

ALGORITMOS PARA DETERMINAÇÃO DE PADRÕES TABULEIROS EXATOS E RESTRITOS: TESTES COMPUTACIONAIS COMPARATIVOS

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.
E-mail: massaru@lac.inpe.br

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.
E-mail: horacio@lac.inpe.br

Resumo

Neste trabalho, testamos dois algoritmos para determinação de padrões tabuleiros exatos, onde o número de itens no padrão é limitado. O primeiro algoritmo baseia-se no algoritmo enumerativo de Yanasse, Soma e Maculan (2000) para determinação das K-melhores soluções para o problema da mochila unidimensional. O segundo algoritmo baseia-se no método da enumeração implícita de Gilmore e Gomory (1963) para resolução do problema da mochila unidimensional irrestrito. Um refinamento no limitante superior do segundo algoritmo é sugerido e seu impacto no tempo de execução é avaliado.

Palavras-chave: padrão tabuleiro, algoritmo enumerativo, algoritmo de enumeração implícita.

Abstract

We test two algorithms for determining exact checkerboard patterns where the number of items in the pattern is limited. The first one is based on an enumerative algorithm proposed by Yanasse, Soma and Maculan (2000) for determining the K-best solutions of the one-dimensional knapsack problem. The second one is based on a Gilmore and Gomory's (1963) implicit enumeration scheme for solving the one dimensional unconstrained knapsack problem. A refinement in the upper bound of the second algorithm is suggested and its impact in the execution time is evaluated.

Keywords: checkerboard pattern, enumerative algorithm, implicit enumeration algorithm.

1 – Introdução

Padrões tabuleiros, também conhecidos como padrões 1-grupo (Gilmore e Gomory, 1965) pertencem a uma classe especial de padrões 2-estágios que podem ser produzidos sem a necessidade de cortar separadamente cada uma das faixas obtidas no primeiro estágio de modo que apenas a serra (ou o objeto, vide Belluzzo, 2002) é girada em 90 graus após o corte do primeiro estágio. Padrões tabuleiros, demandam menos tempo de máquina e são de particular interesse em ambientes de grande

demanda, particularmente, quando a máquina de corte constitui-se num gargalo da produção. Os padrões tabuleiros são exatos quando todos os itens produzidos são obtidos imediatamente após o corte do segundo estágio, como ilustrado na Figura 1a. Padrões tabuleiros podem também ser do tipo não exato (vide Figura 1b), caso haja necessidade de se recortar as peças obtidas no segundo estágio para produzir os itens finais solicitados. Neste caso, muitas vezes, o recorte é feito em outra máquina, possivelmente menos carregada e mais simples, mas certamente incorrendo-se em custos adicionais.

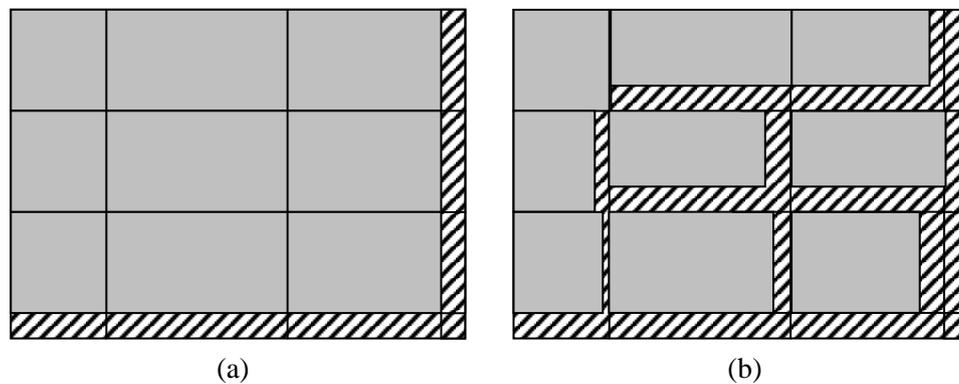


Fig. 1 – Exemplos de padrões tabuleiros: (a) padrão tabuleiro exato, (b) padrão tabuleiro não exato.

São poucos os trabalhos encontrados na literatura que tratam da determinação de padrões tabuleiros. Algoritmos, heurísticas e modelos matemáticos para a determinação de padrões tabuleiros exatos e não exatos estão presentes na literatura, nos trabalhos de Morabito e Arenales (2000), Katsurayama e Yanasse (1999, 2000), Scheithauer (2002) e, mais recentemente, Yanasse e Morabito (2003).

