



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA  
**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**

**INPE-14646-TDI/1206**

**ALGORITMOS ENUMERATIVOS PARA GERAÇÃO DE  
PADRÕES TABULEIROS**

Daniel Massaru Katsurayama

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada, orientada pelo Dr. Horácio Hideki Yanasse, aprovada em 14 de junho de 2006.

Publicado por:

**esta página é responsabilidade do SID**

Instituto Nacional de Pesquisas Espaciais (INPE)

Gabinete do Diretor – (GB)

Serviço de Informação e Documentação (SID)

Caixa Postal 515 – CEP 12.245-970

São José dos Campos – SP – Brasil

Tel.: (012) 3945-6911

Fax: (012) 3945-6919

E-mail: [pubtc@sid.inpe.br](mailto:pubtc@sid.inpe.br)

**Solicita-se intercâmbio  
We ask for exchange**

**Publicação Externa – É permitida sua reprodução para interessados.**



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA  
**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**

**INPE-14646-TDI/1206**

**ALGORITMOS ENUMERATIVOS PARA GERAÇÃO DE  
PADRÕES TABULEIROS**

Daniel Massaru Katsurayama

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada, orientada pelo Dr. Horácio Hideki Yanasse, aprovada em 14 de junho de 2006.

0 519.83

Katsurayama, D. M.

Algoritmos enumerativos para geração de padrões tabuleiros  
/ Daniel Massaru Katsurayama. – São José dos Campos: INPE,  
2006.

190p. ; (INPE-14646-TDI/12060

1.Corte. 2.Produtividade. 3.Algoritmos. 4.Métodos  
heurísticos. 5.Pesquisa operacional. 6. Otimização. 7.Custo de  
operação. 8.Custo de produção. I.Título.

Aprovado (a) pela Banca Examinadora  
em cumprimento ao requisito exigido para  
obtenção do Título de **Doutor(a)** em  
**Computação Aplicada**

Dr. Fernando Manuel Ramos



---

Presidente / INPE / SJC Campos - SP

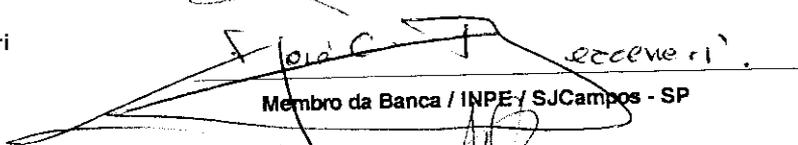
Dr. Horácio Hideki Yanasse



---

Orientador(a) / INPE / SJC Campos - SP

Dr. José Carlos Becceneri



---

Membro da Banca / INPE / SJC Campos - SP

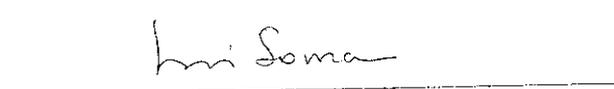
Dr. Reinaldo Morábito Neto



---

Convidado(a) / UFSCAR / São Carlos - SP

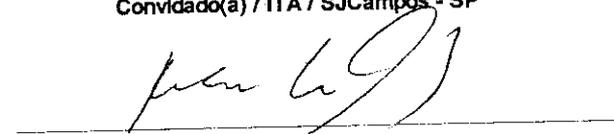
Dr. Nei Yoshihiro Soma



---

Convidado(a) / ITA / SJC Campos - SP

Dr. Marcos Nereu Arenales



---

Convidado(a) / USP-São Carlos / São Carlos - SP

Aluno (a): **Daniel Massaru Katsurayama**

São José dos Campos, 14 de junho de 2006



*“As maiores verdades geralmente são as mais simples”.*

CESARE CATÚ (escritor)



*A minha namorada,  
Nanci.*



## **AGRADECIMENTOS**

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq – Processo 142109/2001-0), pelo auxílio financeiro recebido.

Ao Instituto Nacional de Pesquisas Espaciais (INPE), pela oportunidade de realização deste trabalho e pelos recursos disponibilizados.

Ao meu orientador Prof. Dr. Horacio Hideki Yanasse, pela valiosa orientação e apoio recebidos durante a realização deste trabalho.

Aos docentes da CAP, pelo conhecimento adquirido.

Aos colegas de curso, pela amizade e companheirismo.

A meus familiares, em especial, à minha irmã Sonia Katsurayama dos Santos Pinto e a meu cunhado João Roberto dos Santos Pinto, pelo incentivo e apoio recebidos nos momentos difíceis.

À minha namorada Nanci Naomi Arai, pelo amor e compreensão.



## **RESUMO**

Nesta tese de doutorado focaliza-se o problema da determinação de padrões de corte simples, conhecidos como padrões tabuleiros, ou padrões 1-grupo. São propostos novos métodos exatos e heurísticas para a determinação de padrões tabuleiros restritos e irrestritos. Uma nova heurística para determinação de padrões tabuleiros irrestritos é apresentada. A nova heurística combina mais de uma faixa para compor o melhor padrão, gerando uma maior variedade de padrões do que uma heurística proposta anteriormente na literatura. Propõem-se três métodos exatos para determinação de padrões tabuleiros exatos e restritos. O primeiro método 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 método baseia-se no método da enumeração implícita de Gilmore e Gomory (1963) para resolução do problema da mochila unidimensional irrestrito. O terceiro método é uma versão melhorada do segundo método exato proposto. Os resultados dos testes computacionais realizados indicam que o terceiro método é mais eficiente do que os demais da literatura, em termos de tempos computacionais.



# ENUMERATIVE ALGORITHMS TO GENERATE CHECKERBOARD PATTERNS

## ABSTRACT

We focus on the problem of determining special cutting patterns known as checkerboard patterns or 1-group patterns. We propose new exact methods and heuristics for determining constrained and unconstrained checkerboard patterns. A new heuristic for determining unconstrained checkerboard pattern is presented. The new heuristic combines more than one strip to compose the best pattern, generating a larger variety of patterns than the heuristics proposed previously in the literature. Three exact methods for determining constrained exact checkerboard patterns are proposed. 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. The third one is an improved version of second method. The results obtained from the computational tests indicate that the third method outperforms previous methods of the literature in terms of execution times.



## SUMÁRIO

Pág.

### LISTA DE FIGURAS

### LISTA DE TABELAS

### LISTA DE SÍMBOLOS

### LISTA DE SIGLAS E ABREVIATURAS

<b>CAPÍTULO 1 – INTRODUÇÃO .....</b>	<b>37</b>
<b>CAPÍTULO 2 – REVISÃO BIBLIOGRÁFICA.....</b>	<b>47</b>
2.1 – Redução do Tempo de Corte com Foco na Programação da Produção.....	47
2.2 – Redução do Tempo de Corte com Foco nos Padrões de Corte .....	54
<b>CAPÍTULO 3 – ALGORITMOS PARA DETERMINAÇÃO DE PADRÕES TABULEIROS.....</b>	<b>79</b>
3.1 – Uma Nova Heurística para Determinação de Padrões Tabuleiros Irrestritos.....	80
3.2 – Um Algoritmo Construtivo para Geração de Padrões Tabuleiros Exatos a Partir de uma Combinação Dada de Itens .....	96
3.3 – Métodos Exatos para Determinação de Padrões Tabuleiros Exatos e Restritos .....	106
3.3.1 – Algoritmo Enumerativo.....	107
3.3.2 – Algoritmo de Enumeração Implícita .....	117
3.3.3 – Um Algoritmo Melhorado de Enumeração Implícita.....	134
3.3.4 – Uma Modificação para o Algoritmo Melhorado de Enumeração Implícita.....	170
<b>CAPÍTULO 4 – CONCLUSÃO .....</b>	<b>173</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>179</b>



## LISTA DE FIGURAS

	<u>Pág.</u>
1.1 Exemplo de problema de corte de estoque extraído da indústria de alumínio.....	37
1.2 Exemplos de padrões de corte homogêneos para o exemplo extraído da indústria de alumínio.....	39
1.3 Padrão de corte guilhotinado e não guilhotinado .....	42
1.4 Padrão de corte guilhotinado ortogonal e não ortogonal.....	42
1.5 Padrão de corte guilhotinado ortogonal 2 estágios exato e não exato.....	43
1.6 Exemplos de padrões tabuleiros .....	44
2.1 Preparo do equipamento para corte de 3 diferentes tipos de padrões .....	47
2.2 Corte de pacotes .....	51
2.3 Corte de chapas retangulares de fibra de madeira reconstituída .....	54
2.4 Exemplo de obtenção de padrão homogêneo: repetição do item A ao longo da largura e comprimento do objeto.....	55
2.5 Corte de um padrão tabuleiro .....	56
2.6 Um exemplo de padrão tabuleiro .....	63
2.7 Determinação de um padrão 2-estágios não exato através do método de Gilmore e Gomory (1965) .....	66
2.8 Padrão de corte obtido através da heurística para determinação de padrões 3-estágios proposta por Morabito e Arenales (2000).....	67
2.9 Determinação de um padrão $q$ -estágios utilizando grafos “e/ou” .....	68
2.10 Exemplo de padrão tabuleiro composto .....	69
2.11 Sobras do padrão .....	70
2.12 Padrões T homogêneos .....	72
2.13 Problema do corte guilhotinado não ortogonal de polígonos.....	76

2.14	Corte de polígonos: caso onde os cortes efetuados não precisam ser necessariamente paralelos aos lados do polígono a ser produzido.....	77
3.1	Obtenção de um padrão tabuleiro não exato através da nova heurística proposta .....	82
3.2	Número de objetos cortados e perda de material em função do custo fixo .....	94
3.3	Solução de custo mínimo.....	95
3.4	Proporção de padrões não tabuleiros na solução em função do custo fixo .....	96
3.5	Construção da solução inicial.....	98
3.6	Corte de uma nova faixa vertical.....	99
3.7	Corte de uma nova faixa horizontal.....	100
3.8	Solução parcial após a primeira fase .....	102
3.9	Corte da nova faixa vertical de largura 35.....	103
3.10	Corte da faixa horizontal de largura 20 .....	103
3.11	Corte da faixa horizontal de largura 40 .....	104
3.12	Corte da faixa vertical de largura 25 .....	105
3.13	Corte da faixa vertical de largura 30 .....	105
3.14	Padrão tabuleiro exato final.....	106
3.15	Representação das combinações $\lambda$ para 3 tipos de itens numa árvore de enumeração .....	120
3.16	Padrão tabuleiro exato para o exemplo da geração do retângulo cerne .....	135
3.17	Inclusão do primeiro item A na área do objeto .....	135
3.18	Inclusão do primeiro item tipo B.....	136
3.19	Inclusão do segundo item tipo B e preenchimento da área correspondente ao item A .....	136
3.20	Inclusão do primeiro item tipo C.....	137

3.21 Exemplo de retângulo cerne .....	138
3.22 Escolha do item $r$ para inclusão nos bins acima, à direita e à nordeste do retângulo cerne para cálculo do limitante superior.....	140
3.23 Padrão tabuleiro exato $P^1$ obtido com a inclusão do item 1 .....	148
3.24 Retângulo cerne $Q^1$ associado ao padrão tabuleiro exato da Figura 3.23 .....	149
3.25 Padrão tabuleiro exato $P^2$ obtido com a inclusão do item 2 no nó 2 .....	151
3.26 Retângulo cerne $Q^2$ associado ao padrão tabuleiro exato da Figura 3.25 .....	151
3.27 Padrão tabuleiro exato $P^2$ e seu retângulo cerne $Q^2$ correspondente, obtidos com a remoção de 1 item do tipo 2 no nó 2.....	152
3.28 Padrão tabuleiro exato $P^3$ obtido a partir da remoção do segundo item do tipo 2 no nó 1 e inclusão do item tipo 3 no nó 3 .....	153
3.29 Padrão tabuleiro exato $P^1$ e retângulo cerne $Q^1$ correspondente, obtidos a partir da remoção de 1 item do tipo 1, no nó 1 .....	154
3.30 Padrão tabuleiro exato $P^2$ obtido a partir da remoção de 1 item do tipo 1 no nó 1 e inclusão do item 2 no nó 2.....	154
3.31 Padrão tabuleiro exato $P^2$ e seu retângulo cerne $Q^2$ correspondente, obtidos a partir da remoção de 1 item do tipo 2, no nó 2 .....	155
3.32 Padrão tabuleiro exato $P^3$ obtido com a remoção do segundo item do tipo 2 no nó 4 e inclusão do item 3 no nó 3.....	156
3.33 Padrão tabuleiro exato $P^2$ obtido com a inclusão do item 2 no nó 2 .....	156
3.34 Retângulo cerne $Q^2$ associado ao padrão tabuleiro $P^2$ da Figura 3.33 .....	157
3.35 Padrão tabuleiro exato $P^2$ e seu retângulo cerne $Q^2$ correspondente, obtidos a partir da remoção de 1 item do tipo 2, no nó 2 .....	158
3.36 Solução final: padrão tabuleiro exato obtido.....	158
3.37 Definição de uma nova faixa horizontal (ou vertical) a partir da inclusão do item $s$ acima (ou à direita) do retângulo cerne .....	171

3.38 Retângulo cerne aumentado, obtido a partir da definição de uma nova faixa horizontal (ou vertical).....	171
--	-----

## LISTA DE TABELAS

	<u>Pág.</u>
3.1 Itens a serem produzidos (exemplo para determinação de um padrão tabuleiro exato e irrestrito através da nova heurística) .....	84
3.2 Valor da solução e tempo de execução (em segundos) para a heurística de Morabito e Arenales e para a heurística proposta (determinação de padrões tabuleiros não exatos, instâncias OR-Library) .....	86
3.3 Valor da solução e tempo de execução (em segundos) para a heurística de Morabito e Arenales e para a heurística proposta (determinação de padrões tabuleiros exatos, instâncias OR-Library) .....	87
3.4 Valor da solução e tempo de execução (em segundos) para o algoritmo enumerativo de Katsurayama e Yanasse (1999, 2000), para a heurística de Morabito e Arenales e para a heurística proposta (determinação de padrões tabuleiros exatos, instâncias 1gcut) .....	89
3.5 Valor da solução e tempo de execução (em segundos) para a heurística de Morabito e Arenales e para a heurística proposta (determinação de padrões tabuleiros não exatos, instâncias 1gcut) .....	90
3.6 Itens a serem produzidos .....	93
3.7 Exemplo para geração de um padrão tabuleiro exato .....	101
3.8 Itens a serem produzidos (exemplo numérico para determinação de padrões tabuleiros exatos e restritos) .....	109
3.9 Tabela de enumeração gerada pelo algoritmo YSM para o exemplo numérico da Tabela 3.8 .....	109
3.10 Soluções a serem recuperadas da Tabela 3.9 (em ordem não crescente de valor).....	110
3.11 Resultado obtido pelo procedimento de recuperação das soluções .....	112
3.12 Algoritmo enumerativo x modelo de Yanasse e Morabito (2003) (instâncias extraídas de Yanasse e Morabito, 2003).....	113
3.13 Algoritmo enumerativo x modelo de Yanasse e Morabito (2003) (instâncias extraídas de Yanasse e Morabito, 2003; Vianna <i>et al.</i> , 2002; Christofides e Whitlock, 1977; Wang, 1983; Oliveira e Ferreira, 1990) .....	114

3.14	Algoritmo enumerativo: soluções obtidas (instâncias extraídas de Hifi, 2005) .....	116
3.15	Algoritmo enumerativo x algoritmo de enumeração implícita x modelo de Yanasse e Morabito (2003) (instâncias extraídas de Yanasse e Morabito, 2003; Vianna <i>et al.</i> , 2002; Christofides e Whitlock, 1977; Wang, 1983; Oliveira e Ferreira 1990) .....	125
3.16	Exemplo da fábrica de móveis (instância MA) .....	126
3.17	Algoritmo enumerativo x algoritmo de enumeração implícita (instâncias extraídas de Hifi, 2005) .....	127
3.18	Algoritmo de enumeração implícita x algoritmo de Katsurayama e Yanasse (1999, 2000) (instâncias OR-Library) .....	128
3.19	Versão modificada do algoritmo de enumeração implícita (instâncias extraídas de Yanasse e Morabito, 2003; Vianna <i>et al.</i> , 2002; Christofides e Whitlock, 1977; Wang, 1983; Oliveira e Ferreira, 1990) .....	131
3.20	Versão modificada do algoritmo de enumeração implícita (instâncias extraídas de Hifi, 2005) .....	133
3.21	Informações utilizadas pelo algoritmo melhorado de enumeração implícita e que é conveniente armazená-las em cada um dos nós da árvore de enumeração .....	143
3.22	Itens a serem produzidos (exemplo numérico para o algoritmo melhorado de enumeração implícita) .....	147
3.23	Itens a serem produzidos (em ordem não crescente pela relação valor de utilidade por unidade de área) .....	147
3.24	Resultados obtidos para o conjunto 1 de instâncias extraídas de Yanasse e Morabito (2005) com 5 tipos de itens diferentes e objeto de dimensão 100x100 .....	159
3.25	Resultados obtidos para o conjunto 2 de instâncias extraídas de Yanasse e Morabito (2005) com 5 tipos de itens diferentes e objeto de dimensão 10x10 .....	160
3.26	Resultados obtidos para o conjunto 3 de instâncias extraídas de Yanasse e Morabito (2005) com 10 tipos de itens diferentes e objeto de dimensão 100x100 .....	161

3.27 Resultados obtidos para o conjunto 4 de instâncias extraídas de Yanasse e Morabito (2005) com 20 tipos de itens diferentes e objeto de dimensão 100x100 .....	162
3.28 Resultados obtidos para o conjunto 5 de instâncias extraídas de Yanasse e Morabito (2005) com 50 tipos de itens diferentes e objeto de dimensão 100x100 .....	163
3.29 Resultados obtidos para o conjunto 6 de instâncias extraídas de Yanasse e Morabito (2005) com 100 tipos de itens diferentes e objeto de dimensão 100x100 .....	164
3.30 Tempos computacionais médios obtidos para o conjunto de instâncias extraídas de Yanasse Morabito (2005) com 5, 10, 20 50 e 100 tipos de itens diferentes.....	165
3.31 Resultados obtidos para o exemplo da fábrica de móveis extraída de Yanasse e Morabito (2005).....	165
3.32 Versão melhorada do algoritmo de enumeração implícita (instâncias extraídas de Hifi, 2005) .....	167
3.33 Versão melhorada do algoritmo de enumeração implícita (instâncias c1 gcut) .....	169
3.34 Tempos computacionais médios obtidos pelo algoritmo melhorado e pela versão modificada para o conjunto de instâncias extraídas de Yanasse e Morabito (2005) com 5, 10, 20, 50 e 100 tipos de itens diferentes .....	172



## LISTA DE SÍMBOLOS

### Latinos:

- $\tilde{a}_i$  área do item  $i$  ( $i=1, \dots, n$ );
- $a_{ij}$  quantidade de itens  $i$  ( $i=1, \dots, n$ ) no padrão  $j$  ( $j=1, \dots, m$ );
- $\tilde{a}^N$  área do item mais lucrativo que preenche o bin a nordeste do retângulo cerne;
- $\tilde{a}^R$  área do item mais lucrativo que preenche o bin a direita do retângulo cerne;
- $\tilde{a}^T$  área do item mais lucrativo que preenche o bin acima do retângulo cerne;
- $A$  padrão de corte;
- $b$  largura da faixa horizontal ( $b \in \Omega_w$ ) definida de forma iterativa pelo algoritmo construtivo para geração de padrões tabuleiros exatos;
- $\bar{b}$  lado direito (Algoritmo Enumerativo de Yanasse, Soma e Maculan, 2000 para determinação das K-melhores soluções para o problema unidimensional da Mochila);
- $B$  matriz básica do Método Simplex;
- $\tilde{B}$  área do bin;
- $\tilde{B}^N$  área do bin localizado à nordeste do retângulo cerne;
- $\tilde{B}^R$  área do bin localizado à direita do retângulo cerne;
- $\tilde{B}^T$  área do bin localizado acima do retângulo cerne;
- $c$  largura da faixa vertical ( $c \in \Omega_l$ ) definida de forma iterativa pelo algoritmo construtivo para geração de padrões tabuleiros exatos;

- $C$  comprimento do subpadrão (ou retângulo cerne);
- $d_i$  demanda (frequência) do item  $i$  ( $i=1, \dots, n$ ) no padrão;
- $D$  largura do subpadrão (ou retângulo cerne);
- $D_w$  conjunto de índices de todas as larguras  $w_j$  possíveis dos itens do problema;
- $e_{index}$  vetor cujos elementos são iguais a zero, exceto o elemento da posição  $index$  que é igual a 1;
- $f$  valor inteiro que gera um padrão tabuleiro exato, descrito por  $f = \max(j)$ ,  

$$j = 0, 1, 2, \dots, \min \left\{ \left\lfloor \frac{X}{w_i l_i} \right\rfloor, d_i \right\};$$
- $\overline{G}$  limitante superior do algoritmo de enumeração implícita para determinação de padrões tabuleiros exatos e restritos;
- $\underline{G}$  limitante inferior do algoritmo de enumeração implícita para determinação de padrões tabuleiros exatos e restritos;
- $G_0$  valor total dos itens presentes no cerne;
- $G_1$  valor estimado para os itens acima do cerne;
- $G_2$  valor estimado para os itens à direita do cerne;
- $G_3$  desconto sobre os valores  $G_1$  e  $G_2$ , relativo aos itens não incluídos no retângulo cerne e presentes no padrão;
- $G_4$  valor estimado para os itens à nordeste do retângulo cerne;
- $\underline{h}$  melhor padrão homogêneo restrito;
- $H^{nold}$  padrão tabuleiro exato que leva em conta o preenchimento dos espaços livres referentes aos itens que foram utilizados no preenchimento do retângulo cerne;

