UM ALGORITMO DE ENUMERAÇÃO IMPLÍCITA PARA A GERAÇÃO DE PADRÕES TABULEIROS EXATOS RESTRITOS

Horacio Hideki Yanasse
Instituto Nacional de Pesquisas Espaciais
Laboratório Associado de Computação e Matemática Aplicada
Avenida dos Astronautas, 1758 – Jardim da Granja – C.E.P.: 12227-010
São José dos Campos – S.P.
horacio@lac.inpe.br

Daniel Massaru Katsurayama
Instituto Nacional de Pesquisas Espaciais
Laboratório Associado de Computação e Matemática Aplicada
Avenida dos Astronautas, 1758 – Jardim da Granja – C.E.P.: 12227-010
São José dos Campos – S.P.
massaru@lac.inpe.br

Resumo

Padrões tabuleiros pertencem a uma classe especial de padrões 2-estágios guilhotinados que consomem menos tempo de máquina para serem cortados, portanto, de interesse particular em ambientes de grande demanda. Propomos um algoritmo enumerativo para a geração de padrões tabuleiros exatos restritos. A enumeração é feita priorizando a inclusão de itens no padrão com os maiores valores relativos levando em consideração a viabilidade da combinação dos itens em termos de área total comparada com a área do objeto e, a viabilidade da combinação dos itens para formar um padrão tabuleiro. Limitantes superiores para potenciais padrões que possam vir a ser gerados seguindo-se cada um dos ramos da enumeração são obtidos facilmente. Ramos podem, assim, ser eliminados sempre que estes limitantes forem menores ou iguais aos limitantes inferiores correntes.

Palavras-chave: padrão tabuleiro restrito exato, geração de padrões, algoritmo enumerativo.

Abstract

Checkerboard patterns belong to a special class of two-stage guillotine patterns that require less machine time to be cut, hence, of particular interest in high demand settings. We propose an enumeration scheme to determine

constrained exact checkerboard patterns. The enumeration is made priorizing in the pattern the items with the largest relative values and verifying the feasibility of the combination of the items in terms of their area compared with the area of the object and, the feasibility of the combination of the items to form a checkerboard pattern. Upper bounds on the potential patterns that can be generated following each one of the branches can be easily obtained. Branches can, therefore, be fathomed every time their bounds are smaller or equal to the current lower bound.

Keywords: exact restricted checkerboard pattern, pattern generation, enumerative algorithm.

1. Introdução

Padrões tabuleiros, conhecidos também como *padrões 1-grupo* (Gilmore e Gomory, 1965) pertencem a uma classe especial de padrões 2-estágios. Todas as faixas cortadas no primeiro estágio podem ser simultaneamente cortadas no segundo estágio uma vez que os pontos de corte são os mesmos (vide Figura 1).

Padrões tabuleiro são de particular interesse, em ambientes de grande produção, devido ao tempo reduzido para o seu corte.

