programming. In recent years, research on polynomial optimization is intensified by the several applications found in the areas of Engineering and Sciences, such as signal processing, magnetic resonance imaging problems, low-rank matrix approximation theory, among others. Due to the difficulty of finding exact optimal solutions for this problem, several researcher arise to develop efficient algorithms and find the best approximations of the optimal solution. In this work we show a literature review of the main computational methods that seek to solve polynomial optimization problems. We show the mathematical formulations of these methods, as well as perform numerical experiments implementing some of the algorithms cited in the paper.

**Keywords:** *polynomial optimization, NP-hard problem, computational methods*

### 1. Introducción

Una función polinomial  $f(x)$ , es definida de la forma  $f(x) = \sum f_{\alpha} x^{\alpha}$  donde los coeficientes  $f_{\alpha}$  son escalares y los monomios  $x^{\alpha}$  son definidos por

$$x^{\alpha} := x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n} \text{ con } \alpha \in \mathbb{N}^n.$$

Por ejemplo, para la función polinomial de la forma

$$f(x_1, x_2) = x_1^2 x_2 + 3x_2,$$

tenemos que  $x \in \mathbb{R}^2$  y  $\alpha \in \{(2, 1), (0, 1)\}$ .

La cantidad  $|\alpha| := \alpha_1 + \alpha_2 + \dots + \alpha_n$  es el grado del monomio  $x^\alpha$ , por ejemplo, para el monomio  $x_1^2 x_2$  tenemos  $|\alpha| := \alpha_1 + \alpha_2 = 2 + 1 = 3$ , de esta forma el monomio  $x_1^2 x_2$  es de tercer grado.

En todo el artículo usaremos las notaciones usuales del análisis real, entre las cuales denotamos por  $\mathbb{R}^n$  al conjunto de vectores de la forma

$$x = (x_1, x_2, \dots, x_n), \quad \|x\| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$$

y una matriz  $A$  se denomina semidefinida positiva, denotada por  $A \succeq 0$ , si  $x^T A x \geq 0$ , para todo  $x \in \mathbb{R}^n$ .

Dada una función polinomial  $f(x): \mathbb{R}^n \rightarrow \mathbb{R}$ , un problema de optimización polinomial es definido como

$$(P) \quad f^* = \text{minimizar } \sum f_\alpha x^\alpha \\ \text{sujeto a } x \in X \subseteq \mathbb{R}^n$$

es decir, encontrar un mínimo global  $f^*$  de  $f(x)$ , y un minimizador  $x^* \in X = \{x \in \mathbb{R}^n : g(x) \geq 0\}$  donde  $g(x)$  es una función polinomial.

Claramente resolver un problema de optimización polinomial no es nada fácil, en el caso de un polinomio general, desde que el problema (P) es no convexo, el problema es NP-difícil y generaliza varios problemas de optimización como programación lineal, cuadrático, cúbico, etc.

Shor [23], fue uno de los primeros en introducir la idea de aplicar técnicas de optimización convexa para minimizar funciones polinomiales de varias variables. Así como también, Nesterov [18] fue uno de los primeros en discutir cómo explotar la dualidad de conos de momentos y conos de polinomios no negativos en un marco de optimización convexo. Entre las primeras investigaciones más relevantes sobre el estudio de optimización polinomial podemos mencionar los trabajos realizados por Lasserre [13] y Parrilo [20] ambos aplicando resultados de la teoría algebraica.

Recientemente el estudio de problemas de optimización polinomial se ha intensificado, motivado por las diversas aplicaciones que se encuentran en las áreas de Ingeniería y Ciencias Exactas, como por ejemplo, en procesamiento de señales, problemas de imagen de resonancia magnética (MRI), teoría de aproximación de matrices de bajo rango, entre otros. Dentro de los trabajos que muestran estas aplicaciones podemos sugerir la lectura de los siguientes artículos [1],[2],[3],[12],[21],[27].

En este artículo hacemos una revisión de los principales métodos propuestos en la literatura para

encontrar soluciones aproximadas de un problema de optimización polinomial y mostramos resultados numéricos, implementando en MATLAB, algunos de estos métodos.