- index* tipo de item considerado no nó da árvore de enumeração;
- $I$  número de diferentes tipos de larguras  $w_i$  (modelo de Vianna *et al.* (2002) para padrões 2-estágios não exatos);
- $I1$  número de diferentes tipos de comprimentos  $l_i$  das peças (modelo de Yanasse e Morabito (2000) adaptado por Yanasse e Morabito (2003) para determinação de padrões tabuleiros restritos);
- $I2$  número de diferentes tipos de larguras  $w_k$  das peças (modelo de Yanasse e Morabito (2000) adaptado por Yanasse e Morabito (2003) para determinação de padrões tabuleiros restritos);
- $l_i$  comprimento do item  $i$  ( $i=1,\dots,n$ );
- $L$  comprimento do objeto;
- $L_j$  comprimento da faixa vertical  $j$ ,  $j=1,\dots,\gamma$  (modelo de Scheithauer (2002) para determinação de padrões tabuleiros);
- $L^N$  comprimento do bin localizado à nordeste do retângulo cerne;
- $L^R$  comprimento do bin localizado à direita do retângulo cerne;
- $L^T$  comprimento do bin localizado acima do retângulo cerne;
- $m$  número de diferentes tipos de padrões de corte;
- $M$  número suficientemente grande (modelo linearizado de Yanasse e Morabito (2003) para determinação de padrões tabuleiros exatos e restritos);
- $M_i$  número inteiro  $r$ , tal que  $w_r \leq w_i$ ,  $i=1,\dots,I$  (modelo de Vianna *et al.* (2002) para padrões 2-estágios não exatos);
- $n$  número de diferentes tipos de itens;

- $nold$  valor inteiro que identifica o nó na árvore de decisão;
- $p_{w_r \times l_s}$  valor do item que caiba no retângulo de dimensões  $w_r \times l_s$ . No caso da determinação de padrões tabuleiros exatos,  $p_{w_r \times l_s}$  equivale ao valor do item de dimensão exata  $w_r \times l_s$ . No caso da determinação de padrões tabuleiros não exatos,  $p_{w_r \times l_s}$  equivale ao valor do item mais lucrativo que possua dimensão inferior ou igual a  $w_r \times l_s$ . Caso não seja possível determinar o valor de  $p_{w_r \times l_s}$ , então considera-se  $p_{w_r \times l_s} = 0$ ;
- $P^{nold}$  padrão tabuleiro exato determinado com a inclusão do item  $idx$ , sem levar em conta o preenchimento dos espaços livres compatíveis com os itens viáveis que surgiram no interior do padrão;
- $Q^{nold}$  retângulo cerne associado ao padrão tabuleiro exato  $P^{nold}$ ;
- IR conjunto dos números reais;
- $\tilde{R}_r$  retângulo  $r$  de dimensão  $w_k \times l_i$  (modelo quadrático inteiro de Morabito e Arenales (2000) para determinação de padrões tabuleiros irrestritos);
- $R^{nold}$  retângulo cerne que leva em conta o preenchimento dos espaços livres compatíveis com os itens viáveis;
- $s_i$  valor inteiro, tal que  $2^{s_i-1} \leq \left\lfloor \frac{L}{l_j} \right\rfloor \leq 2^{s_i}$ ,  $i=1, \dots, I1$  (modelo linearizado de Yanasse e Morabito (2003) para determinação de padrões tabuleiros restritos);
- $S_i$  número máximo de padrões da faixa de largura  $w_i$  (modelo de Vianna *et al.* (2002) para padrões 2-estágios não exatos);
- $\tilde{S}_i$  conjunto dos índices de todos os comprimentos possíveis dos itens associados à faixa  $j$ ;

- $T_w^j$  conjunto de índices de todas as larguras  $w_j$  possíveis dos itens do problema, excluindo a largura da faixa  $j$ ;
- $u_i$  número de itens do tipo  $i$  ( $i=1, \dots, n$ ) que podem ser colocados no padrão e, ainda assim, o padrão puder ser gerado
- $\tilde{u}_i$  demanda restante do item tipo  $i$  ( $i=1, \dots, n$ );
- $\tilde{u}^{nold}$  demanda restante para os itens  $index+1, \dots, n$  que foram usados para preencher o retângulo cerne  $R^{nold}$ ;
- $v_i$  valor de utilidade (lucro) por unidade de área do item  $i$  ( $i=1, \dots, n$ );
- $v^T$  valor de utilidade por unidade de área correspondente ao item  $k+1, \dots, n$  mais lucrativo considerado nos bin acima do retângulo cerne;
- $v^R$  valor de utilidade por unidade de área correspondente ao item  $k+1, \dots, n$  mais lucrativo considerado nos bin à direita do retângulo cerne;
- $V_j$  valor da faixa  $j$ , obtida na primeira fase da heurística de Morabito e Arenales (2000) para determinação de padrões tabuleiros irrestritos;
- $V_r^j$  soma dos itens mais lucrativos que podem ser inseridos na faixa horizontal, cuja largura é dada por  $w_r$ , tal que um padrão tabuleiro seja determinado quando a faixa  $r$  for combinada com a faixa  $j$ ;
- $x_i$  quantidade de vezes que o padrão  $i$  ( $i=1, \dots, m$ ) deve ser cortado;
- $X$  espaço restante (sobra) na mochila (ou padrão). Inicialmente  $X=WL$ ;
- $\bar{X}$  espaço restante (sobra) na mochila (ou padrão) considerando a remoção de 1 item do tipo  $k$ ;
- $y$  valor que reduz  $f_{index}$  a um valor que defina um retângulo cerne diferente de  $R^{nold+1}$ ;

- $w_i$  largura do item  $i$  ( $i=1,\dots,n$ );
- $W$  largura do objeto;
- $W_k$  largura da faixa longitudinal  $k$ ,  $k=1,\dots,\delta$  (modelo de Scheithauer (2002) para determinação de padrões tabuleiros);
- $\tilde{W}_j$  conjunto dos diferentes tipos de larguras considerados na determinação de padrões tabuleiros exatos e não exatos, onde:
- $\tilde{W}_j = \{i \mid w_i = w_j\}$  para o caso exato e  $\tilde{W}_j = \{i \mid w_i \leq w_j\}$  para o caso não exato;
- $W^N$  largura do bin localizado à nordeste do retângulo cerne;
- $W^R$  largura do bin localizado à direita do retângulo cerne;
- $W^T$  largura do bin localizado acima do retângulo cerne;
- $Z$  valor da solução corrente obtida;

**Gregos:**

- $\alpha^{nold}$  combinação de itens associado ao padrão tabuleiro exato  $H^{nold}$ ;
- $\beta_s$  número de itens do tipo  $s$  presentes no retângulo cerne,  $s=1,\dots,k$ ;
- $\beta^{nold}$  combinação de itens associado ao retângulo cerne  $Q^{nold}$ ;
- $\gamma$  número de faixas verticais (modelo de Scheithauer (2002) para determinação de padrões tabuleiros);
- $\gamma^T$  parcela a ser descontada do limitante superior quando o item é inserido acima do retângulo cerne;
- $\gamma^R$  parcela a ser descontada do limitante superior quando o item é inserido à direita

do retângulo cerne;

$\delta$  número de faixas longitudinais (modelo de Scheithauer (2002) para determinação de padrões tabuleiros);

$\eta_{ijk}$  variável binária ( $\eta_{ijk} \in \{0,1\}$ ,  $i=1,\dots,n$ ,  $j=1,\dots,\gamma$ ,  $k=1,\dots,\delta$ ), em que  $\eta_{ijk} = 1$  se e somente se a peça do tipo  $i$  é obtida do retângulo  $W_k \times L_j$  e,  $\eta_{ijk} = 0$ , caso contrário (modelo de Scheithauer (2002) para determinação de padrões tabuleiros);

$\theta$  número de itens inseridos no espaço restante calculado por:

$$\theta = \min \left\{ \left\lfloor \frac{X}{w_i l_i} \right\rfloor, d_i \right\};$$

$\kappa_{rik}$  número de peças do tipo  $r$  contidas em todos os retângulos  $w_k \times l_i$  (modelo de Yanasse e Morabito (2000) adaptado por Yanasse e Morabito (2003) para determinação de padrões tabuleiros restritos);

$\lambda$  vetor que representa a combinação de itens presentes no padrão de corte;

$\underline{\lambda}$  vetor que representa a melhor combinação encontrada até o momento;

$\lambda_i$  número de itens do tipo  $i$  ( $i=1,\dots,n$ ) presentes no padrão de corte;

$\lambda^{nold}$  combinação de itens associado ao padrão tabuleiro exato  $P^{nold}$  determinado;

$\mu_k$  número de vezes que a largura  $w_k$  é cortada (modelo quadrático inteiro de Morabito e Arenales (2000) para determinação de padrões tabuleiros irrestritos);

$\pi$  vetor multiplicador Simplex (ou vetor de valor de utilidade dos itens);

$\pi_i$  valor de utilidade do item  $i$  ( $i=1,\dots,n$ );

$\pi^N$	valor de utilidade do item mais lucrativo que preenche o bin à nordeste do retângulo cerne;
$\pi^R$	valor de utilidade do item mais lucrativo que preenche o bin à direita do retângulo cerne;
$\pi^T$	valor de utilidade do item mais lucrativo que preenche o bin acima do retângulo cerne;
$\Pi$	custo do objeto cortado;
$\rho_{ij}^r$	número de vezes que a peça $r$ é cortada no padrão $j$ na faixa de largura $w_i$ (modelo de Vianna <i>et al.</i> (2002) para padrões 2-estágios não exatos);
$\sigma_{ij}$	número de vezes que o padrão $j$ ( $j = 1, \dots, S_i$ ) da faixa de largura $w_i$ ( $i = 1, \dots, I$ ) é repetido ao longo da largura $W$ da chapa (modelo de Vianna <i>et al.</i> (2002) para padrões 2-estágios não exatos);
$\tau_{is}$	variável binária, $\tau_{is} \in \{0,1\}$ , $i=1, \dots, I$ , $s = 1, \dots, S_i$ (modelo linearizado de Yanasse e Morabito (2003) para determinação de padrões tabuleiros restritos);
$\nu_i$	número de vezes que o comprimento $l_i$ é cortado (modelo quadrático inteiro de Morabito e Arenales (2000) para determinação de padrões tabuleiros irrestritos);
$\varphi^{nold}$	combinação de itens associado ao retângulo cerne $R^{nold}$ ;
$\Phi$	valor da solução obtida com a resolução do problema unidimensional da mochila na segunda fase da nova heurística para determinação de padrões tabuleiros irrestritos;
$\chi$	limitante para a perda calculada para o padrão tabuleiro final;
$\psi_k$	número de vezes que a faixa de largura $w_k$ ( $k \in D_w$ ) é usada quando

combinada com a faixa de largura  $w_j$  ;

$\omega_{iks}$  variável do modelo linearizado de Yanasse e Morabito (2003) para determinação de padrões tabuleiros restritos;

$\Omega_w$  conjunto dos diferentes tipos de larguras  $w_k$  ( $k = 1, \dots, n$ ) dos itens existentes na combinação  $\lambda$ ;

$\Omega_l$  conjunto dos diferentes tipos de comprimentos  $l_k$  ( $k = 1, \dots, n$ ) dos itens existentes na combinação  $\lambda$ .



## LISTA DE SIGLAS E ABREVIATURAS

- CECA - Algoritmo Enumerativo para determinação de padrões tabuleiros exatos e restritos
- FFD - Heurística First Fit Decreasing
- GGCECA - Algoritmo de Enumeração Implícita para determinação de padrões tabuleiros exatos e restritos
- NGGCECA - Algoritmo Melhorado de Enumeração Implícita para determinação de padrões tabuleiros exatos e restritos
- PCE - Problema de Corte de Estoque
- PRNP - Problema da Redução do Número de Diferentes Tipos de Padrões de Corte
- YM2005 - Modelo de Yanasse e Morabito (2005) para determinação de padrões tabuleiros exatos e restritos
- YSM - Algoritmo Enumerativo de Yanasse, Soma e Maculan (2000) para determinação das K-melhores soluções para o Problema da Mochila Unidimensional Irrestrito



## CAPÍTULO 1

### INTRODUÇÃO

Nos ambientes de produção de algumas indústrias depara-se freqüentemente com o problema de se cortar peças grandes, conhecidas como objetos, em partes menores, conhecidas como itens. Os itens são requeridos internamente por outros setores da empresa ou por clientes externos. *Problemas de corte* podem ser encontrados, por exemplo, na indústria de papel, de móveis, de alumínio, de vidro, metalúrgica, plástica, têxtil, etc. É importante que os cortes sejam planejados para minimizar os efeitos negativos, tais como o desperdício, que onera os custos de produção.

Considere, por exemplo, uma situação prática extraída da indústria de alumínio, em que uma empresa mantém em estoque, tubos de comprimento  $L$  que devem ser cortados em pedaços menores de comprimentos  $l_i$  para atender uma demanda a eles associados de  $d_i, i=1, \dots, n$ . Neste exemplo, os tubos a serem cortados são os *objetos* (vide Figura 1.1a) e as peças a serem produzidas são os *itens* (vide Figura 1.1b). Devido ao tamanho diferenciado dos itens, os objetos podem ser cortados de diferentes maneiras. Define-se como *padrão de corte*, a maneira como o objeto em estoque é cortado para produzir os itens menores (vide Figura 1.1c).

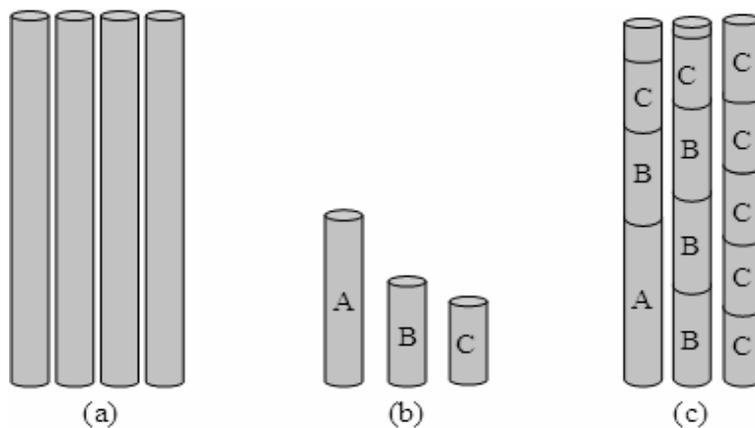


Figura 1.1 – Exemplo de problema de corte de estoque extraído da indústria de alumínio: (a) objetos em estoque, (b) itens demandados, (c) padrões de corte.

A situação prática descrita anteriormente, ilustra um exemplo de *problema de corte de estoque* (PCE). O PCE consiste basicamente em cortar os objetos para produzir os itens de maneira a atender a demanda associada a eles otimizando alguma função objetivo que pode ser, por exemplo, a maximização do lucro, a minimização da perda de material, etc.

Gilmore e Gomory (1961) formularam o PCE como um problema de programação linear inteira da seguinte forma:

$$\text{Minimizar } \sum_{j=1}^m \Pi x_j \quad (1.1.1)$$

$$\text{Sujeito a: } \sum_{j=1}^m A_j x_j \geq d, \quad (1.1.2)$$

$$x_j \geq 0 \text{ e inteiro } (j=1, \dots, m). \quad (1.1.3)$$

Em que:

$A_j$  ( $j = 1, \dots, m$ ), corresponde aos padrões de corte, ou seja,

$$A_1 = \begin{pmatrix} a_{1j} \\ a_{2j} \\ a_{3j} \\ \text{M} \\ a_{nj} \end{pmatrix}, A_2 = \begin{pmatrix} a_{12} \\ a_{22} \\ a_{32} \\ \text{M} \\ a_{n2} \end{pmatrix}, A_3 = \begin{pmatrix} a_{13} \\ a_{23} \\ a_{33} \\ \text{M} \\ a_{n3} \end{pmatrix}, \dots, A_m = \begin{pmatrix} a_{1m} \\ a_{2m} \\ a_{3m} \\ \text{M} \\ a_{nm} \end{pmatrix} \text{ e } a_{ij} \text{ é a quantidade de itens } i \text{ no}$$

padrão  $j$ , onde, no caso da indústria de alumínio anteriormente citado, os  $a_{ij}$ 's devem

respeitar à restrição  $\sum_{i=1}^n l_i a_{ij} \leq L$  ( $j=1, \dots, m$  e  $a_{1j}, a_{2j}, \dots, a_{nj}$  inteiros não negativos);

$x_j$  é a quantidade de vezes que o padrão  $A_j$  deve ser cortado;

$\Pi$  é o custo do objeto cortado;

$d$  é o vetor que representa a demanda associada a cada tipo de item, descrito por  $d = (d_1 \dots d_n)^T$ .

A função (1.1.1) é o custo de produção. A restrição (1.1.2) impõe que o total de itens cortados de cada tipo deve, pelo menos superar a sua demanda. A restrição (1.1.3) impõe a não negatividade e integralidade do número de vezes que cada padrão deverá ser cortado.

Devido às dificuldades em solucionar o problema (1.1.1-1.1.3), eles propuseram relaxar as condições de integralidade do problema original e resolvê-lo utilizando o método Simplex e um procedimento de geração de colunas, em que cada coluna representa um padrão de corte.

Observe que este problema, mesmo relaxado, apresenta dificuldades ao ser resolvido pelo método Simplex, pois o número de possíveis padrões, em casos práticos, pode crescer muito, atingindo valores extremamente elevados. O método proposto por Gilmore e Gomory (1961) consiste basicamente em gerar as colunas do problema à medida do necessário ao invés de gerá-las todas à priori.

Como solução básica inicial, foram escolhidos apenas padrões de corte homogêneos. Denomina-se *padrão de corte homogêneo*, o padrão formado por peças do mesmo tipo (vide Figura 1.2).

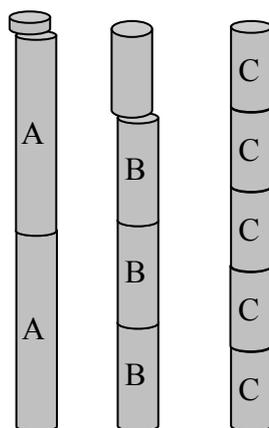


Figura 1.2 - Exemplos de padrões de corte homogêneos para o exemplo extraído da indústria de alumínio.

No caso considerado de corte de tubos, a base inicial composta de  $m$  padrões homogêneos seria:

$$B = \begin{pmatrix} \left\lfloor \frac{L}{l_1} \right\rfloor & 0 & K & 0 \\ 0 & \left\lfloor \frac{L}{l_2} \right\rfloor & K & 0 \\ M & M & O & M \\ 0 & 0 & 0 & \left\lfloor \frac{L}{l_m} \right\rfloor \end{pmatrix},$$

onde  $\lfloor Y \rfloor$  é a função maior (ou igual) inteiro de  $Y \in \mathbb{R}$ .

No caso de chapas retangulares de madeira de tamanhos  $W \times L$ , por exemplo, onde se deseja cortar peças retangulares menores de tamanhos  $w_i \times l_i$ , a base inicial composta de  $m$  padrões homogêneos seria:

$$B = \begin{pmatrix} \left\lfloor \frac{W}{w_1} \right\rfloor \times \left\lfloor \frac{L}{l_1} \right\rfloor & 0 & K & 0 \\ 0 & \left\lfloor \frac{W}{w_2} \right\rfloor \times \left\lfloor \frac{L}{l_2} \right\rfloor & K & 0 \\ M & M & O & M \\ 0 & 0 & 0 & \left\lfloor \frac{W}{w_m} \right\rfloor \times \left\lfloor \frac{L}{l_m} \right\rfloor \end{pmatrix}.$$

Seja  $\pi$ , o vetor multiplicador Simplex. Para a escolha da variável a entrar na base deve-se encontrar a coluna  $A$  que maximiza

$$\{\pi A - \Pi\},$$

em que  $A$  é um padrão de corte viável.

Seja  $A_j$  esta coluna, ela entra na base se

$$\pi A_j - \Pi > 0.$$

Considerando que  $\Pi$  é constante, devemos encontrar a coluna  $A$  que maximiza

$$\{\pi A\},$$

em que  $A$  é um padrão de corte viável.

No problema de corte unidimensional, o padrão de corte  $A$  é determinado levando-se em conta apenas os comprimentos das peças e dos tubos, o que corresponde a resolver o seguinte problema unidimensional da mochila:

$$\text{Maximizar } \sum_{i=1}^n \pi_i a_i$$

$$\text{Sujeito a: } \sum_{i=1}^n l_i a_i \leq L,$$

$$a_i \text{ inteiro e } a_i \geq 0, i=1, \dots, n.$$

No caso de corte de chapas tanto os comprimentos como as larguras dos itens e objetos precisam ser considerados. Padrões de corte podem ser gerados através de algoritmos desenvolvidos para determinação de padrões bidimensionais tais como em Christofides e Whitlock (1977), Wang (1983), Beasley (1985), Oliveira e Ferreira (1990), Morabito e Arenales (1996), Gramani (1997), Morabito e Garcia (1998), Pinto (1999), Vianna *et al.* (2002), Lodi e Monaci (2003).

Uma vez que o problema de geração de padrão tenha sido resolvido, se

$$\pi A_T - \Pi = \text{Máximo}_j \{\pi A_j - \Pi\} \leq 0,$$

então a base corrente é ótima, caso contrário, a nova coluna entra na base e a coluna a sair da base é determinada seguindo-se os passos convencionais do método Simplex.

