

MMDA na resolução da relaxação por Programação Semidefinida do Problema Quadrático de Alocação

Danilo Elias Oliveira¹

Faculdade de Matemática, UFU, Uberlândia, MG

Henry Wolkowicz²

Department of Combinatorics and Optimization, University of Waterloo, Waterloo, ON, Canadá

Yangyang Xu³

Institute for Mathematics and its Applications - IMA, University of Minnesota, Minneapolis, MN, EUA

Resumo. A relaxação por Programação Semidefinida (PSD) já demonstrou ser extremamente útil para muitos problemas difíceis da Otimização Discreta. Em especial, para o problema quadrático de alocação (PQA), conhecido por ser um dos problemas mais difíceis da classe *NP-hard* da Otimização Combinatória. Várias são as dificuldades encontradas ao se resolver a relaxação por PSD através dos métodos atuais. Neste trabalho, propomos a utilização do método dos multiplicadores com direção alternada (MMDA) para resolver a relaxação por PSD do PQA. Obtemos, assim, iterações mais rápidas; um método rápido para se obter soluções com posto deficiente; e, também, uma forma simples de se adicionar desigualdades de planos de corte. Em nossos experimentos numéricos, obtivemos resultados mais robustos, eficientes e melhores aproximações para as soluções do PQA.

Palavras-chave. Problema quadrático de alocação, relaxação por programação semidefinida, método dos multiplicadores com direção alternada.

1 Introdução

A primeira formulação de um Problema Quadrático de Alocação (PQA) foi apresentada por Koopmans e Beckmann em [4], com o objetivo de instalar pares de n fábricas a pares de n locais minimizando o custo total. Em [5], podemos encontrar várias aplicações e formulações para o PQA. Neste trabalho iremos formulá-lo utilizando o produto interno definido pela função traço de uma matriz, $\langle Y, X \rangle = \text{tr}(YX^t)$.

Desta forma, definimos o PQA como sendo o seguinte problema de minimização,

$$p_X^* := \min_{X \in \Pi_n} \langle FXD - 2C, X \rangle, \quad (1)$$

onde F e D são matrizes $n \times n$ reais e simétricas ($F, D \in \mathbb{S}^n$), que representam, respectivamente, o fluxo entre os pares de objetos e a distância entre os pares de locais. Além

¹daniloelias@famat.ufu.br

²hwoikowicz@uwaterloo.ca

³xuyang.gucas@gmail.com

disso, $C \in \mathbb{R}^n$ representa o custo de cada atribuição e Π_n representa o conjunto de todas as matrizes de permutação de ordem n .

Atualmente, o PQA é um dos problemas mais difíceis da classe *NP-hard* da Otimização Combinatória, onde exemplares de ordem $n \geq 30$ são considerados de grande porte e de difícil solução. Por isso, as técnicas atuais consistem em determinar limites inferiores de forma eficiente. Uma importante ferramenta para se determinar tais limites é a relaxação por Programação Semidefinida (PSD), apresentada em [8].

Neste trabalho, apresentamos um método dos multiplicadores com direção alternada (MMDA) para resolver a relaxação por PSD do PQA. Realizamos, também, uma comparação dos resultados obtidos com os melhores limites atuais, apresentados em [6], e com um método de pontos-interiores primal-dual, p-i p-d, que utiliza as direções HKM. Em nossos resultados, podemos ver que o método MMDA é significativamente mais rápido, além de fornecer resultados mais exatos. Também obtivemos soluções com posto deficiente para a relaxação, o que nos proporcionou melhores aproximações viáveis para a solução do PQA, funcionando como um limite superior. Podemos encontrar mais informações sobre o MMDA em [3].