El artículo está organizado de la siguiente forma: en la Sección 2, describimos los diferentes métodos computacionales para optimización polinomial, luego en la Sección 3 presentamos resultados numéricos implementando algunos de de estos métodos. Finalmente, en la Sección 4 damos nuestras conclusiones.

## 2 Metodología

### 2.1 Revisión de métodos en optimización polinomial

En este trabajo realizamos un estudio de los diversos métodos de optimización que buscan resolver un problema de optimización polinomial.

Uno de los primeros artículos publicados sobre optimización polinomial fue elaborado por Lasserre [13], el resultado principal del artículo muestra que la solución global de un problema de minimizar funciones polinomiales sobre un conjunto compacto definido por desigualdades polinomiales puede ser aproximado por las soluciones de problemas convexos (denominados LMI). Mas específicamente, para el problema (P) en el caso  $X = \mathbb{R}^n$ , la secuencia  $\{Q^m\}$

$$Q^m : \inf_x \sum f_\alpha y_\alpha \text{ sujeto a } M_m(y) \succeq 0$$

donde  $y = \{y_\alpha\}$  es el vector de momentos (con  $y_{0,0} = 1$ ) definido por  $y_\alpha := \int x^\alpha d\mu$ . para alguna medida de probabilidad  $\mu$ , el símbolo  $\succeq$  significa que la matriz  $M_m(y)$  es semidefinida positiva y  $M_m(y)$  es la matriz de bloques  $\{M_{ij}(y)\}$ , donde los índices  $i, j = 0, 1, 2, \dots, 2m$ , definida por

$$M_{ij}(y) = \begin{bmatrix} y_{i+j,0} & y_{i+j-1,1} & \dots & y_{ij} \\ y_{i+j-1,1} & y_{i+j-2,2} & \dots & y_{i-1,j+1} \\ \dots & \dots & \dots & \dots \\ y_{j,i} & y_{i+j-1,1} & \dots & y_{0,i+j} \end{bmatrix},$$

converge para  $f^*$  cuando  $m \rightarrow \infty$ , donde  $y_{ij}$  representa el momento de orden  $(i + j)$ , con

$y_{ij} = \int x^i y^j \mu(d(x, y))$  para alguna medida de probabilidad  $\mu$ . Bajo algunas consideraciones adicionales, Lasserre [13] muestra resultados análogos cuando el conjunto de restricciones es un conjunto compacto.

Esta metodología permite resolver el problema de optimización polinomial usando la teoría de momentos, generando una sucesión de soluciones de problemas LMI (convexos), y mostrando que esta sucesión converge a una solución óptima global del problema de optimización polinomial. Basado en esta idea, Henrion y Lasserre [9] desarrollan un programa computacional, llamado GloptiPoly, este software es realizado en Matlab buscando construir y resolver los problemas LMI, que son las relajaciones de problemas de optimización polinomial.

GloptiPoly no intenta resolver globalmente los problemas no convexos pero si permite resolver los problemas LMI iterativamente de modo que estas soluciones generan una sucesión que convergen en el límite a una solución óptima global del problema. Este software fue utilizado en diversas instancias de tamaño mediano obteniendo óptimos resultados a un tiempo computacional bastante corto. En el 2009, D. Henrion et al. [10] presentan la versión mas actualizado del programa llamado GloptiPoly3. Para mas detalles sobre la instalación y uso del software, ver [9],[10].

Debido a la limitación de los solvers para tratar problemas SDP de gran tamaño, GloptiPoly3 puede no ser una buena herramienta para resolver problemas de polinomios de grado alto. Sin embargo, si el modelo de optimización polinomial de alguna manera es esparsa, es decir, si la función objetivo no está definida con todos los monomios posibles, entonces es posible explotar la esparsidad del modelo en GloptiPoly3, ver [15].

Una diferente metodología basada en la teoría de Sumas de Cuadrados (SOS) fue propuesta por Lasserre [14] e Parrilo [20]. La propuesta SOS busca formular el problema de optimización polinomial como un problema de programación semidefinida, posiblemente de dimensión mayor. Debido al desarrollo de eficientes solvers en programación semidefinida, por ejemplo, SeDuMi de Sturm [24], SDPT3 de Toh et al. [25], y SDPA de Fujisawa et al. [7], y aprovechando la esparsidad del modelo de optimización polinomial, la idea de aplicar la metodología SOS es bastante atractiva, obteniendose eficientes resultados como fue mostrado por Waki et al. en [26].