Morabito e Arenales (2000) formularam o problema da determinação de padrões tabuleiros exatos como um problema quadrático inteiro e propuseram uma heurística para sua obtenção. Em Katsurayama e Yanasse (1999, 2000) é apresentado um algoritmo enumerativo para determinação de padrões tabuleiros exatos baseado no algoritmo enumerativo de Yanasse, Soma e Maculan (2000) para determinação das K-melhores soluções para o problema da mochila unidimensional.

Modelos matemáticos para determinação de padrões tabuleiros restritos foram apresentados por Scheithauer (2002) e por Yanasse e Morabito (2003). No entanto, estes modelos podem requerer elevado tempo computacional na resolução do problema de corte devido ao grande número de variáveis envolvidas. Não se tem conhecimento na literatura de outros trabalhos a respeito.

Neste trabalho, implementamos e testamos dois algoritmos para a determinação de padrões tabuleiros exatos, onde o número de itens de cada tipo a serem produzidos no padrão é limitado. A limitação de itens em padrões tabuleiros ocorre, por exemplo, no corte de pedras. Os dois algoritmos se baseiam em procedimentos para resolução do problema da mochila unidimensional presentes na literatura. O primeiro algoritmo, apresentado em Katsurayama e Yanasse (2004b), se baseia no algoritmo enumerativo de Yanasse, Soma e Maculan (2000) para determinação das K-melhores soluções para o problema da mochila unidimensional e, o segundo algoritmo, apresentado em Yanasse e Katsurayama (2004), se baseia no método da enumeração implícita de Gilmore e Gomory (1963). Ambos os algoritmos também utilizam o algoritmo construtivo para geração de padrões tabuleiros a partir de uma combinação dada de itens apresentado em Katsurayama e Yanasse (2004a). O algoritmo construtivo de Katsurayama e Yanasse (2004a) pressupõe uma combinação de itens que poderiam estar no padrão e fornece um padrão tabuleiro exato com estes itens, caso um exista ou conclui que não existe tal padrão.

Testes computacionais são apresentados para comparar os dois algoritmos propostos com os resultados obtidos através do modelo matemático de Yanasse e Morabito (2003). Um refinamento no limitante superior do segundo algoritmo é sugerido e também testado.

2 – Os Algoritmos Implementados

Considere, como dados de entrada para os algoritmos, um objeto retangular de dimensão $W \times L$ a ser cortado de modo a produzir M diferentes tipos de itens retangulares menores de dimensão $w_i \times l_i$. Cada item i ($i = 1, \dots, M$) apresenta um lucro (valor de utilidade) π_i e uma demanda d_i associados.

2.1 – Algoritmo Enumerativo

A área ocupada por cada um dos itens no padrão é dada por $a_i = w_i l_i$. Sabe-se que uma condição necessária que deve ser satisfeita por quaisquer combinações de itens em um padrão é que a soma total de suas áreas não deve exceder a área do objeto, o que corresponde a resolver o seguinte problema relaxado:

$$\begin{aligned} \text{Maximizar} \quad & \sum_{i=1}^M \pi_i \lambda_i \\ \text{Sujeito a:} \quad & \sum_{i=1}^M a_i \lambda_i \leq WL, \\ & \lambda_i \leq d_i, \quad i = 1, \dots, M, \\ & \lambda_i \geq 0, \quad i = 1, \dots, M \text{ e inteiro.} \end{aligned}$$

No primeiro algoritmo, encontra-se a melhor solução para este problema relaxado e verifica-se a possibilidade de gerar um padrão tabuleiro exato para a combinação $\lambda = (\lambda_1, \dots, \lambda_M)$ obtida. Se for possível a geração do padrão, então a solução foi encontrada e o algoritmo é encerrado. Caso contrário, prossegue-se verificando a possibilidade de geração da segunda, da terceira, ..., da K -melhor combinação λ , até que o padrão tabuleiro exato seja obtido. Um resumo deste procedimento é apresentado adiante:

Passo 1: construa uma lista com as K -melhores soluções (combinações) para o problema relaxado;

Passo 2: pegue a melhor combinação da lista construída;

Passo 3: usando o algoritmo de Katsurayama e Yanasse (2004a), tente gerar um padrão tabuleiro exato para a dada combinação. Se o padrão tabuleiro foi gerado com sucesso, então PARE; Caso contrário, retorne ao Passo 2.

2.2 – Algoritmo de Enumeração Implícita

Este segundo algoritmo baseia-se no método da enumeração implícita de Gilmore e Gomory (1963). A vantagem do algoritmo de enumeração implícita sobre o algoritmo enumerativo apresentado anteriormente é a possibilidade de interromper sua execução e mesmo assim dispor de alguma solução. No algoritmo enumerativo anterior uma solução só é obtida no final de sua execução.

