

# MÉTODOS DE GERAÇÃO DE COLUNAS PARA PROBLEMAS DE ATRIBUIÇÃO

**Edson Luiz França Senne**

UNESP – Campus de Guaratinguetá – Faculdade de Engenharia

E-mail: elfsenne@feg.unesp.br

**Luiz Antonio Nogueira Lorena**

INPE – Instituto Nacional de Pesquisas Espaciais – Laboratório Associado de Computação e

Matemática Aplicada

E-mail: lorena@lac.inpe.br

**Silvely Nogueira de Almeida Salomão**

UNESP – Campus de Presidente Prudente – Faculdade de Ciências e Tecnologia

E-mail: silvelysalomao@prudente.unesp.br

## **Resumo**

Este trabalho apresenta métodos de geração de colunas para dois importantes problemas de atribuição: o Problema Generalizado de Atribuição (PGA) e o Problema de Atribuição de Antenas a Comutadores (PAAC). O PGA é um dos mais representativos problemas de Otimização Combinatória e consiste em otimizar a atribuição de  $n$  tarefas a  $m$  agentes, de forma que cada tarefa é atribuída a exatamente um agente e a capacidade de cada agente seja respeitada. O PAAC consiste em atribuir  $n$  antenas a  $m$  comutadores em uma rede de telefonia celular, de forma a minimizar os custos de cabeamento entre antenas e comutadores e os custos de transferências de chamadas entre comutadores. A abordagem tradicional de geração de colunas é comparada com as propostas neste trabalho, que utilizam a relaxação lagrangeana/*surrogate*. São apresentados testes computacionais que demonstram a efetividade dos algoritmos propostos.

## **Palavras-chave**

Otimização Combinatória, Problemas de Atribuição, Relaxação Lagrangeana/*Surrogate*, Geração de Colunas.

## COLUMN GENERATION METHODS FOR ASSIGNMENT PROBLEMS

### Abstract

This work presents column generation methods for two important assignment problems: the Generalized Assignment Problem (GAP) and the problem of assigning cells to switches in cellular mobile networks (PACS). GAP is one of the most representative combinatorial optimisation problems and can be stated as the problem of optimising the assignment of  $n$  jobs to  $m$  agents, such that each job is assigned to exactly one agent and the resource capacity of each agent is not violated. PACS consists of determining a cell assignment pattern which minimizes cabling costs between a cell and a switch and transfer costs between cells assigned to different switches, while respecting certain constraints, especially those related to limited switch's capacity. The traditional column generation process is compared with the proposed algorithms that combine the column generation and lagrangean/surrogate relaxation. Computational experiments are presented in order to confirm the effectiveness of the proposed algorithms.

### Key words

Combinatorial Optimization, Assignment Problems, Langrangean/Surrogate Relaxation, Column Generation

## INTRODUÇÃO

O problema de atribuição é um problema clássico de Otimização Combinatória. De uma forma geral, o problema pode ser estabelecido como: um conjunto de  $n$  tarefas deve ser atribuído a um conjunto de  $m$  agentes. É necessário que todas as tarefas sejam atribuídas e que cada tarefa seja atribuída a apenas um agente, de tal forma que o custo total da atribuição seja minimizado. Este problema tem grande importância prática, estando presente, por exemplo, na distribuição de tarefas entre processadores de um sistema de computação paralela (Bokhari, 1987), no planejamento de tarefas de um telescópio espacial (Nowakovski *et al.*, 1999), na distribuição de pacientes em viagens de assistência médica (Ruland, 1999), na atribuição de frequências em redes de comunicação celular (Fischetti *et al.*, 2000; Wang e Gu, 2004), dentre inúmeras outras situações. Além disto, existem muitos problemas de decisão

que, não sendo diretamente um problema de atribuição, contém um problema de atribuição como subproblema.

Existem diversos tipos de problemas de atribuição. Em sua forma mais simples, o problema não envolve restrições de capacidade. Neste caso, quando  $m = n$ , ou seja, quando o número de agentes é igual ao número de tarefas, o problema é conhecido como Problema Linear de Atribuição (PLA). Neste caso, o problema é um caso especial de outro problema de otimização, conhecido como Problema de Transporte. Devido à sua estrutura especial, o PLA pode ser resolvido eficientemente (em tempo polinomial) por diversos métodos bem conhecidos (Kuhn, 1955; Hung e Rom, 1980; Wright, 1990).

Outro tipo de problema de atribuição é conhecido como Problema Quadrático de Atribuição (PQA). O problema ocorre quando, para designar objetos a locais, deve-se levar em conta as distâncias entre os pares de locais e os fluxos de algum tipo de demanda entre os pares de objetos. O PQA consiste em encontrar uma alocação de custo mínimo dos objetos aos locais, sendo os custos obtidos por produtos distância  $\times$  fluxo. O primeiro trabalho sobre este tipo de problema de atribuição foi publicado por Koopmans e Beckmann (1957) e, desde então, o problema tem aparecido em diversas situações práticas, como em projetos de circuitos eletrônicos, em problemas de escalonamento de horário, em análise estatística, em planejamentos de hospitais, dentre muitas outras. Uma extensa revisão sobre o PQA pode ser vista em Loiola *et al.* (2004).

Outro tipo de problema de atribuição ocorre quando elementos de um número variável de conjuntos devem ser atribuídos mutuamente. Este problema é conhecido como Problema de Atribuição Multidimensional (PAM) (Pierskalla, 1968; Grundel *et al.*, 2005). Diversas situações relatadas na literatura envolvem o PAM, dentre as quais, a determinação de trajetórias de partículas elementares (Pusztaszeri *et al.*, 1996) e a determinação de células fibroblásticas a partir de seqüências de imagens biomédicas (Kirubarajan *et al.*, 2001).

Outras variações conhecidas para o problema de atribuição são: o Problema de Atribuição Dinâmica, onde a atribuição de agentes a tarefas deve ser feita de forma dinâmica sobre períodos de tempo, o que é muito comum em problemas de roteamento e escalonamento; o Problema de Atribuição Aleatória, que corresponde em encontrar em uma matriz  $m \times n$  de variáveis aleatórias,  $k$  elementos tais que sua soma seja mínima; o Problema de Atribuição Axial, que consiste em encontrar um clique (subgrafo completo) de peso mínimo para um grafo tripartido completo. Uma revisão sobre os principais problemas de atribuição pode ser encontrada em Burkard (2002).

Neste trabalho, são apresentados métodos de resolução para duas formas do problema de atribuição, para as quais existem restrições de capacidade e o número de agentes é menor do que o número de tarefas ( $m < n$ ): o Problema Generalizado de Atribuição (PGA), um problema clássico e já bastante estudado, e o Problema de Atribuição de Antenas a Comutadores (PAAC), um problema mais recente e que ganhou importância com o surgimento das redes de telefonia celular. Estes dois tipos de problemas de atribuição fazem parte da classe de problemas combinatórios NP-difíceis (Garey e Johnson, 1979).

Uma das primeiras aplicações do PGA se deve a Balachandran (1976), que estudou o problema de atribuir tarefas em uma rede de computadores. Desde então, muitos outros problemas têm sido apresentados na literatura como aplicações do PGA, dentre as quais podem ser citadas: a alocação de salas de aula (Luan e Yao, 1996), o roteamento de veículos (Baker e Sheasby, 1999), o carregamento de caminhões (Pigatti, 2003) e a recuperação de blocos de dados a partir de discos paralelos (Aerts *et al.*, 2003).