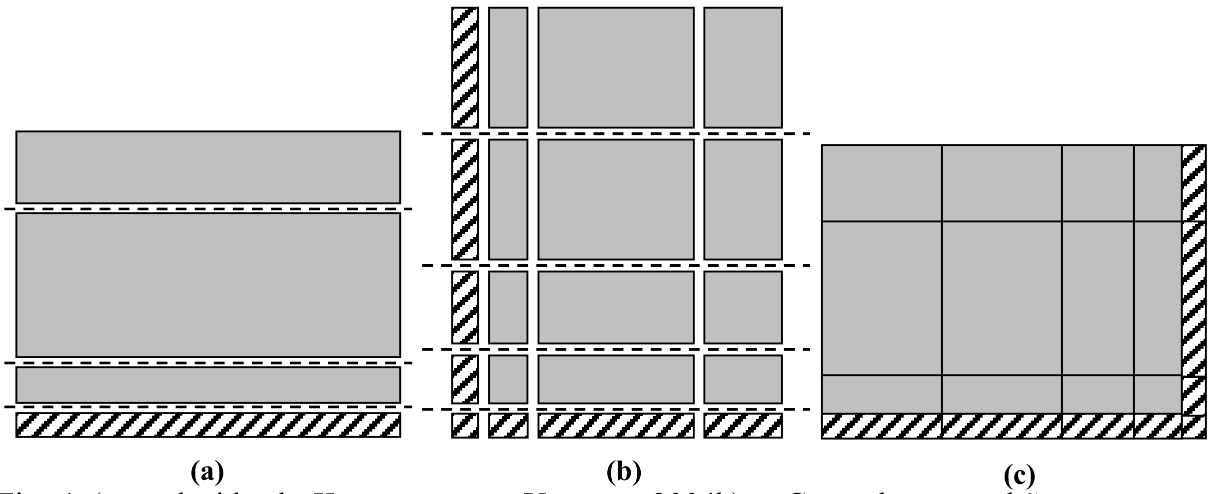

Fig. 1 (reproduzido de Katsurayama e Yanasse, 2004b) – Corte de um padrão tabuleiro: (a) primeiro corte (primeiro estágio), (b) giro em 90 graus (segundo corte, segundo estágio), (c) padrão tabuleiro final.

Denominamos padrões tabuleiros exatos aqueles padrões onde todos os itens produzidos são obtidos imediatamente após o corte do segundo estágio, como ilustrado na Figura 1. Os padrões *não exatos* são aqueles onde algumas das peças finais, para serem obtidas precisam de algum recorte (vide Figura 2). Consideramos que este recorte é feito em outra máquina e, por esta razão, o padrão é também classificado como de 2-estágios. Se o recorte tivesse que ser feito na própria máquina, o padrão tabuleiro não exato é, na realidade, um padrão de 3 ou mais estágios.

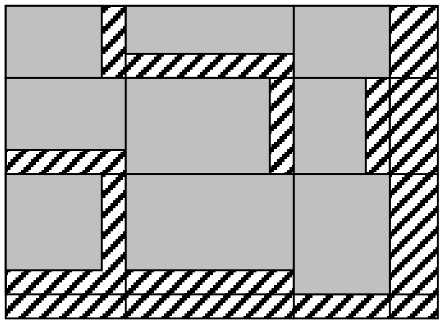

Fig. 2 (reproduzido de Katsurayama e Yanasse, 2004b) – Padrão tabuleiro não exato.

Existem inúmeros trabalhos versando sobre problemas de corte (veja, por exemplo, Gilmore e Gomory (1961, 1963, 1965), Hinxman (1980), Dyckhoff et al. (1985), Dyckhoff (1990), Dyckhoff e Waescher (1990), Dyckhoff e Finke (1992), Lirov (1992), Dowsland e Dowsland (1992), Sweeney e Paternoster (1992), Martello (1994a, 1994b), Bischoff e Waescher (1995), Mukhacheva (1997), Arenales e Morabito (1997), Dyckhoff et al. (1997), Arenales et al. (1999), Wang e Waescher (2002), Hifi (2002), Lodi et al. (2002)). Entretanto, são poucos os que abordam o padrão tabuleiro, apesar deste padrão ser de interesse, por exemplo, em cortes de chapas de madeira (Morabito e Garcia, 1998; Morabito e Arenales, 2000) ou de pedra (Scheithauer, 2002).

Em Morabito e Arenales (2000) o problema de geração de padrões tabuleiro é formulado como um problema inteiro quadrático. Dada a dificuldade em resolvê-lo estes autores propõem uma heurística simples para a geração destes padrões.

Scheithauer (2002) considera a geração de padrões tabuleiros restritos. Ele apresenta uma formulação não-linear e uma formulação de prgramação inteira mista para o problema e fornece alguns resultados de testes computacionais usando o pacote CPLEX.

Em Yanasse e Morabito (2003), modelos inteiros linerares e não-lineares para geração de padrões restritos guilhotinados 2-estágios e 1-grupo são apresentados, incluindo os casos exatos e não exatos. Testes computacionais limitados utilizando o CPLEX são apresentados para instâncias pequenas de até 10 itens.