Nesta tese focaliza-se a geração de padrões de corte bidimensionais especiais. Assim, algumas das definições pertinentes a estes padrões são apresentadas a seguir:

**Definição 1.1:** Um corte *guilhotinado* é um corte reto que vai de um lado a outro do objeto, dividindo-o em 2 partes.

**Definição 1.2:** Denomina-se *padrão guilhotinado* o padrão obtido por sucessivos cortes guilhotinados (vide Figura 1.3a). Caso contrário, ele é classificado como *padrão não guilhotinado* (vide Figura 1.3b).

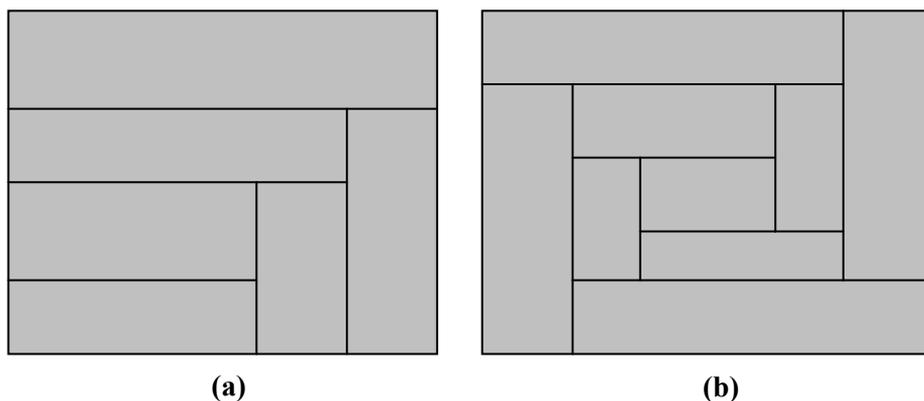


Figura 1.3 – Padrão de corte guilhotinado e não guilhotinado: (a) Padrão de corte guilhotinado, (b) Padrão de corte não guilhotinado.

**Definição 1.3:** Padrões guilhotinados de um objeto retangular são classificados como *ortogonais* se eles são obtidos apenas com cortes guilhotinados paralelos a um dos lados do retângulo (Hinxman, 1980). Caso contrário, os padrões são classificados como guilhotinados *não ortogonais* (vide Figura 1.4b).

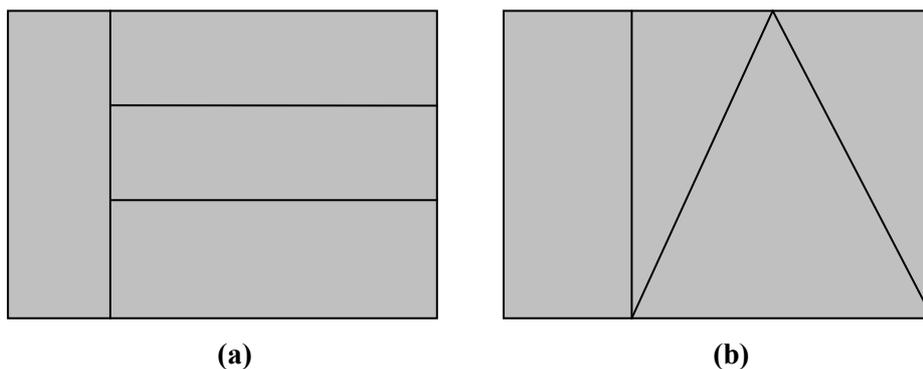


Figura 1.4 – Padrão de corte guilhotinado ortogonal e não ortogonal: (a) Padrão de corte guilhotinado ortogonal, (b) Padrão de corte guilhotinado não ortogonal.

Os padrões de corte podem ser estagiados onde cada estágio indica, normalmente, uma mudança na direção do corte.

**Definição 1.4:** Um padrão guilhotinado ortogonal é dito ser do tipo *2-estágios* se apenas uma única mudança na direção do corte é necessária para cortá-lo (vide Figura 1.4a).

**Definição 1.5:** Um padrão guilhotinado ortogonal 2-estágios é dito ser *exato* se todos os itens são gerados após o segundo estágio de corte (vide Figura 1.5a). Caso haja a necessidade de um recorte para a eliminação de refilos e obtenção dos itens ele é denominado *não exato* (vide Figura 1.5b).

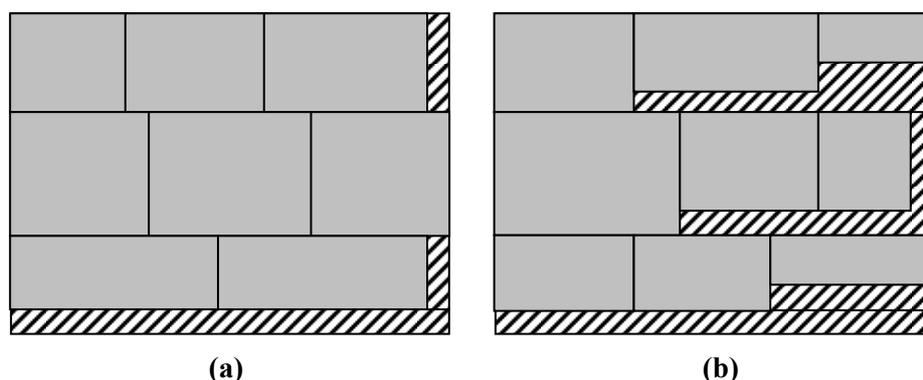


Figura 1.5 – Padrão de corte guilhotinado ortogonal 2-estágios exato (a) e não exato (b).

Na Figura 1.5, as áreas hachuradas representam refilos, ou material desperdiçado. Rigorosamente falando, o padrão 2-estágios não exato é um padrão guilhotinado 3-estágios. Entretanto, o uso do termo 2-estágios não exato é observado em vários artigos da literatura (vide Morabito e Arenales, 2000).

Minimizar a perda de material tem sido um dos objetivos mais frequentes em ambientes de produção de cortes. Outro objetivo frequentemente utilizado é o de minimizar o custo de produção. Em alguns ambientes de produção, nem sempre a solução de menor desperdício implica em uma solução mais econômica para a empresa. Apesar da solução de mínimo desperdício ser desejável, ela pode estar associada a padrões de corte que exigem um alto tempo de utilização da máquina devido à complexidade das operações efetuadas pelo equipamento para produzir os itens requeridos. Se houver uma grande demanda de itens a ser atendida, o equipamento pode tornar-se um “gargalo” na produção, limitando, portanto, o lucro da empresa. A produtividade do equipamento em ambientes de alta produção pode, portanto, ter influência nos lucros.

Padrões tabuleiros pertencem a uma classe especial de padrões guilhotinados ortogonais 2-estágios que são mais simples de serem produzidos pelo equipamento de cortes sendo de interesse particular em ambientes de produção de grande demanda.

Padrões tabuleiros também podem ser classificados como exatos ou, não exatos (vide Figura 1.6).

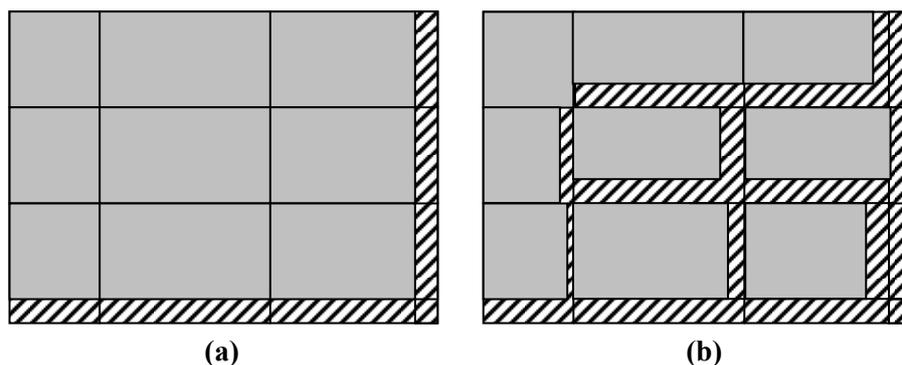


Figura 1.6 – Exemplos de padrões tabuleiros: (a) padrão tabuleiro exato, (b) padrão tabuleiro não exato.

Em ambientes reais de corte, os padrões tabuleiros exatos (Figura 1.6a) são os de maior interesse. Padrões tabuleiros não exatos (Figura 1.6b) requerem operações extras que consomem recursos adicionais em comparação aos padrões tabuleiros exatos.

Neste trabalho foi inicialmente desenvolvida uma nova heurística que se aplica tanto à determinação de padrões tabuleiros irrestritos exatos como não exatos. Nesta heurística, combina-se uma maior variedade de faixas do que em outras heurísticas encontradas na literatura possibilitando obter soluções de melhor qualidade. Testes computacionais são apresentados e comparam-se os resultados obtidos com os da literatura.

Também foram desenvolvidos três métodos exatos para a determinação de padrões tabuleiros do tipo restrito e exato, em que se admite uma limitação no número máximo de itens no padrão de corte. A apresentação destes métodos exatos é feita dentro de um contexto histórico, tentando mostrar a evolução dos resultados obtidos por estes métodos ao longo do desenvolvimento desta tese de doutorado. Na literatura tem-se conhecimento apenas de formulações matemáticas para a resolução do problema da determinação de padrões tabuleiros restritos e exatos. Os três métodos propostos neste

trabalho de tese resolvem diretamente este problema e se baseiam em algoritmos encontrados na literatura para a resolução do problema da mochila unidimensional. Observe que o problema da mochila unidimensional é NP-difícil, o que reflete na complexidade computacional dos métodos exatos propostos.

O primeiro método proposto, denominado Algoritmo Enumerativo, baseia-se no algoritmo de Yanasse Soma e Maculan (2000) para determinação das K-melhores soluções para o problema da mochila unidimensional. O Algoritmo Enumerativo consiste em obter o melhor padrão tabuleiro restrito e exato dentre as melhores combinações de itens enumeradas pelo algoritmo de Yanasse, Soma e Maculan (2000).

O segundo método proposto, denominado Algoritmo de Enumeração Implícita, baseia-se no método da enumeração implícita de Gilmore e Gomory (1963). O Algoritmo de Enumeração Implícita utiliza uma árvore de enumeração para considerar as possíveis combinações de itens que geram um padrão tabuleiro restrito e exato. Limitantes superiores e inferiores são aplicados para avaliar as soluções obtidas em cada nó da árvore de enumeração e reduzir o espaço de soluções a serem investigadas.

O terceiro método exato proposto, denominado Algoritmo Melhorado de Enumeração Implícita, corresponde a uma versão aprimorada do segundo método exato proposto. Um novo limitante superior, mais preciso, foi desenvolvido para se tentar reduzir ainda mais o tempo computacional do algoritmo.

Todos os métodos apresentados utilizam o mesmo algoritmo construtivo para geração de padrões tabuleiros exatos desenvolvido nesta tese. O algoritmo construtivo 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 avaliar os métodos propostos e para comparar os resultados obtidos com os da literatura.

Este texto está organizado nos seguintes Capítulos:

- Capítulo 2 – Revisão Bibliográfica:

Neste Capítulo apresenta-se uma revisão bibliográfica sobre os trabalhos encontrados na literatura que abordam fatores que influenciam na melhoria da produtividade em determinados ambientes de produção de cortes. A apresentação do problema da geração de padrões tabuleiros, tema principal abordado nesta tese é feita dentro de um contexto mais amplo;

- Capítulo 3 – Algoritmos para Determinação de Padrões Tabuleiros:

Neste Capítulo apresenta-se as contribuições feitas para a determinação de padrões tabuleiros. Testes computacionais comparativos também são apresentados;

- Capítulo 4 – Conclusão:

Neste Capítulo apresenta-se as considerações finais sobre o trabalho desenvolvido e propostas para trabalhos futuros.

## CAPÍTULO 2

### REVISÃO BIBLIOGRÁFICA

Embora a literatura sobre problemas de corte seja bastante extensa, os trabalhos que focam sobre produtividade em ambientes de corte são bem limitados. Foram revisados e analisados alguns dos trabalhos encontrados na literatura que tratam de questões relacionadas à melhoria da produtividade em determinados ambientes de corte. Estes trabalhos podem ser divididos em dois grupos de fatores: os trabalhos que focam na programação da produção para tentar reduzir o tempo de corte e, os trabalhos que tentam reduzir o tempo de corte com foco na determinação de padrões de corte.

#### 2.1 – Redução do Tempo de Corte com Foco na Programação da Produção

Em muitos ambientes de corte, o material cortado corresponde a apenas uma parcela do custo relevante. Durante o processo de corte, cada mudança de tipo de padrão (vide Figura 2.1) pode implicar num aumento do custo operacional e perda de produção, por exemplo, devido ao tempo de preparo do equipamento. Quanto maior for o número de padrões diferentes a serem processados, maior será o custo incorrido.

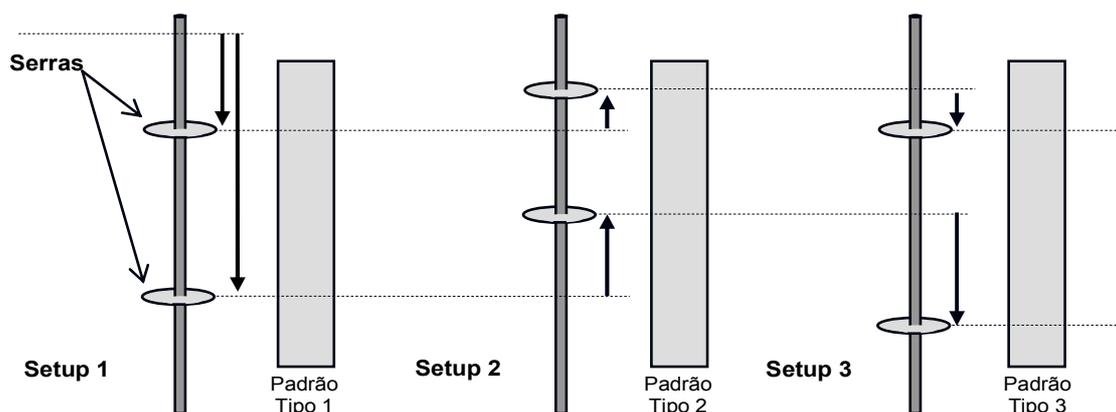


Figura 2.1 – Preparo do equipamento para corte de 3 diferentes tipos de padrões.  
Fonte: adaptado de Umetani *et al.* (2001).

O Problema da Redução do Número de Diferentes Tipos de Padrões (PRNP) é de grande importância em ambientes de produção onde os custos operacionais do

equipamento e da perda de produção com a preparação do equipamento são significativos em comparação com o custo do material cortado.

Haessler (1975) apresentou uma formulação do problema de corte de estoque unidimensional que leva em consideração também o custo associado às mudanças de padrões durante a operação de corte.

Segundo Haessler, o modelo proposto estava distante de ser resolvido de maneira ótima, pois o número de diferentes tipos de padrões pode ser muito grande e a maneira mais conveniente de tratar este tipo de problema segundo aquele autor é através da aplicação de heurísticas. Haessler propôs uma heurística que, seqüencialmente, adiciona novos padrões de corte à solução corrente até que todas as demandas sejam satisfeitas. O procedimento proposto por Haessler baseia-se no uso de descritores e de níveis de aspiração. O uso de descritores serve para analisar a solução em cada estágio. O uso de níveis de aspiração visa a tomada de decisões de acordo com os descritores calculados.

Na literatura encontram-se ainda trabalhos que utilizam outras técnicas para controlar as mudanças de padrões. Alguns destes trabalhos baseiam-se na resolução de problemas similares ao problema da redução do número de diferentes padrões.

Walker (1976) sugere que o PRNP pode ser resolvido como um Problema de Carga Fixa. No trabalho de Walker (1976) o Problema de Carga fixa é resolvido através de uma variação do método Simplex. Walker propôs a resolução deste Problema de Carga Fixa em 2 fases. Na primeira fase é feita uma otimização local modificando-se a regra da seleção da variável a entrar na base que passa a considerar o valor da carga fixa. Na segunda fase, é aplicada uma heurística que percorre os melhores pontos extremos adjacentes ao ponto corrente para encontrar uma solução melhorada para o problema.

Em Farley e Richardson (1984) é proposto um procedimento para a resolução do PRNP baseado no procedimento proposto por Walker (1976) para resolver o Problema de Carga Fixa. O procedimento proposto por Farley e Richardson (1984) tenta reduzir o número de diferentes tipos de padrões de corte através da redução do número de variáveis básicas relacionadas aos padrões utilizados. Esta redução de padrões é feita

iterativamente através do procedimento modificado para o Método Simplex proposto por Walker (1976) e prossegue mantendo o número de objetos utilizados dentro de um limite aceitável. Segundo Farley e Richardson, testes computacionais mostraram que o procedimento proposto consegue ser significativamente mais rápido do que outros trabalhos encontrados na literatura que tratam de problemas mais gerais de Carga Fixa.

Combinar padrões é outra solução aplicada em alguns dos trabalhos encontrados na literatura que trata da redução do número de diferentes padrões. No trabalho de Diegel *et al.* (1993) uma heurística que possibilita combinar dois padrões em um único padrão partindo de uma solução de Programação Linear é apresentada.

Foerster e Wäscher (2000) estenderam o trabalho de Diegel *et al.* (1993), generalizando-o para a determinação de outras possíveis combinações de padrões (combinação 3 em 2 padrões, combinação 4 em 3 padrões, etc ...). A generalização parte da observação de que as frequências dos novos padrões combinados devem ser somas destas frequências iniciais para que o número de objetos cortados se mantenha constante. O procedimento proposto por Foerster e Wäscher demonstrou ser superior ao método de Haessler em uma grande variedade de classes de problemas, principalmente em classes que consideram uma grande demanda e diversidade de itens.

O uso de algoritmos exatos para redução do número de diferentes padrões não tem sido muito freqüente na literatura. Vanderbeck (2000) propôs um algoritmo exato para minimizar o número de ajustes (setup) no problema de corte de estoque unidimensional. O PRNP é formulado como um problema quadrático inteiro e decomposto em vários problemas da mochila restrito. Um algoritmo branch-and-bound é então aplicado para resolver o problema decomposto, em conjunto com uma técnica de geração de colunas. O procedimento proposto por Vanderbeck demonstrou ser eficiente apenas para obtenção de soluções ótimas para problemas pequenos. No caso de problemas maiores, não se conseguiu obter soluções ótimas dentro de um limite de tempo aceitável.

Umetani *et al.* (2003) propuseram um procedimento que encontra uma boa solução a partir de um número fixo  $m$  de diferentes tipos de padrões e busca-se encontrar uma solução que apresente um desvio quadrático pequeno em relação às demandas. Segundo

Umetani *et al.* (2003), o procedimento proposto consegue obter resultados semelhantes ao método proposto por Haessler (1975) e ao método proposto por Foerster e Wäscher (2000).

O uso de procedimentos híbridos é mais uma alternativa na redução do número de diferentes padrões. Recentemente Limeira e Yanasse (2002) sugeriram um procedimento híbrido para a redução de padrões aplicado em três estágios. Num primeiro estágio, obtêm-se bons padrões de corte que completam a demanda de parte dos itens contidos nestes padrões. Em seguida, num segundo estágio, a demanda restante é satisfeita resolvendo-se o problema de corte de estoque residual. Num terceiro estágio, técnicas de redução de padrões tais como a apresentada por Foerster e Wäscher (2000) são aplicadas para tentar melhorar a solução obtida anteriormente. Segundo os autores, através deste procedimento obteve-se um número reduzido de padrões sem que o número de objetos crescesse em demasia.

Em adição ao procedimento híbrido de Limeira e Yanasse (2002), 3 novas heurísticas foram apresentadas por Limeira (2003). A primeira heurística utiliza uma mudança de escala nas demandas dos itens. A segunda heurística é uma variação do procedimento híbrido já apresentado em Limeira e Yanasse (2002). A mudança com relação ao procedimento de Limeira e Yanasse (2002) ocorre no esquema de arredondamento das demandas dos itens no primeiro estágio do procedimento híbrido. A terceira heurística apresentada por Limeira (2003) é uma variação do procedimento de Foerster e Wäscher (2000). A heurística admite que as demandas dos padrões combinados não precisam necessariamente ser satisfeitas de forma exata, apenas deve-se garantir que sejam satisfeitas as demandas dos itens.

Em Poldi (2003), foram desenvolvidas heurísticas residuais que geram uma solução inteira para o problema de corte de estoque unidimensional inteiro. Essas heurísticas tendem a apresentar soluções com um número de padrões de corte menor que outras analisadas. Por isto, a autora realizou testes computacionais para avaliar o desempenho de suas heurísticas residuais, utilizando o procedimento híbrido de Limeira e Yanasse (2003) para resolução do PRNP. Os resultados dos testes computacionais realizados

mostraram que o uso do procedimento híbrido de Limeira e Yanasse (2003) em conjunto com as heurísticas residuais é eficiente apenas nas classes de problemas que apresentam itens pequenos, de fácil combinação. Para as demais classes de problemas, a redução de padrões fica prejudicada devido ao fato dos itens de tamanho maior exigirem a utilização de um número maior de padrões.

Para o PRNP, Salles Neto e Moretti (2003) apresentaram um modelo não linear que minimiza as perdas e reduz o número de diferentes tipos de padrões de corte unidimensional. Com base em um teorema demonstrado por Martinez (2001), Salles Neto e Moretti (2003) resolvem uma seqüência de problemas de programação não linear para obter a solução do problema original.

