

Geração de colunas com divisão em *clusters* para o problema de programação quadrática binária irrestrita

Geraldo Regis Mauri

Luiz Antonio Nogueira Lorena



Resumo

Este trabalho propõe uma nova alternativa de geração de colunas (GC), baseada na relaxação lagrangeana com divisão em clusters (LagClus), para resolução do Problema de Programação Quadrática Binária Irrestrita (PQ). O PQ é um dos problemas clássicos de otimização não-linear, cujo objetivo é resolver uma função quadrática por meio da escolha de valores binários apropriados para as variáveis de decisão. A GC proposta trata um modelo linear inteiro misto (PQL) do PQ, que tem restrições representadas por meio de um grafo e é dividido através de uma heurística de particionamento. Além de encontrar soluções viáveis, o método proposto ainda apresenta duas formas alternativas para obtenção de limitantes para o PQ. Foram realizados vários experimentos computacionais, utilizando-se instâncias de difícil solução com diferentes características. A GC é comparada a métodos tradicionais de relaxação lagrangeana e outros métodos propostos recentemente, sendo que os resultados apresentados são superiores para a maioria das instâncias consideradas.

Palavras-chave: Programação quadrática. Geração de colunas. Relaxação lagrangeana.

1 Introdução

O Problema de Programação Quadrática Binária Irrestrita (PQ) consiste em maximizar (ou minimizar) uma função objetivo quadrática através da escolha de valores apropriados para as variáveis de decisão binárias (BEASLEY, 1998). O PQ é um problema *NP-Hard* (BILLIONNET; ELLOUMI, 2007) clássico na área de otimização não-linear. Esse problema aborda aplicações em diversas áreas, como: biologia molecular (PHILLIPS; ROSEN, 1994); planejamento de investimentos e análise financeira (LAUGHUNN, 1970; McBRIDE; YORMARK, 1980), e alguns problemas do tipo CAD (KRARUP; PRUZAN, 1978). Além disso, o PQ ainda aborda inúmeros problemas modelados através de grafos, como clique máximo, máximo conjunto independente e outros (PARDALOS; PHILLIPS, 1990; PARDALOS; RODGERS, 1992; PARDALOS; XUE, 1994).

Os métodos exatos existentes para resolução do PQ (BILLIONNET; SUTTER, 1994; PARDALOS; RODGERS, 1990a; PARDALOS; RODGERS, 1990b) são restritos a problemas com até 200 variáveis. Já os métodos heurísticos (BEASLEY, 1998; GLOVER et al., 1998; PARDALOS;

JHA, 1992) têm apresentado bons resultados para instâncias com até 2500 variáveis.

Uma prática comum para resolver o PQ é a linearização do seu modelo original (ADAMS et al., 2004; GLOVER; WOOLSEY, 1974; HANSEN; MEYER, 2009), ou seja, a obtenção de um modelo linear equivalente cujas soluções são correspondentes ao modelo quadrático original. Dessa forma, o PQ é transformado em um problema linear inteiro misto, o que permite a relaxação linear de suas variáveis de decisão e, conseqüentemente, a obtenção de um limitante para a solução do problema original. Esse limitante é conhecido como *roof dual* (ADAMS; DEARING, 1994; BOROS et al., 1990; BOROS et al., 1992; HAMMER et al., 1984).

Além desses métodos, a relaxação do problema para obtenção de limitantes (ADAMS; DEARING, 1994; CHARDAIRE; SUTTER, 1995) tem apresentado bons resultados para o PQ. Essa abordagem possui a vantagem de definir limitantes para a solução ótima e pode apresentar uma informação de boa qualidade, o que permite avaliar

a proximidade da melhor solução encontrada em relação à solução ótima do problema. Uma estratégia usada para relaxar o problema original é a sua divisão em problemas menores, com as mesmas características. Essa divisão pode ser realizada através do particionamento do grafo, que representa o problema em *clusters* formados por vértices e arestas. Essa estratégia não garante a obtenção de uma solução viável para o problema completo, pois algumas arestas são ignoradas. Entretanto, uma maneira de considerar essas arestas é relaxá-las no sentido lagrangeano e encontrar um limitante de boa qualidade para o problema original. Essa é a ideia da relaxação lagrangeana baseada em *clusters* - *LagClus* (RIBEIRO, 2007; RIBEIRO; LORENA, 2007; RIBEIRO; LORENA, 2008).

Este trabalho propõe um método de geração de colunas baseado na relaxação lagrangeana do problema através de sua divisão em *clusters* (*LagClus*). Esse método trata um modelo linear inteiro misto (*PQL*) do *PQ* que tem restrições representadas através de um grafo, que é particionado através da heurística METIS (KARYPIS; KUMAR, 1998). É formado, então, um problema coordenador (problema mestre restrito) e um subproblema para cada *cluster*, que são utilizados para geração de novas colunas. Além de encontrar soluções viáveis, o método proposto ainda apresenta duas formas para obtenção de limitantes para o *PQ*.

O restante do artigo está assim organizado: na Seção 2, é apresentada uma breve revisão bibliográfica acerca do problema abordado. Na Seção 3, são apresentados os modelos matemáticos utilizados. Na Seção 4, descreve-se o método proposto e, na Seção 5, são apresentadas algumas formas já existentes para obtenção de limitantes para o *PQ*. Os resultados computacionais são apresentados na Seção 6 e as conclusões são resumidas na Seção 7.

2 Revisão bibliográfica

Uma abordagem trivial para obtenção de limitantes para o *Problema Quadrático da Mochila (PQM)* é apresentada por Billionnet e Calmels (1996). Essa abordagem é baseada na linearização do problema original e na resolução desse modelo linear por meio da relaxação linear de suas variáveis de decisão. Os Autores mostram que esses limitantes são de qualidade moderada. Além disso, eles utilizam a inserção de restrições de corte de *Chvatal-Gomory*, que apresentam melhores limitantes para o *PQM*.

Adams e Dearing (1994) apresentam uma discussão acerca da obtenção de limitantes para o *PQ*. Eles apresentam um modelo linear para o problema, que é obtido através da técnica de linearização de Glover e Woolsey (1974). Várias estratégias de linearização do *PQ* são apresentadas e discutidas em Adams et al. (2004) e Hansen e Meyer (2009).

Vários métodos baseados em busca em árvores para resolver o *PQ* são encontrados na literatura. Gulati et al.

(1984) apresentam um método de busca em árvore, baseado na enumeração de ótimos locais, que resolve problemas com até 40 variáveis. Pardalos e Rodgers (1990a) apresentam um método de busca em árvore que utiliza limitantes baseados na fixação de variáveis em cada nó da árvore. Os resultados tratam problemas com até 200 variáveis. Pardalos e Rodgers (1990b) apresentam uma versão paralelizada de um branch-and-bound capaz de resolver problemas com até 100 variáveis. Billionnet e Sutter (1994) apresentam um método de busca em árvore capaz de resolver problemas com até 100 variáveis. Palubeckis (1995) propõe uma busca em árvore heurística que apresenta resultados para problemas com até 247 variáveis.