2 Relaxação por PSD

A relaxação por PSD do PQA apresentada em [8] é obtida a partir do seguinte formulação equivalente de (1),

$$\begin{aligned} \min_X & \quad \langle FXD - 2C, X \rangle \\ \text{s.a.} & \quad XX^t = I; X^tX = I; X_{ij}^2 - X_{ij} = 0, \forall i, j, \\ & \quad \|Xe - e\|^2 + \|X^t - e\|^2 = 0. \end{aligned}$$

Assim, utilizando uma *redução facial*, para garantir a existência de elementos estritamente viáveis, a relaxação e o respectivo problema dual obtidos são dados por

$$\begin{aligned} p_R^* = \min_R & \quad \langle L_Q, \hat{V}R\hat{V}^t \rangle & d_Y^* = \max_Y & \quad \langle E_{00}, Y \rangle \quad (= Y_{00}) \\ \text{s.a.} & \quad \mathcal{G}_J(\hat{V}R\hat{V}^t) = E_{00} & \text{s.a.} & \quad \hat{V}^t \mathcal{G}_J(Y) \hat{V} \preceq \hat{V}^t L_Q \hat{V} \\ & \quad R \succeq 0, & & \end{aligned} \quad (2)$$

onde

$$L_Q = \begin{bmatrix} 0 & -\text{vec}(C)^t \\ -\text{vec}(C) & D \otimes F \end{bmatrix}, \quad \hat{V} = \begin{bmatrix} 1 & 0 \\ \frac{1}{n}e & V \otimes V \end{bmatrix}, \quad Y = \hat{V}R\hat{V}^t = \begin{bmatrix} y_{00} & y_0^t \\ y_0 & \bar{Y} \end{bmatrix},$$

$\text{vec}(C)$ é a vetorização, de acordo com as colunas, de C , e representa o vetor com todas as componentes iguais à 1, na dimensão apropriada, $V \in \mathbb{R}^{n \times (n-1)}$ uma base matricial do complemento ortogonal de e , por exemplo, $V = \begin{bmatrix} I_{n-1} \\ -e \end{bmatrix}$, o símbolo \otimes denota o produto de Kronecker e \mathcal{G}_J , denominado *operador gângester*, mantém fixo os elementos cujos índices pertencem à J e anula os demais elementos.

O conjunto de índices, J , é definido como sendo o primeiro elemento da matriz Y , y_{00} , e os seguintes elementos com índices $i < j$ de \bar{Y} :

- os que não pertencem à diagonal principal nos n blocos diagonais,
- e os que pertencem à diagonal principal dos blocos que não pertencem à diagonal principal. Os blocos da última coluna que estão fora da diagonal principal e também o bloco $((n - 2), (n - 1))$ podem ser desconsiderados por serem redundantes.

Ainda em [8], temos que as matrizes simétricas \hat{R} , \hat{Y} e \hat{Z} definidas por

$$\hat{R} := \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & \frac{1}{n^2(n-1)} (nI_{n-1} - E_{n-1}) \otimes (nI_{n-1} - E_{n-1}) \end{array} \right]$$

$$\hat{Y} := M \left[\begin{array}{c|c} n & 0 \\ \hline 0 & I_n \otimes (I_n - E_n) \end{array} \right] \quad \text{e} \quad \hat{Z} := \hat{V}^t L_Q \hat{V} - \hat{V}^t \mathcal{G}_J(\hat{Y}) \hat{V}$$

para $M > 0$ suficientemente grande, são estritamente viáveis para (2).

3 Um novo algoritmo MMDA para a relaxação por PSD

O problema primal em (2) pode ser reescrito da seguinte forma:

$$\min_{R, Y} \langle L_Q, Y \rangle, \text{ s.a } \mathcal{G}_J(Y) = E_{00}, Y = \hat{V} R \hat{V}^t, R \succeq 0. \quad (3)$$

Assim, o Lagrangeano aumentado de (3) é dado por,

$$\mathcal{L}_A(R, Y, Z) = \langle L_Q, Y \rangle + \langle Z, Y - \hat{V} R \hat{V}^t \rangle + \frac{\beta}{2} \|Y - \hat{V} R \hat{V}^t\|_F^2. \quad (4)$$