O método proposto por Gilmore e Gomory (1963) resolve o Problema da Mochila Unidimensional Irrestrito. Adaptações foram feitas neste método para lidar com o caso restrito. Além disso, o algoritmo construtivo para gerar padrões tabuleiros exatos a partir de uma combinação dada de itens proposto em Katsurayama e Yanasse (2004) foi embutido no novo algoritmo. O método faz uma busca em profundidade primeiro e aplica limitantes superiores e inferiores para que o processo de enumeração das soluções seja realizado de maneira implícita, sem a necessidade de se verificar todas as combinações do problema.

O algoritmo de enumeração implícita para determinação de padrões tabuleiros exatos e restritos é apresentado a seguir:

Passo 1: ordenação dos itens.

Para cada tipo de item i , defina $v_i = \frac{\pi_i}{w_i l_i}$ ($i = 1, \dots, M$) e reordene os itens, tal que $v_1 \geq v_2 \geq \dots \geq v_M$.

Passo 2: determinação da solução inicial.

Determine $a = (a_1, a_2, \dots, a_M)$, tal que $a_i = \min \left\{ \left\lfloor \frac{X}{w_i l_i} \right\rfloor, d_i, u_i \right\}, i = 1, \dots, M$, onde:

X = espaço restante da mochila (inicialmente, $X = WL$);

u_i = número de itens do tipo i que podem ser colocados no padrão e, ainda assim, um padrão tabuleiro pode ser gerado.

Passo 3: avaliação da solução corrente e armazenamento da solução mais valiosa.

Determine $g(a) = \sum_{i=1}^M \pi_i a_i$;

Se $\underline{G} < g(a)$ (inicialmente, $\underline{G} = 0$), então faça $\underline{G} = g(a)$ e guarde $\underline{a} = a$.

Passo 4: teste de otimalidade e cálculo do limitante superior.

Determine o maior índice k , tal que $a_k > 0$;

Se $k = 0$ então PARE, \underline{a} é a solução ótima;

Senão, faça:

Início

$$\overline{WL} = WL - w_1 l_1 a_1 - \dots - w_k l_k (a_k - 1);$$

$$\text{Determine } j, \text{ tal que: } \begin{cases} \sum_{i=k+1}^{j-1} w_i l_i d_i \leq \overline{WL} \\ \sum_{i=k+1}^j w_i l_i d_i > \overline{WL} \end{cases};$$

Se j for indeterminado, então faça:

$$\text{Se } (w_{k+1} l_{k+1} d_{k+1} > \overline{WL}) \text{ então faça } \overline{G} = \pi_1 a_1 + \dots + \pi_k (a_k - 1) + \pi_{k+1} \left(\frac{\overline{WL}}{w_{k+1} l_{k+1}} \right);$$

$$\text{Senão faça } \overline{G} = \pi_1 a_1 + \dots + \pi_k (a_k - 1) + \pi_{k+1} d_{k+1} + \dots + \pi_M d_M;$$

Senão faça:

Início

$$a_i = d_i, i = k + 1, \dots, j - 1;$$

$$a_j = \left\lfloor \frac{\overline{WL} - \sum_{i=k+1}^{j-1} w_i l_i a_i}{w_j l_j} \right\rfloor;$$

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

Fim;

Fim.

Passo 5: backtracking.

Se $\bar{G} \leq \underline{G}$, então faça:

Início

$$a_k = 0;$$

$$a_{i+1} = \min \left(\left\lfloor \frac{X}{w_{i+1} l_{i+1}} \right\rfloor, d_{i+1}, u_{i+1} \right), \quad i = k, \dots, M - 1, \text{ onde:}$$

X = espaço restante da mochila (inicialmente, $X = WL - w_1 l_1 a_1 - \dots - w_{k-1} l_{k-1} a_{k-1}$);

u_{i+1} = número de itens do tipo $i+1$ ($i = k, \dots, M - 1$) que podem ser colocados no padrão e, ainda assim, um padrão tabuleiro pode ser gerado;

Volte para o Passo 4;

Fim;

Senão, faça:

Início:

$$a_k = a_k - 1;$$

Se ($a_k < 0$) faça $a_k = 0$ e volte para o Passo 4;

$$a_{i+1} = \min \left(\left\lfloor \frac{X}{w_{i+1} l_{i+1}} \right\rfloor, d_{i+1}, u_{i+1} \right), \quad i = k, \dots, M - 1, \text{ onde:}$$

X = espaço restante da mochila (inicialmente, $X = WL - w_1 l_1 a_1 - \dots - w_k l_k a_k$);

u_{i+1} = número de itens do tipo $i+1$ ($i = k, \dots, M - 1$) que podem ser colocados no padrão e, ainda assim, um padrão tabuleiro pode ser gerado;

Fim Enquanto ($a_{k+1} = 0, \dots, a_M = 0$)

e volte para o Passo 3.