Mesmo o PGA se apresenta de várias formas. Uma forma bem conhecida é o Problema Generalizado de Atribuição Multinível (Laguna *et al.* 1995; French e Wilson, 2002; Osorio e Laguna, 2003) onde, na atribuição de tarefas a agentes, leva-se em conta que os agentes podem executar as tarefas em mais de um nível de eficiência. Outra forma é o Problema Generalizado de Atribuição Estocástico (Albareda-Sambola *et al.*, 2002). Neste caso, a natureza estocástica do problema pode ser devida à falta de informação a priori sobre a quantidade de recursos necessários para os agentes realizarem as tarefas ou à incerteza sobre a presença ou a ausência de tarefas individuais, ou seja, quando existe um conjunto potencial de tarefas, mas somente algumas delas serão realmente realizadas (o que acontece, por exemplo, em serviços de emergência).

Uma revisão geral das técnicas de solução propostas para o PGA pode ser vista em (Cattrysse e Van Wassenhove, 1992). Com relação aos métodos aproximados, em geral, as abordagens propõem heurísticas de duas fases, em que em primeiro lugar, as atribuições são calculadas de modo a satisfazer um determinado critério (por exemplo, minimizar uma função de penalização, como em Martello e Toth, 1981) e, numa segunda fase, as atribuições são melhoradas. Amini e Racer (1994) também propõem uma abordagem heurística em duas fases para o problema e fazem uma cuidadosa comparação de vários métodos. Metaheurísticas também já foram propostas para o PGA, dentre as quais podem ser citados os algoritmos genéticos (Wilson, 1997; Chu e Beasley, 1997), a busca tabu (Laguna *et al.*, 1995) e *simulated annealing* (Osman, 1995). Narciso (1998) aplicou a relaxação lagrangeana/*surrogate* combinada com otimização por subgradientes para a solução do PGA.

Nauss (2003) propõe a utilização de cortes, relaxação lagrangeana e otimização por subgradientes. Outras abordagens são propostas por Yagiura *et al.* (2004a, 2004b).

O outro problema de atribuição abordado neste trabalho é o PAAC. Este problema difere do PGA pela necessidade de considerar novos custos devidos às transferências de chamadas entre comutadores, o que torna o problema mais difícil.

Nos trabalhos de Merchant e Sengupta (1995) e Saha *et al.* (2000), o PAAC é formulado como um problema de Programação Inteira e resolvido por meio de heurísticas gulosas de atribuição das antenas aos comutadores. Abordagens que utilizam metaheurísticas também já foram propostas para o PAAC. Podem ser citados os trabalhos de Pierre e Houéto (2002), que usaram a busca tabu, de Quintero e Pierre (2002), que empregaram *simulated annealing* (SA) e algoritmos genéticos, e de Menon e Gupta (2004), que combinaram SA com um método de geração de colunas. Uma comparação de abordagens para o PAAC pode ser encontrada em Quintero e Pierre (2003).

Neste trabalho são apresentados métodos de geração de colunas para a solução aproximada do PGA e do PAAC. Os métodos apresentados diferem das abordagens tradicionais por utilizarem a relaxação lagrangeana/*surrogate*, o que resulta em algoritmos mais eficientes.

## FORMULAÇÕES E BASES TEÓRICAS

Considerando  $c_{ij}$  como o custo associado de atribuir a tarefa  $j$  ao agente  $i$ ,  $r_{ij}$  como a quantidade de recursos necessários ao agente  $i$  para realizar a tarefa  $j$ ,  $b_i$  como a capacidade total de recursos do agente  $i$ , e  $x_{ij}$  como uma variável de decisão binária, tal que  $x_{ij} = 1$ , se a tarefa  $j$  está atribuída ao agente  $i$ , e  $x_{ij} = 0$ , caso contrário, o Problema Generalizado de Atribuição pode ser formulado como:

$$\text{Min} \quad \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (1)$$

sujeito a:

$$\sum_{i=1}^m x_{ij} = 1 \quad \forall j \in N = \{1, \dots, n\} \quad (2)$$

$$\sum_{j=1}^n r_{ij} x_{ij} \leq b_i \quad \forall i \in M = \{1, \dots, m\} \quad (3)$$

$$x_{ij} \in \{0,1\} \quad \forall i \in M; \forall j \in N \quad (4)$$

A restrição (2) elimina a possibilidade da tarefa  $j$  ser atribuída a mais de um agente e a restrição (3) garante que a capacidade do agente  $i$  será respeitada. Esta formulação foi usada por Cattrysse *et al.* (1994) para desenvolver um algoritmo heurístico para o PGA.

Para a formulação do PAAC outras considerações são necessárias. Geralmente, uma área geográfica atendida por serviços de comunicação móvel é dividida em unidades geográficas denominadas células. Para efeito de estudo, as células são representadas por hexágonos, conforme mostra a Figura 1.

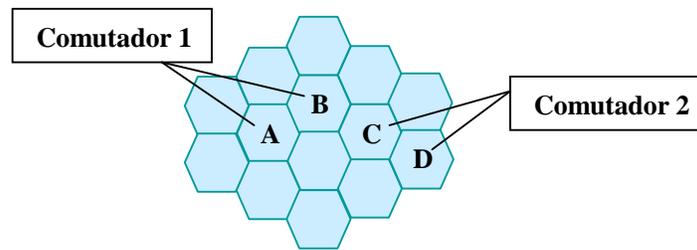


Figura 1 – Representação de Células em uma Rede de Comunicação

Cada célula possui uma antena de transmissão e recepção, também conhecida como estação rádio base, responsável pela comunicação entre o equipamento celular móvel e um comutador, que tem a função de encaminhar o tráfego de chamadas entre antenas. Durante uma chamada, conforme aumenta a distância entre a unidade móvel (celular) e a estação rádio base, o sinal torna-se fraco e aumentam os ruídos provenientes de interferências da vizinhança. Para evitar isto, é preciso fazer uma transferência da chamada de uma antena para outra adjacente que esteja mais próxima do usuário. Esta transferência deve ser realizada para que o assinante não perca a qualidade do sinal enquanto estiver realizando uma chamada. Quando o assinante se move entre antenas atendidas por um mesmo comutador como, por exemplo, quando sai da célula A e entra na célula B da Figura 1, a transferência é simples e não acarreta custos consideráveis. No entanto, quando o assinante se move entre antenas atendidas por comutadores diferentes como, por exemplo, quando sai da célula B e entra na

célula C da Figura 1, tem-se o denominado *rooming* e a transferência acarreta um custo, denominado custo de *handoff*, que precisa ser considerado.

Assim, considerando  $M = \{1, \dots, m\}$  como o conjunto de índices de comutadores,  $N = \{1, \dots, n\}$  como o conjunto de índices de antenas,  $c_{ij}$  como o custo de cabeamento entre o comutador  $i$  e a antena  $j$  ( $i \in M, j \in N$ ),  $h_{jk}$  como o custo de *handoff* entre as células  $j$  e  $k$  ( $j, k \in N$ ),  $v_j$  como o volume de chamadas da célula  $j$  ( $j \in N$ ),  $b_i$  como a capacidade de atendimento de chamadas do comutador  $i$  ( $i \in M$ ), e  $x_{ij}$  como uma variável de decisão binária, tal que  $x_{ij} = 1$  se a antena  $j$  está atribuída ao comutador  $i$  e  $x_{ij} = 0$ , caso contrário ( $i \in M, j \in N$ ), o PAAC pode ser formulado como:

$$\text{Min } \sum_{i \in M} \sum_{j \in N} c_{ij} x_{ij} + \sum_{j \in N} \sum_{k \in N} [1 - \sum_{i \in M} x_{ij} x_{ik}] h_{jk} \quad (5)$$

sujeito a:

$$\sum_{i \in M} x_{ij} = 1 \quad \forall j \in N \quad (6)$$

$$\sum_{j \in N} v_j x_{ij} \leq b_i \quad \forall i \in M \quad (7)$$

$$x_{ij} \in \{0,1\} \quad \forall i \in M, \forall j \in N \quad (8)$$

A função-objetivo (5) busca minimizar os custos de cabeamento e de *handoff*. Observe que os custos de *handoff* só devem ser considerados para antenas atribuídas a comutadores diferentes. As restrições de atribuição (6) evitam que uma antena seja atribuída a mais de um comutador e as restrições de capacidade (7) impossibilitam que as capacidades de atendimento dos comutadores sejam violadas.