Uma heurística baseada na Busca Tabu é proposta por Glover et al. (1998). Essa heurística resolve problemas com até 500 variáveis. Beasley (1998) apresenta uma comparação entre duas metaheurísticas para resolver o *PQ*. Ele utiliza a Busca Tabu e o *Simulated Annealing* para resolver problemas com até 2500 variáveis (com baixa densidade). A Busca Tabu apresenta os melhores resultados para a maioria dos problemas utilizados. Já o *Simulated Annealing* supera a Busca Tabu para os problemas com maior número de variáveis.

Pardalos e Jha (1992) discutem a complexidade computacional de vários problemas relacionados com o *PQ* e apresentam uma heurística de busca local para resolvê-los. Os resultados são apresentados para problemas com até 100 variáveis.

Glover et al. (2002) apresentam várias heurísticas baseadas no algoritmo DDT (BOROS et al., 1989) para resolução do *PQ*. Essas heurísticas são testadas através de várias instâncias com até 2500 variáveis. Billionnet e Elloumi (2007) aplicam técnicas de convexificação da função objetivo do *PQ* e as resolvem via CPLEX (ILOG, 2006). Os métodos propostos apresentam resultados para instâncias com até 200 variáveis.

A *LagClus*, apesar de recente, já apresenta bons resultados para vários problemas, como: carregamento de paletes (RIBEIRO; LORENA, 2007), rotulação cartográfica (RIBEIRO; LORENA, 2008), alocação de facilidades (CORRÊA et al., 2006), entre outros.

3 Formulações para o Programação Quadrática Binária Irrestrita

Seja $Q = [q_{ij}]_{m \times m}$ uma matriz de números reais. O *PQ* pode ser formulado pela Equação (1).

$$PQ : v(PQ) = \max_{x \text{ binário}} \sum_{i=1}^m \sum_{j=1}^m q_{ij} x_i x_j \quad (1)$$

Sem perder a generalidade do problema, a matriz Q pode ser considerada simétrica (BILLIONNET; ELLOUMI, 2007). Logo, de forma análoga à apresentada por Adams

e Dearing (1994), o *PQ* pode ser reescrito da seguinte forma:

$$PQ : v(PQ) = \max_{x \text{ binário}} \left(\sum_{i=1}^m q_{ii}x_i + \sum_{(i,j) \in P \cup N} 2q_{ij}x_ix_j \right) \quad (2)$$

em que $N = \{(i,j) : i < j, q_{ij} < 0\}$ e $P = \{(i,j) : i < j, q_{ij} > 0\}$. Pode-se, então, aplicar a técnica de linearização de Glover e Woolsey (1974) em (2), substituindo os termos quadráticos x_ix_j pela variável contínua w_{ij} e por restrições que garantam que $w_{ij} = x_ix_j$. Logo, tem-se uma versão linear inteira mista de *PQ* (3-8). Por convenção, esse modelo será chamado *PQL*.

$$PQL : v(PQL) = \text{Max} \sum_{i=1}^m q_{ii}x_i + \sum_{(i,j) \in P \cup N} 2q_{ij}w_{ij} \quad (3)$$

Sujeito a

$$w_{ij} - x_i \leq 0 \quad (i, j) \in P \quad (4)$$

$$w_{ij} - x_j \leq 0 \quad (i, j) \in P \quad (5)$$

$$x_i + x_j - w_{ij} \leq 1 \quad (i, j) \in N \quad (6)$$

$$w_{ij} \geq 0 \quad (i, j) \in N \quad (7)$$

$$x_i \in \{0,1\} \quad i = 1, \dots, m \quad (8)$$

Fica claro nos modelos (1), (2) e (3-8) que o número de elementos da matriz **Q** utilizados em (2) e (3-8) será significativamente menor do que em (1), pois nesses casos são considerados apenas os elementos não nulos situados na metade superior e na diagonal principal dessa matriz.

4 Geração de colunas com divisão em clusters

Dada a matriz **Q** descrita na seção anterior, pode-se criar um grafo $G = (V, E)$ com $V = \{1, \dots, m\}$ e uma matriz de adjacências $E = [e_{ij}]_{m \times m}$, $e_{ij} = 1$ se $q_{ij} \neq 0$ e $e_{ij} = 0$ se $q_{ij} = 0$. Particionando o grafo G em n ($n \leq m$) *clusters* independentes, tem-se $V = V_1 \cup V_2 \cup \dots \cup V_n$, em que $V_i \cap V_j = \emptyset, \forall i, j = 1, \dots, n, i \neq j, G_i = (V_i, E_i), i = 1, \dots, n$, e $X_i = V - V_i, i = 1, \dots, n$. Logo, o *PQL* pode ser reescrito da seguinte forma:

$$PQL^n : v(PQL^n) = \text{Max} \sum_{k=1}^n \left(\sum_{i \in V_k} q_{ii}x_i + \sum_{(i,j) \in P \cup N; i \in V_k; j \in V} 2q_{ij}w_{ij} \right) \quad (9)$$

Sujeito a

$$w_{ij} - x_i \leq 0 \quad (i, j) \in P, i \in V_k, j \in V, k = 1, \dots, n \quad (10)$$

$$w_{ij} - x_j \leq 0 \quad (i, j) \in P, i \in V_k, j \in V_k, k = 1, \dots, n \quad (11)$$

$$x_i + x_j - w_{ij} \leq 1 \quad (i, j) \in N, i \in V_k, j \in V_k, k = 1, \dots, n \quad (12)$$

$$w_{ij} \geq 0 \quad (i, j) \in N, i \in V_k, j \in V, k = 1, \dots, n \quad (13)$$

$$w_{ij} - x_j \leq 0 \quad (i, j) \in P, i \in V_k, j \in X_k, k = 1, \dots, n \quad (14)$$

$$x_i + x_j - w_{ij} \leq 1 \quad (i, j) \in N, i \in V_k, j \in X_k, k = 1, \dots, n \quad (15)$$

$$x_i \in \{0,1\} \quad i \in V_k, k = 1, \dots, n \quad (16)$$

As restrições (10) e (13) tratam as arestas (i,j) , cujos vértices i são internos e os vértices j internos ou externos, ao *cluster* (subgrafo) k . As restrições (11) e (12) tratam as arestas (i,j) , cujos vértices i e j são internos ao *cluster* k . Já as restrições (14) e (15) tratam apenas as arestas (i,j) , cujos vértices i e j estão em *clusters* distintos. A Figura 1 apresenta o exemplo de um grafo formado para um problema com seis variáveis ($m = 6$) e seu respectivo particionamento. Nesse exemplo, o grafo é particionado em dois *clusters* ($n = 2$) e duas arestas são “cortadas”, o que resulta na relaxação das restrições referentes a essas arestas. Um exemplo completo da aplicação da Geração de Colunas proposta neste artigo pode ser encontrado em Mauri (2008).