Existem outros estudos que modelam padrões estagiados, por exemplo, em Beasley (1985), Christofides e Hadjiconstantinou (1995), Morabito e Arenales (1996), Hifi e Roucairol (2001) e Lodi e Monaci (2003). Entretanto, observamos que estas formulações não podem ser facilmente estendidas para padrões tabuleiros pois existe uma dificuldade em impor a restrição de que os cortes a serem realizados no segundo estágio devem ser nas mesmas posições em todas as tiras.

Yanasse e Katsurayama (2003), propõem um esquema enumerativo para gerar padrões tabuleiros exatos que apresenta um desempenho melhor em termos de tempos computacionais que os métodos de Scheithauer (2002) e Yanasse e Morabito (2003).

Em Katsurayama e Yanasse (2004b) é apresentado um algoritmo que constrói um padrão tabuleiro dada um combinação de itens ou conclui-se que não é possível obter um padrão tabuleiro com aquela combinação. Em Yanasse e Katsurayama (2004) e Katsurayama e Yanasse (2004a) um algoritmo é proposto para encontrar o melhor padrão tabuleiro gerando-se em ordem decrescente de valor, todas as combinações possíveis dos itens que tenham uma área total menor ou igual à área do objeto e, analisando-se sequencialmente cada uma dessas combinações, aplica-se o algoritmo descrito em Katsurayama e Yanasse (2004b) até que uma combinação que gera uma solução viável é obtida.

Neste trabalho, propomos uma abordagem diferente, um processo de enumeração implícita, para a determinação de padrões tabuleiros exatos restritos, e que se utiliza de resultados e desenvolvimentos apresentados nestes trabalhos recentes citados anteriormente. Uma enumeração é proposta priorizando a inclusão de itens no padrão com os maiores valores relativos. As possíveis ramificações são realizadas levando-se em consideração também a viabilidade da combinação dos itens em termos de área total comparada com a área do objeto e se a viabilidade da combinação dos itens para formar um padrão tabuleiro também é verificada.

2. Um algoritmo para a geração de padrões tabuleiros exatos restritos

Nosso problema consiste em dado um objeto retangular de dimensão Wx C, obter o melhor padrão tabuleiro composto de um subconjunto de M tipos diferentes de itens retangulares menores de dimensão w_k x c_k ($k = 1,...,M$), sendo que o número máximo de itens do tipo k que pode aparecer no padrão é dado por d_k, $k = 1,...,M$. Determina-se o melhor padrão tabuleiro exato segundo uma função que associa a cada item um valor ou utilidade.

O algoritmo proposto utiliza a o procedimento de enumeração implícita sugerido por Gilmore e Gomory (1963) para resolução do problema da mochila combinado com o método de construção de geração de padrões tabuleiros exatos de Katsurayama e Yanasse (2004b).

Seja l_i a área do item i, i.e., $l_i = c_i.w_i$ e π_i o valor, lucro ou utilidade associado ao item i. O problema P que desejamos resolver é

P: Maximize $\sum_i \pi_i a_i$ (1)

Sujeito a: a_i, $i \in$ I forma um padrão tabuleiro (2)

$a_i \leq d_i$, para todo $i \in$ I (3)

$a_i \geq 0$, para todo $i \in$ I, e inteiro, (4)

onde I = {$1,2,\ldots,M$}

Uma condição necessária que precisa ser satisfeita por qualquer combinação dos itens em qualquer padrão viável é que a soma de suas áreas não pode exceder a área do objeto, ou seja, $\sum_i l_i a_i \leq W.C = L$. Assim, podemos incluir mais esta restrição no problema (P) obtendo:

P: Maximize $\sum_i \pi_i a_i$ (1)

Sujeito a: a_i, $i \in$ I forma um padrão tabuleiro (2)

$a_i \leq d_i$, para todo $i \in$ I (3)

$a_i \geq 0$, para todo $i \in$ I, e inteiro, (4)

onde I = $\{1,2,\ldots,M\}$

$\sum_i l_i a_i \leq L$. (5)

Considere o problema da mochila unidimensional restrito formado por (1), (3), (4) e (5). Este problema tem sido bastante estudado na literatura e diversos algoritmos já foram propostos para a sua resolução (vide, por exemplo, Yanasse e Soma, 1987; Martello e Toth, 1990; yanasse e Soma, 1990; Yanasse et al. 2000). Gilmore e Gomory (1963) sugerem uma enumeração implícita para sua resolução. Nossa proposta é adaptar esta mesma enumeração implícita destes autores, levando em consideração a restrição (2) durante a enumeração feita. A seguir, reproduzimos de Pinto (2004), o algoritmo de enumeração implícita de Gilmore e Gomory (1963) para solução do problema da mochila restrito.