3 – Testes Computacionais

Foram realizados testes computacionais para avaliar estes algoritmos e compará-los com os da literatura. O modelo 4 de Yanasse e Morabito (2003) foi utilizado para comparação. Este modelo é baseado na formulação não linear de Morabito e Arenales (2000) para padrões tabuleiros irrestritos. Segundo Yanasse e Morabito (2003), o modelo 4 foi o que apresentou o melhor desempenho dentre os modelos lineares para padrões tabuleiros restritos testados pelos autores.

Os algoritmos foram implementados em linguagem C++. Uma grande dificuldade do algoritmo enumerativo é a geração das K-melhores soluções do Problema da Mochila. No algoritmo de Yanasse, Soma e Maculan (2000) que foi utilizado na implementação computacional, é construída uma matriz de dimensão $m \times LW$ para enumerar as possíveis combinações de itens que devem ser recuperadas. O algoritmo, portanto, apresenta um requisito de memória na ordem de $O(LWm)$. Em problemas práticos, onde as dimensões dos objetos pode facilmente ultrapassar centenas de unidades e onde existem dezenas de tipos de itens de tamanhos relativamente pequenos a serem cortados, o problema de se manipular matrizes grandes pode tornar-se crítico. Uma solução encontrada para resolver este problema foi realizar uma mudança de escala nos dados de entrada do problema. Neste caso, as áreas dos itens e objeto são divididas por um fator de escala, calculado pela menor área dentre os itens considerados.

Os testes foram realizados num microcomputador Pentium III de 1GHz com 128MB RAM rodando o sistema operacional Linux. O equipamento utilizado possui configurações similares às do equipamento utilizado por Yanasse e Morabito (2003) nos testes computacionais realizados por estes autores.

Nos testes computacionais foram utilizados dois conjuntos de instâncias extraídas da literatura. O primeiro conjunto de instâncias reúne dados de Yanasse e Morabito (2003), Vianna *et al.* (2002), Christofides e Whitlock (1977), Wang (1983) e Oliveira e Ferreira (1990). O segundo conjunto de instâncias foi extraído de Hifi (2005).

A Tabela 1 apresenta os resultados obtidos para o primeiro conjunto de instâncias. Os tempos computacionais estão expressos em segundos:

TABELA 1 – ALGORITMO ENUMERATIVO X ALGORITMO DE ENUMERAÇÃO IMPLÍCITA X MODELO DE YANASSE E MORABITO (2003) (INSTÂNCIAS EXTRAÍDAS DE YANASSE E MORABITO, 2003; VIANNA ET AL., 2002; CHRISTOFIDES E WHITLOCK, 1977; WANG, 1983; OLIVEIRA E FERREIRA, 1990)

Instância	M	Valor da solução ótima	Tempo Algoritmo Enumerativo	Tempo Algoritmo de Enumeração Implícita	Tempo Modelo de Yanasse e Morabito (2003)
1	5	7800*	0.40	0.15	0.32
2	5	8280*	0.17	0.07	0.35
3	5	7000*	0.14	0.06	0.29
4	5	7566	0.08	0.04	0.39
5	5	7936*	0.07	0.05	0.35
6	5	9306	0.03	0.01	0.29
7	5	4770	0.04	0.03	0.55
8	5	7904*	0.03	0.01	0.35
9	5	7800*	0.03	0.01	0.26
10	5	5810	0.01	0.01	0.35
11	5	63*	0.002	0.002	0.25
12	5	90	0.005	0.004	0.26
13	5	70	0.001	0.001	0.29
14	5	80	0.001	0.002	0.23
15	5	90	0.012	0.022	0.26
16	5	78	0.0006	0.001	0.29
17	5	100*	0.0002	0.004	0.14
18	2	82	0.002	0.002	0.28
19	5	100*	0.003	0.008	0.16
20	5	70	0.002	0.001	0.28
VAG	5	6450	0.04	0.015	0.34
MA	13	6715500	**	147.52	9.99
CW1	7	240	0.009	0.009	0.19
CW2	10	1687	1.28	0.041	0.38
CW3	20	1080*	1.44	0.034	1.16
W	20	2277*	1.45	0.021	1.96
OF1	10	1769	0.33	0.013	0.47
OF2	10	1684	0.21	0.016	0.43

* Solução homogênea, ** não encontrou uma solução no limite de tempo fixado em 3600 segundos.