Com o intuito de facilitar a compreensão do método proposto (*GC*), esse modelo (9)-(16) pode ser representado matricialmente da seguinte forma:

$$PQL^n : v(PQL^n) = \text{Max} \sum_{k=1}^n (q_x^k x^k + q_w^k w^k)$$

$$\text{Sujeito a} \begin{bmatrix} A_1 & A_2 & \dots & A_n \\ B_1 & 0 & 0 & 0 \\ 0 & B_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & B_n \end{bmatrix} \times \begin{bmatrix} x^1 \\ w^1 \\ x^2 \\ w^2 \\ \vdots \\ x^n \\ w^n \end{bmatrix} \sim \begin{bmatrix} b^A \\ b^B \end{bmatrix}$$

em que: \mathbf{q}_x^k : é um vetor linha com os coeficientes das variáveis $x_i, i \in V_k$; \mathbf{q}_w^k : é um vetor linha com os coeficientes

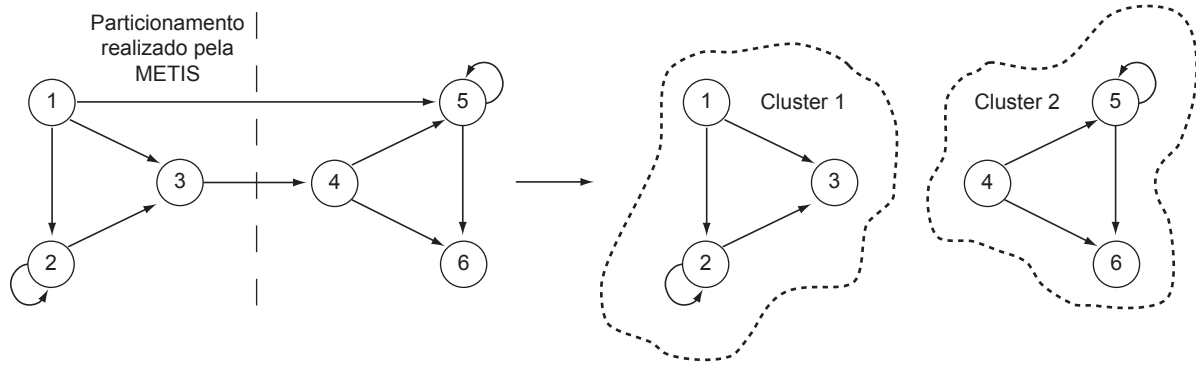


Figura 1. Exemplo de particionamento de grafo.

das variáveis w_{ij} , $i \in V_k$ e $j \in V$; \mathbf{B}_k : é uma matriz com os coeficientes das variáveis (pertencentes ao *cluster* k) presentes nas restrições (10)-(13); \mathbf{A}_k : é uma matriz com os coeficientes das variáveis (pertencentes ao *cluster* k) presentes restrições (14)-(15); \mathbf{x}^k : é um vetor coluna com os valores das variáveis de decisão x_i , $i \in V_k$; \mathbf{w}^k : é um vetor coluna com os valores das variáveis de decisão w_{ij} , $i \in V_k$ e $j \in V$; \sim : são os operadores relacionais \leq ou \geq , variando de acordo com as restrições; \mathbf{b}^A : vetor coluna com os valores (0 ou 1) do lado direito das restrições (14) e (15); e \mathbf{b}^B : vetor coluna com os valores (0 ou 1) do lado direito das restrições (10)-(13).

Relaxando as restrições (14) e (15) no sentido lagrangeano através do vetor de multiplicadores μ , ($\mu \geq 0$), o problema PQL^n (indiretamente o PQ) pode ser dividido em n subproblemas independentes. Considerando R_k como sendo as restrições associadas ao subproblema k , $k = 1, \dots, n$ e d o número de restrições (15) relaxadas, cada subproblema pode ser representado como descrito em (17), e a solução da relaxação do problema PQL em n subproblemas (*clusters*) é dada pela Equação (18).

$$LC_{\mu} PQL_k : v(LC_{\mu} PQL_k) = \max \left\{ \left(\begin{bmatrix} q_x^k & q_w^k \end{bmatrix} - \mu^T A_k \right) \begin{bmatrix} x^k \\ w^k \end{bmatrix} : x^k, w^k \text{ satisfazem } R_k \right\} \quad (17)$$

$$LC_{\mu} PQL^n : v(LC_{\mu} PQL^n) = \sum_{k=1}^n v(LC_{\mu} PQL_k) - \sum_{p=1}^d \mu_p \quad (18)$$

A implementação clássica da técnica de geração de colunas utiliza um problema coordenador – ou *Problema Mestre Restrito (PMR)* – e subproblemas geradores de colunas para formá-lo. O *PMR*, através de suas variáveis duais, direciona os subproblemas na busca de novas colunas. Assim, aplicando a decomposição *Dantzig-Wolfe* (BAZAARA et al., 1990) para a relaxação linear (*PL*) do problema PQL^n , tem-se (Equação 19, 20, 21, 22):

$$PMR_{PL} : v(PMR_{PL}) = \max \sum_{k=1}^n \sum_{s \in S_k} \lambda_{sk} (q_x^k x^{sk} + q_w^k w^{sk}) \quad (19)$$

Sujeito a

$$\sum_{k=1}^n \sum_{s \in S_k} \lambda_{sk} \begin{bmatrix} A_k \begin{bmatrix} x^{sk} \\ w^{sk} \end{bmatrix} \end{bmatrix} \sim \mathbf{b}^A \quad (20)$$

$$\sum_{s \in S_k} \lambda_{sk} = 1 \quad \forall k \in \{1, \dots, n\} \quad (21)$$

$$\lambda_{sk} \geq 0 \quad \forall k \in \{1, \dots, n\}, s \in S_k \quad (22)$$

O conjunto de pontos extremos de R_k associados com as colunas geradas no *PMR* é dado por S_k . β_k é a variável dual associada com a k -ésima restrição de convexidade de (21). \mathbf{x}^{sk} e \mathbf{w}^{sk} são vetores que definem os pontos extremos $s \in S_k$, ou seja, as soluções viáveis do subproblema definido pelo *cluster* k . λ_{sk} é a variável de decisão correspondente ao ponto extremo $s \in S_k$, ou seja, $\lambda_{sk} = 1$ se a coluna s (uma solução para o subproblema k) pertencer a solução do *PMR*, e $\lambda_{sk} = 0$, caso contrário. Para cada subproblema k (17), pode-se substituir o vetor de multiplicadores lagrangeanos μ pelo vetor de variáveis duais α associado com as restrições (20) e, de uma forma alternativa, cada subproblema pode ser descrito pela Equação (23) e a relaxação do PQL , em n *clusters*, por (24).