Algoritmo de Gilmore e Gomory (1963)
(reproduzido de Pinto, 2004)

Passo 1 : {Defina o problema segundo as variáveis mais valiosas}
Para cada item do tipo i (à qual estão associados um valor π_i e uma área l_i), defina: $v_i = \frac{\pi_i}{l_i}, i = 1,2,\ldots,M$. Os itens mais valiosos são aqueles com os maiores valores de v_i. Desta forma, reordene as variáveis em ordem não crescente, segundo os v_i's. Sem perda de generalidade, admita que: $v_1 \geq v_2 \geq \ldots \geq v_M$.
Passo 2 : {Determine a solução inicial, utilizando busca em profundidade primeiro}
Determine a solução $\boldsymbol{a} = (a_1, a_2, \ldots, a_M)$, tal que:

$$a_1 = \left\lfloor \frac{L}{l_1} \right\rfloor$$

$$a_{k+1} = \left\lfloor \frac{L - \sum_{i=1}^{k} l_i a_i}{l_{k+1}} \right\rfloor, \quad k = 1, \ldots, M-1$$

Passo 3: {Avalie a solução corrente e armazene a mais valiosa}

Determine: $g(\boldsymbol{a}) = \sum_{i=1}^{m} \pi_i a_i$

Se $\underline{G} < g(\boldsymbol{a})$ (considerando inicialmente $\underline{G} = 0$); **então** faça:

$\underline{G} = g(\boldsymbol{a})$

e guarde a solução correspondente: $\underline{\boldsymbol{a}} = \boldsymbol{a}$

Passo 4: {Teste a otimalidade e calcule o limitante superior}

Determine k o maior índice tal que $a_k \neq 0$, ou seja, o item k corresponderá ao item menos valioso que pertence à solução avaliada.

Se $k = 0$ **então** PARE , a melhor solução guardada em $\underline{\boldsymbol{a}}$ é solução ótima.

Senão, calcule: (se $k = M$, considere $\pi_{M+1} = 0$ e $l_{M+1} = 1$)

$$\overline{G} = \pi_1 a_1 + \pi_2 a_2 + \ldots + \pi_k (a_k - 1) + \frac{\pi_{k+1}}{l_{k+1}} \left(L - l_1 a_1 - l_2 a_2 - \ldots - l_k (a_k - 1) \right)$$

Passo 5 : {*Backtracking*}

Com o valor de $\overline{G}$, calculado no passo 4, verifica-se se compensaria retirar um item do tipo k e substituí-lo por itens do tipo $k+1$, ou seja, se $\overline{G} > \underline{G}$. Se compensar, o passo 5.2 é realizado, senão o passo 5.1 será feito.

5.1 {Retorno longo}

Se $\overline{G} \leq \underline{G}$ faça $a_k = 0$ (retire todos os itens do tipo k) e volte ao passo 4.

5.2 {Retorno ao nó precedente e nova busca em profundidade}

Se $\overline{G} > \underline{G}$ então faça $a_k \leftarrow a_k - 1$ e defina a nova solução $\boldsymbol{y}$:

$$a_{j+1} = \left\lfloor \frac{L - \sum_{i=1}^{j} l_i a_i}{l_{j+1}} \right\rfloor, \quad j = k, \ldots, M-1$$

e volte ao passo 3.

O método descrito anteriormente resolve o problema da mochila irrestrito com relação à demanda. Para o caso restrito, introduz-se as seguintes modificações:

Modificação 1:

No passo 2, a solução inicial será determinada alocando na mochila o mínimo entre o que cabe do item i e a sua demanda, ou seja:

$a_i = \min\left\{ \left\lfloor \frac{X}{l_i} \right\rfloor, d_i \right\}$ onde: X = espaço restante da mochila.
Inicialmente, $X = L$.
Esta condição deverá ser considerada também no passo 5.2, quando uma nova solução $\boldsymbol{a}$ é determinada.