Nie [19], también presenta un método basado en la idea de relajaciones via la teoría de sumas de cuadrados (SOS) para casos particulares de problemas de optimización con polinomios, que

también pueden ser convertidas en relajaciones via programación semidefinida (SDP).

En la literatura surgen diversos métodos para resolver algunos casos particulares de problemas de optimización polinomial. Entre estos métodos podemos mencionar los siguientes.

He et al. [8], presentan un método de optimización cuyos resultados numéricos son obtenidos para el caso de minimizar funciones homogéneas sujeto a restricciones definidas por funciones cuadráticas. El objetivo de este trabajo es la búsqueda de un esquema general que permita obtener soluciones aproximadas con razón de rendimiento garantizada en el peor de los casos. Mas especificamente, considerando la función multilineal

$$F(x^1, x^2 \dots x^d) = \sum a_{i_1 i_2 \dots i_d} x_{i_1}^{1} x_{i_2}^{2} \dots x_{i_d}^d$$

donde  $x^k \in \mathbb{R}^{n_k}$ ,  $k = 1, \dots, d$ .

Dado los modelos:

$$(A_1) \text{ maximizar } F(x^1, x^2, \dots, x^d) \\ \text{s.a } ||x^k|| = 1, x^k \in \mathbb{R}^{n_k}, \\ k=1,2,\dots,d$$

$$(A_2) \text{ maximizar } F(x, x, \dots, x) \\ \text{s.a } ||x|| = 1, x \in \mathbb{R}^n$$

aquí, en la función objetivo se repite d veces la variable x.

$$(A_3) \text{ maximizar } F(x^1, x^2, \dots, x^d) \\ \text{s.a } (x^k)^T Q_{i_k} x^k \leq 1, k=1,2,\dots,d; \\ i_k = 1,2,\dots,m_k; \\ x^k \in \mathbb{R}^{n_k}, k=1,2,\dots,d.$$

$$(A_4) \text{ maximizar } F(x, x, \dots, x) \\ \text{s.a } x^T Q_i x \leq 1, i=1,2,\dots,m \\ x \in \mathbb{R}^n$$

aquí, en la función objetivo se repite d veces la variable x.

Para cada modelo se obtienen los siguientes resultados:

Modelo	Razón de aproximación
(A <sub>1</sub> )	$(n_1 n_2 \dots n_{d-2})^{-1/2}$
(A <sub>2</sub> )	$d!d^{-d}n^{-(d-2)/2}$
(A <sub>3</sub> )	$\Omega( (n_1 n_2 \dots n_{d-2})^{1/2} (\log \max m_k)^{d-1} )^{-1}$
(A <sub>4</sub> )	$\Omega( d!d^{-d}(n^{(d-2)/2} \log^{d-1} m)^{-1} )$

donde la notación  $\Omega(\lambda)$  se debe entender como “al menos en el orden de  $\lambda$ ”.

Por otro lado, debido a la natural estructura de no convexidad de los problemas en optimización polinomial, surgen diversos estudios que buscan algoritmos eficientes para encontrar buenas soluciones de puntos KKT (puntos estacionarios que son optimizadores locales). Uno de estos métodos es el propuesto por Chen et al. [5], llamado maximun block improvement (MBI), este método garantiza que cualquier punto de acumulación de la sucesión generada por el algoritmo es un punto estacionario. Además esta método se ha mostrado bastante eficiente en la práctica, véase [5],[6].

En esta dirección, Jiang et al. [11] proponen aplicar el método llamado alternating direction method of multipliers (ADMM), para resolver problemas de optimización polinomial en el caso real y complejo, con el objetivo de encontrar buenos puntos de KKT.

Cuando la función objetivo es cúbica y la restricción es definida por una esfera, el problema (P) es denominado problema cúbico esférico, el cual fue introducido por Zhang et al. [28]. Recientemente, Buchheim et al. [4], presentan un método para obtener limitantes inferiores del óptimo global del problema cúbico esférico, aplicando la metodología de Descomposición Lagrangeana. La relevancia de este método es la practicidad de su implementación así como también su utilidad para resolver problemas de gran tamaño.