$$Z_k = \max \left\{ \left(\begin{bmatrix} q_x^k & q_w^k \end{bmatrix} - \alpha^T A_k \right) \begin{bmatrix} x^k \\ w^k \end{bmatrix} : x^k, w^k \text{ satisfazem } R_k \right\} \quad (23)$$

$$LC_{\alpha} PQL^n : v(LC_{\alpha} PQL^n) = \sum_{k=1}^n Z_k + \sum_{p=1}^d \alpha_p \quad (24)$$

A partir de então, uma nova coluna gerada pelo subproblema k é inserida no *PMR* se o seu custo reduzido

θ_k for positivo, isto é, $\theta_k = Z_k - \beta_k > 0$. Assim, o *PMR* coordena as soluções dos problemas por meio de suas variáveis duais, buscando uma solução para o problema original.

O *PMR* inicial é gerado através da heurística apresentada na Figura 2. Essa heurística utiliza a busca local apresentada em Beasley (1998). Em seguida, novas colunas são geradas até que um critério de parada seja satisfeito e o *PMR* final, formado por todas as colunas geradas, é resolvido de forma inteira ($\lambda_{sk} \in \{0,1\}$); conseqüentemente, sua solução será equivalente a uma solução viável para o *PQL* e, indiretamente, para o *PQ*. Por convenção, esse problema será tratado como *PMR_{PLI}* e o valor de sua solução como $v(\text{PMR}_{PLI})$. A Figura 3 apresenta os procedimentos para execução da *GC*.

- | | |
|-----|--|
| 1. | para $v = 1$ até m faça |
| 2. | $x_v \leftarrow 1$; |
| 3. | Aplicar uma busca local e encontrar a melhor solução vizinha com x_v fixo igual a 1; |
| 4. | para cada variável w_{ij} faça |
| 5. | se ($x_i = 1$ e $x_j = 1$) então |
| 6. | $w_{ij} \leftarrow 1$; |
| 7. | senão |
| 8. | $w_{ij} \leftarrow 0$; |
| 9. | fim-se; |
| 10. | fim-para; |
| 11. | Dividir as variáveis da solução encontrada nos respectivos clusters; |
| 12. | Criar uma coluna para cada cluster com os valores das variáveis da solução; |
| 13. | Inserir as colunas no <i>PMR</i> ; |
| 14. | fim-para. |

Figura 2. Formação do *PMR* inicial.

- | | |
|-----|---|
| 1. | Encontrar um conjunto inicial de colunas e formar o <i>PMR</i> (Figura 2); |
| 2. | Otimizar o <i>PMR</i> (via CPLEX) e obter o conjunto ótimo de variáveis duais; |
| 3. | Atualizar e resolver os subproblemas (23) via CPLEX; |
| 4. | se ($\forall k, \theta_k \leq 0$) ou $v(LC_\alpha PQL^n) = v(\text{PMR}_{PLI})$ então |
| 5. | Parar, pois a solução ótima para o <i>PMR</i> foi encontrada; |
| 6. | Resolver o <i>PMR</i> de forma inteira via CPLEX – obter $v(\text{PMR}_{PLI})$; |
| 7. | senão |
| 8. | Inserir toda coluna cujo $\theta_k > 0$ no <i>PMR</i> ; |
| 9. | Voltar ao passo 2; |
| 10. | fim-se; |

Figura 3. Execução do método proposto.

A solução do *PMR* de forma inteira – $v(\text{PMR}_{PLI})$ – apresenta uma solução viável para o *PQ*. Já a solução do *PMR* através de sua relaxação linear – $v(\text{PMR}_{PL})$ – e a solução da relaxação com divisão em clusters – $v(LC_\alpha PQL^n)$ – apresentam limitantes para o *PQ*.

5 Outros limitantes para o Programação Quadrática Binária Irrestrita

O valor da solução da relaxação linear do *PQL* (substituindo a restrição (8) por $0 \leq x_i \leq 1$) é um limitante trivial para o *PQ* (BILLIONNET; CALMELS, 1996). Como apresentado em Adams e Dearing (1994), esse limitante é conhecido como *roof dual*.

Outro limitante para o *PQ* pode ser obtido através da inserção de uma restrição de corte de *Chvatal-Gomory* em *PQL*. Essa restrição é descrita na Equação (25).

$$x_i + x_j + x_k - w_{ij} - w_{jk} - w_{ik} \leq \quad (25)$$

$$(i, j), (j, k), (i, k) \in P \cup N$$

Essa restrição de corte é baseada em uma das restrições de corte apresentadas em Billionnet e Calmels (1996) para o *Problema Quadrático da Mochila - PQM*. Como mostrado pelos Autores, os limitantes obtidos com a inserção de restrições de corte são melhores do que os apresentados pela relaxação linear do *PQM*. Por convenção, o modelo *PQL* com a restrição de corte será tratado como *PQLC*.

A relaxação lagrangeana tradicional das restrições (4), (5) e/ou (6) também pode ser utilizada para obtenção de outros limitantes para o *PQ*. Essas restrições podem ser combinadas de forma a gerar sete diferentes modelos para o problema (relaxando as restrições 4; 5; 6; 4 e 5; 4 e 6; 5 e 6; 4 e 5 e 6), sendo que a minimização de cada dual correspondente apresenta um limitante para o *PQ*. Todos esses sete modelos foram implementados e os melhores resultados obtidos entre eles são apresentados na coluna *Lag*, na Tabela 1. Mais detalhes acerca da relação lagrangeana são apresentados em Guignard (2003).

6 Experiência computacional

Vários experimentos computacionais trataram um conjunto de 45 instâncias disponíveis na *OR-Library* (BEASLEY, 1990). Essas instâncias foram criadas através do gerador proposto por Pardalos e Rodgers (1990a) e separadas em seis classes (A, B, C, D, E e F) com diferentes características (m , densidade e intervalo dos elementos da matriz Q). Além de fornecer os critérios para a criação das instâncias, os problemas das classes A, B e C foram apresentados neste mesmo estudo, e os demais em Glover et al. (1998). Segundo estes Autores, devido a suas características, essas instâncias estão entre as mais difíceis encontradas na literatura. Todos os experimentos

Tabela 1. Resultados obtidos pelos limitantes apresentados.