Sejam (R, Y, Z) o ponto obtido na iteração k e \mathbb{S}^{rn} o conjunto das matrizes que pertencem à \mathbb{S}^n e que possuem posto igual, ou menor, do que r . Nosso algoritmo é uma aplicação do método dos multiplicadores com direção alternada, MMDA, que utiliza o Lagrangeano aumentado em (4) para calcular as seguintes iterações:

$$R_{k+1} = \arg \min_{R \in \mathbb{S}^{rn}} \mathcal{L}_A(R, Y, Z), \quad (5)$$

$$Y_{k+1} = \arg \min_{Y \in \mathcal{P}_i} \mathcal{L}_A(R_{k+1}, Y, Z), \quad (6)$$

$$Z_{k+1} = Z + \gamma \cdot \beta (Y_{k+1} - \hat{V} R_{k+1} \hat{V}^t), \quad (7)$$

onde o caso mais simples para as restrições do poliedro \mathcal{P}_i é o subespaço afim obtido pelas restrições do gângster: $\mathcal{P}_1 = \{Y \in \mathbb{S}^{n^2-1} : \mathcal{G}_J(Y) = E_{00}\}$. O segundo caso é o politopo $\mathcal{P}_2 = \mathcal{P}_1 \cap \{0 \leq Y \leq 1\}$.

Seja \hat{V} tal que $\hat{V}^t \hat{V} = I$. Assim, se $r = n$, podemos calcular R_{k+1} de forma explícita,

$$R_{k+1} = \mathcal{P}_{\mathbb{S}_+} \left(\hat{V}^t \left(Y + \frac{Z}{\beta} \right) \hat{V} \right), \quad (8)$$

onde \mathbb{S}_+ representa o cone PSD, e $\mathcal{P}_{\mathbb{S}_+}$ é a projeção em \mathbb{S}_+ . Para qualquer matriz simétrica W , temos $\mathcal{P}_{\mathbb{S}_+}(W) = U_+ \Sigma_+ U_+^t$, onde (U_+, Σ_+) contem os autopares (autovalores e autovetores) positivos de W e (U_-, Σ_-) os autopares negativos.

Se $i = 1$ em (6), podemos, então, calcular Y_{k+1} de forma explícita,

$$Y_{k+1} = E_{00} + \mathcal{G}_{J^c} \left(\hat{V}R_+\hat{V}^t - \frac{L_Q + Z}{\beta} \right),$$

onde J^c representa o complementar do conjunto de índices J .

A vantagem de se utilizar o MMDA é que a sua complexidade não aumenta muito quando acrescentamos restrições ao problema primal em (2) para melhorar a relaxação por PSD. Por exemplo, ao acrescentarmos a restrição $0 \leq \hat{V}R\hat{V}^t \leq 1$, temos, então, a restrição $0 \leq Y \leq 1$ em (3), de onde obtemos o problema

$$P_{RY}^* = \min_{R,Y} \langle L_Q, Y \rangle, \text{ s.a } \mathcal{G}_J(Y) = E_{00}, 0 \leq Y \leq 1, Y = \hat{V}R\hat{V}^t, R \succeq 0. \quad (9)$$

Desta forma, o MMDA para resolver (9) possui as mesmas expressões que (5) e (7) para atualizar R e Z , respectivamente. Porém, a expressão para atualizar Y é dada por,

$$Y_{k+1} = E_{00} + \min \left(1, \max \left(0, \mathcal{G}_{J^c} \left(\hat{V}R_+\hat{V}^t - \frac{L_Q + Z}{\beta} \right) \right) \right),$$

Com a restrição de não-negatividade, a restrição “menor do que um” se torna redundante, mas faz com que o algoritmo convirja mais rapidamente.

4 Limite inferior e solução viável para o PQA

Ao resolvermos os problemas primal em (2) ou (3) com uma alta exatidão obtemos um limite inferior para o PQA original em (1). No entanto, a dimensão destes problemas podem ser extremamente grandes, o que requerem um altíssimo custo computacional.

Apresentamos, agora, uma forma eficiente para se obter um limite inferior para p_X^* a partir do resultado de nosso algoritmo, que fornece uma solução para (3) com uma exatidão moderada. Sejam (R^*, Y^*, Z^*) a solução obtida para (9) pelo MMDA.