### 3. Experimentos Computacionales

En esta sección mostramos resultados numéricos del trabajo de investigación realizados por Sarmiento y Fampa [22], al aplicar el método alternating direction method of multipliers (ADMM), GloptiPoly3 y el solver fmincon del Matlab, para resolver un problema de optimización esférico cúbico.

Las implementaciones son realizadas en MATLAB R2017b en una laptop Intel core i5-8250U 1.60 GHz con 8GB de RAM, compilando bajo el sistema

operacional Ubuntu 18.04. En cada instancia consideramos los puntos iniciales iguales para la comparación de los métodos, siguiendo una distribución normal en el intervalo [-1,1].

Lista de abreviaciones mencionadas en las tablas.

**fmincon** : aplicando el solver fmincon con un punto inicial aleatorio  $\bar{x}$ ;

**ADMM** : aplicando el método ADMM con punto inicial  $(x^0)^{(0)} = (x^1)^{(0)} = (x^2)^{(0)} = (x^3)^{(0)} = \bar{x}$  aleatorios.

**GLP** : valor obtenido al aplicar GloptiPoly3;

**val** : valor de la función en la solución obtenida por el método correspondiente;

**t** : tiempo dado en segundos.

Tabla 1: Comparación del algoritmo ADMM, fmincon y GloptiPoly3 para 10 problemas aleatorias de dimensión  $n=5$ .

I	fmincon		ADMM		GLP	
	val	t	val	t	val	t
1	<b>-1.671</b>	0.158	<b>-1.671</b>	0.005	<b>-1.671</b>	0.637
2	<b>-2.098</b>	0.274	<b>-2.098</b>	0.005	<b>-2.098</b>	0.282
3	<b>-2.325</b>	0.158	<b>-2.325</b>	0.004	<b>-2.325</b>	0.217
4	-0.911	0.148	<b>-1.495</b>	0.022	<b>-1.495</b>	0.290
5	-1.280	0.132	-1.280	0.003	<b>-1.866</b>	0.226
6	-1.347	0.147	<b>-1.800</b>	0.168	<b>-1.800</b>	0.210
7	-1.185	0.193	-1.185	0.008	<b>-1.874</b>	0.217
8	-0.691	0.103	-0.911	0.225	<b>-1.100</b>	0.303
9	<b>-2.075</b>	0.148	<b>-2.075</b>	0.003	<b>-2.075</b>	0.211
10	<b>-1.424</b>	0.127	<b>-1.424</b>	0.009	<b>-1.424</b>	0.228

Tabla 2: Comparación del algoritmo ADMM, fmincon y GloptiPoly3 para 10 problemas aleatorias de dimensión  $n=10$ .

I	fmincon		ADMM		GLP	
	val	t	val	t	val	t
1	<b>-3.078</b>	0.562	<b>-3.078</b>	0.032	<b>-3.078</b>	4.965
2	-2.344	0.374	-2.343	0.006	<b>-3.221</b>	3.728
3	-2.352	0.391	<b>-2.848</b>	0.058	<b>-2.848</b>	3.713
4	-2.121	0.414	<b>-2.364</b>	0.035	<b>-2.364</b>	5.068
5	-1.864	0.286	-2.564	0.036	<b>-2.819</b>	3.975
6	-2.280	0.390	-2.302	0.009	<b>-2.591</b>	3.932
7	-2.705	0.473	-2.629	0.324	<b>-2.705</b>	4.483
8	-2.058	0.274	<b>-2.927</b>	0.032	<b>-2.927</b>	4.250
9	-1.907	0.252	-1.907	0.007	<b>-2.767</b>	4.270
10	<b>-2.993</b>	0.300	<b>-2.993</b>	0.007	<b>-2.993</b>	3.973

Tabla 3: Comparación del algoritmo ADMM, fmincon y GloptiPoly3 para 10 problemas aleatorias de dimensión  $n=15$ .