Inst	m	Dsde (%)	Roof dual		PQLC		Lag		GC		
			Gap	CPU	Gap	CPU	Gap	CPU	LC _α PQL ⁿ	PMR _{PL}	CPU
1a	50	10	0	0	0	0	0	0,22	0	0	0,17
2a	60	10	0	0	0	0	0	0,36	0	0	0,91
3a	70	10	6,77	0	6,36	0	3,37	24,75	0,31	0,31	9,37
4a	80	10	0,10	0,02	0,10	0,01	0,10	31,58	0,10	0,10	77,39
5a	50	20	16,12	0,02	11,59	0,02	4,17	48,08	0,36	0,36	20,72
6a	30	40	28,38	0	20,67	0,03	8,48	21,51	0,14	0,14	10,36
7a	30	50	28,51	0	10,58	0,05	7,69	31,19	0	0	20,28
8a	100	6,25	0	0	0	0	0	1,00	0	0	4,48
1b	20	100	193,98	0	95,99	0	194,66	0,22	38,72	38,72	0,06
2b	30	100	328,51	0,01	185,67	0,06	329,68	0,36	49,59	49,59	0,09
3b	40	100	429,24	0	252,82	0,08	429,66	0,74	52,54	52,54	0,13
4b	50	100	537,60	0,01	325,06	0,16	538,65	0,95	47,67	47,67	0,28
5b	60	100	532,67	0,02	321,78	0,39	534,36	1,03	39,00	39,00	0,67
6b	70	100	678,77	0,03	419,18	0,64	680,98	1,53	52,05	52,05	1,08
7b	80	100	687,50	0,05	425,00	1,06	690,11	2,01	25,94	25,94	1,59
8b	90	100	890,00	0,06	560,00	1,56	892,05	3,22	66,21	66,21	2,20
9b	100	100	1109,12	0,08	706,08	2,48	1109,76	4,97	74,82	74,82	3,31
10b	125	10	1175,97	0,12	750,65	6,12	1176,72	6,69	63,85	63,85	8,94
1c	40	80	65,57	0,02	21,31	1,08	17,27	2154,89	13,77	12,67	3600,08
2c	50	60	63,79	0,05	25,51	0,91	20,43	3604,84	10,60	9,43	3600,35
3c	60	40	44,04	0,03	22,27	1,17	14,24	3603,95	2,11	0	3602,95
4c	70	30	36,62	0,03	20,03	0,61	9,11	3614,12	0,48	0,48	3251,85
5c	80	20	21,64	0,05	16,35	0,25	6,90	694,34	0,35	0,35	1760,81
6c	90	10	1,48	0,01	0,30	0,03	0	389,25	0	0	50,22
7c	100	10	0	0,03	0	0,05	0	54,52	0	0	852,69
1d	100	10	11,53	0,02	8,27	0,08	1,73	206,73	0	0	2713,41
2d	100	20	86,91	0,06	64,31	0,67	44,96	3702,94	33,69	18,58	3611,64
3d	100	30	94,94	0,14	58,08	4,67	57,69	3812,78	70,83	6,25	3619,18
4d	100	40	134,52	0,28	73,17	4,73	92,15	3848,99	142,92	16,50	3729,77
5d	100	50	164,34	0,53	86,05	16,55	112,79	3639,78	192,79	24,86	3647,80
6d	100	60	162,79	0,64	80,52	50,44	113,68	3613,28	218,45	23,96	3831,31
7d	100	70	205,14	0,64	110,48	66,06	162,54	3738,97	284,08	50,10	3706,08
8d	100	80	207,24	1,42	105,85	106,72	151,40	5382,86	288,40	49,13	3627,82
9d	100	90	252,13	1,74	134,76	128,42	194,06	3616,89	350,46	66,32	3739,25
10d	100	100	234,16	2,16	132,04	276,67	183,00	5044,47	338,49	74,88	3611,41
1e	200	10	44,27	0,30	35,26	1,33	28,26	3796,38	41,20	1,90	3605,04
2e	200	20	119,48	1,49	83,86	26,58	94,4	4445,22	186,67	27,94	3613,85
3e	200	30	188,73	4,06	110,69	49,75	144,66	3604,11	304,70	66,69	3608,29
4e	200	40	186,88	2,95	98,50	438,45	187,58	5878,97	343,44	72,78	3601,36
5e	200	50	263,42	4,17	148,84	951,31	228,72	4109,58	451,90	122,26	3630,80
1f	500	10	159,28	29,55	126,12	189,61	124,66	5306,53	359,56	0,38	3840,75
2f	500	25	297,20	235,80	164,58	3606,83	297,34	5411,75	577,28	0,22	3793,00
3f	500	50	474,68	1078,94	-	-	19,33	3715,66	1292,80	0,23	5827,56
4f	500	75	587,96	2857,45	-	-	11,08	4335,98	-	-	-
5f	500	100	500,32	3605,06	-	-	9,41	3917,95	-	-	-
Tempo médio:			173,96		141,32		2120,58		1959,05		

foram realizados em um PC com processador AMD Athlon de 2.2 GHz com 1GB de memória RAM e o código-fonte foi implementado em C++.

Para resolver a relaxação linear do problema (*roof dual*), a relaxação linear com a restrição de corte (*PQLC*) e o modelo *PQL*, foi utilizado o CPLEX 10.0.1 (ILOG, 2006), com o tempo limite de uma hora de processamento para cada instância.

O dual lagrangeano das relaxações lagrangeanas tradicionais, apresentadas na seção anterior, foi otimizado através do algoritmo de subgradiente apresentado em Narciso e Lorena (1999), que aproxima as soluções do problema no sentido euclidiano de distância (PARKER; RARDIN, 1988). O CPLEX (ILOG, 2006) também foi utilizado para resolver os problemas de forma exata a cada iteração do algoritmo de subgradiente. O tempo limite para execução desse algoritmo também foi de uma hora.

Na geração de colunas proposta, o *PMR* e os subproblemas também foram resolvidos através do CPLEX. Para a divisão do grafo *G*, foi utilizada a heurística METIS (KARYPIS; KUMAR, 1998), que, segundo Hicks et al. (2005), apresenta bons resultados para o particionamento de grafos. Essa heurística divide o grafo minimizando o número de arestas com terminações em clusters distintos.

$$gap = \frac{v(\text{Solução}) - v(OPT)}{v(OPT)} \times 100 \quad (26)$$

Na Tabela 1, os *gaps* apresentados são calculados de acordo com a Equação (26), na qual $v(OPT)$ é o valor das melhores soluções conhecidas (GLOVER et al., 1998) para as instâncias utilizadas e $v(\text{Solução})$ é o valor dos limitantes obtidos pelas abordagens apresentadas. As colunas CPU apresentam o tempo de processamento, em segundos. A coluna *Dsde* apresenta a densidade da matriz *Q* de cada instância e a coluna *Lag* apresenta os melhores *gaps* obtidos pelas relaxações lagrangeanas tradicionais (ver Seção 5). Os limitantes ($LC_{\alpha}PQL^n$ e PMR_{PLI}) obtidos pela geração de colunas proposta (*GC*) apresentam os mesmos valores para as instâncias menores (A, B e C), o que indica a convergência completa do método. Diversamente, para as maiores, essa convergência não acontece no tempo limite de uma hora de processamento. Entretanto, os limitantes obtidos apresentam excelentes *gaps* para todas as instâncias. Os melhores resultados estão destacados em negrito.