Lemma 4.1. *Sejam $\mathcal{R} = \{R : R \succeq 0\}$, $\mathcal{Y} = \{Y : \mathcal{G}_J(Y) = E_{00}, 0 \leq Y \leq 1\}$, e $\mathcal{Z} = \{Z : \hat{V}^t Z \hat{V} \preceq 0\}$. Defina $g(Z) = \min_{Y \in \mathcal{Y}} \langle L_Q + Z, Y \rangle$ como sendo a função dual do MMDA. Então, o problema dual de (9) é dado por $d_Z^* := \max_{Z \in \mathcal{Z}} g(Z)$ e $d_Z^* \leq P_R^*$.*

Demonstração: O problema dual de (9) pode ser derivado da seguinte forma:

$$\begin{aligned} d_Z^* &:= \max_Z \min_{R \in \mathcal{R}, Y \in \mathcal{Y}} \langle L_Q, Y \rangle + \langle Z, Y - \hat{V}R\hat{V}^t \rangle \\ &= \max_Z \min_{Y \in \mathcal{Y}} \langle L_Q, Y \rangle + \langle Z, Y \rangle + \min_{R \in \mathcal{R}} \langle \hat{V}^t Z \hat{V}, -R \rangle = \max_{Z \in \mathcal{Z}} g(Z) \end{aligned}$$

A dualidade fraca segue ao se permutar max e min. □

Assim, $g(Z)$ é um limite inferior de (9), e, também, do PQA original, para toda matriz $Z \in \mathcal{Z}$. Em nossos experimentos, utilizamos como limite inferior para o PQA a função dual da projeção $\mathcal{P}_{\mathcal{Z}}(Z^*)$, onde

$$\mathcal{P}_{\mathcal{Z}}(\tilde{Z}) = \arg \min_{Z \in \mathcal{Z}} \|Z - \tilde{Z}\|_F^2 = \arg \min_{W_{11} \leq 0} \|W - \tilde{V}^t \tilde{Z} \tilde{V}\|_F^2 = \begin{bmatrix} \mathcal{P}_{S_-}(\tilde{W}_{11}) & \tilde{W}_{12} \\ \tilde{W}_{21} & \tilde{W}_{22} \end{bmatrix},$$

\hat{V}_\perp é uma base ortonormal do espaço nulo de \hat{V} , \bar{V} é a matriz ortogonal ($\hat{V} \hat{V}_\perp$), \mathcal{S}_- representa o cone semidefinido negativo e $\bar{V}^t \tilde{Z} \bar{V} = \begin{bmatrix} \tilde{W}_{11} & \tilde{W}_{12} \\ \tilde{W}_{12} & \tilde{W}_{22} \end{bmatrix}$.

Agora, assumindo que o maior autovalor e o correspondente autovetor de Y^* são λ e v , respectivamente, a solução viável para o PQA é obtida resolvendo o problema linear

$$\max_X \langle \tilde{X}, X \rangle, \text{ s.a } Xe = e, X^t e = e, X \geq 0, \tag{10}$$

pelo método simplex, onde \tilde{X} é uma matriz quadrada obtida da reconfiguração do segundo até o último elemento da primeira coluna da matriz λvv^t .

Por outro lado, ao invés de se obter uma solução viável para o PQA resolvendo (10), podemos obtê-la diretamente ao restringirmos R a uma matriz com posto igual a 1. Desta forma, a expressão em (5) é modificada para

$$R_{k+1} = \mathcal{P}_{\mathcal{S}_+ \cap \mathcal{R}_1} \left(\hat{V}^t \left(Y + \frac{Z}{\beta} \right) \hat{V} \right), \tag{11}$$

onde \mathcal{R}_1 denota o conjunto das matrizes com posto igual a 1. Se W é uma matriz simétrica e (λ, w) é autopar de W onde λ é o maior autovalor positivo, então $\mathcal{P}_{\mathcal{S}_+ \cap \mathcal{R}_1} = \lambda w w^t$.