Os métodos de geração de colunas propostos neste trabalho baseiam-se em tópicos abordados por Lorena *et al.* (2003), dentre os quais: a reformulação dos problemas de atribuição como problemas de partição de conjuntos, a relaxação lagrangena/*surrogate* e a decomposição de Dantzig-Wolfe. Um resumo destes tópicos é apresentado a seguir.

### O Problema de Particionamento de Conjuntos

O problema de determinar uma partição de um conjunto de modo a satisfazer algumas restrições e otimizar uma função linear é conhecido como Problema de Particionamento de Conjuntos (PPC). Sejam:  $M = \{1, \dots, m\}$ , um conjunto finito,  $P = \{P_1, \dots, P_n\}$ , o conjunto de todos os subconjuntos de  $M$ , e  $N = \{1, \dots, n\}$ , o conjunto de índices de  $P$ . Um subconjunto  $F$

de  $N$  ( $F \subseteq N$ ) é uma partição de  $M$  se  $\bigcup_{j \in F} P_j = M$  e  $P_j \cap P_k = \emptyset$ ,  $\forall j, k \in F$ . Se o subconjunto

$P_j$  ( $j \in N$ ) é representado pelo vetor  $z^j = (z_{1j}, \dots, z_{mj})^t$ , tal que  $z_{ij} = 1$  se o elemento  $i \in M$  está no conjunto  $P_j$ , e  $z_{ij} = 0$ , caso contrário, o PPC pode ser formulado como:

$$\text{Min } \sum_{j \in N} c(z^j) \quad (9)$$

sujeito a:

$$\sum_{j \in N} z_{ij} = 1 \quad \forall i \in M \quad (10)$$

$$z^j \in P \quad (11)$$

onde  $c(z^j)$  é o custo associado ao subconjunto  $P_j$ .

Em geral, os subconjuntos  $P_j$  factíveis de um problema são tais que:

$P_j = \{z^j / A_j z^j \leq b_j\}$  ( $j \in N$ ), ou seja, para cada subconjunto existe um conjunto de restrições a serem satisfeitas, como no caso dos problemas de atribuição considerados neste trabalho.

### **A Relaxação Lagrangeana/Surrogate**

A relaxação lagrangeana/surrogate consiste em aplicar a relaxação lagrangeana sobre restrições já relaxadas pela relaxação surrogate. Por exemplo, a relaxação surrogate das restrições (2) do problema (1)-(4), conduz ao seguinte problema:

$$\text{Min } \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (12)$$

sujeito a:

$$\sum_{j=1}^n \lambda_j \sum_{i=1}^m x_{ij} = \sum_{j=1}^n \lambda_j \quad (13)$$

$$\sum_{j=1}^n r_{ij} x_{ij} \leq b_i \quad \forall i \in M = \{1, \dots, m\} \quad (14)$$

$$x_{ij} \in \{0,1\} \quad \forall i \in M; \forall j \in N \quad (15)$$

onde  $\lambda = (\lambda_1, \dots, \lambda_n)$  é o vetor formado pelos multiplicadores *surrogate*, que transformam as restrições (2) numa única restrição. Seja t o multiplicador lagrangeano aplicado sobre esta restrição. Assim, a relaxação lagrangeana/*surrogate* do problema (12)-(15) pode ser escrita como:

$$\text{Min} \quad \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + t \sum_{j=1}^n (1 - \sum_{i=1}^m x_{ij}) \lambda_j \quad (16)$$

sujeito a:

$$\sum_{j=1}^n r_{ij} x_{ij} \leq b_i \quad \forall i \in M = \{1, \dots, m\} \quad (17)$$

$$x_{ij} \in \{0,1\} \quad \forall i \in M; \forall j \in N \quad (18)$$

Se  $t = 1$ , a relaxação lagrangeana/*surrogate* é idêntica à relaxação lagrangeana tradicional. No entanto, um valor conveniente para t, que pode ser obtido por uma heurística de busca como a proposta por Senne e Lorena (2000), pode melhorar o limitante produzido pela relaxação.

### A Decomposição de Dantzig-Wolfe

Seja um problema de Programação Linear inteiro da forma:

$$\text{Min} \quad c x \quad (19)$$

sujeito a:

$$A x = b \quad (20)$$

$$A' x \leq b' \quad (21)$$

$$x \text{ inteiro} \quad (22)$$

onde: A (uma matriz  $m \times n$ ), A' (uma matriz  $m' \times n$ ), c (um vetor  $n \times 1$ ), b (um vetor  $m \times 1$ ) e b' (um vetor  $m' \times 1$ ) são conhecidos, e a variável de decisão x é um vetor  $n \times 1$ . A decomposição de Dantzig-Wolfe (Dantzig e Wolfe, 1960) para problemas deste tipo é recomendada quando as restrições (20) e (21) podem ser trabalhadas separadamente. Neste caso, tais restrições são chamadas de restrições gerais e restrições especiais, respectivamente. As restrições especiais, em geral, têm uma interpretação natural (restrições da mochila, por exemplo).

Definindo  $S = \{ x \text{ inteiro} / A'x \leq b' \}$ , seja  $\{ x^k / k \in K \}$  o conjunto dos pontos extremos de  $S$ , onde  $K$  é o conjunto de índices destes pontos. Se  $S$  é limitado, então  $x \in S$  pode ser representado como uma combinação linear convexa de pontos extremos de  $S$ , ou seja:

$$x = \sum_{k \in K} \alpha_k x^k \quad (23)$$

onde:

$$\sum_{k \in K} \alpha_k = 1 \text{ e } \alpha_k \geq 0, k \in K \quad (24)$$

Com isto, o problema original (19)-(22) pode ser escrito como:

$$\text{Min } c \sum_{k \in K} \alpha_k x^k \quad (25)$$

sujeito a:

$$A \left( \sum_{k \in K} \alpha_k x^k \right) = b \quad (26)$$

$$\sum_{k \in K} \alpha_k = 1 \quad (27)$$

$$\alpha_k \geq 0, k \in K \quad (28)$$

Na decomposição de Dantzig-Wolfe, o problema de encontrar soluções viáveis (extremos) de  $S$  é chamado de subproblema e o problema (25)-(28) é denominado de problema-mestre.

### O Método de Geração de Colunas

Em geral, o número de pontos extremos  $x^k$  é grande e, por razões práticas, em um algoritmo opta-se por trabalhar apenas com um número restrito de extremos. Seja  $K^i \subseteq K$ , o conjunto de índices de pontos extremos de  $S$  gerados até a iteração  $i$  do algoritmo. O problema-mestre (25)-(28) quando se considera a substituição de  $K$  por  $K^i$ , ou seja, quando se considera apenas um subconjunto dos extremos de  $S$ , passa a ser denominado problema-mestre restrito. Neste caso, o subproblema pode ser chamado de subproblema de geração de colunas, pois as variáveis obtidas pela resolução deste subproblema constituirão novas colunas para o problema-mestre restrito. A fim de gerar variáveis que melhorem a função-objetivo do problema-mestre restrito, os índices destas variáveis são determinados pelo cálculo dos custos reduzidos:

$$z_k - \hat{c}_k = \text{Min}_{1 \leq j \leq s} (c - \lambda A) x^j - \mu \quad (29)$$

onde  $\lambda$  e  $\mu$  são as variáveis duais do problema-mestre restrito referentes às restrições (26) e (27), respectivamente, e  $x^j$  ( $j = 1, \dots, s$ ) os extremos de  $S$ . Desta forma, o subproblema torna-se um problema de custo reduzido, também conhecido como problema *pricing*.