Em alguns ambientes de produção, a produtividade do equipamento está ligada aos ciclos da máquina. Cada ciclo de máquina corresponde ao corte de um padrão. A novidade está no fato de se poder cortar mais de um objeto de cada vez, ou seja, o corte pode ser feito em pacotes (vide Figura 2.2).

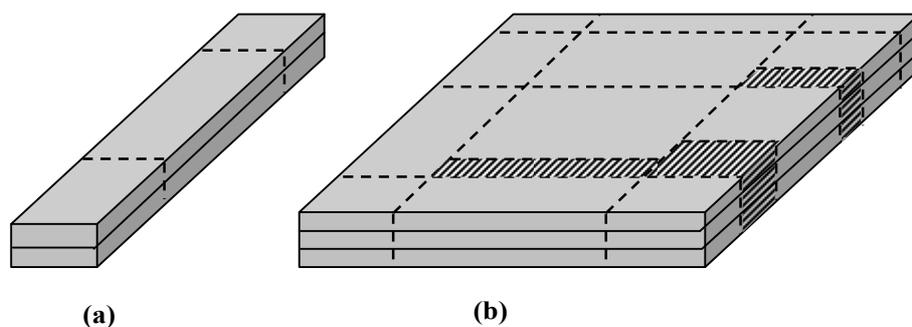


Figura 2.2 – Corte de pacotes: (a) caso unidimensional, (b) caso bidimensional.

O tempo de corte de um “pacote” com um único objeto ou com vários objetos pode não ser muito diferente. Portanto, cortar individualmente os objetos pode implicar em muitos ciclos e longo tempo de máquina, o que seria indesejável do ponto de vista da produtividade.

A redução do número de ciclos de máquina aumenta a produtividade da máquina, mas pode provocar um maior desperdício de material. Têm-se, portanto um balanço natural entre produtividade e perda.

Não temos conhecimento da existência de muitos trabalhos na literatura que tratam especificamente da redução do número de ciclos de máquina.

Antonio *et al.* (1996), resolveram um problema de corte de estoque unidimensional em pacotes. Para aumentar a produtividade da máquina de corte, as barras (objetos) de mesmo tipo são separadas e empilhadas uma sobre as outras de maneira a formar os pacotes.

O método de resolução utilizado varia de acordo com o uso e necessidades do ambiente de produção. Para ambientes de produção que requerem computação em tempo real, os autores sugerem a aplicação de uma heurística FFD (First Fit Decreasing). Para o caso onde o tempo computacional para se obter uma solução não é tão importante é sugerida a utilização de um algoritmo adaptado de Programação Dinâmica.

No trabalho de Chu e Antonio (1999), o problema de corte discutido em Antonio *et al.* (1996) é novamente explorado, desta vez, através da resolução de um problema prático de corte de tubos. Os autores apresentaram um modelo matemático para resolver este problema que leva em consideração os custos associados à perda de material e o tempo de corte da máquina e também o fato de ser possível reaproveitar parte da perda de material na confecção de outros produtos. A função objetivo é não linear e o problema é resolvido através de uma extensão do algoritmo de programação dinâmica proposto em Antonio *et al.* (1996).

Wagner (1999) também considera que o corte dos objetos seja feito em pacotes e propôs um procedimento, baseado em algoritmo genético para resolver este problema.

Yanasse *et al.* (1993) propuseram uma heurística para minimizar o número de ciclos de máquina mantendo algum controle sobre a perda de material. A heurística tenta definir, seqüencialmente, padrões de corte que apresentem itens com grande demanda de

maneira que os padrões formados possam ser repetidos várias vezes. Esta estratégia é utilizada enquanto o desperdício dos padrões for aceitável. O procedimento heurístico proposto por estes autores pode ser estendido facilmente para problemas de corte de diversas dimensões. O mesmo não pode ser dito do procedimento de Antonio *et al.* (1996) e de Wagner (1999).

Não se tem conhecimento de outros trabalhos que tratam do problema da redução de ciclos na determinação de padrões unidimensionais e bidimensionais. Acredita-se que esta escassez de trabalhos esteja relacionada ao fato do PRNP apresentar uma certa similaridade com o problema da redução de ciclos de máquina.

Reduzir o número de diferentes tipos de padrões implica numa redução do número de ciclos de máquina no caso de se poder cortar cada tipo de padrão de uma só vez. No caso de ser necessário cortar um tipo de padrão em mais de um ciclo, esta redução de padrões nem sempre atinge o menor número de ciclos possível.

No trabalho de Diegel *et al.* (2006), o conceito de redução do número de ciclos de máquina está relacionado com o aumento do comprimento da corrida (run length). Os autores sugerem que na resolução do problema de corte de estoque seja imposto um valor mínimo para o comprimento da corrida.

Alguns dos trabalhos encontrados na literatura apresentam modelos matemáticos que integram os custos relacionados ao tempo de preparo do equipamento.

Westerlund *et al.* (1998), estudaram um problema prático da indústria de papel que leva em consideração a perda de material e o tempo de corte. O modelo inicialmente apresentado por Westerlund *et al.* (1998) apresentava uma função objetivo bilinear e algumas restrições bilineares, o que o tornava um modelo não convexo. Para contornar o problema da não linearidade, os autores propuseram a resolução do modelo em duas etapas. Numa primeira etapa, um algoritmo de enumeração explícita é usado para gerar todos os padrões de corte. Na segunda etapa, os padrões gerados pelo algoritmo de enumeração explícita são inseridos no modelo de programação linear inteira mista e o modelo é executado com o auxílio de uma linguagem de modelagem matemática.

## 2.2 – Redução do Tempo de Corte com Foco nos Padrões de Corte

Em alguns ambientes de produção pode existir uma diferença significativa de tempo quando se cortam diferentes padrões. Dada as características da máquina, alguns padrões de corte são cortados mais rapidamente que outros. Por exemplo, no processo de produção de corte de chapas retangulares de fibras de madeira reconstituída, estudada por Belluzzo (2002), as chapas são giradas pela máquina de corte para que cortes transversais e longitudinais (denominados cortes cabeças) ao longo do objeto sejam aplicados. O corte pode ser efetuado em vários estágios, onde cada estágio implica num giro de 90 graus do objeto a ser cortado (vide Figura 2.3).

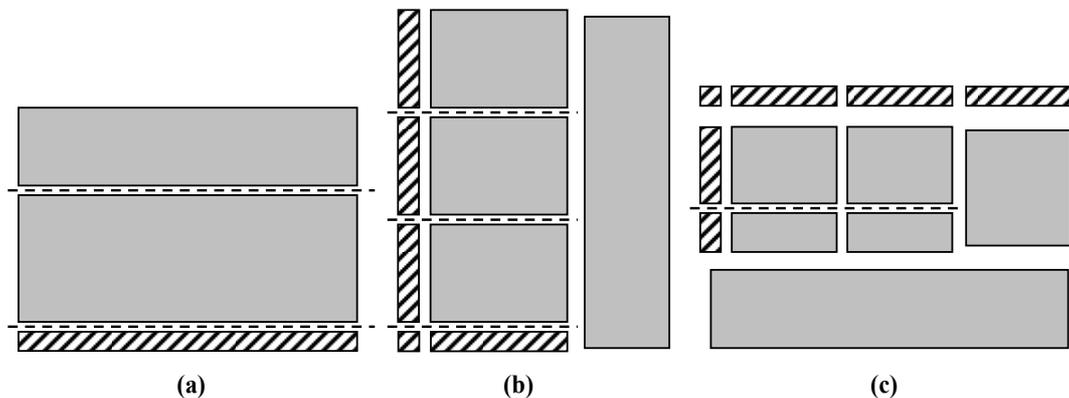


Figura 2.3 – Corte de chapas retangulares de fibras de madeira reconstituída: (a) primeiro corte (primeiro estágio); (b) primeiro giro (segundo corte, segundo estágio); (c) segundo giro (terceiro corte, terceiro estágio).

Padrões mais complexos podem ser cortados utilizando-se um maior número de estágios para se tentar reduzir ainda mais as perdas de material. No entanto, estes padrões exigem um maior número de giros da chapa aumentando o tempo de corte dos objetos. Belluzzo (2002), por exemplo, estima em cerca de 2 minutos o tempo necessário para que a máquina considerada em seus estudos efetue cada giro. É desejável, portanto, utilizar padrões de corte mais simples para se reduzir o tempo de corte dos objetos.

O uso de padrões de corte mais simples para melhorar a produtividade do equipamento de corte foi analisado em Morabito e Arenales (2000) e Katsurayama e Yanasse (2001) focalizando problemas de corte bidimensionais em chapas de madeira. Em problemas de

corte unidimensionais, tridimensionais, ou de maiores dimensões, não se tem conhecimento de trabalhos na literatura que tratam especificamente deste tema, nem de trabalhos relatando aplicações práticas em ambientes de produção.

Um dos padrões bidimensionais mais simples é o homogêneo. Este é facilmente obtido bastando repetir um único tipo de item ao longo da largura e comprimento do objeto (vide Figura 2.4).

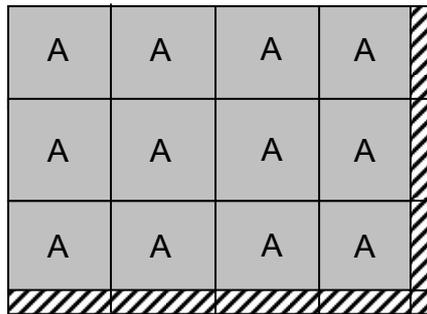


Figura 2.4 – Exemplo de obtenção de padrão homogêneo: repetição do item A ao longo da largura e comprimento do objeto.

No caso irrestrito para a determinação do melhor padrão homogêneo, basta selecionar o item  $i$  dentre os  $n$  tipos de itens de dimensão  $w_i \times l_i$  e valor de utilidade  $\pi_i$  que resulte no padrão homogêneo mais lucrativo, ou seja:

$$Max_i \left\{ \pi_i \left\lfloor \frac{W}{w_i} \right\rfloor \left\lfloor \frac{L}{l_i} \right\rfloor \right\}.$$

Para o caso restrito, a determinação do padrão homogêneo é feita através de:

$$Max_i \left\{ \min \left\{ \pi_i \left\lfloor \frac{W}{w_i} \right\rfloor \left\lfloor \frac{L}{l_i} \right\rfloor, d_i \right\} \right\},$$

em que  $d_i$  corresponde à demanda associada ao item  $i$ .

Um outro padrão bidimensional simples é o padrão tabuleiro. Os *padrões tabuleiros*, também conhecidos como *padrões 1-grupo* (Gilmore e Gomory, 1965) 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 (vide Figura 2.5).

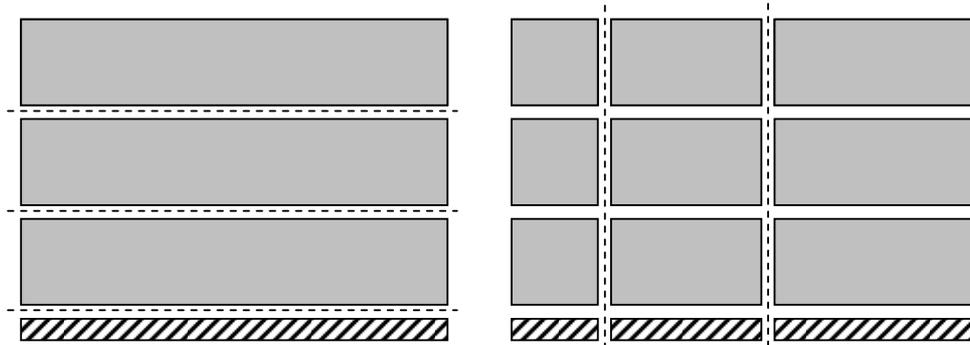


Figura 2.5 – Corte de um padrão tabuleiro: (a) primeiro corte (primeiro estágio), (b) único giro (segundo corte, segundo estágio).

Morabito e Arenales (2000) formularam o problema da determinação de padrões tabuleiros como um problema quadrático inteiro e sugeriram uma heurística para a sua obtenção. O seguinte modelo matemático para padrões tabuleiros foi sugerido por eles:

$$\text{Maximizar } \sum_{i=1}^n \sum_{k=1}^n \pi_{ik} \nu_i \mu_k \quad (2.1.1)$$

$$\text{Sujeito a: } \sum_{i=1}^n l_i \nu_i \leq L, \quad (2.1.2)$$

$$\sum_{k=1}^n w_k \mu_k \leq W, \quad (2.1.3)$$

$$\nu_i \geq 0, \mu_k \geq 0 \quad (i, k=1, \dots, n), \text{ inteiro.} \quad (2.1.4)$$

Em que:

$\nu_i$  é o número de vezes que o comprimento  $l_i$  é cortado;

$\mu_k$  é o número de vezes que a largura  $w_k$  é cortada;

$\pi = \{\pi_{ik}\}$  é uma matriz  $n \times n$  com a utilidade/valor da peça demandada de dimensão  $w_k \times l_i$ , sendo:

$$\pi_{ik} = \begin{cases} \pi_r \text{ se } \tilde{R}_r = (w_k \times l_i) \text{ para algum } r \\ \text{(ou seja, existe uma peça de dimensões } w_k \times l_i \text{ e } \pi_r \text{ é o valor de utilidade desta peça)} \\ 0, \text{ caso contrário} \end{cases}$$

Em (2.1.1) maximiza-se o valor total das peças cortadas no padrão. As restrições (2.1.2) e (2.1.3) impõem que a soma das peças de comprimento e de largura não devem exceder, respectivamente, o comprimento e largura do objeto. A restrição (2.1.4) impõe a não negatividade e integralidade das variáveis.

Este programa não linear inteiro modela o caso exato de padrões tabuleiros. Para o caso não exato, basta considerar a seguinte modificação sugerida por Morabito (2001):

$$\pi_{ik} = \begin{cases} \text{Max}_j \{ \pi_j \text{ tal que } l_i \geq l_j \text{ e } w_k \geq w_j \} \\ 0, \text{ caso contrário} \end{cases}$$

Morabito e Arenales não exploram esta formulação e sugerem uma heurística simples para gerar os padrões tabuleiros. A heurística sugerida por Morabito e Arenales consiste em obter o mais valioso padrão dentre todos os padrões compostos de faixas do mesmo tipo. O procedimento heurístico é descrito por:

**Passo 1:**

$$V_j = \text{Maximizar } \sum_{i \in \tilde{W}_j} \pi_i \nu_i$$

$$\text{Sujeito a: } \sum_{i \in \tilde{W}_j} l_i \nu_i \leq L,$$

$$\nu_i \geq 0 \text{ (} i=1, \dots, n \text{) e inteiro.}$$

Em que  $\tilde{W}_j = \{i \mid w_i = w_j\}$  para o caso exato e  $\tilde{W}_j = \{i \mid w_i \leq w_j\}$  para o caso não exato.

**Passo 2:**

$$\text{Maximizar } \lfloor W / w_j \rfloor V_j, j \in D_w.$$

**Em que  $D_w$  corresponde ao conjunto de índices de todas as larguras  $w_j$  possíveis dos itens do problema.**

Analogamente, a composição das faixas poderá ser feita no comprimento do objeto. O melhor padrão tabuleiro é obtido determinando-se o padrão mais lucrativo em ambas as dimensões.

Recentemente, surgiram outras formulações para o problema da determinação de padrões tabuleiros. Em Scheithauer (2002), foi estudado o problema da determinação de padrões tabuleiros no processo de produção de corte de pedras. Scheithauer propôs a resolução do problema através de um modelo de programação linear inteira mista que leva em consideração a determinação de padrões tabuleiros do tipo exato e não exato.

O modelo proposto por Scheithauer considera o corte de no máximo  $\gamma$  faixas verticais de comprimentos  $L_1, \dots, L_\gamma$  e de  $\delta$  faixas longitudinais de larguras  $W_1, \dots, W_\delta$ .

Seja  $n$ , o número de diferentes tipos de itens a serem produzidos, o número máximo  $\gamma$  de faixas verticais presente no padrão pode ser calculado através de

$$\gamma = \lfloor L/l_{\min} \rfloor, \text{ em que } l_{\min} = \min_{i=1, \dots, n} l_i,$$

e o número máximo  $\delta$  de faixas longitudinais presente no padrão pode ser calculado através de

$$\delta = \lfloor W/w_{\min} \rfloor, \text{ em que } w_{\min} = \min_{i=1, \dots, n} w_i.$$

Seja  $\eta_{ijk} \in \{0,1\}$  ( $i = 1, \dots, n$ ,  $j = 1, \dots, \gamma$ ,  $k = 1, \dots, \delta$ ), em que  $\eta_{ijk} = 1$  se e somente se a peça do tipo  $i$  é obtida do retângulo  $W_k \times L_j$  e,  $\eta_{ijk} = 0$ , caso contrário. O modelo proposto por Scheithauer (2002) pode então ser descrito como se segue:

$$\text{Maximizar } \sum_{i=1}^n \sum_{j=1}^{\gamma} \sum_{k=1}^{\delta} \pi_i \eta_{ijk} \quad (2.2.1)$$

$$\text{Sujeito a: } \sum_{j=1}^{\gamma} L_j \leq L, \quad (2.2.2)$$

$$\sum_{k=1}^{\delta} W_k \leq W, \quad (2.2.3)$$

$$\sum_{i=1}^n \eta_{ijk} \leq 1, \quad j = 1, \dots, \gamma, \quad k = 1, \dots, \delta, \quad (2.2.4)$$

$$\sum_{i=1}^n l_i \eta_{ijk} \leq L_j, \quad j = 1, \dots, \gamma, \quad k = 1, \dots, \delta, \quad (2.2.5)$$

$$\sum_{i=1}^n w_i \eta_{ijk} \leq W_k, \quad j = 1, \dots, \gamma, \quad k = 1, \dots, \delta, \quad (2.2.6)$$

$$\sum_{j=1}^{\gamma} \sum_{k=1}^{\delta} \eta_{ijk} \leq d_i, \quad i = 1, \dots, n, \quad (2.2.7)$$

$$L_j \geq L_{j+1}, \quad j = 1, \dots, \gamma - 1, \quad (2.2.8)$$

$$W_k \geq W_{k+1}, \quad k = 1, \dots, \delta - 1, \quad (2.2.9)$$

$$L_j, W_k \geq 0, \quad \forall j, k. \quad (2.2.10)$$

Em (2.2.1) maximiza-se o valor total das peças cortadas no padrão. As restrições (2.2.2) e (2.2.3) impõem que o arranjo de faixas de comprimentos e larguras ao longo do padrão não deve exceder, respectivamente, o comprimento e largura do objeto. A restrição (2.2.4) impõe que no máximo 1 peça é colocada em cada retângulo de dimensão  $W_k \times L_j$ . As restrições (2.2.5) e (2.2.6) impõem que o arranjo de peças no comprimento e na largura não devem exceder, respectivamente, os comprimentos e larguras das faixas. A restrição (2.2.7) limita a disponibilidade (demanda) das peças a serem cortadas. As restrições (2.2.8-2.2.10) limitam o grau de liberdade das faixas sendo consideradas e valores não negativos para as variáveis de decisão.

Observe que no modelo proposto por Scheithauer, o autor considera a determinação de padrões tabuleiros restritos, onde  $d_i$  representa a demanda das peças do tipo  $i$  a serem produzidas. O modelo introduzido por Morabito e Arenales (2000) para determinação de padrões tabuleiros leva em consideração apenas padrões irrestritos. Para o caso restrito, Yanasse e Morabito (2003) propuseram o seguinte modelo, adaptado a partir do modelo proposto por Morabito e Arenales (2000):

$$\text{Maximizar } \sum_{r=1}^n \sum_{i=1}^{I1} \sum_{k=1}^{I2} \pi_{rik} \kappa_{rik} \quad (2.3.1)$$

$$\text{Sujeito a: } \sum_{i=1}^{I1} l_i v_i \leq L, \quad (2.3.2)$$

$$\sum_{k=1}^{I2} w_k \mu_k \leq W, \quad (2.3.3)$$

$$\sum_{r=1}^n \kappa_{rik} \leq v_i \mu_k, \text{ para todos os } i, k, \quad (2.3.4)$$

$$\sum_{i=1}^{I1} \sum_{k=1}^{I2} \kappa_{rik} \leq d_r, \text{ para todos os } r, \quad (2.3.5)$$

$$v_i \geq 0, \mu_k \geq 0, \kappa_{rik} \geq 0, \text{ inteiro } (i=1, \dots, I1; k=1, \dots, I2). \quad (2.3.6)$$

Em que:

$I1, I2$  representam, respectivamente, o número de diferentes tipos de comprimentos  $l_i$  e larguras  $w_k$  das peças;

$\kappa_{rik}$  é o número de peças do tipo  $r$  contidas em todos os retângulos  $w_k \times l_i$ ;

$d_r$  é a demanda da peça tipo  $r$ ;

$\pi_{rik}$  representa a utilidade/valor da peça demandada  $r$  de dimensão  $w_k \times l_i$ , definido de

acordo com o tipo de padrão tabuleiro a ser determinado. Para o caso da determinação de padrões tabuleiros não exatos têm-se:

$$\pi_{rik} = \begin{cases} \pi_r & \text{se } l_r \leq l_i \text{ e } w_r \leq w_k, r = 1, \dots, n \\ 0, & \text{caso contrário} \end{cases},$$

e para o caso da determinação de padrões tabuleiros exatos têm-se:

$$\pi_{rik} = \begin{cases} \pi_r & \text{se } l_r = l_i \text{ e } w_r = w_k, r = 1, \dots, n \\ 0, & \text{caso contrário} \end{cases}.$$

Em (2.3.1) maximiza-se o valor total das peças cortadas no padrão. As restrições (2.3.2) e (2.3.3) impõem que as peças de comprimentos e larguras não devem exceder, respectivamente, o comprimento e largura do objeto. A restrição (2.3.4) limita o total de peças de todos os tipos cortados no padrão ao número máximo de retângulos  $v_i \mu_k$  definidos pelas faixas. A restrição (2.3.5) refere-se à demanda das peças. A restrição (2.3.6) diz respeito à não negatividade e integralidade das variáveis.