5 Experimentos numéricos

Em nossos experimentos utilizamos 45 problemas da coleção QAPLIB [2], os resultados estão apresentados na Tabela 1. Os valores ótimos de cada problema estão na segunda coluna, na terceira coluna temos os melhores limites inferiores, atualmente, apresentados em [6]. Os limites inferiores apresentados na quarta coluna foram obtidos por um método p-i p-d, o qual se encontra implementado no código SDPT3 [7]. Este código não conseguiu resolver os 11 problemas que estão marcados com -1111.

Os limites inferiores obtidos pelo algoritmo MMDA estão na quinta coluna. A coluna seguinte apresenta os melhores limites superiores obtidos pela nossa heurística. Finalmente, as duas últimas colunas apresentam o tempo computacional de nosso algoritmo onde utilizamos, respectivamente, R com posto completo, (8), e $\text{posto}(R) = 1$, (11).

Utilizamos em nossos experimentos o MATLAB versão 8.6.0.267246 (R2015b) em um computador Dell Optiplex 9020 64-bit, com 16 Gigabytes de memória e com o sistema operacional Windows 7.

Utilizamos, de forma heurística, em nosso algoritmo MMDA, $\gamma = 1.618$, $\beta = \frac{n}{3}$ e duas tolerâncias, $1e - 5$, $1e - 12$. No entanto, resolver a relaxação por PSD com uma alta exatidão não produziu melhores limites para o PQA, porém o fato do algoritmo resolver a relaxação com esta exatidão merece ser destacado.

Quanto ao tempo computacional, destacamos que o maior tempo computacional foi de 2 horas para o problema NUG30, enquanto o mesmo problema leva 10 horas para ser resolvido pelo método bundle em [6].

Analisando a Tabela 1 podemos verificar que todos os limites foram melhorados. Em especial, o algoritmo MMDA encontrou a solução exata para seis problemas: HAD12, 14, 16, 18, ROU12 e TAI12A.

6 Conclusões

Podemos destacar a eficiência da abordagem MMDA para resolver a relaxação por PSD do PQA. Em especial, mostramos que é possível obter soluções da relaxação por PSD com uma alta exatidão e a um custo computacional menor do que as abordagens mais utilizadas atualmente. Além disso, a inclusão da restrição de não-negatividade não apresentou um grande aumento no tempo computacional do algoritmo. Por estes motivos, o algoritmo pode nos fornecer melhores limites inferiores, e superiores, para o PQA de forma mais eficiente.

Agradecimentos

Os autores agradecem o Natural Sciences and Engineering Research Council - NSERC, a AFOSR e ao CNPq.

Referências

- [1] K. M. Anstreicher and N. W. Brixius. A new bound for the quadratic assignment problem based on convex quadratic programming. *Mathematical Programming*, 89(3, Ser. A):341–357, 2001. DOI: 10.1007/PL00011402.
- [2] R.E. Burkard, S.E. Karisch, and F. Rendl. QAPLIB—a quadratic assignment problem library. *Journal of Global Optimization*, 10(4):391–403, 1997. DOI: 10.1023/A:1008293323270.
- [3] S. Boyd, N. Parikh, E. Chua, B. Peleato and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. trends Machine Learning*, 3(1):1–122, 2011. DOI: 10.1561/22000000016.
- [4] T. C. Koopmans and M. J. Beckmann. Assignment problems and the location of economics activities. *Econometrica*, 25:53–76, 1957. DOI: 10.2307/1907742
- [5] E. M. Loiola, N. M. M. Abreu and P. O. B. Netto. Uma revisão comentada das abordagens do problema quadrático de alocação. *Pesquisa Operacional*, 9:73–108, 2004. ISSN: 0101-7438.
- [6] F. Rendl and R. Sotirov. Bounds for the quadratic assignment problem using the bundle method. *Mathematical Programming*, 109(2-3, Ser. B):505–524, 2007. DOI: 10.1007/s10107-006-0038-8
- [7] K. C. Toh, M. J. Todd e R. H. Tütüncü. SDPT3 - a MATLAB software package for semidefinite programming, version 1.3. *Optim. Methods Software*, 11/12(1-4):545–581, 1999. DOI: 10.1080/10556789908805762
- [8] Q. Zhao, S.E. Karisch, F. Rendl and H. Wolkowicz. Semidefinite programming relaxations for the quadratic assignment problem. *J. Comb. Optim.*, 2(1):71–109, 1998. DOI: 10.1023/A:1009795911987.