Assim, se  $z_j - \hat{c}_j \geq 0, \forall j \in \{1, \dots, s\}$ , então a solução é ótima. Se existir algum índice

$\hat{k}$  tal que  $z_{\hat{k}} - \hat{c}_{\hat{k}} < 0$ , então a coluna  $\begin{bmatrix} Ax^{\hat{k}} \\ 1 \end{bmatrix}$  é selecionada para entrar no problema-mestre

restrito e, desta forma, o conjunto de índices pode ser atualizado como:  $K^{i+1} = K^i \cup \{\hat{k}\}$ .

Como a casca convexa de  $S$  é poliédrica e limitada, e  $x^j, j = 1, \dots, s$  são os extremos de  $S$ , então a solução do problema linear-inteiro será um dos extremos de  $S$ , ou seja, a solução ótima de (29) é equivalente à solução ótima do problema:

$$\begin{aligned} \text{Min } (c - \lambda A)x - \mu \\ x \in S \end{aligned} \quad (30)$$

Assim, o algoritmo de geração de colunas pode ser estabelecido como:

1. Seja  $i = 0$ . Determine um conjunto inicial  $K^0$  de índices de colunas para o problema-mestre restrito;
2. Resolva a relaxação de Programação Linear do problema-mestre restrito (PMR);
3. Utilize os valores duais ótimos obtidos no passo anterior e resolva o subproblema para obter novas colunas  $x^{\hat{k}}$  em  $S$ ;
4. Adicione as colunas  $x^{\hat{k}}$  com custos reduzidos negativos à formulação do problema-mestre restrito. Se não houver tais colunas, então pare, pois a solução é ótima;
5. Caso contrário, faça  $K^{i+1} = K^i \cup \{\hat{k}\}$ ,  $i = i + 1$  e volte para o passo 2.

Uma vez evitadas soluções cíclicas no problema-mestre restrito e a degeneração no subproblema, o algoritmo converge em um número finito de iterações, pois consiste do algoritmo Simplex Revisado com a inclusão dos custos reduzidos dados pela resolução do subproblema. O subproblema não precisa ser completamente otimizado em cada iteração. É necessário apenas que as colunas  $x^k$  geradas tenham custos reduzidos negativos para que seus respectivos  $\lambda_k$  sejam candidatos a entrar na base.

## OS MÉTODOS DE GERAÇÃO DE COLUNAS PROPOSTOS

O método de geração de colunas proposto para o PGA baseia-se na reformulação deste

problema como um PPC. Para isto, seja  $K_i = \{x_i^1, \dots, x_i^{k_i}\}$  o conjunto de padrões de atribuição viáveis para o agente  $i$ , onde  $x_i^k = \{x_{ij}^k, j=1, \dots, n\}$  é uma solução viável para as restrições (2) e (4). Considerando  $y_i^k$ ,  $i \in M$  e  $k \in K_i$ , como uma variável binária indicando se o padrão de atribuição  $x_i^k$  está sendo utilizado, ou seja,  $y_i^k = 1$ , se o padrão  $x_i^k$  faz parte das atribuições do agente  $i$ , e  $y_i^k = 0$ , caso contrário, o PGA pode ser reformulado como:

$$\text{Min} \quad \sum_{i=1}^m \sum_{k=1}^{k_i} \left( \sum_{j=1}^n c_{ij} x_{ij}^k \right) y_i^k \quad (31)$$

sujeito a:

$$\sum_{i=1}^m \sum_{k=1}^{k_i} x_{ij}^k y_i^k = 1, \quad \forall j \in N \quad (32)$$

$$\sum_{k=1}^{k_i} y_i^k \leq 1, \quad \forall i \in M \quad (33)$$

$$y_i^k \in \{0,1\}, \quad \forall i \in M \quad (34)$$

A formulação (31)-(34) define o problema-mestre a ser considerado, no sentido da decomposição de Dantzig-Wolfe. Nesta formulação, para determinar as atribuições  $x_i^k$  possíveis a cada agente  $i$  é necessário resolver os seguintes  $m$  Problemas da Mochila, para  $i = 1, \dots, m$ :

$$\text{Min} \quad \sum_{j=1}^n c_{ij} x_{ij} \quad (35)$$

sujeito a:

$$\sum_{j=1}^n r_{ij} x_{ij} \leq b_i \quad (36)$$

$$x_{ij} \in \{0,1\}, \quad \forall j \in N \quad (37)$$

Por outro lado, considerando  $S = \{ x^i = (x_{ij}, j=1, \dots, n) / x_{ij} \in \{0,1\} \text{ e } \sum_{j=1}^n r_{ij} x_{ij} \leq b_i, i \in M \}$

e  $t = t^*$ , onde  $t^*$  é o multiplicador que resolve o problema dual:

$$\max_{t \geq 0} \quad \min_{x \in S} \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + t \sum_{j=1}^n \lambda_j \left( 1 - \sum_{i=1}^m x_{ij} \right) \quad (38)$$

a relaxação lagrangeana/surrogate (16)-(18) pode ser escrita como:

$$\text{Min}_{x \in S} \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + t^* \sum_{j=1}^n \lambda_j \left( 1 - \sum_{i=1}^m x_{ij} \right) \quad (40)$$

e pode ser resolvido separadamente por  $m$  problemas da forma:

$$\text{Min}_{x \in S_i} \sum_{j=1}^n [c_{ij} x_{ij} + t^* \lambda_j (1 - x_{ij})] \quad (41)$$

onde  $S_i = \{x^i = (x_{i1}, \dots, x_{in}) / x_{ij} \in \{0,1\} \text{ e } \sum_{j=1}^n r_{ij} x_{ij} \leq b_i\}$ .

Assim, após resolver a relaxação de Programação Linear do problema-mestre (31)-(34), obtida substituindo-se a restrição (34) por  $y_i^k \in [0,1]$ , a busca por novas colunas pode ser realizada resolvendo-se o problema de custo reduzido:

$$\text{Max}_{i \in M} (v(M_i) - \mu_i) \quad (42)$$

onde  $\mu_i$  é o custo dual ótimo associado à restrição (33) e  $v(M_i)$  é o valor da solução ótima do seguinte Problema da Mochila:

$$\text{Min} \sum_{j=1}^n (c_{ij} - t^* \lambda_j) x_j^i \quad (43)$$

sujeito a:

$$\sum_{j=1}^n r_{ij} x_j^i \leq b_i \quad (44)$$

$$x_j^i \in \{0,1\}, j \in N \quad (45)$$

onde  $\lambda_j$  é o custo dual ótimo associado às restrições (32).

Desta forma, o método de geração de colunas para o PGA pode ser estabelecido pelo seguinte algoritmo:

1. Determine um conjunto inicial de colunas para o problema (31)-(34);
2. Resolva a relaxação de Programação Linear do problema (31)-(34) considerando o conjunto atual de colunas, obtendo as variáveis duais  $\lambda_j$  ( $j \in N$ ), referentes às restrições (32) e  $\mu_i$  ( $i \in M$ ), referentes às restrições (33);
3. Utilize as variáveis duais  $\lambda_j$  ( $j \in N$ ) para determinar o multiplicador  $t^*$  que resolve o problema (38);

4. Utilize as variáveis duais  $\lambda_j$  ( $j \in N$ ) e o multiplicador  $t^*$  para resolver os subproblemas (43)-(45) para cada  $i \in M$ . Sejam  $x^{\hat{k}} \in S$  as colunas referentes às soluções destes subproblemas;
5. Adicione as colunas  $x^{\hat{k}}$  com custos reduzidos negativos ao problema-mestre restrito (31)-(34);
6. Se, no passo anterior, nenhuma nova coluna foi acrescentada ao problema-mestre restrito, então pare. Caso contrário, volte para o passo 2.