A última linha da Tabela 1 apresenta os tempos médios necessários para obtenção dos limitantes para cada instância através do método correspondente. O número de *clusters* (*n*) utilizado para as instâncias das classes A, B, C e D foi igual a 2 e, para as instâncias das classes E e F, foram 3 e 5, respectivamente.

A Tabela 2 apresenta as soluções inteiras obtidas para as instâncias das classes B e F. Nessa tabela, a coluna *Perc* apresenta o percentual da solução obtida em relação à melhor solução conhecida ($(v(\text{Solução})/v(OPT))*100$). Os

resultados obtidos pela *GC* são comparados aos resultados apresentados pelas diferentes heurísticas apresentadas em Glover et al. (2002) e com o CPLEX (coluna CPX). Como pode ser observado, em relação às heurísticas de Glover et al. (2002), a *GC* apresentou os melhores resultados para todas as instâncias da classe B e para as instâncias da classe F com densidade de até 50%. Já o CPLEX superou a *GC* em apenas três das dez instâncias da classe B, porém foi expressivamente pior para as primeiras instâncias da classe F. Para as demais, a *GC* não foi capaz de encontrar uma solução em um tempo aproximado de uma hora.

O método descrito em Billionnet e Elloumi (2007) não foi capaz de resolver as instâncias da classe F e, segundo os Autores, todas as soluções ótimas foram encontradas para as instâncias da classe B com um limite de tempo de duas horas de processamento, em um PC com processador Intel Pentium IV de 1.6 GHz com 1GB de memória RAM. Glover et al. (2002) não apresentam os tempos computacionais utilizados pelas heurísticas propostas. A *GC* proposta apresentou um tempo médio inferior a um segundo para resolver o PMR_{PLI} para as instâncias da classe B e 2483,23 segundos, para a classe F. Logo, considerando os tempos para formar esse *PMR* (Tabela 1), a *GC* apresentou um tempo total médio de 0,18 e 6970,33 segundos para as instâncias das classes B e F, respectivamente.

A Tabela 3 apresenta as soluções obtidas para as instâncias da classe D. Os *gaps* obtidos pela *GC* proposta são comparados com os apresentados por Billionnet e Elloumi (2007). O tempo médio de execução para resolver o *PMR* de forma inteira foi de 10,30 segundos, além do tempo para formá-lo (Tabela 1), e o método apresentado pelos Autores utilizou um tempo médio de dez minutos por instância. O tempo médio utilizado pelo CPLEX foi de uma hora de processamento. Como apresentado na Tabela 2, os resultados apresentados pela *GC* proposta foram expressivamente melhores. Glover et al. (2002) não apresentam resultados para essas instâncias. A coluna *Gap* apresenta o *gap* referente à solução inteira do *PMR*. Esses *gaps* são calculados de acordo com a Equação (27).

$$gap = \frac{v(OPT) - v(PMR_{PLI})}{v(OPT)} \times 100 \quad (27)$$

A Tabela 4 apresenta as soluções obtidas para as instâncias das classes A, C e E. De acordo com Billionnet e Elloumi (2007), seu método obteve as soluções ótimas para as instâncias das classes A e C em um tempo limite de duas horas por instância; entretanto, não foi capaz de resolver os problemas da classe E. O CPLEX foi capaz de encontrar as soluções ótimas para as instâncias das classes A e C em um tempo médio de 3,28 segundos e o tempo médio para obter as soluções para as instâncias da classe E foi de uma hora. A *GC* proposta utilizou um tempo médio de 17,96, 2426,56 e 119,98 segundos para resolver o PMR_{PLI} para

Tabela 2. Resultados obtidos para as instâncias das classes B e F.

Inst.	m	Dsde (%)	Melhores soluções	Glover et al. (2002) – Perc (%)					CPX Perc	GC		
				DDT	A2n	A2t	V3n	V3t		n	v(PMR _{PLP})	Perc
1b	20	100	133	73,7	100,0	78,9	10,5	10,5	100	2	133	100,0
2b	30	100	121	95,0	75,2	86,8	5,0	5,0	100	2	121	100,0
3b	40	100	118	47,5	86,4	80,5	0,0	0,0	100	2	118	100,0
4b	50	100	129	66,7	78,3	78,3	0,0	0,0	100	2	129	100,0
5b	60	100	150	70,0	100,0	60,0	0,7	0,7	100	2	150	100,0
6b	70	100	146	43,2	77,4	72,6	2,7	2,7	100	2	133	91,1
7b	80	100	160	56,3	100,0	100,0	0,6	0,6	100	2	160	100,0
8b	90	100	145	61,4	80,7	80,7	0,0	0,0	100	2	145	100,0
9b	100	100	137	70,1	92,7	75,9	0,0	0,0	100	2	135	98,5
10b	125	10	154	65,6	78,6	78,6	0,0	0,0	100	2	147	95,5
1f	500	10	61194	99,2	77,9	76,9	92,9	92,8	57,5	5	60959	99,6
2f	500	25	100161	99,3	80,4	80,8	92,7	90,4	16,0	5	99836	99,7
3f	500	50	138035	98,7	78,6	79,7	92,7	92,1	23,6	5	137716	99,8
4f	500	75	172771	98,7	82,0	83,9	94,4	94,4	15,5	-	-	-
5f	500	100	190507	98,9	87,8	77,5	95,4	94,2	10,5	-	-	-

Tabela 3. Resultados obtidos para as instâncias da classe D.

Inst.	m	Dsde (%)	Melhores soluções	Billionnet e Eloumi (2007) Gap (%)	CPX Gap	GC		
						n	v(PMR _{PLP})	Gap (%)
1d	100	10	6333	4,1	0	2	6333	0
2d	100	20	6579	10,0	5,21	2	6579	0
3d	100	30	9261	7,6	3,54	2	9261	0
4d	100	40	10727	8,1	6,25	2	10727	0
5d	100	50	11626	8,7	17,82	2	11604	0,19
6d	100	60	14207	7,2	9,90	2	14157	0,35
7d	100	70	14476	8,3	14,02	2	14476	0
8d	100	80	16352	6,1	17,48	2	16352	0
9d	100	90	15656	8,7	14,65	2	15647	0,06
10d	100	100	19102	6,9	11,99	2	19102	0

as instâncias das classes A, C e E, respectivamente. Logo, considerando os tempos para formar esse *PMR* (Tabela 1), a *GC* apresentou um tempo total médio de 17,96, 2426,56 e 6970,33 segundos para as instâncias das classes A, C e E, respectivamente. Glover et al. (2002) não apresentam resultados para essas instâncias.