Para resolver o problema da determinação de padrões tabuleiros exatos/não exatos restritos, uma formulação linear para o modelo (2.3.1-2.3.6) foi apresentado em Yanasse e Morabito (2003). A formulação é derivada do método proposto por Harjunkoski *et al.* (1997) que utiliza representações binárias para reescrever as restrições não lineares do modelo sob uma forma linear. Segundo o novo modelo linear proposto por Yanasse e Morabito (2003) para determinação de padrões tabuleiros exatos e restritos, as equações (2.3.1-2.3.6) podem ser reescritas através do seguinte conjunto de equações:

$$\text{Maximizar } \sum_{r=1}^n \sum_{i=1}^{I1} \sum_{k=1}^{I2} \pi_{rik} K_{rik} \quad (2.4.1)$$

$$\text{Sujeito a: } \sum_{i=1}^{I1} l_i \sum_{s=1}^{s_i} 2^{s-1} \tau_{is} \leq L, \quad (2.4.2)$$

$$\sum_{k=1}^{I2} w_k \mu_k \leq W, \quad (2.4.3)$$

$$\sum_{r=1}^n \kappa_{rik} \leq \sum_{s=1}^{s_i} 2^{s-1} \omega_{iks}, \text{ para todos os } i, k, \quad (2.4.4)$$

$$\sum_{i=1}^{I1} \sum_{k=1}^{I2} \kappa_{rik} \leq d_r, \text{ para todos os } r, \quad (2.4.5)$$

$$\omega_{iks} \leq \mu_k, \text{ para todos } i, k, s, \quad (2.4.6)$$

$$\omega_{iks} \geq \mu_k - M(1 - \tau_{is}), \text{ para todos } i, k, s, \quad (2.4.7)$$

$$\omega_{iks} \leq M\tau_{is}, \text{ para todos } i, k, s, \quad (2.4.8)$$

$$\tau_{is} \in \{0,1\}, \mu_k \geq 0, \kappa_{rik} \geq 0, \text{ inteiros, } \omega_{iks} \geq 0$$

$$(r = 1, \dots, n; i = 1, \dots, I1; k = 1, \dots, I2; s = 1, \dots, s_i,$$

$$s_i \text{ inteiro, tal que } 2^{s_i-1} \leq \left\lfloor \frac{L}{l_j} \right\rfloor \leq 2^{s_i}), \quad (2.4.9)$$

em que  $M$  corresponde a um número suficientemente grande, por exemplo,  $M=WL$ .

O modelo linearizado (2.4.1-2.4.9) foi codificado na linguagem de modelagem GAMS. Testes computacionais realizados por Yanasse e Morabito (2003) mostraram tempos computacionais bastante satisfatórios na resolução deste modelo linear, melhores até que os tempos computacionais obtidos com a resolução do modelo de Scheithauer (2002).

O primeiro algoritmo exato para geração de padrões tabuleiro parece ter sido o proposto em Katsurayama e Yanasse (1999, 2000). Eles desenvolveram um algoritmo enumerativo baseado no algoritmo de Yanasse, Soma e Maculan (2000) para determinação das K-melhores soluções para o problema da mochila unidimensional. O algoritmo proposto por Katsurayama e Yanasse se aplica a padrões tabuleiros exatos irrestritos e funciona em duas fases, primeiro, buscando uma melhor combinação nas faixas e, depois, buscando uma boa combinação de faixas.

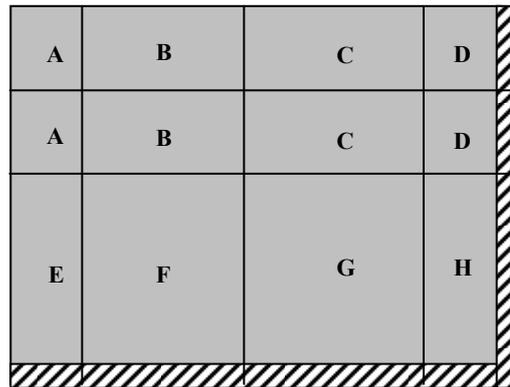


Figura 2.6 – Um exemplo de padrão tabuleiro.  
 Fonte: Katsurayama (2002).

Por exemplo, para obtenção do padrão apresentado na Figura 2.6, gera-se na primeira fase, a faixa (E, F, G, H) e, na segunda fase, geram-se faixas que podem formar padrões tabuleiro quando compostas com a faixa gerada na primeira fase. Na segunda fase seria gerada a faixa (A, B, C, D) e, possivelmente outras. Uma vez geradas todas estas faixas, determina-se a melhor combinação delas, gerando finalmente o padrão tabuleiro final. O algoritmo segue, portanto, o procedimento sugerido por Gilmore e Gomory (1965) para geração de padrões 2-estágios.

Embora o algoritmo proposto por Katsurayama e Yanasse (1999, 2000) possa ser adaptado para a determinação de padrões tabuleiros não exatos e outros, os autores concluíram que a aplicação de heurísticas ainda é uma excelente alternativa para a determinação deste tipo de padrão, pois oferece soluções de boa qualidade em um tempo de processamento muito inferior aos métodos exatos conhecidos.

Os próximos padrões bidimensionais mais simples são os padrões do tipo *2-estágios exato* e *2-estágios não exato*.

Vianna *et al.* (2002) apresentaram o seguinte modelo não linear inteiro para padrões 2-estágios não exatos:

$$\text{Maximizar } \sum_{i=1}^I \sum_{j=1}^{S_i} \sum_{r \in M_i} \pi_r \rho_{ij}^r \sigma_{ij} \tag{2.5.1}$$

$$\text{Sujeito a: } \sum_{r \in M_i} l_r \rho_{ij}^r \leq L, \quad i = 1, \dots, I, \quad j \in S_i, \quad (2.5.2)$$

$$\sum_{i=1}^I w_i \sum_{j=1}^{S_i} \sigma_{ij} \leq W, \quad (2.5.3)$$

$$\rho_{ij}^r, \sigma_{ij} \geq 0, \text{ inteiro}, \quad r = 1, \dots, n, \quad i = 1, \dots, I, \quad j = 1, \dots, S_i. \quad (2.5.4)$$

Em que:

$n$  é o número de diferentes tipos de itens;

$I$  é o número de diferentes tipos de larguras  $w_i$ ;

$S_i$  é o número máximo de padrões da faixa de largura  $w_i$ ;

$$M_i = \{r \mid w_r \leq w_i\};$$

$\pi_r$  é o valor de utilidade da peça  $r$ ;

$l_r$  é o comprimento da peça  $r$ ;

$\rho_{ij}^r$  é o número de vezes que a peça  $r$  é cortada no padrão  $j$  na faixa de largura  $w_i$ ;

$\sigma_{ij}$  é o número de vezes que o padrão  $j$  da faixa de largura  $w_i$  é repetido ao longo da largura  $W$  da chapa.

Em (2.5.1) maximiza-se o valor total das peças cortadas no padrão. As restrições (2.5.2) e (2.5.3) impõem que o arranjo de faixas de comprimentos e larguras ao longo do padrão não devem exceder, respectivamente, o comprimento e largura do objeto. A restrição (2.5.4) diz respeito à não negatividade e integralidade das variáveis.

Para o caso restrito de padrões 2-estágios não exatos, a seguinte restrição é incluída:

$$\sum_{i=1}^I \sum_{j=1}^{S_i} \rho_{ij}^r \sigma_{ij} \leq d_r, \quad r = 1, \dots, n,$$

em que  $d_r$  é a demanda associada à peça  $r$ .

Modelos matemáticos para padrões 2-estágios restritos também foram apresentados por Yanasse e Morabito (2003). Uma formulação linear inteira foi proposta pelos autores como uma extensão do modelo de Scheithauer (2002), descrito anteriormente, para padrões tabuleiros não exatos. A extensão proposta por Yanasse e Morabito (2003) consiste em redefinir a variável  $\eta_{ijk}$  do modelo de Scheithauer (2002), de maneira que ela passa a indicar se a peça do tipo  $i$  é obtida na posição  $j$  da faixa  $k$  ( $\eta_{ijk} = 1$ ) ou não ( $\eta_{ijk} = 0$ ), ao invés de indicar se ela é ou não obtida no retângulo  $W_k \times L_j$ . Segundo Yanasse e Morabito (2003), com base nesta suposição, o modelo apresentado por Scheithauer (2002) passa a ser reescrito como:

$$\text{Maximizar } \sum_{i=1}^n \sum_{j=1}^{\gamma} \sum_{k=1}^{\delta} \pi_i \eta_{ijk}$$

$$\text{Sujeito a: } \sum_{k=1}^{\delta} W_k \leq W,$$

$$\sum_{i=1}^n \eta_{ijk} \leq 1, \quad j = 1, \dots, \gamma, \quad k = 1, \dots, \delta,$$

$$\sum_{i=1}^n \sum_{j=1}^{\gamma} l_i \eta_{ijk} \leq L, \quad k = 1, \dots, \delta,$$

$$\sum_{i=1}^n w_i \eta_{ijk} \leq W_k, \quad j = 1, \dots, \gamma, \quad k = 1, \dots, \delta,$$

$$\sum_{j=1}^{\gamma} \sum_{k=1}^{\delta} \eta_{ijk} \leq d_i, \quad i = 1, \dots, n,$$

$$W_k \geq W_{k+1}, \quad k = 1, \dots, \delta - 1,$$

$$\eta_{ijk} \in \{0,1\}, W_k \geq 0, \quad \forall i, j, k.$$

Yanasse e Morabito (2003) também propuseram uma linearização do modelo apresentado por Vianna *et al.* (2002) para padrões 2-estágios. No entanto, os tempos computacionais com uso do modelo linearizado foram bastante elevados em alguns casos.

Padrões 2-estágios podem ser obtidos através do procedimento proposto por Gilmore e Gomory (1965). No entanto, este procedimento não se baseia em nenhum modelo matemático para padrões 2-estágios, tais como os apresentados em Vianna *et al.* (2002) e Yanasse e Morabito (2003). O procedimento de Gilmore e Gomory (1965) utiliza 2 fases na determinação de padrões 2-estágios. Na primeira fase, resolve-se  $k$  problemas da mochila (onde  $k$  é o total de diferentes tipos de largura). No caso da determinação de padrões 2-estágios exatos, cada problema da mochila corresponde a um grupo de peças de mesma largura (ou comprimento). No caso da determinação de padrões 2-estágios não exatos, cada problema da mochila corresponde a um grupo de peças de largura (ou comprimento) inferior ou igual a cada um dos tipos de larguras (ou comprimentos). Na segunda fase um problema da mochila é resolvido considerando as  $k$  larguras de faixas obtidas na primeira fase. Na Figura 2.7 ilustra-se a determinação de um padrão 2-estágios (tipo não exato) através do procedimento proposto por Gilmore e Gomory (1965):

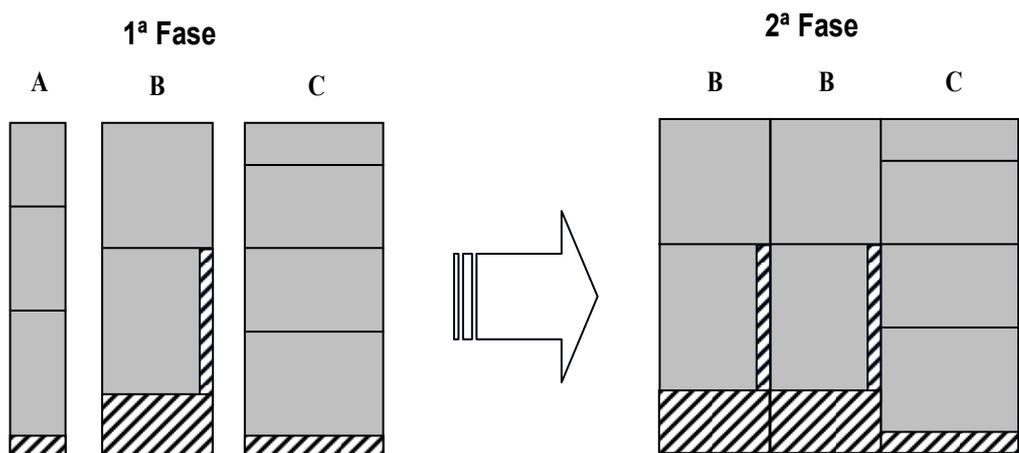


Figura 2.7 – Determinação de um padrão 2-estágios não exato através do método de Gilmore e Gomory (1965).

Muitos outros trabalhos tratam da geração de padrões 2-estágios, por exemplo, Beasley (1985), Hifi e Roucairol (2003), Lodi e Monaci (2003). Limitamo-nos à descrição dos anteriores, pois o foco da tese é em padrões tabuleiros e os trabalhos encontrados na literatura que tratam de padrões 2-estágios parecem não ser adaptáveis, ou extensíveis (pelo menos diretamente) para gerar padrões 1-grupo.

Obviamente, crescendo em ordem de dificuldade de corte existem os padrões 3-estágios, 4-estágios, 5-estágios, ...,  $q$ -estágios.

No caso particular de padrões 3-estágios, Morabito e Arenales (2000) sugerem a utilização de uma heurística baseada no trabalho de Gilmore e Gomory (1965). A heurística também funciona em 2 fases. A diferença com relação ao método de Gilmore e Gomory (1965) está na primeira fase da heurística que passa a permitir a repetição dos itens se houver espaço disponível ao longo da largura (ou comprimento) faixa.

Segundo Morabito e Arenales, a heurística proposta tem produzido bons resultados na indústria de móveis estudada por eles. Na Figura 2.8 ilustra-se um padrão de corte obtido através da heurística para determinação de padrões 3-estágios, proposta em Morabito e Arenales (2000). Note que a peça de largura  $w_j$  é repetida ao longo da largura de faixa  $w_i$ .

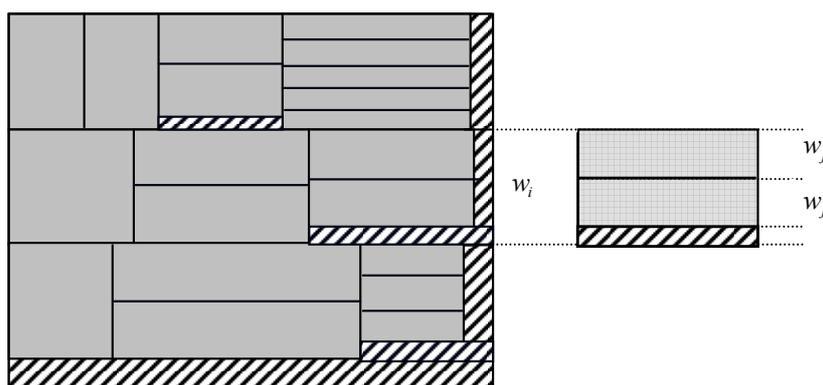


Figura 2.8 – Padrão de corte obtido através da Heurística para determinação de padrões 3-estágios proposta por Morabito e Arenales (2000).

O procedimento sugerido por Morabito e Arenales (1996) utilizando grafos “e/ou” permite a obtenção de padrões sem qualquer limitação no número de estágios, ou de padrões com  $q$ -estágios ( $q$  qualquer). Uma versão deste mesmo procedimento de grafo “e/ou” para lidar com a determinação de padrões não guilhotinados também está presente na literatura e foi apresentado anteriormente em Arenales e Morabito (1995). A estrutura de grafo “e” é utilizada para representar um padrão. O grafo “ou” serve para indicar caminhos diferentes de decomposição. Na Figura 2.9 ilustra-se a aplicação do procedimento de Morabito e Arenales (1996):

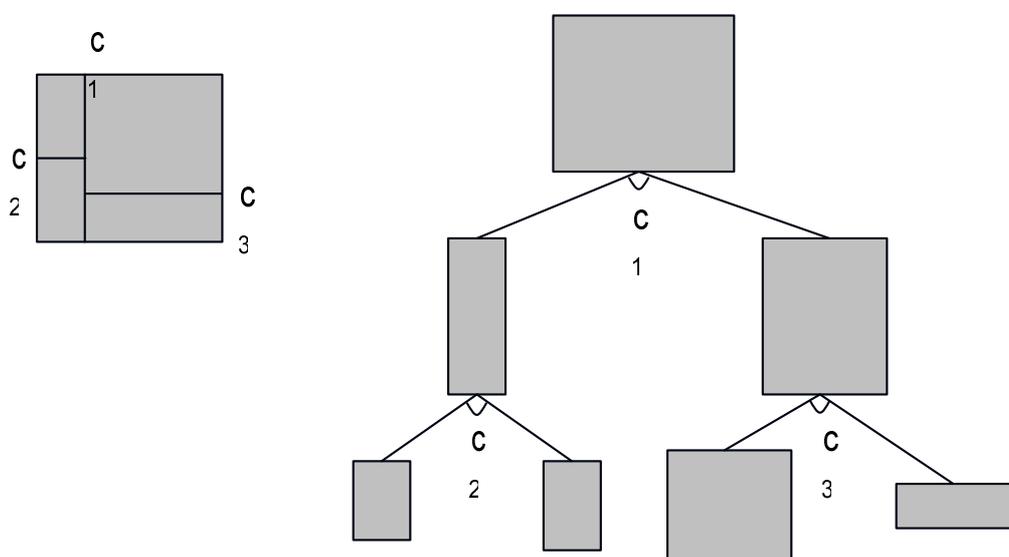


Figura 2.9 – Determinação de um padrão  $q$ -estágios utilizando grafos “e/ou”.  
Fonte: adaptado de Morabito e Arenales (1996).

Eventualmente, padrões tabuleiros podem resultar em elevado desperdício de material, inviabilizando a utilização destes padrões em determinados ambientes reais de produção de cortes. Figueiredo e Rangel (2005) sugerem a utilização de padrões tabuleiros compostos num caso real, extraído da indústria de móveis estudada pelos autores. Os padrões tabuleiros compostos de Figueiredo e Rangel são formados por grupos distintos de 2 ou mais padrões tabuleiros, ou padrões homogêneos, dispostos ao longo da área do objeto (vide Figura 2.10).

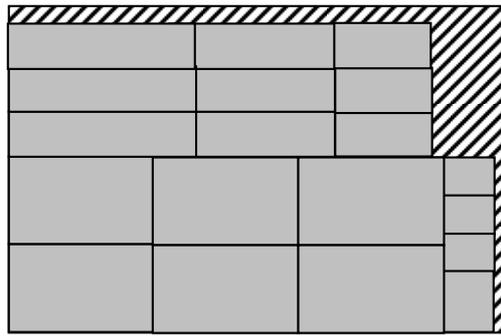


Figura 2.10 – Exemplo de padrão tabuleiro composto.

O corte dos padrões tabuleiros compostos é, obviamente, mais complicado do que o corte de padrões tabuleiros convencionais, porém mais simples do que o corte de muitos dos padrões 2 ou 3 estágios utilizados na fábrica de móveis e, portanto, podem ser utilizados como alternativa para melhorar a produtividade do equipamento de cortes e, ao mesmo tempo, manter o desperdício de material num nível aceitável pela indústria. O corte de padrões tabuleiros compostos em ambientes reais de produção pode ser feito em duas etapas. Numa primeira etapa, os grupos de padrões tabuleiros são separados um a um e, numa segunda etapa, cada um dos grupos obtidos na primeira etapa são individualmente cortados, seguindo o procedimento já tradicionalmente utilizado no corte de padrões tabuleiros.

Devido ao fato dos padrões tabuleiros compostos serem formados por grupos de padrões tabuleiros, Gilmore e Gomory (1965) definiu-os como padrões 2, 3 ou, n-grupos, em referência ao número de grupos de padrões tabuleiros presentes no padrão composto. Para determinação do padrão tabuleiro composto, Figueiredo e Rangel (2005) propuseram uma heurística que parte de padrões homogêneos e insere padrões tabuleiros exatos e não exatos nas sobras dos padrões homogêneos obtidos, com a remoção de algumas de suas faixas (vide Figura 2.11).

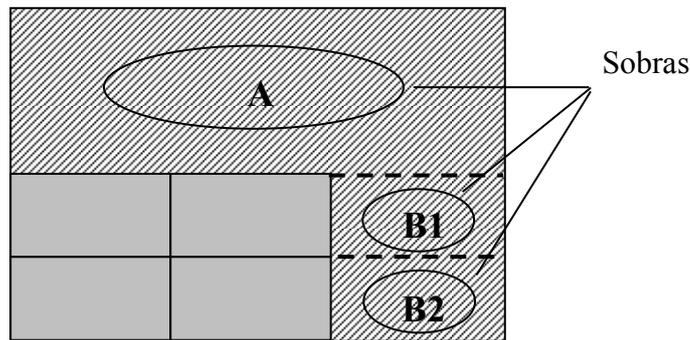


Figura 2.11 – Sobras do padrão.

Fonte: Figueiredo e Rangel (2005).