Este método pode ser usado também para resolver, de forma aproximada, o PAAC, pois este problema tem estrutura semelhante à do PGA, exceto pelos custos de *handoff* na função-objetivo. Assim, para o PAAC pode-se considerar a seguinte formulação:

$$\text{Min } \sum_{s \in S} c_s x_s \quad (46)$$

sujeito a:

$$\sum_{s \in S} a_{js} x_s = 1 \quad \forall j \in N \quad (47)$$

$$\sum_{s \in S} b_{is} x_s \leq 1 \quad \forall i \in M \quad (48)$$

$$x_s \in \{0,1\} \quad \forall s \in S \quad (49)$$

onde  $S$  é o conjunto de colunas que satisfazem (7)-(8), onde cada coluna  $s \in S$  é da forma:  $(a_{1s}, \dots, a_{ns}, b_{1s}, \dots, b_{ms})^t$ , tal que  $a_{js} \in \{0,1\}$  ( $j \in N$ ) e  $\exists k \in M$  tal que  $b_{ks} = 1$  e  $b_{is} = 0, \forall i \in M, i \neq k$ , ou seja,  $\{a_{1s}, \dots, a_{ns}\}$  é um conjunto de índices de células atribuídas a um comutador  $k$ , e  $c_s$  é o custo da coluna  $s$ , dado por:

$$c_s = \sum_{i=1}^n (c_{ki} + \sum_{j=1}^n h_{ij}) a_{is} - \sum_{i=1}^n \sum_{j=1}^n h_{ij} a_{is} a_{js} \quad (50)$$

Neste caso, a relaxação lagrangeana/*surrogate* do problema leva aos seguintes  $m$  subproblemas geradores de colunas, para  $k = 1, \dots, m$ :

$$\text{Min } \sum_{i=1}^n [(c_{ki} + \sum_{j=1}^n h_{ij}) - t \lambda_i] x_{ki} - \sum_{i=1}^n \sum_{j=1}^n h_{ij} x_{ki} x_{kj} \quad (51)$$

sujeito a:

$$\sum_{j=1}^n v_j x_{kj} \leq b_k \quad (52)$$

$$x_{kj} \in \{0,1\} \quad \forall j \in N \quad (53)$$

onde:  $\lambda_j$  são as variáveis duais associadas às restrições (47) da relaxação de Programação Linear do problema-mestre restrito (46)-(49).

Assim, o método de geração de colunas para o PAACC pode ser estabelecido pelo seguinte algoritmo:

1. Determine um conjunto inicial de colunas para o problema (46)-(49);
2. Resolva a relaxação de Programação Linear do problema (46)-(49) considerando o conjunto atual de colunas, obtendo as variáveis duais  $\lambda_j$  ( $j \in N$ ), referentes às restrições (47) e  $\mu_i$  ( $i \in M$ ), referentes às restrições (48);
3. Estipule um valor apropriado para o multiplicador  $t$ ;
4. Utilize as variáveis duais  $\lambda_j$  ( $j \in N$ ) e o valor do multiplicador  $t$  e resolva os subproblemas (51)-(53) para cada  $k \in M$ . Seja  $S'$  o conjunto de colunas referentes às soluções destes subproblemas;
5. Para cada coluna  $s \in S'$ , determine o custo reduzido da coluna por:  
$$cr_s = c_s - \sum_{i=1}^n \pi_i a_{is} - \mu_k$$
, onde  $c_s$  é o custo da coluna  $s$ , conforme estabelecido em (50), e acrescente as colunas  $s$  tais que  $cr_s < 0$  ao problema-mestre restrito (46)-(49).
6. Se, no passo anterior, nenhuma nova coluna foi acrescentada ao problema-mestre restrito, então pare. Caso contrário, volte para o passo 2.

## RESULTADOS COMPUTACIONAIS

Os algoritmos propostos foram implementados na linguagem C e executados em um microcomputador Pentium III com 1.1 GHz e 512 MB de RAM, utilizando a versão 7.5 do software CPLEX para a solução dos problemas de Programação Linear.

Para o PGA, utilizou-se o algoritmo de Horowitz-Sahni (Martello e Toth, 1990) para a solução dos problemas da mochila. Neste caso, o conjunto inicial de colunas do PMR é constituído por  $n$  colunas artificiais e por um conjunto de colunas referentes a uma primeira solução inteira encontrada para o problema (1)-(4) pelo software CPLEX, ou seja, colunas que representam uma solução factível para o problema original. Cada coluna artificial corresponde a um vetor nulo, exceto por um elemento unitário que correspondente à restrição de associação de uma das tarefas. O custo de cada coluna artificial é estabelecido como o custo da atribuição (agente, tarefa) mais cara.

Nos testes computacionais, considerou-se como 5000 o número máximo de colunas para o PMR e foi implementado um procedimento para remover colunas do problema sempre que este limite para o número de colunas for excedido. Este procedimento remove as colunas cujos custos reduzidos são maiores do que o custo reduzido médio, consideradas todas as colunas do problema. Para os testes realizados, não se buscou o melhor valor para o multiplicador  $t$ . Em vez disto, a cada iteração do algoritmo, foram considerados os seguintes valores para  $t$ : 0.50, 0.60, 0.70, 0.80, 0.85, 0.90, 0.93, 0.95, 0.97 e 1.00. É interessante notar que para cada valor de  $t$  tem-se um novo conjunto de  $m$  problemas da mochila e, conseqüentemente, um novo conjunto de  $m$  possíveis colunas candidatas a entrarem no PMR.

Para os experimentos computacionais foram consideradas as seguintes classes de problemas para as quais as soluções ótimas são conhecidas: classes A, B, C, D e E, disponíveis na OR-Library (Beasley, 2004), e classe Y, em Yagiura (2003). Cada classe contém 6 problemas. Para as classes da OR-Library,  $m \in \{5, 10, 20\}$  e  $n \in \{100, 200\}$ . Para a classe Y,  $m \in \{20, 40\}$  e  $n \in \{200, 400\}$ . A Tabela 1 mostra os resultados médios obtidos para cada classe com o algoritmo proposto (que utiliza a relaxação lagrangeana/*surrogate*) e, entre colchetes, os resultados médios obtidos pelo método tradicional de geração de colunas (que utiliza a relaxação lagrangeana e que, portanto, pode ser obtido a partir do algoritmo proposto fixando-se  $t = 1$ ).

Tabela 1 – Resultados Computacionais para o PGA

Classe	Iter	Cols-a	Cols-r	Gap	Cpu	S/L
A	621 [1132]	4099 [4146]	15037 [6742]	0.000 [0.000]	1368.51 [2144.33]	0.59
B	324 [596]	4459 [4153]	26536 [8286]	0.096 [0.097]	687.73 [1147.86]	0.60
C	328 [590]	4148 [3924]	27838 [8225]	0.079 [0.077]	630.26 [1219.86]	0.58
D	293 [353]	4340 [3822]	14749 [7578]	0.115 [0.114]	857.91 [721.55]	0.93
E	396 [609]	3855 [4157]	22512 [7792]	0.040 [0.040]	829.62 [1132.64]	0.80
Y	250 [357]	4320 [4059]	74087 [21067]	0.006 [0.006]	4467.63 [4931.00]	0.86

Nesta tabela:

- **Iter** é o número de iterações do algoritmo;
- **Cols-a** é o número de colunas aproveitadas;
- **Cols-r** é o número de colunas removidas do PMR pelo procedimento de remoção de colunas;

- **Gap** é calculado como:  $\text{Gap} = 100 * | \text{Soluc} - \text{LCpx} | / \text{Soluc}$ , onde **Soluc** é o valor da solução ótima do problema e **LCplx** é o limite obtido para o PMR;
- **Cpu** é o tempo computacional do algoritmo proposto, em segundos;
- **S/L** é a relação entre o tempo de execução do algoritmo proposto e o tempo de execução do algoritmo tradicional de geração de colunas.