Como pode ser observado nas tabelas anteriores, o método proposto se destaca na resolução dos problemas de maior porte (a partir de 100 vértices), principalmente aqueles cuja matriz *Q* possui baixa densidade (até 50%), apresentando resultados melhores que outros métodos encontrados na literatura.

7 Conclusões

Este trabalho apresentou uma nova estratégia de geração de colunas para resolver o problema de programação

quadrática binária irrestrita. O método proposto, além de encontrar soluções viáveis, também apresenta duas alternativas para obtenção de limitantes para o *PQ*. Instâncias de difícil solução e com diferentes características foram utilizadas para avaliar o método proposto.

A *GC* tratou o *PQ* por meio de um modelo linear com restrições representadas através de um grafo, cuja divisão é realizada por meio da heurística de particionamento METIS (KARYPIS; KUMAR, 1998). O *PMR* e os subproblemas foram resolvidos através do CPLEX 10.0.1 (ILOG, 2006).

Foram obtidos excelentes resultados para as instâncias com até 200 variáveis e várias densidades, e também para instâncias com 500 variáveis e baixas e médias densidades. Diversamente, para as instâncias com 500 variáveis e alta densidade, a *GC* não apresentou soluções, pois o CPLEX

Tabela 4. Resultados obtidos para as instâncias das classes A, C e E.

Inst.	m	Dsde (%)	Melhor conhecida	CPX		GC			
				LQP	Gap	PMR_{PL}	$LC_{\alpha} PQL^n$	PMR_{PLI}	Gap
1a	50	10	3414	3414	0	3414,00	3414,00	3414	0
2a	60	10	6063	6063	0	6063,00	6063,00	6063	0
3a	70	10	6037	6037	0	6056,00	6056,00	6035	0,03
4a	80	10	8598	8598	0	8606,50	8606,50	8582	0,19
5a	50	20	5737	5737	0	5757,50	5757,50	5737	0
6a	30	40	3980	3980	0	3985,50	3985,50	3980	0
7a	30	50	4541	4541	0	4541,00	4541,00	4541	0
8a	100	6,25	11109	11109	0	11109,00	11109,00	11109	0
1c	40	80	5058	5058	0	5699,30	5754,72	5058	0
2c	50	60	6213	6213	0	6799,08	6871,31	6213	0
3c	60	40	6665	6665	0	6665,00	6805,93	6665	0
4c	70	30	7398	7398	0	7433,76	7433,76	7398	0
5c	80	20	7362	7362	0	7387,88	7387,88	7362	0
6c	90	10	5824	5824	0	5824,00	5824,00	5824	0
7c	100	10	7225	7225	0	7225,00	7225,00	7225	0
1e	200	10	16464	15724	4,49	16776,20	23247,60	16447	0,10
2e	200	20	23395	21464	8,25	29931,63	67067,54	23376	0,08
3e	200	30	25243	21836	13,50	42078,45	102159,19	25228	0,06
4e	200	40	35594	31848	10,52	61499,90	157839,61	35594	0
5e	200	50	35154	27820	20,86	78134,28	194014,96	35149	0,01

não foi capaz de resolver os subproblemas dentro de um tempo aceitável.

O método proposto foi comparado diretamente com métodos propostos recentemente e com o CPLEX, e apresentou excelentes resultados para praticamente todas as instâncias. A GC proposta contribui com a exploração

do PQ e apresenta uma nova alternativa para resolução de problemas representados por grafos. Além disso, acredita-se que uma técnica mais eficiente para resolução dos subproblemas geradores de colunas pode resultar em melhores soluções para as instâncias tratadas neste trabalho e, possivelmente, para instâncias ainda maiores.

Column generation with clusters for the unconstrained binary quadratic programming problem

Abstract

This paper proposes a new alternative of column generation (GC) based on the lagrangean relaxation with clusters (LagClus) to solve the Unconstrained Binary Quadratic Programming Problem (PQ). The PQ is a classical non-linear problem of optimizing a quadratic function by suitable choices of binary decisions variables. The proposed GC treats a mixed binary linear model (PQL) of PQ with constraints represented by a graph and divided through a partitioning heuristic. Besides finding feasible solutions the proposed method still presents two alternative ways to find bounds for PQ. Several computational experiments were performed using hard instances with different features. GC is compared to traditional lagrangean relaxation and other methods recently proposed presenting improved results for most of these instances.

Keywords: Quadratic programming. Column generation. Lagrangean relaxation.