Segundo Figueiredo e Rangel (2005), o interesse em se partir da determinação de padrões homogêneos está fundamentada numa análise prática previamente feita pelos autores sobre os padrões de corte que já vinham sendo utilizados dentro da indústria estudada. Figueiredo e Rangel (2005) constataram que os padrões preferidos pelos operadores eram padrões tabuleiros, derivados de padrões homogêneos e que estes padrões, quase sempre, apresentavam bons resultados operacionais com índice de perda dentro do limite aceitável pela indústria de móveis. A determinação dos padrões tabuleiros a serem alocados nas sobras se baseou no modelo linear apresentado por Yanasse e Morabito (2003) para determinação de padrões tabuleiros exatos e não exatos restritos. Um resumo da heurística de Figueiredo e Rangel (2005) para determinação de padrões tabuleiros compostos é apresentado a seguir:

**Passo 1:**

*{ Inicialização, leitura e rearranjo dos dados relativos ao objeto e aos itens }*

**1.1 - Itens rotacionados são gerados como novos itens;**

**1.2 - Itens são ordenados (largura/comprimento);**

**Passo 2:**

*{ Geração dos padrões homogêneos e derivados }*

**2.1 - Padrões homogêneos são gerados para cada tipo de item;**

**2.2 - É criada uma matriz de sobras para os padrões homogêneos gerados. As sobras são obtidas a partir da eliminação de algumas das faixas do padrão homogêneo (área A da Figura 2.11) e a partir das sobras que surgem nas faixas horizontais do padrão (áreas B1 e B2 da Figura 2.11);**

### **Passo 3:**

#### *{ Geração dos padrões tabuleiros compostos }*

- 3.1 – As sobras identificadas no Passo 2 são consideradas como um novo objeto (áreas A, B1 e B2 da Figura 2.11);**
- 3.2 - Para cada uma das sobras identificadas no Passo 2, aplica-se o modelo de Yanasse e Morabito (2005) para a determinação de padrões tabuleiros. Sobre a sobra superior (área A da Figura 2.11), é aplicado o modelo para determinação de padrões tabuleiros exatos. Sobre a sobra das faixas horizontais (áreas B1 e B2 da Figura 2.11), é aplicado o modelo para determinação de padrões tabuleiros não exatos.**

No trabalho de Figueiredo e Rangel (2005) , os padrões tabuleiros compostos foram utilizados na resolução do problema de corte de estoque que busca minimizar o número de objetos cortados. Testes computacionais realizados por Figueiredo e Rangel (2005) mostraram que a utilização de padrões tabuleiros compostos na resolução do problema de corte da indústria de móveis estudada resulta em índices de perda melhores do que os índices que vinham sendo obtidos pela empresa e, muito próximos aos índices de perda do software CorteBi (Perin e Rangel, 1989). No entanto, é importante observar que os padrões obtidos pelo software não apresentam as mesmas facilidades operacionais (diga-se produtividade no corte) oferecida pelos padrões tabuleiros compostos que foram obtidos pela heurística proposta.

Em determinados ambientes de produção de cortes, o uso de padrões homogêneos é fundamental. Cui (2007) relata a utilização destes padrões homogêneos, como subconjuntos dos padrões utilizados na indústria de chapas de metal. Nestes ambientes de produção, são utilizados padrões 3-estágios. No entanto, os padrões 3-estágios destes ambientes possuem características particulares que visam a simplificação do processo de corte. A produção dos cortes é feita em duas etapas. Numa primeira etapa, faixas horizontais (ou verticais) são produzidas através de cortes guilhotinados sob a superfície da chapa. Na segunda etapa, é realizado um processo de estampagem para extrair os itens das faixas. O processo de estampagem requer que cada faixa apresente apenas itens do mesmo tipo. Os padrões pertencem, portanto, a uma classe especial de padrões 3-estágios, composto de faixas que apresentam padrões homogêneos. Devido aos cortes do padrão serem semelhantes a um “T” e devido as faixas corresponderem a padrões

homogêneos, Cui (2007) denominou-os padrões “T” homogêneos (homogeneous T-shape patterns). A Figura 2.12 ilustra 2 tipos de padrões T homogêneos:

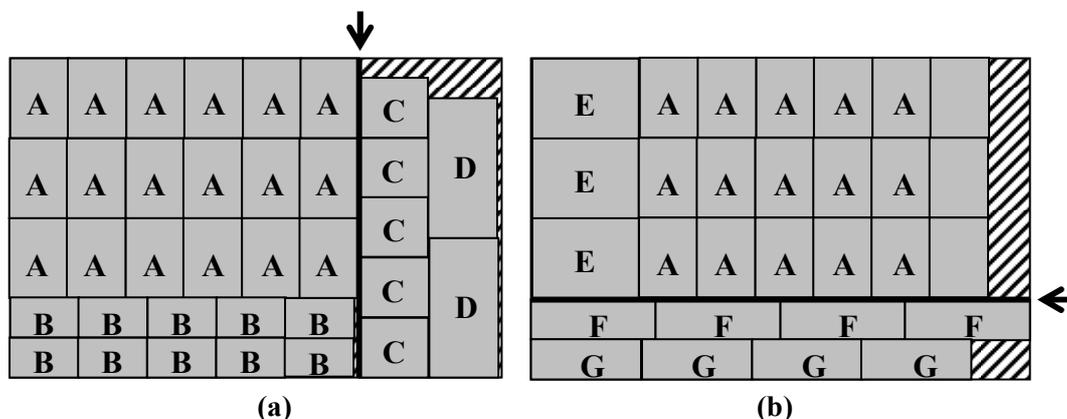


Figura 2.12 – Padrões T homogêneos: (a) padrão TX, (b) padrão TY. Fonte: adaptado de Cui (2007).

Na Figura 2.12a e 2.12b, ilustram-se, respectivamente um padrão T homogêneo tipo TX e um padrão T homogêneo tipo TY. Os padrões da Figura 2.12 utilizam um corte principal (indicado pela seta, na Figura) que os divide em 2 grupos de faixas que são perpendiculares entre si. No padrão TX, o corte principal é feito na direção vertical e, no padrão TY, este corte é feito na direção horizontal. O grupo de faixas dispostas horizontalmente num padrão T é denominado segmento (ou seção) X e, o grupo de faixas dispostas verticalmente denomina-se segmento (ou seção) Y. Para ilustrar melhor, na Figura 2.12a, o grupo de faixas dispostas horizontalmente à esquerda do corte principal corresponde a um segmento (ou seção) X e, o grupo de faixas dispostas verticalmente à direita do corte principal corresponde a um segmento (ou seção) Y.

No trabalho de Cui (2007) foi proposto um algoritmo exato para geração de padrões T homogêneos restritos que se baseia num procedimento branch-and-bound combinado com algumas técnicas de programação dinâmica. O algoritmo se aplica a padrões T homogêneos do tipo TX. Uma extensão para o caso da determinação de padrões T homogêneos tipo TY poderia ser feita, bastando apenas rotacionar o objeto e todos os itens em 90 graus. Seja um objeto de dimensão  $W \times L$  e  $n$  diferentes tipos de itens retangulares menores de dimensão  $w_j \times l_j$  ( $j = 1, \dots, n$ ) e valor de utilidade  $\pi_j$ , Cui

(2007) propôs os seguintes passos básicos para a obtenção de uma solução ótima para o problema da determinação de padrões T homogêneos restritos:

**Passo 1:**

*{ Geração das possíveis faixas horizontais e verticais }*

Para o item tipo  $j$  ( $j = 1, \dots, n$ ) serão geradas  $\min\left\{d_j, \left\lfloor \frac{L}{l_j} \right\rfloor\right\}$  faixas horizontais de largura

$w_j$ , tal que a faixa  $k$  contenha  $k$  itens e seu valor seja  $k\pi_j, 1 \leq k \leq \min\left\{d_j, \left\lfloor \frac{L}{l_j} \right\rfloor\right\}$ . De

forma similar, para o item  $j$ , serão geradas  $\min\left\{d_j, \left\lfloor \frac{W}{w_j} \right\rfloor\right\}$  faixas verticais de largura  $l_j$ ,

tal que a faixa  $k$  contenha  $k$  itens e seu valor seja  $k\pi_j, 1 \leq k \leq \min\left\{d_j, \left\lfloor \frac{W}{w_j} \right\rfloor\right\}$ .

**Passo 2:**

*{ Obtenção dos possíveis segmentos X a partir das faixas horizontais geradas }*

Dada 2 faixas horizontais representadas, respectivamente, por  $a = w_a \times l_a$  e por  $b = w_b \times l_b$ , então, um segmento X, definido por  $c = w_c \times l_c$  é obtido a partir da combinação entre  $a$  e  $b$ , tal que:

$$\begin{aligned} l_c &= \max(l_a, l_b), \\ w_c &= w_a + w_b, \\ l_c &\leq L \text{ e } w_c \leq W. \end{aligned}$$

**Passo 3:**

*{ Obtenção dos possíveis segmentos Y a partir das faixas verticais geradas }*

Dada 2 faixas verticais representadas, respectivamente, por  $a = w_a \times l_a$  e por  $b = w_b \times l_b$ , então, um segmento Y, definido por  $c = w_c \times l_c$  é obtido a partir da combinação entre  $a$  e  $b$ , tal que:

$$\begin{aligned} l_c &= l_a + l_b, \\ w_c &= \max(w_a, w_b), \\ l_c &\leq L \text{ e } w_c \leq W. \end{aligned}$$

#### **Passo 4:**

*{ Geração dos padrões T homogêneos restritos combinando os segmentos X e Y gerados no Passo 3 }*

**Dado o segmento X, definido por  $a = w_a \times l_a$  e o segmento Y, definido por  $b = w_b \times l_b$ , então, um padrão T homogêneo, definido por  $c = w_c \times l_c$  é gerado a partir da combinação entre  $a$  e  $b$ , tal que:**

$$\begin{aligned}l_c &= l_a + l_b, \\w_c &= \max(w_a, w_b), \\l_c &\leq L \text{ e } w_c \leq W.\end{aligned}$$

**Além disto, o número de itens presentes em  $c$  não deverá exceder as demandas.**

Os passos indicados por Cui (2007) podem resultar em elevado tempo computacional, caso a busca pela solução ótima seja feita de forma exaustiva. O autor propõe o uso de limitantes superiores e inferiores para reduzir o número de combinações a serem investigadas. Os limitantes são calculados resolvendo-se, por programação dinâmica, o problema da mochila que maximiza o valor total dos diferentes tipos de faixas homogêneas horizontais/verticais a serem inseridas nas áreas disponíveis anteriores e à direita de um segmento previamente considerado no padrão. Os resultados dos testes computacionais realizados por Cui (2007) mostraram que o algoritmo é eficiente tanto no tempo computacional como no índice de perda de material.

O problema explorado por Cui (2007) limita-se à obtenção de padrões que apresentam, no máximo, 2 segmentos (ou seções), contendo faixas compostas de itens do mesmo tipo. Obter padrões com 3 ou mais segmentos pode não ser interessante sob o ponto de vista da produtividade do equipamento de cortes. Em Fayard e Zissimopoulos (1995) é apresentada uma heurística para determinação de padrões com, no máximo, 2 segmentos irrestritos. A heurística de Fayard e Zissimopoulos se aplica à um caso mais geral da determinação dos padrões T de Cui (2007), onde as faixas verticais e horizontais não precisam ser necessariamente homogêneas. O funcionamento da heurística se baseia na resolução de uma série de problemas da mochila unidimensionais que geram conjunto de faixas que são preenchidas para a obtenção do padrão. Cui *et al.* (2006) propuseram um algoritmo exato que corresponde a uma versão melhorada da

heurística apresentada por Fayard e Zissimopoulos (1995) para determinação de padrões guilhotinados com 2 segmentos. O algoritmo exato de Cui *et al.* (2006) utiliza um método de enumeração implícita para considerar todos os possíveis tamanhos para os segmentos. Segundo os autores, em problemas pequenos, o algoritmo proposto conseguiu obter soluções com valor equivalente às soluções ótimas obtidas para padrões 3 estágios e, em problemas médios e grandes, as soluções obtidas pelo algoritmo ficaram muito próximas destas soluções ótimas.

Algumas observações merecem ser colocadas com relação aos trabalhos desenvolvidos por Fayard e Zissimopoulos (1995), Cui (2007) e Cui *et al.* (2006). É importante notar que embora as soluções obtidas nestes trabalhos apresentem, no máximo, 2 segmentos de faixas por padrão de corte, não existe nenhuma restrição que limita o corte conjunto das faixas obtidas para cada segmento. A ausência desta restrição pode resultar na obtenção de padrões de corte que apresentem um grande número de faixas que precisam ser cortadas individualmente para a produção dos itens. Sob o ponto de vista da produtividade do equipamento de cortes, soluções melhores poderiam ser obtidas se esta restrição fosse levada em consideração, tais como nos métodos e modelos matemáticos explorados na literatura para determinação de padrões tabuleiros.

Os trabalhos citados anteriormente tratam da obtenção de padrões de corte obtidos através de cortes guilhotinados ortogonais. Na literatura sobre padrões de corte especiais, também é possível encontrar trabalhos que tratam da determinação de padrões guilhotinados não ortogonais.

Considere o problema de se cortar uma placa poligonal  $g$  menor a partir de uma placa poligonal  $G$  maior (vide Figura 2.13). Admitindo que a placa  $g$  se encontra numa posição fixa dentro da área de  $G$ , o problema a ser resolvido consiste em encontrar uma seqüência de cortes guilhotinados, cujo comprimento total dos cortes efetuados ao longo da placa seja o menor possível.

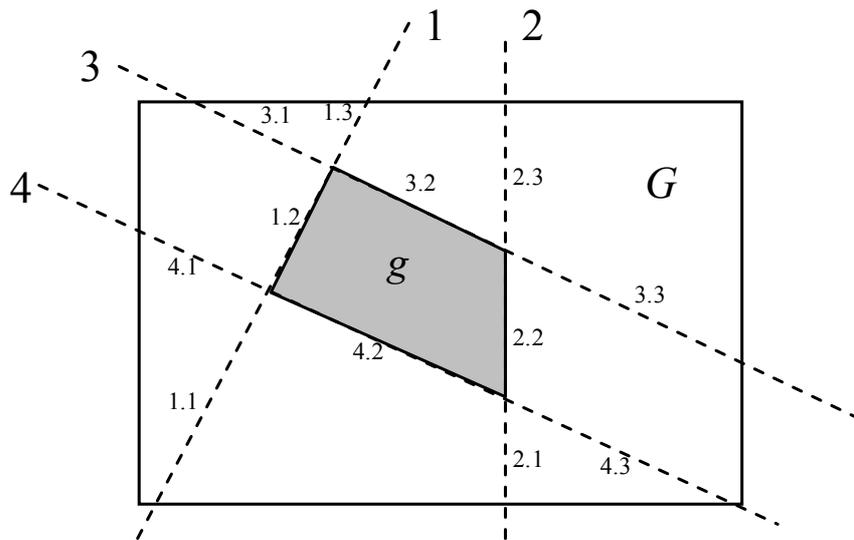


Figura 2.13 – Problema do corte guilhotinado não ortogonal de polígonos.

O problema apresentado aparenta ser de grande aplicação prática. A minimização do comprimento da seqüência de cortes, além de reduzir o tempo de corte de materiais, possibilita também reduzir o desgaste provocado pelo uso da serra.

Em Overmars e Welzl (1985) o problema do corte guilhotinado de polígonos é resolvido considerando-se que os cortes efetuados são sempre paralelos aos lados do polígono convexo  $g$  a ser produzido (vide Figura 2.13). Overmars e Welzl (1985) apresentaram e provaram uma série de propriedades geométricas e teoremas que envolvem o corte de polígonos. Em seguida, um algoritmo de programação dinâmica foi proposto para resolver o problema.

Uma extensão do trabalho de Overmars e Welzl (1985) foi feita por Bhadury e Chandrasekaran (1996) para um caso mais geral, onde os cortes efetuados não precisam ser necessariamente paralelos aos lados do polígono a ser produzido (vide Figura 2.14).

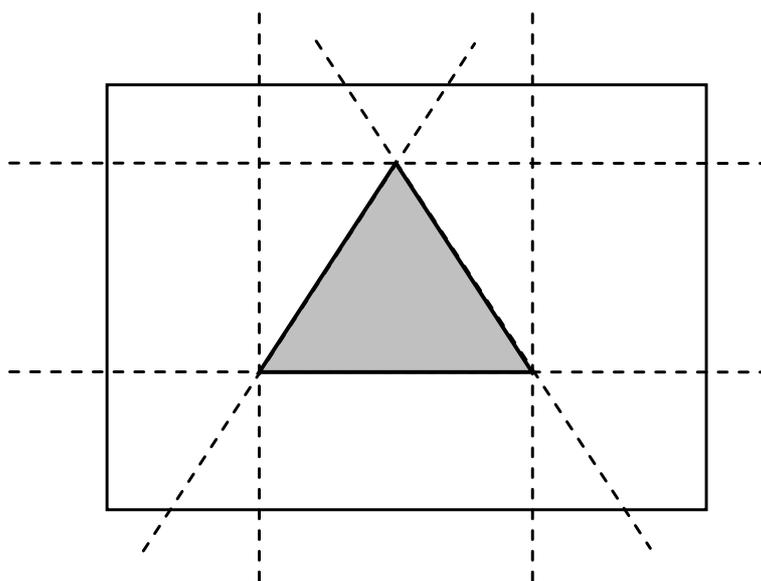


Figura 2.14 – Corte de polígonos: caso onde os cortes efetuados não precisam ser necessariamente paralelos aos lados do polígono a ser produzido.  
Fonte: adaptado de Bhadury e Chandrasekaran (1996).

Segundo Bhadury e Chandrasekaran o caso explorado pode levar à obtenção de uma seqüência de cortes ótima. Bhadury e Chandrasekaran estenderam as propriedades e teoremas propostos por Overmars e Welz (1985) numa série de propriedades trigonométricas que comprovam a possibilidade de obter esta seqüência ótima de cortes. O algoritmo de programação dinâmica também foi estendido para se adequar às novas propriedades introduzidas pelos autores. O problema do corte guilhotinado de polígonos e da redução do comprimento da seqüência de cortes continua sendo explorado na literatura e, recentemente, foram realizados novos avanços sobre este assunto, mais especificamente na área da Geometria Computacional. Para uma revisão bibliográfica mais atualizada, o leitor poderá consultar os trabalhos de Dumitrescu (2004), Daescu e Luo (2004) e Chandrasekaran *et al.* (2005).

Existe obviamente um compromisso do uso de padrões mais simples que melhoram a produtividade da máquina e o desperdício de material. A escolha dos tipos de padrões a serem utilizados nos ambientes de produção depende de uma análise deste compromisso.

Morabito e Arenales (2000) e Katsurayama e Yanasse (2001) foram os únicos trabalhos encontrados na literatura onde é feita esta análise no caso de chapas de madeira bidimensionais. Eles realizaram esta análise para o caso de padrões tabuleiros.

Morabito e Arenales (2000) sugeriram a adição de um custo fixo  $\Delta$  aos padrões mais complexos de serem processados (padrões não tabuleiros) associados à mão de obra ou ao tempo médio extra de processamento de corte. Portanto, seja  $\Pi$ , o custo do objeto cortado, se o padrão não for tabuleiro então o seu custo associado é  $\Pi + \Delta$ , caso contrário, o seu custo é simplesmente  $\Pi$ .

Katsurayama e Yanasse (2001) sugeriram uma função com custo variável, associado ao tempo extra de corte da chapa utilizando padrões 2-estágios não exatos e padrões tabuleiros. O tempo de corte e conseqüentemente seu custo leva em consideração o tempo para rotação e deslocamento da serra ao ponto inicial do corte de cada faixa e a velocidade de corte do equipamento nas direções longitudinal e transversal. Foi também levado em consideração o tempo gasto nas operações de corte extras quando existe a presença de refilos.

Com as funções custo definidas foi realizada uma análise de balanço entre maior produtividade ou menor perda. A análise feita utiliza o procedimento para resolução do problema de corte sugerido por Gilmore e Gomory (1961, 1963, 1965) com geração de colunas. A utilização da função custo de Katsurayama e Yanasse torna a resolução do subproblema de geração de padrões mais difícil pois a função objetivo do subproblema passa a depender agora do padrão de corte a ser gerado o que não acontece com a função custo proposta em Morabito e Arenales. Para contornar esta dificuldade, Katsurayama e Yanasse resolveram o subproblema de geração maximizando o “lucro” dos itens cortados no padrão e apenas posteriormente o custo referente ao corte do padrão foi considerado para verificar se o padrão deve ou não entrar na base. A função custo de Katsurayama e Yanasse, embora ainda simplificada, permite captar com maior riqueza o custo dos padrões tabuleiros e não tabuleiros, comparado com a função custo fixo sugerida em Morabito e Arenales, apesar das soluções obtidas serem “sub-ótimas”.

## CAPÍTULO 3

### ALGORITMOS PARA DETERMINAÇÃO DE PADRÕES TABULEIROS

Como pode ser constatado pela revisão feita, 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 restritos e irrestritos foram encontrados apenas nos trabalhos de Morabito e Arenales (2000), Katsurayama e Yanasse (1999, 2000), Katsurayama (2002), Scheithauer (2002), Yanasse e Morabito (2003) e, mais recentemente, em Yanasse e Morabito (2005).