Da Tabela 1, observamos que apenas na instância MA (de Morabito e Arenales, 2000) não se determinou uma solução pelo algoritmo enumerativo no limite de tempo fixado em 3600 segundos. O algoritmo enumerativo foi mais rápido do que o modelo 4 de Yanasse e Morabito (2003) em 13 das 18 instâncias avaliadas. O algoritmo de enumeração implícita foi mais rápido em todas as instâncias testadas, exceto na instância MA. A instância MA apresenta itens com grande demanda, portanto sem limitações de itens no padrão. A existência de vários itens relativamente pequenos em comparação ao tamanho do objeto também contribuiu para dificultar a resolução desta instância por métodos de

enumeração. O tamanho relativo, a frequência e o número de diferentes tipos de itens são alguns dos fatores que influenciam no aumento ou redução de combinações possíveis que precisam ser investigadas para se determinar o melhor padrão tabuleiro.

A Tabela 2 apresenta os resultados obtidos para o segundo conjunto de instâncias:

TABELA 2 – ALGORITMO ENUMERATIVO X ALGORITMO DE ENUMERAÇÃO IMPLÍCITA
(INSTÂNCIAS EXTRAÍDAS DE HIFI, 2005)

Instância	M	Valor da solução ótima	Tempo Algoritmo Enumerativo	Tempo Algoritmo de Enumeração Implícita
HH	5	7020*	0.02	0.02
2	10	1687	1.28	0.04
3	20	1080*	1.44	0.04
A1	20	1040*	2.20	0.06
A2	20	1560	11.88	0.09
STS2	30	3800	**	0.65
CHL1	30	6579	**	2.43
CHL2	10	1226	0.28	0.02
CW1	25	5358	922.45	0.23
CW2	35	2964*	526.98	0.19
CW3	40	4428*	**	0.28
Hchl2	35	7408	**	21.45
Hchl9	35	4250	**	2.06
2s	10	1689	1.26	0.07
3s	20	2277*	1.37	0.03
A1s	20	1750*	2.20	0.05
A2s	20	2714	11.85	0.07
STS2s	30	3952	**	0.62
STS4s	20	7614	**	1.83
CHL1s	30	12276	**	2.43
CHL2s	10	1922	0.27	0.02
A3	20	4524	115.26	0.29
A4	20	3180	195.24	0.32
A5	20	12276	**	0.88
CHL5	10	220	0.099	0.04
CHL6	30	13113	**	7.26
CHL7	35	12958	**	25.61
CU1	25	12200	381.349	0.25
CU2	35	20250	1205.92	0.36
Hchl3s	10	7388	352.01	3.10
Hchl4s	10	7388	13.37	1.06
Hchl6s	22	40835	**	4.93
Hchl7s	40	41869	**	37.40
Hchl8s	10	416	0.68	0.06

* Solução homogênea, ** não encontrou uma solução no limite de tempo fixado em 3600 segundos.

Da Tabela 2, observa-se que o algoritmo enumerativo falhou na determinação do padrão tabuleiro restrito, no tempo limite de 1 hora, em 13 das 34 instâncias testadas. A eficiência do algoritmo enumerativo na determinação dos padrões está relacionada com algumas características dos dados de entrada do problema. O algoritmo de enumeração implícita conseguiu determinar todos os padrões destas instâncias no limite de tempo fixado em 3600 segundos sendo que o tempo médio para determinação destes padrões ficou abaixo de 1 minuto e foi sempre mais rápido do que o algoritmo enumerativo.

Yanasse e Morabito (2003) não testaram as instâncias da Tabela 2. Prevê-se alguma dificuldade em se resolver as instâncias maiores desta Tabela com o modelo deles devido ao grande número de variáveis de decisão.

Observamos que é possível melhorar o limitante superior \bar{G} calculado no Passo 4 do algoritmo de enumeração implícita se considerarmos apenas a inclusão dos itens restantes que ainda podem ser alocados integralmente na solução. Assim, as seguintes modificações nos Passos 1 e 4 são sugeridas:

Modificação no Passo 1: os itens passam a ser ordenados também em função de suas áreas.