Modificação 2:
No passo 4, as seguintes relações devem ser consideradas:
Seja: $\overline{L} = L - l_1 a_1 - \ldots - l_k(a_k - 1)$;

Determine j tal que: $\begin{cases} \sum_{i=k+1}^{j-1} l_i\, d_i \le \overline{L} \\ \sum_{i=k+1}^{j} l_i\, d_i > \overline{L} \end{cases}$

Agora, a solução do problema da mochila contínua restrita é
$a_i = d_i \qquad i = k+1, \ldots, j-1$

$$a_j = \frac{\overline{L} - \sum_{i=k+1}^{j-1} l_i\, a_i}{l_j}$$

e, o limitante superior é
$\overline{G} = \pi_1 a_1 + \ldots + \pi_k(a_k - 1) + \pi_{k+1} a_{k+1} + \ldots + \pi_j a_j$

O algoritmo anterior resolve apenas o problema (1), (3), (4) e (5). Para que resolva o problema (1), (2), (3), (4) e (5) é preciso que a combinação obtida produza um padrão tabuleiro. Para isso, sugerimos a incorporação das seguines modificações para levar em conta a restrição (2):

Modificação 3:
No passo 2, a solução inicial será determinada alocando na mochila o mínimo entre o que cabe do item i, sua demanda, e o número máximo que se pode ter do item i para que um padrão tabuleiro possa ser gerado, ou seja:

$a_i = \min \{\lfloor X/l_i \rfloor, d_i, u_i\}$

onde
X = espaço restante da mochila e
u_i é o número máximo de itens tipo i que pode ser incluído na combinação corrrente e ainda assim, é possível gerar um padrão tabuleiro exato.
Inicialmente, $X = L$.

A verificação se é possível gerar um padrão tabuleiro exato é feita com o algoritmo descrito em Katsurayama e Yanasse (2004b).

Condição similar é também considerada no passo 5.2, quando uma nova solução $\boldsymbol{a}$ é determinada.

Modificação 4:
No passo 4, as seguintes relações devem ser consideradas:
Seja: $\overline{L} = L - l_1 a_1 - \ldots - l_k(a_k - 1)$;

Determine j tal que: $$\begin{cases} \sum_{i=k+1}^{j-1} l_i\, h_i \leq \overline{L} \\ \sum_{i=k+1}^{j} l_i\, h_i > \overline{L} \end{cases}$$

onde
$h_i = \min\ \{d_i, u_i\}$ e
u_i é como definido anteriormente.

A solução do problema da mochila contínua restrita é

$$a_i = h_i \qquad i = k+1, \ldots, j-1$$

$$a_j = \frac{\overline{L} - \sum_{i=k+1}^{j-1} l_i\, a_i}{l_j}$$

e, o limitante superior é

$$\overline{G} = \pi_1 a_1 + \ldots + \pi_k(a_k - 1) + \pi_{k+1} a_{k+1} + \ldots + \pi_j a_j$$

Modificação 5:

No passo 5.2, a seguinte solução deve ser considerada:
$a_1, a_2, a_3, \ldots, a_k - 1, a_i = h_i, i = k+1, \ldots, j-1,$

$$a_j = \lfloor (L - \sum_{i=k+1}^{j-1} l_i a_i)/l_j) \rfloor$$

$$a_{s+1} = \min\ \{ \lfloor (L - \sum_{i=1}^{s} l_i a_i)/l_{s+1}) \rfloor, d_{s+1}, u_{s+1}\}, \qquad s = j, \ldots, M-1.$$

Com estas simples modificações o algoritmo acima gera o melhor padrão tabuleiro exato restrito.