Pelos resultados mostrados na Tabela 1, nota-se que, mesmo sem buscar  $t^*$  (melhor valor para o multiplicador  $t$ ) e considerando 10 valores pré-estabelecidos para  $t$ , o que leva a ter que resolver 10 problemas da mochila a cada iteração e, conseqüentemente, à geração de um número bem maior de colunas, o algoritmo proposto supera em velocidade o método tradicional de geração de colunas. Deve-se observar que este ganho em velocidade não se dá às custas da qualidade da solução obtida, medida pelo valor de **Gap**.

No caso do algoritmo proposto para o PAAC foram consideradas como colunas iniciais do PMR as colunas relativas à primeira solução inteira obtida pelo software CPLEX para o problema sem os custos de *handoff* (que é denominada de solução de referência) e outras colunas geradas aleatoriamente até completar 1000 colunas para o PMR. A dificuldade no método proposto para o PAAC está no fato dos  $m$  subproblemas serem problemas da mochila com função-objetivo não-linear. As seguintes heurísticas foram consideradas para a solução dos subproblemas, a cada iteração do algoritmo:

- Heurística **Sub**: consiste na solução pelo CPLEX de cada um dos  $m$  subproblemas separadamente, levando-se em conta as equações de linearização consideradas por Merchant e Sengupta (1995). Deve-se observar que, como os subproblemas são resolvidos separadamente, ao final pode haver antenas atribuídas a mais de um comutador e mesmo antenas não atribuídas a comutador algum.
- Heurística **LSH**: consiste em resolver, separadamente, cada subproblema  $k$  por meio do seguinte algoritmo:
  1. Ordenar as células em ordem crescente dos valores  $(c_{ki} - t \pi_i)$ , onde  $t$  é o multiplicador lagrangeano/*surrogate*;
  2. Alocar as células ao comutador  $k$ , segundo a ordenação, enquanto isso for possível, considerando a capacidade  $b_k$  do comutador;
  3. Para as células não atribuídas, tentar a atribuição de menor custo ainda possível, respeitando as restrições de capacidade dos comutadores.
- Heurística **GSH**: consiste em resolver todos os  $m$  subproblemas por meio do seguinte algoritmo:

1. Ordenar as células em ordem crescente segundo os valores de  $(c_{ki} - t \pi_i)$ , onde  $t$  é o multiplicador lagrangeano/*surrogate*;
  2. Seja  $\{o_1, \dots, o_{n*m}\}$  a ordenação dos pares  $(i, k)$  de índices das  $n$  células atribuídas aos  $m$  comutadores;
  3. Para  $j = 1, \dots, n*m$ , faça:
    - Seja  $o_j = (i, k)$ ;
    - Alocar a célula  $i$  ao comutador  $k$ , caso a célula  $i$  ainda não esteja atribuída à comutador algum e a capacidade do comutador  $k$  não seja violada;
  4. Para as células não atribuídas, tentar a atribuição de menor custo ainda possível, respeitando as restrições de capacidade dos comutadores.
- Heurística **Ref**: usada para encontrar soluções viáveis melhores para o problema original, com base nas atribuições obtidas pela heurística **Sub** e pela solução de referência. A heurística Ref consiste nos seguintes passos:
1. Selecionar dentre as atribuições obtidas através da heurística Sub, aquelas em que uma célula está atribuída a um único comutador;
  2. Para cada célula  $i$  ainda não atribuída, atribuí-la ao comutador  $k$  ao qual esta célula está atribuída na solução de referência, caso isto seja possível, ou seja, se a restrição de capacidade do comutador  $k$  não for violada. Caso contrário, procurar atribuir a célula  $i$  a um comutador com capacidade para recebê-la ao menor custo possível;
  3. Realizar trocas entre as  $n$  células e os  $m$  comutadores considerando o melhor ganho para a função-objetivo do problema original.
  4. Substituir a solução de referência pela solução obtida, caso esta solução seja viável e melhor do que a solução de referência atual.

O método de geração de colunas proposto para o PAAC foi implementado utilizando três estratégias de resolução para os subproblemas:

- Estratégia **LG**: a cada iteração, os  $m$  subproblemas são resolvidos pela heurística LSH. Caso o valor da solução obtida pelo CPLEX para o PMR não se altere, utiliza-se também a heurística GSH;
- Estratégia **GSR**: a cada iteração, os  $m$  subproblemas são resolvidos pelas heurísticas GSH, Sub e Ref;
- Estratégia **GLR**: a cada iteração, os  $m$  subproblemas são resolvidos pelas heurísticas GSH, LSH e Ref.

Para o experimento computacional foram considerados os mesmos problemas testados por Menon e Gupta (2004). Para estes problemas,  $m$  (número de computadores) varia de 2 a 5, e os problemas estão divididos nas seguintes classes, segundo o valor de  $n$  (número de antenas):

- Classe P:  $n \in \{15, 30, 45, 60\}$ ;
- Classe M:  $n \in \{75, 100, 125, 150\}$ ;
- Classe G:  $n \in \{175, 200, 225, 250\}$ .

Nos testes realizados não se buscou o melhor multiplicador lagrangeano/*surrogate*. O valor do multiplicador  $t$  foi estabelecido inicialmente em 0.50 e, a cada iteração, este valor foi incrementado de 0.01 até atingir o valor 1.00, quando então foi mantido inalterado. A Tabela 2 mostra os resultados médios obtidos para cada classe e para cada uma das estratégias de resolução dos subproblemas.

Tabela 2 – Resultados computacionais para o PAAC

Estratégia	Classe	Tempo	Iter	FCols	GapC	GapV	Cpu
LG	P	26,79	8 [8]	40 [44]	0,336 [0,349]	0,336 [0,349]	0,01 [0,01]
	M	203,09	11 [10]	56 [54]	0,211 [0,210]	0,211 [0,210]	0,06 [0,05]
	G	327,61	12 [12]	65 [65]	0,146 [0,140]	0,146 [0,141]	0,18 [0,18]
GSR	P	26,79	7 [7]	48 [48]	0,048 [0,060]	0,210 [0,247]	0,89 [1,67]
	M	203,09	8 [9]	69 [71]	0,167 [0,169]	0,167 [0,169]	10,33 [10,50]
	G	327,61	10 [11]	72 [77]	0,122 [0,122]	0,122 [0,122]	56,73 [63,12]
GLR	P	26,79	9 [8]	58 [61]	0,125 [0,237]	0,216 [3,255]	0,31 [0,37]
	M	203,09	12 [9]	100 [77]	0,141 [0,171]	0,141 [1,216]	10,01 [6,88]
	G	327,61	11 [12]	77 [90]	0,122 [0,125]	0,122 [3,019]	50,56 [56,41]

Nesta tabela:

- **Tempo** é o tempo gasto pelo CPLEX para determinar a solução ótima do problema linearizado;
- **Iter** é o número de iterações do algoritmo;
- **FCols** é o número final de colunas do PMR;
- **GapC** é calculado como  $\text{GapC} = 100 * | \text{Soluc} - \text{LCpx} | / \text{Soluc}$ , onde **Soluc** é o valor da solução ótima do problema linearizado e **LCpx** é o limite obtido para o PMR;

- **GapV** é calculado como  $\text{GapV} = 100 * | \text{Soluc} - \text{Viável} | / \text{Soluc}$ , onde **Viável** é o valor da melhor solução viável encontrada;
- **Cpu** é o tempo computacional do algoritmo proposto, em segundos.

Pelos resultados mostrados na Tabela 2 pode-se constatar que os algoritmos propostos obtêm soluções de melhor qualidade (medida pelo indicador GapV) do que as obtidas pelo método tradicional de geração de colunas e, em geral, em menos tempo. Na comparação entre as três estratégias de resolução dos subproblemas, a estratégia LG é muito superior às outras duas em relação ao tempo computacional, embora a qualidade das soluções, tanto relaxada (medida pelo GapC), como viável (medida pelo GapV), seja melhor para a estratégia GSR.