Katsurayama (2002) sugere que a aplicação de heurísticas, tais como a de Morabito e Arenales (2000), represente uma boa alternativa para se determinar padrões tabuleiros irrestritos, pois apresenta soluções eficientes em termos de qualidade da solução e tempo de processamento. De fato, o uso de heurísticas para determinação de padrões tabuleiros exatos e não exatos irrestritos é justificado quando os itens a serem produzidos apresentam dimensões bastante diferenciadas, pois os padrões ótimos obtidos estão quase sempre associados a soluções triviais, formadas por itens ou faixas do mesmo tipo. Neste caso, procedimentos heurísticos encontram boas soluções, quase sempre ótimas, sem grande esforço computacional. Em outros casos, quando as dimensões dos itens se repetem, a qualidade das soluções ainda pode ser melhorada se forem utilizados procedimentos que possibilitem combinar uma variedade maior de faixas.

Modelos matemáticos para determinação de padrões tabuleiros restritos foram apresentados por Scheithauer (2002) e Yanasse e Morabito (2003, 2005). No entanto, a resolução de problemas de corte reais usando estes modelos requer, geralmente elevado tempo computacional devido ao grande número de variáveis envolvidas. Não se tem conhecimento na literatura de métodos exatos que determinam diretamente padrões tabuleiros restritos, o que é explorado nesta tese. Estes métodos exatos mostraram-se competitivos comparados com a resolução via formulações matemáticas e uso de pacotes comerciais.

### 3.1 – Uma Nova Heurística para Determinação de Padrões Tabuleiros Irrestritos

Propõe-se uma nova heurística para determinação de padrões tabuleiros irrestritos que combina uma maior variedade de faixas em comparação à heurística de Morabito e Arenales (2000). O procedimento heurístico proposto funciona basicamente em duas fases, sendo que na primeira delas repete-se o primeiro passo da heurística de Morabito e Arenales. Mudanças significativas em relação à heurística de Morabito e Arenales ocorrem na combinação das faixas obtidas, quando outras faixas são geradas para serem combinadas com cada uma das faixas  $j$  obtidas na primeira fase. Admitindo-se que a combinação dos itens obtidos no primeiro passo da heurística de Morabito e Arenales tenha sido feita na largura da chapa, as peças que compõem as novas faixas de largura são escolhidas aplicando-se o seguinte procedimento de geração para cada uma das faixas obtidas:

#### **Passo 1:**

**Repita o primeiro passo da heurística de Morabito e Arenales (2000):**

$$V_j = \text{Maximizar } \sum_{i \in \tilde{W}_j} \pi_i v_i$$

$$\text{Sujeito a: } \sum_{i \in \tilde{W}_j} l_i v_i \leq L,$$

$$v_i \geq 0 \quad (i=1, \dots, n) \text{ e inteiro.}$$

Em que  $\tilde{W}_j = \{i \mid w_i = w_j\}$ , para o caso exato e  $\tilde{W}_j = \{i \mid w_i \leq w_j\}$ , para o caso não exato.

#### **Passo 2:**

**Para ( $r \in T_w^j$ ) faça:**

**Início:**

$$V_r^j = 0;$$

$$\text{Para } (s \in \tilde{S}_j) \text{ faça } V_r^j = V_r^j + p_{w_r \times l_s};$$

**Fim-Para.**

**Em que:**

$T_w^j$  corresponde ao conjunto de índices de todas as larguras  $w_j$  possíveis dos itens do problema, excluindo a largura da faixa  $j$ ;

$\tilde{S}_j$  corresponde ao conjunto dos índices de todos os comprimentos possíveis dos itens associados à faixa  $j$ ;

$p_{w_r \times l_s}$  corresponde ao valor do item que caiba no retângulo de dimensões  $w_r \times l_s$ . No caso da determinação de padrões tabuleiros exatos,  $p_{w_r \times l_s}$  equivale ao valor do item de dimensão exata  $w_r \times l_s$ . No caso da determinação de padrões tabuleiros não exatos,  $p_{w_r \times l_s}$  equivale ao valor do item mais lucrativo que possua dimensão inferior ou igual a  $w_r \times l_s$ . Caso não seja possível determinar o valor de  $p_{w_r \times l_s}$ , então considera-se  $p_{w_r \times l_s} = 0$ ;

$V_r^j$  corresponde à soma dos itens mais lucrativos que podem ser inseridos na faixa horizontal, cuja largura é dada por  $w_r$ , tal que um padrão tabuleiro seja determinado quando a faixa  $r$  for combinada com a faixa  $j$ .

Prossegue-se com a segunda fase da heurística resolvendo o problema da mochila para cada uma das faixas obtidas na primeira fase, combinadas com os conjuntos de faixas associados obtidos pelo procedimento de geração das novas faixas. A resolução do problema da mochila é feita na largura do chapa e o lucro associado a cada uma das faixas é calculado pela soma dos lucros das peças que a compõem. Determina-se o padrão tabuleiro escolhendo a melhor solução dentre os problemas da mochila resolvidos na segunda fase. O procedimento seguinte descreve a segunda fase da heurística proposta:

**SolucaoAtual=Melhor padrão homogêneo;**  
**ValorAtual=Valor do padrão associado a SolucaoAtual;**

**Para ( $j \in D_w$ ) faça:**

**Início:**

$$V_j^j = V_j;$$

$$\Phi = \text{Maximizar } \sum_{k \in D_w} V_k^j \psi_k$$

$$\text{Sujeito a: } \sum_{k \in D_w} w_k \psi_k \leq W,$$

$$\psi_k \geq 0, \text{ e inteiro, } k \in D_w.$$

**Se ( $\text{ValorAtual} < \Phi$ ), então faça:**

**Início:**

$$\text{ValorCorrente} = \Phi;$$

$$\text{SolucaoAtual} = \text{Padrão tabuleiro relacionado a } \Phi;$$

**Fim-Se;**

**Fim-Para.**

Em que  $D_w$  corresponde ao conjunto de índices de todas as larguras  $w_j$  possíveis dos itens do problema.

Na Figura 3.1 ilustra-se a obtenção de um padrão tabuleiro não exato com a aplicação da primeira e segunda fase da nova heurística proposta:

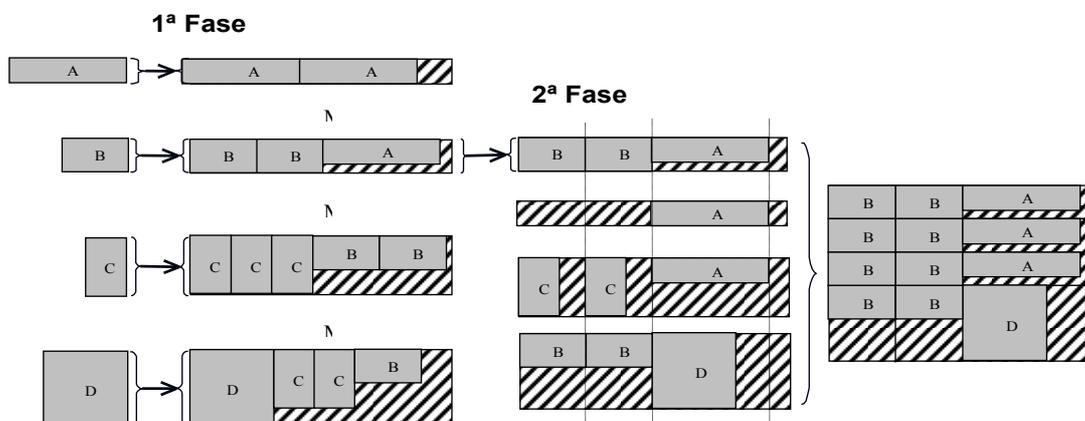


Figura 3.1 – Obtenção de um padrão tabuleiro não exato através da nova heurística proposta.

De forma similar ao apresentado pela heurística de Morabito e Arenales, a composição das faixas poderá ser feita no comprimento do objeto e o melhor padrão tabuleiro é obtido avaliando-se o padrão mais lucrativo em ambas as dimensões.

Cabe salientar que dependendo do tamanho do problema, os conjuntos de itens a serem combinados na primeira fase da nova heurística resultam num problema da mochila composto de um grande número de itens, podendo exigir alto tempo de processamento e grande espaço de memória para armazenamento dos dados. Se for de interesse é possível realizar um pré-processamento selecionando os itens com maior lucro para reduzir o tamanho do problema.

Uma outra adaptação que pode ser feita para reduzir o tempo de processamento da heurística é impor um limitante durante a resolução do problema da mochila na segunda fase. Este limitante utiliza, inicialmente, o valor do melhor padrão homogêneo por unidade de largura ( $\text{Valor do padrão homogêneo}/W$ ). Na primeira fase da heurística, os valores de utilidades (lucros) associados aos itens  $i$  passam a ser calculados pela relação  $\pi_i / w_i$ . O problema da mochila é então resolvido considerando estes novos valores de utilidade e as faixas obtidas, aqui denominadas faixas bases, são colocadas em ordem

não crescente pelo seu valor  $V_j$  calculado. Prossegue-se com a segunda fase da heurística, avaliando-se esta seqüência ordenada de faixas bases e comparando-se os valores  $V_j$  destas faixas com os valores associados aos seus respectivos padrões tabuleiros por unidade de largura. Se o valor do padrão tabuleiro por unidade de largura, associado à faixa base de valor  $V_j$ , for superior ao valor do limitante calculado, então o melhor padrão encontrado até o momento é atualizado para o padrão tabuleiro desta faixa base e o limitante é atualizado pelo valor deste padrão por unidade de largura. Caso contrário, o limitante não será atualizado e a verificação é repetida para a próxima faixa base e seu padrão tabuleiro associado. Este procedimento é executado até que o valor  $V_j$  da faixa base  $j$  avaliada seja inferior ou equivalente ao valor do limitante atualizado. Estas adaptações, realizadas na segunda fase da heurística, podem ser resumidas através do seguinte procedimento:

**SolucaoAtual=Melhor padrão homogêneo;**  
**ValorAtual=Valor do padrão associado a SolucaoAtual;**  
**Limitante=ValorAtual/W;**

**Enquanto ( $V_j > \text{Limitante}$ ,  $j \in D_w$ ) faça:**

**Início:**

$$V_j^j = V_j w_j;$$

$$V_k^j = V_k^j w_k, k \in D_w \text{ e } k \neq j;$$

$$\Phi = \text{Maximizar } \sum_{k \in D_w} V_k^j \psi_k$$

$$\text{Sujeito a: } \sum_{k \in D_w} w_k \psi_k \leq W,$$

$$\psi_k \geq 0, \text{ e inteiro, } k \in D_w.$$

**Se ( $\text{Limitante} < \Phi / W$ ), então faça:**

**Início:**

**ValorAtual =  $\Phi$  ;**

**SolucaoAtual = Padrão tabuleiro relacionado a  $\Phi$  ;**

**Limitante =  $\Phi / W$ ;**

**Fim-Se;**

**Fim-Enquanto.**

Para ilustrar o funcionamento deste novo procedimento, considere um objeto de largura  $W=10$  e comprimento  $L=10$ . O objetivo será determinar o melhor padrão tabuleiro exato e irrestrito que produza os itens da Tabela 3.1. Considere também que os itens da tabela apresentem orientação fixa, ou seja, não é permitido rotacioná-los para a geração do padrão.

Tabela 3.1 – Itens a serem produzidos (exemplo para determinação de um padrão tabuleiro exato e irrestrito através da nova heurística).

$i$	$w_i$	$l_i$	$\pi_i$
1	4	3	20
2	6	4	30
3	6	3	18
4	4	4	15
5	3	4	12
6	4	6	25
7	2	7	14
8	2	4	12
9	3	3	13

Aplicando-se a nova heurística, têm-se as seguintes faixas bases, obtidas a partir da resolução do problema da mochila na primeira fase:

Faixa base de largura 2 (itens compatíveis: 2x4, 2x7):

Solução encontrada: 2 itens 2x4 com valor de utilidade  $\pi = 12 / 2 = 6$

Valor da solução encontrada= $V_1 = 12$

Faixa base de largura 3 (itens compatíveis: 3x4, 3x3):

Solução encontrada: 3 itens 3x3 com valor de utilidade  $\pi = 13 / 3 = 4.3333$

Valor da solução encontrada= $V_2 = 13$

Faixa base de largura 4 (itens compatíveis: 4x3, 4x4, 4x6):

Solução encontrada: 3 itens 4x3 com valor de utilidade  $\pi = 20 / 4 = 5$

Valor da solução encontrada= $V_3 = 15$

Faixa base de largura 6 (itens compatíveis: 6x3, 6x4):

Solução encontrada: 2 itens 6x3 com valor de utilidade  $\pi = 18/6 = 3$ , 1 item 6x4 com valor de utilidade  $\pi = 30/6 = 5$

Valor da solução encontrada= $V_4 = 11$

Colocando as faixas bases em ordem não crescente pelos seus valores  $V_j, j = 1, \dots, 4$ , obtém-se:

Faixa base de largura 4,  $V_1 = 15$

Faixa base de largura 3,  $V_2 = 13$

Faixa base de largura 2,  $V_3 = 12$

Faixa base de largura 6,  $V_4 = 11$

O melhor padrão homogêneo é composto por 10 itens tipo 2x4 e seu valor total é de 120. Portanto, temos:

**SolucaoAtual**=melhor padrão homogêneo,

**Limitante**=valor do melhor padrão homogêneo/ $W=120/10=12$

A primeira faixa base a ser avaliada é a faixa de largura 4, valor 15. O valor desta faixa base é superior ao valor do limitante inicial calculado. Um padrão tabuleiro exato deve, portanto, ser determinado a partir desta faixa base. Obtém-se um padrão tabuleiro de valor 120, cujo valor é equivalente ao padrão homogêneo associado ao limitante inicial calculado. A solução e o limitante não serão atualizados e avalia-se a próxima faixa base. A próxima faixa base apresenta largura 3 e valor 13. O valor desta faixa base é superior ao valor do limitante. Portanto, determina-se o padrão tabuleiro associado a esta faixa. O padrão tabuleiro determinado é composto por 6 itens 3x3 e 3 itens 4x3 e seu valor é 138. O valor deste padrão, por unidade de largura é 13,8, superior ao valor do limitante calculado. Neste caso, atualiza-se a melhor solução encontrada e o limitante para:

**SolucaoAtual**=padrão tabuleiro determinado a partir da faixa base de largura 3,

**Limitante**=13,8

A faixa base seguinte a ser avaliada apresenta largura 2, valor 12. O valor da faixa base é inferior ao valor do limitante atualizado (13.2). Portanto, a heurística é encerrada, sem a necessidade de avaliar as próximas faixas bases. A solução final encontrada pela nova heurística corresponde ao padrão armazenado em **SolucaoAtual**, cujo valor obtido é 138.

Foram realizados testes computacionais com a nova heurística proposta. Os testes já incorporam o limitante proposto anteriormente para reduzir o tempo de processamento da heurística. Para realização dos testes computacionais, foi utilizada uma estação de trabalho Sun UltraSPARC II 296 MHz com 384 MB de RAM, rodando o sistema operacional SunOS 5.6. A heurística proposta foi implementada em linguagem C++. Na Tabela 3.2 apresenta-se os resultados obtidos pela nova heurística e pela heurística de Morabito e Arenales (2000) para instâncias GCUT extraídas do site OR-Library (OR-Library, 2003). Os testes foram realizados visando à determinação de padrões tabuleiros não exatos. Considera-se que os itens possam ser produzidos em quaisquer orientações. Para efeito de comparação, na Tabela 3.2 apresentam-se também os valores ótimos obtidos por Beasley (1985) para padrões 2-estágios exatos e os valores ótimos para padrões tabuleiros exatos obtidos através do algoritmo enumerativo de Katsurayama e Yanasse (1999, 2000).

Tabela 3.2 – Valor da solução e tempo de execução (em segundos) para a heurística de Morabito e Arenales e para a heurística proposta (determinação de padrões tabuleiros não exatos, instâncias OR-Library).

Instância	$n$	Valor Ótimo 2-Estágios Exatos	Algoritmo de Katsurayama e Yanasse (1999, 2000)	Heurística de Morabito e Arenales (2000)	Heurística Proposta	Tempo Heurística de Morabito e Arenales (2000)	Tempo Heurística Proposta
gcut1	10	56460	53808	53808	53808	0.000319	0.000799
gcut2	20	60076	45114	59476	59476	0.001563	0.004186
gcut3	30	60133	52392	56742	57140	0.003394	0.034063
gcut4	50	61698	61008	61034	61034	0.012637	0.066212
gcut5	10	246000	246000	246000	246000	0.000407	0.000865
gcut6	20	235058	207855	209764	219282	0.001295	0.008374

(Continua)

Tabela 3.2 – Conclusão.

<b>Instância</b>	<b><i>n</i></b>	<b>Valor Ótimo 2-Estágios Exatos</b>	<b>Algoritmo de Katsurayama e Yanasse (1999, 2000)</b>	<b>Heurística de Morabito e Arenales (2000)</b>	<b>Heurística Proposta</b>	<b>Tempo Heurística de Morabito e Arenales (2000)</b>	<b>Tempo Heurística Proposta</b>
gcut7	30	242567	216372	230620	230620	0.00326	0.021751
gcut8	50	245758	226000	236822	242656	0.017288	0.058611
gcut9	10	971100	971100	971100	971100	0.000316	0.000864
gcut10	20	982025	894222	934548	934548	0.001561	0.007118
gcut11	30	974638	810540	960148	960148	0.003681	0.019574
gcut12	50	977768	933120	945226	953783	0.016373	0.071459
gcut13	32	-	8.806e+06	8.8725e+06	8.8725e+06	0.00539	0.028869

Conforme pode ser observado na Tabela 3.2, a nova heurística proposta apresentou o melhor padrão tabuleiro não exato em 4 das 13 instâncias testadas. Nas 9 instâncias restantes, as heurísticas obtiveram o mesmo padrão. O tempo de processamento da heurística proposta foi um pouco mais elevado em comparação com o tempo de processamento da heurística de Morabito e Arenales (2000). Testes computacionais também foram realizados com as instâncias OR-LIBRARY usando as heurísticas adaptadas para a determinação de padrões tabuleiros exatos. Os resultados são apresentados na Tabela 3.3:

Tabela 3.3 – Valor da solução e tempo de execução (em segundos) para a heurística de Morabito e Arenales e para a heurística proposta (determinação de padrões tabuleiros exatos, instâncias OR-Library).

<b>Instância</b>	<b><i>n</i></b>	<b>Algoritmo de Katsurayama e Yanasse (1999, 2000)</b>	<b>Heurística de Morabito e Arenales (2000)</b>	<b>Heurística Proposta</b>	<b>Tempo Algoritmo de Katsurayama e Yanasse (1999, 2000)</b>	<b>Tempo Heurística de Morabito e Arenales</b>	<b>Tempo Heurística Proposta</b>
gcut1	10	53808	53808	53808	0.002082	0.000197	0.000429
gcut2	20	45114	45114	45114	0.003364	0.000296	0.00147
gcut3	30	52392	52392	52392	0.004466	0.000617	0.002179
gcut4	50	61008	61008	61008	0.006518	0.000786	0.003256

(Continua)

Tabela 3.3 – Conclusão.

<b>Instância</b>	<b><i>n</i></b>	<b>Algoritmo de Katsurayama e Yanasse (1999, 2000)</b>	<b>Heurística de Morabito e Arenales (2000)</b>	<b>Heurística Proposta</b>	<b>Tempo Algoritmo de Katsurayama e Yanasse (1999, 2000)</b>	<b>Tempo Heurística de Morabito e Arenales</b>	<b>Tempo Heurística Proposta</b>
gcut5	10	246000	246000	246000	0.003774	0.00017	0.000514
gcut6	20	207855	207855	207855	0.006182	0.000293	0.001413
gcut7	30	216372	216372	216372	0.008148	0.000519	0.002032
gcut8	50	226000	226000	226000	0.012656	0.000785	0.003891
gcut9	10	971100	971100	971100	0.007392	0.000171	0.000444
gcut10	20	894222	894222	894222	0.011501	0.000291	0.001277
gcut11	30	810540	810540	810540	0.0157	0.000529	0.002712
gcut12	50	933120	933120	933120	0.024767	0.000807	0.003307
gcut13	32	8.806e+06	8.806e+06	8.806e+06	0.051716	0.000639	0.001834

Pela Tabela 3.3 observa-se que ambas as heurísticas apresentaram os mesmos resultados sendo que elas atingiram o valor ótimo obtido pelo algoritmo enumerativo de Katsurayama e Yanasse (1999, 2000). Isto se deve ao fato dos padrões tabuleiros exatos serem mais restritivos do que padrões tabuleiros não exatos e também pelo fato dos itens das instâncias OR-LIBRARY apresentarem dimensões bastante diferenciadas, ocasionando a determinação de padrões formados por peças ou faixas do mesmo tipo. As duas heurísticas levam em consideração a possibilidade de repetição de faixas do mesmo tipo, portanto, soluções ótimas associadas à padrões tabuleiros exatos puderam ser facilmente obtidos para as instâncias testadas.

Também foram realizados testes computacionais usando as instâncias 1GCUT. Estas instâncias foram geradas exclusivamente para a determinação de padrões tabuleiros exatos e podem ser baixadas na Internet, através do endereço <http://www.lac.inpe.br/~massaru/downloads>. As instâncias 1GCUT apresentam itens com dimensões pouco diferenciadas. Na Tabela 3.4 apresentam-se os resultados obtidos pelo algoritmo enumerativo de Katsurayama e Yanasse (1999, 2000), pela nova heurística e pela heurística de Morabito e Arenales (2000) para as instâncias 1GCUT:

Tabela 3.4 – Valor da solução e tempo de execução (em segundos) para o algoritmo enumerativo de Katsurayama e Yanasse (1999, 2000), para a heurística de Morabito e Arenales e para a heurística proposta (determinação de padrões tabuleiros exatos, instâncias 1GCUT).