Referências bibliográficas

- ADAMS, W. P.; DEARING, P. M. On the equivalence between roof duality and lagrangean duality for unconstrained 0-1 quadratic programming problems. **Discrete Applied Mathematics**, v. 48, n. 1, p. 1-20, 1994.
- ADAMS, W. P.; FORRESTER, R. J.; GLOVER, F. W. Comparisons and enhancement strategies for linearizing mixed 0-1 quadratic programs. **Discrete Optimization**, v. 1, p. 99-120, 2004.
- BAZAARA, M. S.; JARVIS, J. J.; SHERALI, H. D. Linear programming and network flows. New York: John Wiley & Sons, 1990.
- BEASLEY, J. E. **Heuristic algorithms for the unconstrained binary quadratic programming problem**. London, UK: Management School, Imperial College, 1998. Technical Report.
- BEASLEY, J. E. Or-library: Distributing test problems by electronic mail. **Journal of the Operational Research Society**, v. 41, n. 11, p. 1069-1072, 1990.
- BILLIONNET, A.; CALMELS, F. Linear programming for the 0-1 quadratic knapsack problem. **European Journal of Operational Research**, v. 92, n. 2, p. 310-325, 1996.
- BILLIONNET, A.; ELLOUMI, S. Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem. **Mathematical Programming**, v. 109, p. 55-68, 2007.
- BILLIONNET, A.; SUTTER, A. Minimization of a quadratic pseudo-boolean function. **European Journal of Operational Research**, v. 78, p. 106-115, 1994.
- BOROS, E.; CRAMA, Y.; HAMMER, P. L. Chvatal cuts and odd cycle inequalities in quadratic 0-1 optimization. **SIAM - Journal on Discrete Mathematics**, v. 5, n. 2, p. 163-177, 1992.
- BOROS, E.; CRAMA, Y.; HAMMER, P. L. Upper bounds for quadratic 0-1 maximization. **Operations Research Letters**, v. 9, p. 73-79, 1990.
- BOROS, E.; HAMMER, P.; SUN, X. **The DDT method for quadratic 0-1 minimization**. New Brunswick, USA: RUTCOR Research Center, 1989. RRR 39-89.
- CHARDAIRE, P.; SUTTER, A. A decomposition method for quadratic zero-one programming. **Management Science**, v. 41, n. 4, p. 704-712, 1995.
- CORRÊA, F. A.; LORENA, L. A. N.; SENNE, E. L. F. Lagrangean relaxation with clusters for the uncapacitated facility location problem. In: Congresso Latino-Iberoamericano de Investigación Operativa, 13, 2006, Montevideo - Uruguay. **Anais...**
- GLOVER, F. et al. One-pass heuristics for large-scale unconstrained binary quadratic problems. **European Journal of Operational Research**, v. 137, n. 2, p. 272-287, 2002.
- GLOVER, F.; KOCHENBERGER, G.; ALIDAEI, B. Adaptive memory tabu search for binary quadratic programs. **Management Science**, v. 44, n. 3, p. 336-345, 1998.
- GLOVER, F.; WOOLSEY, E. Converting a 0-1 polynomial programming problem to a 0-1 linear program. **Operations Research**, v. 22, p. 180-182, 1974.
- GUIGNARD, M. Lagrangean relaxation. **TOP**, v. 11, n. 2, p. 151-200, 2003.
- GULATI, V. P.; GUPTA, S. K.; MITTAL, A. K. Unconstrained quadratic bivalent programming problem. **European Journal of Operational Research**, v. 15, p. 121-155, 1984.
- HAMMER, P. L.; HANSEN, P.; SIMEONE, B. Roof duality, complementation and persistency in quadratic 0-1 optimization. **Mathematical Programming**, v. 28, p. 121-195, 1984.
- HANSEN, P.; MEYER, C. Improved compact linearizations for the unconstrained quadratic 0-1 minimization problem. **Discrete Applied Mathematics**, v. 157, n. 6, p. 1267-1290, 2009.
- HICKS, I. V. et al. A branch-and-price approach for the maximum weight independent set problem. **Networks**, v. 46, n. 4, p. 198-209, 2005.
- ILOG. **ILOG CPLEX 10.0: user's manual**. France, 2006. 478 p.
- KARYPIS, G.; KUMAR, V. Multilevel k-way partitioning scheme for irregular graphs. **Journal of Parallel and Distributed Computing**, v. 48, p. 96-129, 1998.
- KRARUP, J.; PRUZAN, P. A. Computer aided layout design. **Mathematical Programming Study**, v. 9, p. 75-94, 1978.
- LAUGHUNN, D. J. Quadratic binary programming with application to capital budgeting problems. **Operations Research**, v. 18, p. 454-461, 1970.
- MAURI, G. R. **Novas abordagens para representação e obtenção de limitantes e soluções para alguns problemas de otimização combinatória**. São José dos Campos - SP, 2008. Tese (Doutorado em Computação Aplicada) - Instituto Nacional de Pesquisas Espaciais.
- MCBRIDE, R. D.; YORMARK, J. S. An implicit enumeration algorithm for quadratic integer programming. **Management Science**, v. 26, p. 282-296, 1980.
- NARCISO, M. G.; LORENA, L. A. N. Lagrangean/surrogate relaxation for generalized assignment problems. **European Journal of Operational Research**, v. 114, p. 165-177, 1999.
- PALUBECKIS, G. A heuristic-based branch and bound algorithm for unconstrained quadratic 0-1 programming. **Computing**, v. 54, n. 4, p. 283-301, 1995.
- PARDALOS, P. M.; JHA, S. Complexity of uniqueness and local search in quadratic 0-1 programming. **Operations Research Letters**, v. 11, p. 119-123, 1992.
- PARDALOS, P. M.; PHILLIPS, A. T. A global optimization approach for solving the maximum clique problem. **International Journal of Computer Mathematics**, v. 33, p. 209-216, 1990.
- PARDALOS, P. M.; RODGERS, G. P. A branch and bound algorithm for the maximum clique problem. **Computers & Operations Research**, v. 19, p. 363-375, 1992.
- PARDALOS, P. M.; RODGERS, G. P. Computational aspect of a branch and bound algorithm for quadratic 0-1 programming. **Computing**, v. 45, p. 131-144, 1990a.
- PARDALOS, P. M.; RODGERS, G. P. Parallel branch and bound algorithms for quadratic zero-one programs on the hypercube architecture. **Annals of Operations Research**, v. 22, p. 271-292, 1990b.
- PARDALOS, P. M.; XUE, J. The maximum clique problem. **Journal of Global Optimization**, v. 4, p. 301-328, 1994.
- PARKER, R. G.; RARDIN, R. L. **Discrete Optimization**. New York, USA: Academic Press, 1988.
- PHILLIPS, A. T.; ROSEN, J. B. A quadratic assignment formulation of the molecular conformation problem. **Journal of Global Optimization**, v. 4, p. 229-241, 1994.
- RIBEIRO, G. M. Relaxação lagrangeana com divisão em clusters para alguns problemas de otimização modelados em grafos de conflitos. São José dos Campos - SP, 2007. Tese (Doutorado em Computação Aplicada) - Instituto Nacional de Pesquisas Espaciais.
- RIBEIRO, G. M.; LORENA, L. A. N. Lagrangean relaxation with clusters and column generation for the manufacturer's pallet loading problem. **Computers & Operations Research**, v. 34, n. 9, p. 2695-2708, 2007.
- RIBEIRO, G. M.; LORENA, L. A. N. Lagrangean relaxation with clusters for point-feature cartographic label placement problems. **Computers & Operations Research**, v. 35, p. 2129-2140, 2008.

Sobre os autores

Geraldo Regis Mauri

Centro de Ciências Agrárias, Departamento de Engenharia Rural, Universidade Federal do Espírito Santo – UFES
CEP 29500-000, Alegre, ES, Brasil
e-mail: mauri@cca.ufes.br

Luiz Antonio Nogueira Lorena

Laboratório Associado de Computação e Matemática Aplicada, Instituto Nacional de Pesquisas Espaciais – INPE
CEP 12227-010, São José dos Campos, SP, Brasil,
e-mail: lorena@lac.inpe.br

Agradecimentos: Os autores agradecem aos dois revisores anônimos pelos valiosos comentários e sugestões, e à Fundação de Amparo à Pesquisa do Estado de São Paulo – FAPESP (processo 04/11053-9) e ao Conselho Nacional de Pesquisas – CNPq (processo 304598/2003-8) pelo apoio financeiro parcial dado ao desenvolvimento deste trabalho.

Recebido em 28/4/2008
Aceito em 5/10/2009