A implementação deste algoritmo deve ser feita utilizando-se uma árvore de enumeração onde é acrescentado/modificado, a cada vez, apenas um novo item à combinação corrente avaliada. Para a combinação corrente, já se tem determinado se ela gera um padrão tabuleiro e, a análise da viabilidade da nova combinação com o acréscimo de um novo item à combinação corrente é facilitada, uma vez que se pode partir já da configuração presente já construída. Com isto, entende-se que economias em computação são ganhos em comparação com o algoritmo de Yanasse e Katsurayama (2004), uma vez que, neste último algoritmo, a cada nova combinação da lista ordenada, a verificação se esta gera um padrão tabuleiro parte do início, nada aproveitando das soluções construídas anteriormente.

3. Considerações finais

Um novo algoritmo para o problema de geração de padrões tabuleiro, exato restrito foi proposto baseado na composição de dois algoritmos: o algorimo de Gilmore e Gomory (1963) para resolução do problema da mochila unidimensional restrito e o algoritmo descrito em Katsurayama e Yanasse (2004c) que constrói um padrão tabuleiro exato dada a combinação de itens que a compõe.

Além da diminuição dos tempos computacionais, uma outra vantagem do presente algoritmo é que se uma solução viável é rapidamente obtida e depois, o algoritmo evolui tentando achar soluções melhores. No algoritmo de Yanasse e Katsuryama (2004a) não se dispõe de nenhuma solução até que o algoritmo pare sua execução.

Em continuidade a este trabalho, pretende-se implementar este novo algoritmo e realizar testes computacionais visando comprovar a eficiência dele em comparação ao algoritmo anterior destes mesmos autores.

Agradecimentos: Este trabalho foi parcialmente financiado pela FAPESP e CNPq.

Referências

[1] Arenales, M.N.; Morabito, R. (editores). (1997). Problemas de corte e empacotamento e aplicações industriais. XX Congresso Nacional de Matemática Aplicada e Computacional (XX CNMAC), Gramado.

[2] Arenales, M., Morabito, R., Yanasse, H. (eds.), 1999, Special issue: Cutting and packing problems. **Pesquisa Operacional** 19(2), 107-299.

[3] Beasley, J.E. (1985). Algorithms for unconstrained two-dimensional guillotine cutting. **Journal of the Operational Research Society** 36(4), 297-306.

[4] Bischoff, E. and Waescher, G. (eds.), 1995, Special issue: Cutting and packing. **European Journal of Operational Research**, 84(3).

[5] Christofides, N. and Hadjiconstantinou, E., 1995, An exact algorithm for orthogonal 2-D cutting problems using guillotine cuts. **European Journal of Operational Research**, 83, 21-38.

[6] Dowsland, K. e Dowsland, W. (1992). Packing problems. **European Journal of Operations Research** 56, 2-14.

[7] Dyckhoff, H. (1990). A typology of cutting and packing problems. **European Journal of Operations Research** 44, 145-159.

[8] Dyckhoff, H. e Finke, U. (1992). Cutting and packing in production and distribution: typology and bibliography. Springler-Verlag Co, Heidelberg.