I	fmincon		ADMM		GLP	
	val	t	val	t	val	t
1	-3.286	0.447	-3.286	0.009	<b>-3.406</b>	314.9
2	-3.314	0.481	-3.163	0.060	<b>-3.314</b>	254.6
3	-3.096	0.455	<b>-3.150</b>	0.433	<b>-3.150</b>	273.9
4	-3.256	0.610	<b>-3.420</b>	0.442	<b>-3.420</b>	233.1
5	-3.032	0.455	-2.962	0.236	<b>-3.113</b>	233.1
6	-3.645	0.471	<b>-3.735</b>	0.057	<b>-3.735</b>	214.5
7	-3.244	0.427	-3.244	0.013	<b>-3.371</b>	249.2
8	-2.560	0.396	-2.262	0.436	<b>-3.145</b>	232.9
9	-3.355	0.433	-3.355	0.010	<b>-3.740</b>	250.1
10	-2.915	0.501	-2.915	0.011	<b>-3.127</b>	215.4

En las tablas 1-3, resaltamos en negrita los mejores valores obtenidos entre los 3 métodos, de los resultados mostrados podemos deducir que el método ADMM es una óptima alternativa para la obtención de óptimos locales para el problema de optimización cúbico esférico. El método ADMM obtuvo mejores resultados para la mayor parte de las instancias, con un menor tiempo de ejecución, notando que el 50% de los problemas, el método ADMM obtiene el óptimo global.

Por otro lado, es muy importante resaltar que el método GloptiPoly3 resuelve todas las instancias hasta conseguir el óptimo global de cada problema. La comparación se dió hasta  $n=15$  variables debido que no fue posible ejecutar el método GloptiPoly3 dentro de un periodo limite determinado de 30 minutos, verificando las limitaciones del método cuando no hay esparcidad en la función polinomial.

#### 4. Conclusiones

En este trabajo de investigación se ha realizado una revisión de los principales métodos propuestos en la literatura para resolver problemas de optimización definidas por polinomios. Se ha podido constatar que existen diversos métodos, todos muy bien estructurados tanto en la parte teórica como en la parte computacional. A pesar que existe aún la dificultad de tener soluciones óptimas globales para estos problemas, algunos métodos presentados obtienen eficientes aproximaciones de la solución a un costo computacional bastante corto, como son los métodos GloptiPoly3 [10] y SparsePOP [26], todos ellos aprovechando la esparcidad de cada problema.

Por otro lado, se há podido constatar que métodos que buscan buenos puntos de KKT (minimizadores locales) son bastante eficientes en la práctica, como por ejemplo, el método MBI [5] y el método ADMM [11]. Estos resultados motivan a seguir desarrollando nuevas metodologías con la finalidad de mejorar las soluciones aproximadas locales, intentando buscar el óptimo global. Asi como también, a partir de los buenos resultados obtenidos, repetirlos en la práctica para problemas de optimización con funciones polinomiales completos (sin esparcidad de sus monomios) y de grado alto.

#### Agradecimientos

Los autores agradecen a las instituciones UFRJ – Campus Duque de Caxias, y UERJ por disponibilizar sus instalaciones durante la realización de este trabajo.

#### Referencias

- [1] A. Barmpoutis, B. Jian, B. C. Vemuri, and T. M. Shepherd, Symmetric positive 4th order tensors and their estimation from diffusion weighted MRI. In: Karssemijer, N., Lelieveldt, B. (eds.) IPMI 2007, LNCS 4584 (2007), 308 - 319.
- [2] P. J. Basser and D. K. Jones. Diffusion-tensor MRI: theory, experimental design and data analysis-a technical review, NMR Biomed. 15 (2002), 456 - 467.
- [3] P. J. Basser, J. Mattiello and D. LeBihan. Estimation of the effective seldiffusion tensor from the NMR spin echo. Journal of Magnetic Resonance, 8 (1994), 247 - 254.
- [4] C. Buccheim, M. Fampa and O. Sarmiento, Lower Bounds for Cubic Optimization over the Sphere, Journal of Optimization Theory and Applications, 3 (2021) 823-846.
- [5] B. Chen, S. He, Z. Li, S. Zhang, Maximum block improvement and polynomial optimization. SIAM J. Optim. (2012), 87 -107.
- [6] B. Chen. Optimization with block variables: theory and applications [PhD Thesis]. Hong Kong: The Chinese University of Hong Kong; 2012.
- [7] K. Fujisawa, M. Kojima, K. Nakata, and M. Yamashita, SDPA (SemiDefinite Programming Algorithm) User's Manual - version 6.2.0, Research Report B-308, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Japan (1995).