Para os problemas das classes A, B, C e D da OR-Library, pode-se fazer uma comparação do método proposto para o PGA com os métodos propostos por Narciso (1998). A Tabela 3 apresenta os valores médios obtidos para essas classes de problemas. Nesta tabela:

- **Método** corresponde ao método utilizado, onde GC é o método de geração de colunas proposto, e  $(L_t * \text{SGAP}^\lambda)_a$ ,  $(L_t * \text{SGAP}^\lambda)_b$  e  $(L_t * \text{SGAP}^\lambda)_c$  são os métodos propostos por Narciso (1998), que utilizam otimização por subgradientes e relaxação lagrangeana/*surrogate*;
- **Iter** corresponde ao número de iterações;
- **Gap** é o desvio percentual que, para o método proposto, é calculado como para a Tabela 1 e no caso dos métodos propostos por Narciso e Lorena (1999) é calculado como:  

$$\text{Gap} = \frac{|\text{Soluc} - \text{LB}|}{\text{Soluc}}$$
, onde **Soluc** é o valor da solução ótima do problema e **LB** é o melhor limite inferior obtido para o problema;
- **Cpu** é o tempo computacional, em segundos, que para os métodos propostos por Narciso (1998), foi obtido em uma estação Sun SPARC 5.

Tabela 3 – Comparação de Resultados para o PGA

Método	Iter	Gap	Cpu
GC	391,42	0,073	886,10
$(L_t * \text{SGAP}^\lambda)_a$	148,71	0,899	2,60
$(L_t * \text{SGAP}^\lambda)_b$	433,71	1,992	115,83
$(L_t * \text{SGAP}^\lambda)_c$	541,96	2,643	278,84

Nota-se pelos resultados da Tabela 3 que, embora os métodos baseados na otimização por subgradientes sejam muito rápidos, a qualidade da solução obtida pelo método proposto é muito superior.

Para o PAAC, a Tabela 4 compara os resultados obtidos pelos métodos propostos com os resultados apresentados por Menon e Gupta (2004). Nesta tabela:

- **Método** corresponde ao método utilizado, onde GC/LG, GC/GSR e GC/GLR são os métodos propostos usando as estratégias LG, GSR e GLR, respectivamente, e MG é o método de Menon e Gupta (2004);
- **Gap** corresponde aos desvios percentuais, que no caso dos métodos propostos é calculado como:  $Gap = 100 * \left| \frac{LCpx - Viável}{Viável} \right|$ , onde **LCpx** e **Viável** são conforme estabelecidos para a Tabela 2, e no caso do método MG é calculado como:  $Gap = 100 * \left| \frac{UB - LB}{LB} \right|$ , onde **UB** é o valor da melhor solução viável obtida e **LB** é o valor obtido pela relaxação linear do problema;
- **Cpu** é o tempo computacional, em segundos.

Tabela 4 – Comparação de Resultados para o PAAC

Método	Gap	Cpu
GC/LG	0,23	0,08
GC/GSR	0,17	22,65
GC/GLR	0,16	20,29
MG	0,89	0,26

Embora esta comparação não possa ser considerada perfeita, pois os valores de Gap são calculados de forma diferente e os valores de Cpu foram obtidos em ambientes computacionais distintos (microcomputador Pentium III com 1.1 GHz e 512 MB de RAM e software CPLEX versão 7.5, para os métodos propostos, e microcomputador Pentium IV 2.4 GHz e software LINDO, para o método MG), os valores da Tabela 4 apresentam alguma evidência sobre o desempenho dos métodos propostos, indicando que a utilização da relaxação lagrangeana/*surrogate* pode ser vantajosa também para o PAAC.

## CONCLUSÃO

Este trabalho apresentou métodos de solução para dois problemas difíceis de

atribuição: o Problema Generalizado de Atribuição (PGA), um dos mais representativos problemas de Otimização Combinatória e para o qual existem muitos trabalhos publicados na literatura, e o Problema de Atribuição de Antenas a Comutadores (PAAC), um problema mais recente e ainda relativamente pouco estudado. Tanto para o PGA como para o PAAC, foram propostos métodos de geração de colunas para a solução aproximada do problema. Os métodos propostos utilizam a relaxação lagrangeana/*surrogate* que, como para outros problemas estudados anteriormente (Narciso e Lorena, 1999; Senne e Lorena, 2000; Lorena e Senne, 2004), demonstrou sua superioridade em relação à relaxação lagrangeana tradicional.

Em relação aos resultados computacionais para problemas de atribuição generalizados (Tabela 1), existem ganhos de desempenho em relação ao tempo de CPU, mas estes ganhos dependem da dimensionalidade do problema. Uma pesquisa interessante seria analisar este efeito em termos da estratégia *surrogate* e do uso de uma estrutura fixa de valores para o multiplicador  $t$ .

Em relação aos problemas de atribuição de antenas a comutadores (Tabela 2), existem ganhos em relação aos indicadores de qualidade das soluções e tempo de CPU para alguns problemas-teste, ainda que aparentemente não se possa afirmar que os algoritmos propostos sejam substancialmente melhores do que os métodos tradicionais de geração de colunas. Da mesma forma, pela falta de um índice de *gap* único, ainda não é possível concluir pelo melhor desempenho dos algoritmos propostos para o PAAC, em relação aos algoritmos propostos por Menon e Gupta (2004), embora existam indicações de que a relaxação lagrangeana/*surrogate* possa ser vantajosa também para o PAAC.

## REFERÊNCIAS BIBLIOGRÁFICAS

AERTS, J.; KORST, J.; SPIEKSMAN, F. Approximation of a retrieval problem for parallel disks. **Lecture Notes in Computer Science**, v. 2653, p. 178-188, 2003.

ALBAREDA-SAMBOLA, M.; VAN DER VLERK, M.H.; FERNÁNDEZ, E. **Exact Solutions to a Class of Stochastic Generalized Assignment Problems**. Research report no. 02A11, University of Groningen, Research Institute of Systems, Organisations and Management, Feb. 2002.

AMINI, M.M.; RACER, M. A rigorous comparison of alternative solution methods for the generalized assignment problem. **Management Science**, v. 40, p. 868-890, 1994.

BAKER, B.M. E SHEASBY, J. Accelerating the convergency of subgradient optimisation. **European Journal of Operational Research**, v. 117, p. 136-144, 1999.

BALACHANDRAN, V. **An integer generalized transportation model for optimal log assignment in computer networks**. Working Paper 34-72-3. Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburg, 1976.

BEASLEY, J.E. **OR-Library**. Disponível em: <http://www.brunel.ac.uk/depts/ma/research/jeb/orlib/gapinfo.html>. Acesso em 23 de dezembro de 2004.

BOKHARI, S.H. **Assignment Problems in Parallel and Distributed Computing**. Kluwer Academic Publisher, Boston, USA, 1987.

BURKARD, R.E. Selected topics on assignment problems. **Discrete Applied Mathematics**, v. 123, n. 1-3, p. 257-302, Nov. 2002.

CATTRYSSE, D.; SALOMON, M.; VAN WASSENHOVE, L. A set partitioning heuristic for the generalized assignment problem. **European Journal of Operational Research**, v. 72 p. 167-174, 1994.

CATTRYSSE, D.G.; VAN WASSENHOVE, L.N. A survey of algorithms for the generalized assignment problem. **European Journal of Operational Research**, v. 60, p. 260-272, 1992.

CERNY, V. Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. **Journal Optimization Theory Applications**, v. 45, n. 1, p. 41-51, 1985.

CHU, P.C.; BEASLEY, J.E. A genetic algorithm for the generalised assignment problem. **Computers and Operations Research**, v. 24, p. 17-23, 1997.

DANTZIG. G.B.; WOLFE, P. Decomposition principle for linear programs. **Operations Research**, v. 8, p. 101-111, 1960.