Tabela 1: Limites inferiores e tempo computacional para problemas da QAPLIB.

	valor ótimo	Bundle [6] Lim.Inf	HKM Lim.Inf	ADMM Lim.Inf	sol.viável Lim.Sup	cpusec Post.Compl	cpusec Post.1
Esc16a	68	59	50	64	72	2.30e+01	4.02
Esc16b	292	288	276	290	300	3.87e+00	4.55
Esc16c	160	142	132	154	188	1.09e+01	8.09
Esc16d	16	8	-12	13	18	2.14e+01	3.69
Esc16e	28	23	13	27	32	3.02e+01	4.29
Esc16g	26	20	11	25	28	4.24e+01	4.27
Esc16h	996	970	909	977	996	4.91e+00	3.53
Esc16i	14	9	-21	12	14	1.37e+02	4.30
Esc16j	8	7	-4	8	14	8.95e+01	4.80
Had12	1652	1643	1641	1652	1652	1.02e+01	1.08
Had14	2724	2715	2709	2724	2724	3.23e+01	1.69
Had16	3720	3699	3678	3720	3720	1.75e+02	3.15
Had18	5358	5317	5287	5358	5358	4.49e+02	6.00
Had20	6922	6885	6848	6922	6930	3.85e+02	12.15
Kra30a	149936	136059	-1111	143576	169708	5.88e+03	149.32
Kra30b	91420	81156	-1111	87858	105740	4.36e+03	170.57
Kra32	88700	79659	-1111	85775	103790	3.57e+03	200.26
Nug12	578	557	530	568	632	2.60e+01	1.04
Nug14	1014	992	960	1011	1022	7.15e+01	1.87
Nug15	1150	1122	1071	1141	1306	9.10e+01	3.31
Nug16a	1610	1570	1528	1600	1610	1.81e+02	3.06
Nug16b	1240	1188	1139	1219	1356	9.35e+01	3.19
Nug17	1732	1669	1622	1708	1756	2.31e+02	4.34
Nug18	1930	1852	1802	1894	2160	4.16e+02	5.47
Nug20	2570	2451	2386	2507	2784	4.76e+02	11.56
Nug21	2438	2323	2386	2382	2706	1.41e+03	15.32
Nug22	3596	3440	3396	3529	3940	2.07e+03	21.82
Nug24	3488	3310	-1111	3402	3794	1.20e+03	29.64
Nug25	3744	3535	-1111	3626	4060	3.12e+03	39.23
Nug27	5234	4965	-1111	5130	5822	5.11e+03	78.18
Nug28	5166	4901	-1111	5026	5730	4.11e+03	83.38
Nug30	6124	5803	-1111	5950	6676	7.36e+03	133.38
Rou12	235528	223680	221161	235528	235528	2.76e+01	0.93
Rou15	354210	333287	323235	350217	367782	3.12e+01	2.70
Rou20	725522	663833	642856	695181	765390	1.67e+02	10.31
Scr12	31410	29321	23973	31410	38806	4.40e+00	1.17
Scr15	51140	48836	42204	51140	58304	1.38e+01	2.41
Scr20	110030	94998	83302	106803	138474	1.53e+03	9.61
Tai12a	224416	222784	215637	224416	224416	1.79e+00	0.90
Tai15a	388214	364761	349586	377101	412760	2.74e+01	2.35
Tai17a	491812	451317	441294	476525	546366	6.50e+01	4.52
Tai20a	703482	637300	619092	671675	750450	1.28e+02	10.10
Tai25a	1167256	1041337	-1111	1096657	1271696	3.09e+02	38.48
Tai30a	1818146	1652186	-1111	1706871	1942086	1.25e+03	142.55
Tho30	88900	77647	-1111	86838	102760	2.83e+03	164.86