Para cada tipo de item i , defina $v_i = \frac{\pi_i}{w_i l_i}$ ($i = 1, \dots, M$) e reordene os itens, tal que $v_1 \geq v_2 \geq \dots \geq v_M$

e, tal que $w_i l_i \geq w_{i+1} l_{i+1} \geq \dots \geq w_{i+s} l_{i+s}$, se $v_i = v_{i+1} = \dots = v_{i+s}$ (para algum i ou s , inteiros não negativos).

Modificação no Passo 4: novo limitante superior é calculado a partir da determinação (ou não) de um valor positivo inteiro para o índice j .

Se j for indeterminado, então faça:

Se $(w_{k+1} l_{k+1} d_{k+1} > \bar{WL})$ então faça

Início

Determine o menor índice r ($r = k, \dots, M - 1$), tal que $\left\lfloor \frac{\bar{WL}}{w_{r+1} l_{r+1}} \right\rfloor > 0$;

Se r for indeterminado, então faça $\bar{G} = \pi_1 a_1 + \dots + \pi_k (a_k - 1)$;

Senão faça

$$\bar{G} = \pi_1 a_1 + \dots + \pi_k (a_k - 1) + \pi_{r+1} \left\lfloor \frac{\bar{WL}}{w_{r+1} l_{r+1}} \right\rfloor + \pi_{r+2} \left(\frac{\bar{WL} - \left\lfloor \frac{\bar{WL}}{w_{r+1} l_{r+1}} \right\rfloor w_{r+1} l_{r+1}}{w_{r+2} l_{r+2}} \right);$$

Fim;

Senão faça $\bar{G} = \pi_1 a_1 + \dots + \pi_k (a_k - 1) + \pi_{k+1} d_{k+1} + \dots + \pi_M d_M$;

Senão faça:

Início

$a_i = d_i, i = k + 1, \dots, j - 1$;

$$a_j = \left\lfloor \frac{\bar{WL} - \sum_{i=k+1}^{j-1} w_i l_i a_i}{w_j l_j} \right\rfloor;$$

Determine o menor índice r ($r = j, \dots, M - 1$), tal que $\left\lfloor \frac{\bar{WL} - \sum_{i=k+1}^j w_i l_i a_i}{w_{r+1} l_{r+1}} \right\rfloor > 0$;

Se r for indeterminado, então faça $\bar{G} = \pi_1 a_1 + \dots + \pi_k (a_k - 1) + \pi_{k+1} a_{k+1} + \dots + \pi_j a_j$;

Senão faça

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

$$+ \pi_{r+1} \left[\frac{\bar{WL} - \sum_{i=k+1}^j w_i l_i a_i}{w_{r+1} l_{r+1}} \right] + \pi_{r+2} \left[\frac{\bar{WL} - \sum_{i=k+1}^j w_i l_i a_i - \left[\frac{\bar{WL} - \sum_{i=k+1}^j w_i l_i a_i}{w_{r+1} l_{r+1}} \right] w_{r+1} l_{r+1}}{w_{r+2} l_{r+2}} \right];$$

Fim.

Os tempos computacionais obtidos com esta versão refinada do algoritmo de enumeração implícita para o primeiro conjunto de instâncias estão apresentados na Tabela 3:

TABELA 3 – VERSÃO REFINADA DO ALGORITMO DE ENUMERAÇÃO IMPLÍCITA (INSTÂNCIAS EXTRAÍDAS DE YANASSE E MORABITO, 2003; VIANNA ET AL, 2002; CHRISTOFIDES E WHITLOCK, 1977; WANG, 1983; OLIVEIRA E FERREIRA, 1990)

Instância	M	Valor da solução ótima	Tempo Algoritmo de Enumeração Implícita	Tempo Versão Refinada do Algoritmo de Enumeração Implícita
1	5	7800*	0.15	0.12
2	5	8280*	0.07	0.05
3	5	7000*	0.06	0.04
4	5	7566	0.04	0.02
5	5	7936*	0.05	0.02
6	5	9306	0.01	0.01
7	5	4770	0.03	0.02
8	5	7904*	0.01	0.01
9	5	7800*	0.01	0.01
10	5	5810	0.01	0.01
11	5	63*	0.002	0.001
12	5	90	0.004	0.003
13	5	70	0.001	0.001
14	5	80	0.002	0.002
15	5	90	0.022	0.014
16	5	78	0.001	0.001
17	5	100*	0.004	0.001
18	2	82	0.002	0.002
19	5	100*	0.008	0.001
20	5	70	0.001	0.001
VAG	5	6450	0.015	0.014
MA	13	6715500	147.52	140.15
CW1	7	240	0.009	0.008
CW2	10	1687	0.041	0.034
CW3	20	1080*	0.034	0.034
W	20	2277*	0.021	0.016
OF1	10	1769	0.013	0.008
OF2	10	1684	0.016	0.011

* Solução homogênea.