[9] Dyckhoff, H.; Kuse, H.-J; Abel, D.; Gal, T. (1985). Trim loss and related problems. **Omega** 13, 59-72.
[10] Dyckhoff, H., Scheithauer, G., Terno, J., 1997, Cutting and packing. In M. Amico, F. Maffioli, F., S. Martello (eds.), Annotated bibliographies in combinatorial optimisation. John Wiley & Sons, New York, NY, 393-414.
[11] Dyckhoff, H. and Waescher, G. (eds.), 1990, Special issue: Cutting and packing. **European Journal of Operational Research**, 44(2).
[12] Gilmore, P.; Gomory, R. (1961). A linear programming approach to the cutting-stock problem. **Operations Research** 9, 848-859.
[13] Gilmore, P.; Gomory, R. (1963). A linear programming approach to the cutting-stock problem II. **Operations Research** 11, 863-888.
[14] Gilmore, P.; Gomory, R. (1965). Multistage cutting stock problems of two and more dimensions. **Operations Research** 13, 94-120.
[15] Hifi, M. (ed), 2002, Special issue on cutting and packing. **Studia Informatica Universalis**, 2, 1-161.
[16] Hifi, M. and Roucairol, C., 2001, Approximate and exact algorithms for constrained (un)weighted two-dimensional two-staged cutting stock problems. **Journal of Combinatorial Optimization**, 5, 465-494.
[17] Hinxman, A. (1980). The trim-loss and assortment problems: a survey. **European Journal of Operational Research** 5, 8-18.
[18] Katsurayama, D. M.; Yanasse, H. H. Um algoritmo enumerativo para determinação de padrões tabuleiros exatos e restritos. IV WORCAP, São José dos Campos, SP, 20 a 21 de outubro de 2004a.
[19] Katsurayama, D. M.; Yanasse, H. H. Um algoritmo para geração de padrões tabuleiros exatos a partir de uma combinação de itens. XXXVI Simpósio Brasileiro de Pesquisa Operacional, São João del Rei (MG), 23 a 26 de novembro de 2004b.
[20] Lirov, Y. (ed), 1992, Special issue: Cutting stock: Geometric resource allocation, **Mathematical and Computer Modelling**, 16(1).
[21] Lodi, A., Martello, S. and Monaci, M., 2002, Two-dimensional packing problems: a survey. **European Journal of Operational Research**, 141, 241-252.
[22] Lodi, A. and Monaci, M., 2003, Integer programming models for 2-staged two-dimensional knapsack problems. **Mathematical Programming**, 94, 257-278. 23.
[23] Martello, S. (ed), 1994a, Special issue: Knapsack, packing and cutting, Part I: One-dimensional knapsack problems. **INFOR**, 32(3).
[24]. Martello, S. (ed), 1994b, Special issue: Knapsack, packing and cutting, Part II: Multidimensional knapsack and cutting stock problems. **INFOR**, 32(4).
[25] Martello, S.; Toth, P. (1990). **Knapsack problems – Algorithms and computer implementations**. John Wiley & Sons, Chichester.
[26] Morabito, R. e Arenales, M. (1996). Staged and constrained two-dimensional guillotine cutting problems: An and/or-graph approach, **European Journal of Operational Research**, 94, 548-560.

[27] Morabito, R.; Arenales, M.N. (2000). Optimizing the cutting of stock plates in a furniture company. **International Journal of Production Research** 12(38), 2725-2742.
[28] Morabito, R. and Garcia, V. (1998). The cutting stock problem in a hardboard industry: A case study. **Computers & Operations Research**, 25(6), 469-485.
[29] Mukhacheva, E. A. (ed), 1997, Decision making under conditions of uncertainty: cutting–packing problems. The International Scientific Collection, Ufa, Russia.
[30] Pinto, M.J. (2004). Algumas contribuições à resolução do problema de corte integrado ao problema de sequenciamento dos padrões, Tese de Doutorado, Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP.
[31] Scheithauer, G. (2002). On a two dimensional guillotine cutting problem. IFORS XVI, Edinburgh, UK.
[32] Sweeney, P.; Patternoster, E. (1992). Cutting and packing problems: a categorised, application-oriented research bibliography. **Journal of the Operational Research Society** 43, 691-706.
[33] Wang, P. and Waescher, G. (eds.), 2002, Special issue on cutting and packing problems. **European Journal of Operational Research**, 141, 239-469.
[34] Yanasse, H.H.; Katsurayama, D.M. (2003) Checkerboard patterns: proposals for its generation. To appear in **International Transactions in Operations Research**.
[35] Yanasse, H.H.; Katsurayama, D.M. (2004). An enumeration scheme to generate constrained exact checkerboard patterns. EURO XX Conference, realizado em Rhodes, Grécia. Book of Abstracts, p. 36.
[36] Yanasse, H.H.; Morabito, R. (2003) Linear models for two-stage constrained two-dimensional guillotine cutting problems. Submitted.
[37] Yanasse, H.H.; Soma, N.Y. (1987) A new enumeration scheme for the knapsack problem. **Discrete Applied Mathematics** 18, 235-245.
[38] Yanasse, H.H.; Soma, N.Y. (1990) Finding the k-best solutions to a value independent knapsack problem. IFORS XII, Athens, Greece.
[39] Yanasse, H.H.; Soma, N.Y.; Maculan N. (2000). An algorithm for determining the k-best solutions of the one-dimensional knapsack problem. **Pesquisa Operacional** 1(20), 117-134.