- [8] S. He, Z. Li and S. Zhang, Approximation algorithms for homogeneous polynomial optimization with quadratic constraint, 2 (2010), 353-383.
- [9] D. Henrion, J. B. Lasserre, GloptiPoly: Global optimization over polynomials with Matlab and SeDuMi, ACM Transactions on Mathematical Software, 29-2, (2003), 165 - 194.
- [10] D. Henrion, J. B. Lasserre and J. Loeferberg, GloptiPoly3: moments, optimization and semidefinite programming, Optim. Meth. Softw, 24 (2009), 761 - 779.
- [11] B. Jiang, S. Ma y S. Zhang, Alternating direction method of multipliers for real and complex polynomial optimization models, Optimization: Journal of Mathematical Programming and Operations Research, 6 (2014), 883-898.
- [12] E. Jondeau, M. Rockinger, Optimal portfolio allocation under higher moments. Eur. Financ. Manage. 12 (2006), 29 - 55.
- [13] J. B. Lasserre, Global optimization with polynomials and the problem of moments, SIAM Journal on Optimization, 3 (2001) 796 - 817.
- [14] J. B. Lasserre, Polynomials nonnegative on a grid and discrete representations. Trans. Am. Math. Soc. 354, (2001), 631 – 649.
- [15] J. B. Lasserre, Convergent SDP relaxations in polynomial optimization with sparsity, SIAM J. Optim., 17 (2006), 822 - 843.
- [16] J. B. Lasserre, A semidefinite programming approach to the generalized problem of moments. *Math. Program*, 112, (2008) 65 - 92.
- [17] B. Maricic, Z. Q. Luo, T. N. Davidson, Blind constant modulus equalization via convex optimization. IEEE Trans. Signal Process. 51, (2003), 805 - 818.
- [18] Y. Nesterov, Squared functional systems and optimization problems, in J.B.G. Frenk, C. Roos, T. Terlaky, and S. Zhang, editors, High Performance Optimization, 405 - 440. Kluwer Academic Publishers (2000)
- [19] J. Nie, Sum of squares methods for minimizing polynomial forms over spheres and hypersurfaces, Frontiers of Mathematics in China, 2 (2012), 321 - 346.
- [20] P. A. Parrilo, Semidefinite programming relaxations for semialgebraic problems, Math. Programming, 96 (2003), 293 - 320.
- [21] L. Qi, L. K. L. Teo, Multivariate polynomial minimization and its applications in signal processing. J. Glob. Optim. 26 (2003), 419 - 433.
- [22] O. Sarmiento e M. Fampa. O método ADMM para um problema de otimização polinomial não convexo, Proceeding Series of the Brazilian Society of Computational and Applied Mathematics (Preprint), 2022.
- [23] N. Z. Shor, Class of global minimum bounds of polynomial functions, Cybernetics, 23: 6 (1987), 731 - 734.
- [24] J. F. Sturm, SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. Optim. Methods Softw. 11 & 12, (1999) 625 - 653.
- [25] K. C. Toh, M. J. Todd, R. H. Tutuncu., SDPT3—a Matlab software package for semidefinite programming. Optim. Methods Softw. 11, (1999), 545 - 581.
- [26] H. Waki, S. Kim, M. Kojima, M. Muramatsu, and H. Sugimoto, Algorithm 883: SparsePOP: A sparse semidefinite programming relaxation of polynomial optimization problems, ACM Trans. Math. Software, 35 (2008), 1 - 13.
- [27] X. Zhang, C. Ling and L. Qi, The best rank-1 approximation of a symmetric tensor and related spherical optimization problems, SIAM L. Matrix Anal. Appl, 33 (2012), 806 - 821.
- [28] X. Zhang, L. Qi and Y. Ye, The cubic spherical optimization problems, Mathematics of Computation, 81 (2012), 1513 - 1525.