Da Tabela 3, com o refinamento sugerido, observa-se uma redução do tempo de processamento em todas as instâncias testadas. Na Tabela 4 mostra-se os resultados obtidos pela versão refinada para o segundo conjunto de instâncias:

TABELA 4 – VERSÃO REFINADA DO ALGORITMO DE ENUMERAÇÃO IMPLÍCITA (INSTÂNCIAS EXTRAÍDAS DE HIFI, 2005)

Instância	M	Valor da solução ótima	Tempo Algoritmo de Enumeração Implícita	Tempo Versão Refinada do Algoritmo de Enumeração Implícita
HH	5	7020*	0.02	0.01
2	10	1687	0.04	0.03
3	20	1080*	0.04	0.03
A1	20	1040*	0.06	0.06
A2	20	1560	0.09	0.08
STS2	30	3800	0.65	0.63
CHL1	30	6579	2.43	2.40
CHL2	10	1226	0.02	0.02
CW1	25	5358	0.23	0.21
CW2	35	2964*	0.19	0.18
CW3	40	4428*	0.28	0.25
Hchl2	35	7408	21.45	21.43
Hchl9	35	4250	2.06	2.05
2s	10	1689	0.07	0.03
3s	20	2277*	0.03	0.01
A1s	20	1750*	0.05	0.03
A2s	20	2714	0.07	0.04
STS2s	30	3952	0.62	0.46
STS4s	20	7614	1.83	0.87
CHL1s	30	12276	2.43	1.19
CHL2s	10	1922	0.02	0.01
A3	20	4524	0.29	0.13
A4	20	3180	0.32	0.14
A5	20	12276	0.88	0.26
CHL5	10	220	0.04	0.01
CHL6	30	13113	7.26	3.92
CHL7	35	12958	25.61	14.57
CU1	25	12200	0.25	0.13
CU2	35	20250	0.36	0.25
Hchl3s	10	7388	3.10	1.01
Hchl4s	10	7388	1.06	0.21
Hchl6s	22	40835	4.93	3.46
Hchl7s	40	41869	37.40	18.71
Hchl8s	10	416	0.06	0.03

* Solução homogênea.

A redução no tempo de processamento também é observada na Tabela 4 em todas as instâncias avaliadas sendo que, em alguns casos, esta redução chegou a mais de 50% (p. ex: instâncias STS4s, CHL1s, Hchl3s, Hchl4s, Hchl7s).

4 – Considerações Finais

Neste trabalho foram testados dois algoritmos para a determinação de padrões tabuleiros exatos e restritos. Estes métodos utilizam o algoritmo construtivo para geração de padrões tabuleiros exatos de Katsurayama e Yanasse (2004) e são baseados em algoritmos para resolução do problema da

mochila unidimensional encontrados na literatura. O primeiro algoritmo baseia-se no algoritmo enumerativo de Yanasse, Soma e Maculan (2000) para determinação das K-melhores soluções para o problema da mochila unidimensional. O segundo algoritmo baseia-se no método da enumeração implícita de Gilmore e Gomory (1963) para resolução do problema da mochila unidimensional irrestrito.

Testes computacionais realizados com instâncias extraídas da literatura mostraram que os dois algoritmos têm desempenhos em termos de tempos de processamento, em média, melhores do que os indicados na literatura utilizando-se o modelo de Yanasse e Morabito (2003). Dentre os dois algoritmos testados, o de enumeração implícita apresentou o melhor desempenho.

O algoritmo de enumeração implícita tem a vantagem de permitir obter uma solução para o problema mesmo que sua execução seja interrompida, ao contrário do que acontece com o algoritmo enumerativo que só obtém uma solução ao final de sua execução. Um refinamento proposto no limitante superior do algoritmo de enumeração implícita possibilitou uma pequena redução do tempo computacional em uma boa parte das instâncias utilizadas nos testes computacionais. Acredita-se que outros refinamentos possam ser introduzidos para se tentar reduzir ainda mais este tempo. Por exemplo, pode-se tentar levar em consideração também restrições com respeito às larguras e comprimentos dos itens, além de suas respectivas áreas, no problema da mochila utilizado, para limitar ainda mais a procura.