FISCHETTI, M.; LEPSCHY, C.; MINERVA, G.; ROMANIN-JACUR, G.; TOTO, E. Frequency assignment in mobile radio systems using branch-and-cut techniques. **European Journal of Operational Research**, v. 123, p. 241-255, 2000.

FRENCH, A.P.; WILSON, J.M. Heuristic Solution Methods for the Multilevel Generalized Assignment Problem. **Journal of Heuristics**, v. 8, n. 2, p. 143-153, Mar. 2002.

GAREY, M.R.; JOHNSON, D.S. Computers and intractability: A guide to the theory of NP-completeness. San Francisco: W.H. Freeman and Co., 1979. 340p.

GRUNDEL, D.A.; KROKHMAL, P.; OLIVEIRA, C.A.S.; PARDALOS, P.M. On the average case behavior of the multidimensional assignment problem **Pacific Journal of Optimization**, v. 1, n. 1, p. 39-57, 2005.

HUNG, M.S.; ROM, W.O. Solving the assignment problem by relaxation. **Operations Research**, v. 28, n. 4, p. 969-982, 1980.

KIRUBARAJAN, T.; BAR-SHALOM, Y.; PATTIPATI, K.R. Multiassignment for tracking a large number of overlapping objects. **IEEE Transactions on Aerospace and Electronic Systems**, v. 37, n. 1, p. 2-21, Jan. 2001.

KOOPMANS, T.C.; BECKMANN, M.J. Assignment problems and the location of economic activities. **Econometrica**, v. 25, p. 53-76, 1957.

KUHN, H.W. The Hungarian method for the assignment problem. **Naval Research Logistic Quarterly**, v. 2, p. 83-97, 1955.

LAGUNA, M.; KELLY, J.P.; GONZALEZ-VELARDE, J.L.; GLOVER, F. Tabu search for the multilevel generalized assignment problem. **European Journal of Operations Research**, v. 42, p. 677-687, 1995.

LOIOLA, E.M.; ABREU, N.M.M.; BOAVENTURA NETTO, P.O. Uma revisão comentada das abordagens do problema quadrático de alocação. **Pesquisa Operacional**, v. 24, n. 1, p. 73-109, Jan.-Abr. 2004.

LORENA, L.A.N., PEREIRA, M.A.; SALOMÃO, S.N.A. A relaxação lagrangeana/*surrogate* e o método de geração de colunas: novos limitantes e novas colunas. **Revista Pesquisa Operacional**, v. 23, n. 1, p. 29-47, 2003.

LORENA, L.A.N.; SENNE, E.L.F. A column generation approach to capacitated p-median problems. **Computers and Operations Research**, v. 31, n. 6, p. 863-876, 2004.

LUAN, F.; YAO, X. **Solving real-world lecture room assignment problems by genetic algorithms**, Complex Systems - From Local Interactions to Global Phenomena, IOS Press, Amsterdam, p.148-160, 1996.

MARTELLO, S.; TOTH, P. An algorithm for the generalized assignment problem. **Operational Research**, v. 81, p. 589-603, 1981.

MARTELLO, S.; TOTH, P. **Knapsack problems: algorithms and computer implementations**. New York, Wiley, 1990.

- MENON, S.; GUPTA, R. Assigning cells to switches in cellular networks by incorporating a pricing mechanism into simulated annealing. **IEEE Transactions on Systems, Man, and Cybernetics, Part B**, v. 34, n. 1, p. 558-565, 2004.
- MERCHANT, A.; SENGUPTA, B. Assignment of cells to switches in PCS networks. **IEEE/ACM Transactions on Networks**, v. 3, n. 5, p. 521-526, 1995.
- NARCISO, M.G. **A relaxação lagrangeana/surrogate e algumas aplicações em otimização combinatória**. 1998. 134p Tese (Doutorado em Computação Aplicada) Instituto Nacional de Pesquisas Espaciais, São José dos Campos. 1998.
- 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.
- NAUSS, R.M. Solving the generalized assignment problem: An optimizing and heuristic approach. **INFORMS Journal on Computing**, v. 15, n. 3, p. 249-266, 2003.
- NOWAKOVSKI, J.; SCHWÄRZLER, W.; TRIESCH, E. Using the generalized assignment problem in scheduling the ROSAT space telescope. **European Journal of Operational Research**, v. 112, n. 3, p. 531-541, 1999.
- OSMAN, I.H. Heuristics for the Generalized Assignment Problem: Simulated Annealing and Tabu Search Approaches, **OR Spektrum**, v. 17, p. 211-225, 1995.
- OSORIO, M.A.; LAGUNA M. Logic cuts for multilevel generalized assignment problems. **European Journal of Operational Research**, v. 151, n. 1, p. 238-246, Nov. 2003.
- PIERRE, S.; HOUÉTO, F. Assigning cells to switches in cellular mobile networks using taboo search. **IEEE Transactions on Systems, Man and Cybernetics, Part B**, v. 32, n. 3, p. 209-224, 2002.
- PIERSKALLA, W.P. The multidimensional assignment problem. **Operations Research**, v. 16, p. 422-431, 1968.
- PIGATTI, A.A. **Modelos e algoritmos para o problema de alocação generalizada e aplicações**. Dissertação de Mestrado, Pontifícia Universidade Católica do Rio de Janeiro, Julho 2003.

- PUSZTASZERI, J.; RENSING, P.E.; LIEBLING, T.M. Tracking elementary particles near their primary vertex: A combinatorial approach. **Journal of Global Optimization**, v. 9, n. 1, p. 41-64, July 1996.
- QUINTERO, A.; PIERRE S. A memetic algorithm for assigning cells to switches in cellular mobile networks. **IEEE Communication Letters**, v. 6, n. 11, p. 484-486, 2002.
- QUINTERO, A.; PIERRE S. Assigning cells to switches in cellular mobile networks: a comparative study. **Computer Communications**, v. 26, n. 9, p. 950-960, 2003.
- RULAND, K.S. A model for aeromedical routing and scheduling. **International Transactions in Operational Research**, v. 6, n. 1, p. 57-73, 1999.
- SAHA, D.; MUKHERJEE, A.; BHATTACHARYA, P. A simple heuristic for assignment of cells to switches in a PCS network. **Wireless Personal Communications**, v. 12, n. 3, p. 209-223, 2000.
- SENNE, E.L.F.; LORENA, L.A.N. **Lagrangean/surrogate heuristics for p-median problems**. In: Computing tools for modeling, optimization and simulation: interfaces in computer science and operations research. M. Laguna and J.L. Gonzalez-Velarde (eds.). Kluwer Academic Publishers, 2000, p. 115-130.
- WANG, L.; GU, W. Genetic algorithms with stochastic ranking for optimal channel assignment in mobile communications. **Lecture Notes in Computer Science**, v. 3314, p. 154-159, 2004.
- WILSON, J.M. A genetic algorithm for the generalised assignment problem. **Journal of the Operational Research Society**, v. 48, n. 8, p. 804-809, 1997.
- WRIGHT, M.B. Speeding up the Hungarian algorithm. **Computers and Operations Research**, v. 17, n. 1, p. 95-96, 1990.
- YAGIURA, M. **Generalized assignment problem instances**. Disponível em: <http://www-or.amp.i.kyoto-u.ac.jp/members/yagiura/gap/>. Acesso em 10 de janeiro de 2003.
- YAGIURA, M.; IBARAKI, T.; GLOVER, F. **A path relinking approach with ejection chains for the generalized assignment problem**. Technical Report 2004-002, Department of Applied Mathematics and Physics, Kyoto University, 2004a.
- YAGIURA, M.; IBARAKI, T.; GLOVER, F. An ejection chain approach for the generalized assignment problem. **Inform Journal on Computing**, v. 16, n. 2, p. 133-151, Spring 2004b.

## **AGRADECIMENTOS**

Os autores agradecem a dois revisores anônimos pelos valiosos comentários e sugestões.