Os algoritmos testados neste trabalho apenas determinam padrões tabuleiros exatos. Infelizmente, uma extensão deles para o caso da determinação de padrões tabuleiros não exatos não parece ser trivial. Os algoritmos geram também padrões tabuleiros irrestritos. No entanto, seus desempenhos são inferiores, em termos do tempo de processamento, aos algoritmos desenvolvidos especificamente para a determinação de padrões tabuleiros irrestritos (vide, por exemplo: Morabito e Arenales, 2000; Katsurayama e Yanasse 1999, 2000). Na prática, podemos utilizar os dois algoritmos de maneira combinada. Inicialmente o algoritmo para o problema irrestrito é aplicado. Se o padrão gerado for viável, o problema está resolvido. Se for inviável, utilizamos o seu valor como limitante superior para o problema restrito e aplicamos o algoritmo de geração do padrão restrito.

5 – Reconhecimento

Este trabalho foi parcialmente financiado pelo CNPq (Conselho Nacional de Pesquisa e Desenvolvimento Científico e Tecnológico) e pela FAPESP (Fundação de Amparo a Pesquisa do Estado de São Paulo).

6 – Referências Bibliográficas

- Belluzzo, L. Otimização nos planos de corte de chapas de fibra de madeira reconstituída: Um estudo de caso. Dissertação de Mestrado em Engenharia de Produção. Universidade Federal de São Carlos – São Carlos (SP), 2002.
- Christofides, N.; Whitlock, C. An algorithm for two-dimensional cutting problems. **Operations Research**, v.25, n.1, p.30-44, 1977.
- Gilmore, P.; Gomory, R. A linear programming approach to the cutting-stock problem II. **Operations Research**, v.11, n.6, p.863-888, 1963.
- Gilmore, P.; Gomory, R. Multistage cutting stock problems of two and more dimensions. **Operations Research**, v.13, n.1, p.94-120, 1965.
- Hifi, Mhand. <ftp://cermse.univ-paris1.fr/pub/CERMSEM/hifi/2Dcutting>. 10/03/2005. 2005.

- Katsurayama, D. M.; Yanasse, H. H. Um algoritmo enumerativo para determinação de padrões tabuleiros. [CDROM]. In: Simpósio Brasileiro de Pesquisa Operacional, 31., Juiz de Fora (MG), 1999. Anais. Tec Art Editora. Seção de Corte e Empacotamento, 1999.
- Katsurayama, D. M.; Yanasse, H. H. Um algoritmo enumerativo para determinação de padrões tabuleiros: aspectos computacionais de implementação. [CDROM]. In: Simpósio Brasileiro de Pesquisa Operacional, 32., Viçosa (MG). Anais. Tec Art Editora. Seção de Corte e Empacotamento, 2000.
- Katsurayama, D. M.; Yanasse, H. H. Um algoritmo para geração de padrões tabuleiros exatos a partir de uma combinação dada de itens. [CDROM]. In: Simpósio Brasileiro de Pesquisa Operacional, 36., São João Del-Rei (MG), 2004a.
- Katsurayama, D. M.; Yanasse, H. H. Um algoritmo enumerativo para determinação de padrões tabuleiros exatos e restritos. IV Workshop do Curso de Computação Aplicada. INPE. São José dos Campos (SP), 2004b.
- Morabito, R.; Arenales, M. N. Optimizing the cutting of stock plates in a furniture company. **International Journal of Production Research**, v.38, n.12, p.2725-2742, 2000.
- Oliveira, J. F.; Ferreira, J. S. An improved version of Wang's algorithm for two-dimensional cutting problems. **European Journal of Operational Research**, v.44, p.256-266, 1990.
- Scheithauer, G. On a two-dimensional guilhotine cutting problem. IFORS 2002. Edinburgh, Scotland, U. K. Cutting and Packing Workshop, 2002.
- Yanasse, H. H.; Katsurayama, D. M. Um algoritmo de enumeração implícita para geração de padrões tabuleiros exatos e restritos. SPOLM 2004 – VII Simpósio de Pesquisa Operacional e Logística da Marinha. Rio de Janeiro (RJ), 2004.
- Yanasse, H. H.; Soma, N. Y.; Maculan N. An algorithm for determining the k-best solutions of the one-dimensional knapsack problem. **Pesquisa Operacional**, v.20, n.1, p.117-134, 2000.
- Yanasse, H. H.; Morabito, R. Linear Models for two-stage constrained two-dimensional guillotine cutting problems. Working paper, INPE/UFSCAR (submetido para publicação), 2003.
- Wang, P. Two algorithms for constrained two-dimensional cutting stock problems. **Operations Research**, v.31, p.573-586. 1983.