<b>Instância</b>	<b><i>n</i></b>	<b>Algoritmo de Katsurayama e Yanasse (1999, 2000)</b>	<b>Heurística de Morabito e Arenales (2000)</b>	<b>Heurística Proposta</b>	<b>Tempo Algoritmo de Katsurayama e Yanasse (1999, 2000)</b>	<b>Tempo Heurística de Morabito e Arenales</b>	<b>Tempo Heurística Proposta</b>
lgcut1	15	9801	9504	9801	0.010046	0.000328	0.000802
lgcut2	15	10000	9900	10000	0.002871	0.00033	0.000598
lgcut3	21	250000	244000	250000	0.007645	0.000475	0.000872
lgcut4	28	9801	9702	9801	0.002888	0.000577	0.001623
lgcut5	28	10000	9900	10000	0.10262	0.000429	0.000902
lgcut6	36	992016	900384	992016	0.021972	0.000914	0.002567
lgcut7	45	10000	9200	10000	0.036943	0.000665	0.001565
lgcut8	45	996004	992012	996004	0.037979	0.003772	0.009809
lgcut9	55	250000	246000	250000	0.054732	0.00254	0.00421
lgcut10	78	10000	9900	10000	0.980424	0.001004	0.002316
lgcut11	91	986049	983070	986049	0.143939	0.010221	0.027243
lgcut12	105	998001	983016	998001	0.264733	0.033185	0.084302
lgcut13	105	1e+06	984000	1e+06	0.082605	0.017641	0.020555
lgcut14	105	1e+06	990000	1e+06	2.41278	0.028313	0.030633
lgcut15	105	2.247e+06	2.23951e+06	2.247e+06	0.222094	0.03113	0.068045
lgcut16	105	2.25e+06	2.214e+06	2.25e+06	0.117022	0.005247	0.007322
lgcut17	561	1e+06	992000	1e+06	2.30109	0.552233	0.560912
lgcut18	820	998001	977022	998001	1.09305	0.418839	0.880009
lgcut19	1035	1e+06	998000	1e+06	1.10963	0.20764	0.227268
lgcut20	1081	998001	959040	998001	1.60578	0.355036	1.01527
lgcut21	1128	2.25e+06	2.2365e+06	2.25e+06	12.4243	0.330573	0.355441
lgcut22	1128	1e+06	998000	1e+06	27.7571	0.254898	0.280106
lgcut23	28	289700	289700	289700	0.010479	0.000522	0.001001
lgcut24	28	976998	929690	976998	0.016991	0.000455	0.001102
lgcut25	45	278980	276425	276425	0.015797	0.00084	0.001988
lgcut26	45	985290	950830	985290	0.027115	0.000845	0.001981
lgcut27	45	2.43775e+06	2.41186e+06	2.43775e+06	0.036394	0.000752	0.001587
lgcut28	55	342796	342796	342796	0.017454	0.003865	0.00473
lgcut29	66	445983	435642	445983	0.095628	0.001421	0.003033
lgcut30	78	2.41713e+06	2.38378e+06	2.39738e+06	0.065665	0.001938	0.004191

(Continua)

Tabela 3.4 – Conclusão.

Instância	$n$	Algoritmo de Katsurayama e Yanasse (1999, 2000)	Heurística de Morabito e Arenales (2000)	Heurística Proposta	Tempo Algoritmo de Katsurayama e Yanasse (1999, 2000)	Tempo Heurística de Morabito e Arenales	Tempo Heurística Proposta
1gcut31	91	319168	312516	319168	0.243199	0.002189	0.004625
1gcut32	105	292894	292894	292894	0.050942	0.002731	0.005103
1gcut33	105	2.52445e+06	2.52445e+06	2.52445e+06	0.086999	0.002548	0.004326
1gcut34	120	373946	362528	373946	0.073303	0.003435	0.005496
1gcut35	136	372944	366450	372944	0.207577	0.005307	0.009596
1gcut36	210	1.12196e+06	1.12196e+06	1.12196e+06	0.120647	0.006964	0.012813
1gcut37	630	1.1837e+06	1.1837e+06	1.1837e+06	1.16616	0.042136	0.061701
1gcut38	1035	2.46784e+06	2.41242e+06	2.45672e+06	0.954547	0.069983	0.115362

Os resultados apresentados na Tabela 3.4 mostram que a heurística proposta só não atingiu o valor ótimo em 3 das 38 instâncias testadas (instâncias 1gcut25, 1gcut30, 1gcut38). Em comparação com a heurística de Morabito e Arenales (2000), a heurística proposta obteve soluções melhores em 31 instâncias e obteve o mesmo resultado em outras 7 instâncias restantes. O tempo computacional da heurística proposta ficou, em média, pouco acima do tempo computacional da heurística de Morabito e Arenales (2000) e abaixo do tempo computacional do algoritmo enumerativo de Katsurayama e Yanasse (1999, 2000). Na Tabela 3.5 apresenta-se os resultados obtidos pelas duas heurísticas, no caso da determinação de padrões tabuleiros não exatos:

Tabela 3.5 – Valor da solução e tempo de execução (em segundos) para a heurística de Morabito e Arenales e para a heurística proposta (determinação de padrões tabuleiros não exatos, instâncias 1GCUT).

Instância	$n$	Heurística de Morabito e Arenales (2000)	Heurística Proposta	Tempo Heurística de Morabito e Arenales	Tempo Heurística Proposta
1gcut1	15	9504	9801	0.000924	0.001944
1gcut2	15	9900	10000	0.001328	0.00192
1gcut3	21	244000	250000	0.003121	0.004291

(Continua)

Tabela 3.5 – (Continuação)

<b>Instância</b>	<b><i>n</i></b>	<b>Heurística de Morabito e Arenales (2000)</b>	<b>Heurística Proposta</b>	<b>Tempo Heurística de Morabito e Arenales</b>	<b>Tempo Heurística Proposta</b>
lgcut4	28	9702	9801	0.003538	0.007491
lgcut5	28	9900	10000	0.00207	0.003384
lgcut6	36	900384	992016	0.008795	0.016246
lgcut7	45	9200	10000	0.011597	0.015102
lgcut8	45	992012	996004	0.05979	0.078396
lgcut9	55	246000	250000	0.038735	0.044398
lgcut10	78	9900	10000	0.01502	0.024292
lgcut11	91	983070	986049	0.884361	0.975084
lgcut12	105	983016	998001	2.05809	2.2538
lgcut13	105	984000	1e+06	0.783955	0.827551
lgcut14	105	990000	1e+06	1.49983	1.53089
lgcut15	105	2.23951e+06	2.247e+06	1.93807	2.11775
lgcut16	105	2.214e+06	2.25e+06	0.229537	0.260392
lgcut17	561	992000	1e+06	178.163	178.843
lgcut18	820	977022	998001	268.749	350.976
lgcut19	1035	998000	1e+06	203.706	208.815
lgcut20	1081	959040	998001	759.818	935.508
lgcut21	1128	2.2365e+06	2.25e+06	274.312	280.018
lgcut22	1128	998000	1e+06	188.927	194.839
lgcut23	28	289700	289700	0.004146	0.005845
lgcut24	28	929690	994534	0.003326	0.006434
lgcut25	45	276425	277271	0.008065	0.019571
lgcut26	45	950830	1.00542e+06	0.019896	0.026604
lgcut27	45	2.41186e+06	2.43775e+06	0.008206	0.011694
lgcut28	55	345500	345616	0.334604	0.434864
lgcut29	66	435642	449087	0.028676	0.045052
lgcut30	78	2.38378e+06	2.44627e+06	0.04754	0.060583
lgcut31	91	312516	323837	0.142276	0.194857
lgcut32	105	292894	292894	0.052593	0.123617
lgcut33	105	2.52445e+06	2.52445e+06	0.084805	0.125793
lgcut34	120	366670	378411	0.072449	0.108656
lgcut35	136	366485	374615	0.239893	0.402665

(Continua)

Tabela 3.5 – Conclusão.

<b>Instância</b>	<b><i>n</i></b>	<b>Heurística de Morabito e Arenales (2000)</b>	<b>Heurística Proposta</b>	<b>Tempo Heurística de Morabito e Arenales</b>	<b>Tempo Heurística Proposta</b>
1gcut36	210	1.13001e+06	1.13041e+06	0.349902	0.849833
1gcut37	630	1.1837e+06	1.1837e+06	13.6674	18.3248
1gcut38	1035	2.45318e+06	2.49004e+06	66.8123	75.5392

Observa-se na Tabela 3.5 que a heurística proposta apresentou resultados ainda melhores do que os resultados apresentados na Tabela 3.4 para o caso da determinação de padrões tabuleiros não exatos. Na determinação de padrões tabuleiros não exatos para as instâncias 1GCUT, a heurística proposta foi melhor que a heurística de Morabito e Arenales (2000) em 34 das 38 instâncias testadas e teve resultado equivalente nas 4 instâncias restantes (instâncias 1gcut23, 1gcut32, 1gcut33, 1gcut37). Padrões tabuleiros não exatos são menos restritivos do que padrões tabuleiros exatos, assim, na determinação de padrões tabuleiros não exatos, o número de combinações de faixas a serem determinadas na segunda fase da heurística é maior, se comparado com este mesmo número de combinações na determinação de padrões tabuleiros exatos. Esta característica parece ter favorecido a heurística proposta nos testes da Tabela 3.5.

Testes computacionais também foram realizados para analisar o impacto da utilização da nova heurística na análise de balanço entre padrões tabuleiros e padrões não tabuleiros como realizada em Morabito e Arenales (2000). Para realização desta análise de balanço, estes autores propuseram a adição de um custo fixo  $\delta$  associado à geração dos padrões não tabuleiros (padrões 2-estágios não exatos) na resolução do Método Simplex com geração de colunas. Os padrões tabuleiros não exatos utilizados na análise de balanço de Morabito e Arenales (2000) foram gerados pela heurística baseada na repetição das faixas, proposta por eles. Nos testes realizados utiliza-se o mesmo conjunto de dados usados em Morabito e Arenales (2000). Considera-se o corte de chapas retangulares de dimensões 1850x3670mm. que possuem um custo unitário de \$1,00. Os dados dos itens a serem produzidos estão apresentados na Tabela 3.6:

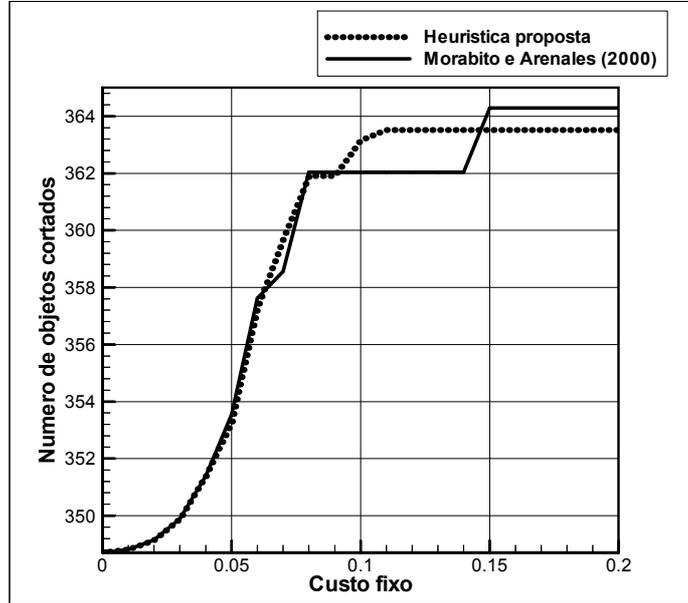
Tabela 3.6 – Itens a serem produzidos.

<i>i</i>	<i>w<sub>i</sub></i>	<i>l<sub>i</sub></i>	<i>d<sub>i</sub></i>
1	274	609	630
2	274	380	1260
3	330	425	630
4	361	650	630
5	270	348	315
6	270	705	893
7	328	718	2520
8	300	705	90
9	330	465	5040
10	330	480	315
11	250	1956	112
12	302	674	118
13	270	674	181
14	270	636	493

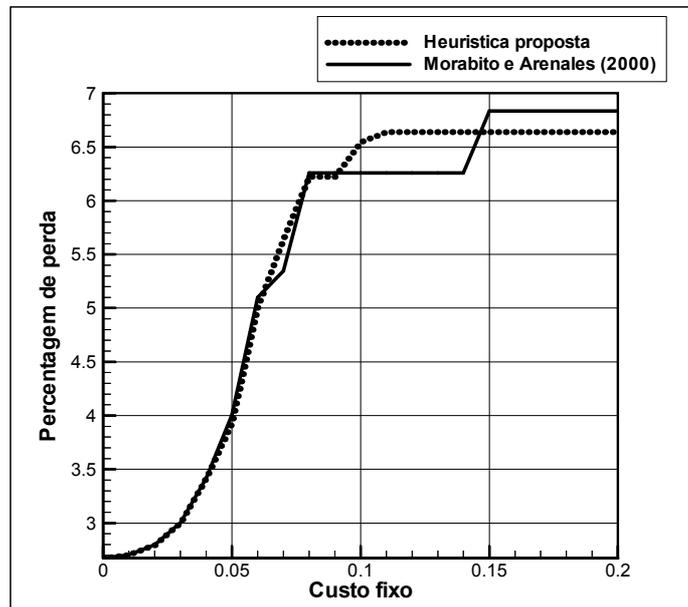
(Reproduzida de Morabito e Arenales, 2000)

De forma similar a Morabito e Arenales (2000), considera-se que os itens possam ser cortados em quaisquer orientações e que o número de chapas em estoque seja suficiente para atender a toda demanda de itens. Deve-se também levar em consideração uma espessura de 4mm. da lâmina de corte que acarreta desperdícios na largura e comprimento da chapa. Nos testes realizados, adotou-se o mesmo critério de parada, tipo de solução e tolerância a erros sugeridos por Morabito e Arenales (2000) na resolução do Método Simplex com geração de colunas. Utilizou-se a solução relaxada do problema para comparação para evitar variações locais devido ao arredondamento das variáveis. O critério de parada foi interromper a execução do algoritmo se nas 10 últimas iterações, o valor da função objetivo não tiver sido reduzido em pelo menos 0.1%. Foi considerada uma tolerância de  $\pm 10^{-6}$  para erro em ponto flutuante. A biblioteca de otimização CPLEX, versão 6.5 foi utilizada para auxiliar na resolução do modelo de programação linear proposto. Nas Figuras 3.2a e 3.2b apresentam-se os resultados da análise de balanço realizada, usando, respectivamente, o número de objetos cortados e a percentagem de perda dos padrões obtidos pela heurística proposta

e pela heurística de Morabito e Arenales (2000) em função de um custo fixo que varia de 0 a 0.2.



(a)



(b)

Figura 3.2 – Número de objetos cortados e perda de material em função do custo fixo: (a) número de objetos cortados em função do custo fixo, (b) perda de material em função do custo fixo.

Como já era de se esperar, nas Figuras 3.2a e 3.2b verifica-se o aumento no número de objetos cortados e na perda de material quando o custo fixo aumenta. Observe, no entanto, que os valores das soluções obtidas com uso da heurística proposta e com uso da heurística de Morabito e Arenales (2000) foram as mesmas para um custo fixo variando entre 0 e 0.05. Isto ocorre devido ao fato dos padrões gerados neste intervalo terem sido os mesmos para ambas as heurísticas. A partir de um custo fixo superior a 0.05, houve variações nas soluções apresentadas. Embora, em alguns casos, o uso da heurística proposta tenha apresentado soluções com maior número de objetos cortados e perda de material do que com uso da heurística de Morabito e Arenales (2000), as soluções de custo mínimo permaneceram sempre melhores ou equivalentes às soluções apresentadas pela heurística de Morabito e Arenales (2000) (vide Figura 3.3).

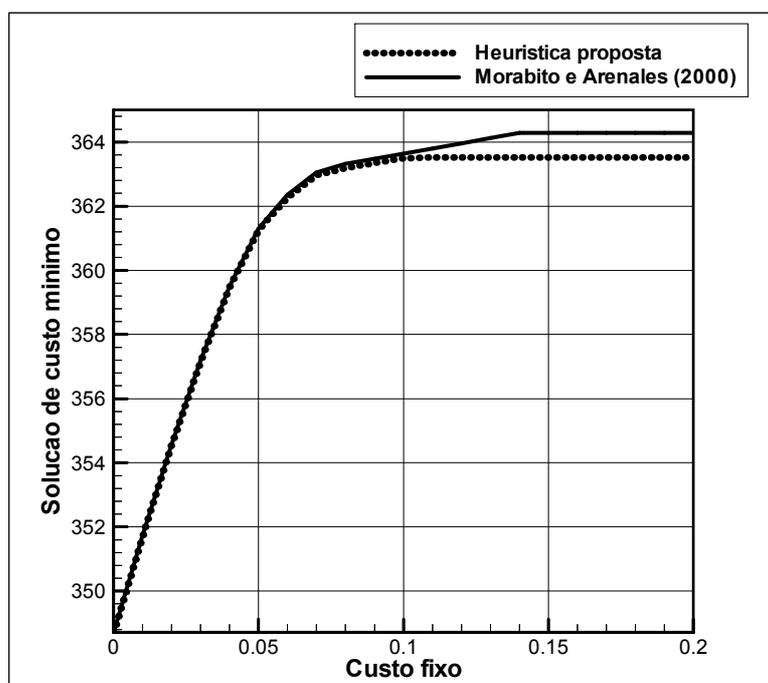


Figura 3.3 – Solução de custo mínimo.

O aumento do número de objetos cortados e da perda de material apresentados na Figura 3.2a e 3.2b se deve ao fato da proporção de padrões não tabuleiros ser reduzida em função do custo fixo. Esta análise é apresentada na Figura 3.4:

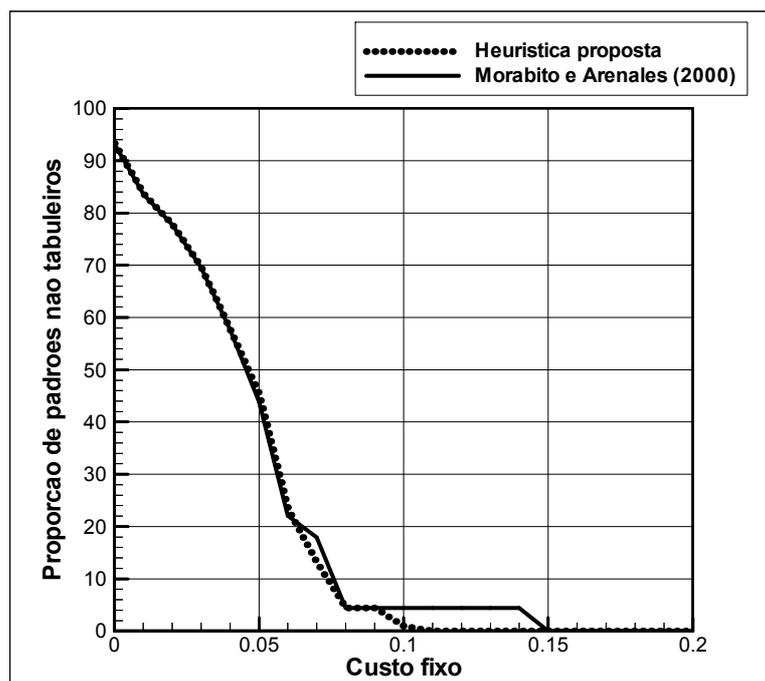


Figura 3.4 – Proporção de padrões não tabuleiros na solução em função do custo fixo.

Na Figura 3.4 também se pode observar que com o uso da heurística proposta é possível manter a proporção de padrões não tabuleiros sempre equivalente ou abaixo do que quando usada a heurística de Morabito e Arenales (2000). Isto ocorre porque a heurística proposta possibilita a geração de uma variedade maior de padrões tabuleiros, se comparada com a heurística de Morabito e Arenales (2000). Certamente, esta possibilidade de gerar uma variedade maior de padrões tabuleiros influenciou na obtenção de soluções de custo mínimo melhores por parte da heurística proposta.

### 3.2 – Um Algoritmo Construtivo para Geração de Padrões Tabuleiros Exatos a Partir de uma Combinação Dada de Itens

Nesta seção, apresenta-se um algoritmo construtivo para a geração de padrões tabuleiros exatos, onde o número de itens de cada tipo a serem produzidos no padrão é dado. O algoritmo fornece um padrão tabuleiro exato com estes itens, caso um exista ou conclui que não existe tal padrão.

Observe que no caso de problemas de corte unidimensionais, a geração do padrão de corte dado o número de itens a serem produzidos é um problema trivial, mas o mesmo

não acontece em problemas com mais dimensões. No caso bi-dimensional, por exemplo, este problema corresponde a resolver um quebra-cabeça, com algumas regras particulares.

Além do interesse inerente de se resolver o problema da geração de padrões tabuleiros a partir de uma combinação dada de itens, o algoritmo proposto resolve também o problema mais geral de geração de padrões tabuleiros exatos restritos. Este problema consiste em, dado um conjunto de itens, determinar o melhor padrão tabuleiro exato que os contenha (parte ou todos estes itens), com alguma restrição adicional sobre os itens. Duas das restrições mais comuns consideradas na prática são, por exemplo, uma limitação no número máximo de itens diferentes no padrão e, uma limitação no número de itens de cada tipo no padrão. No algoritmo construtivo a ser apresentado focaliza-se a segunda restrição.

Considere um objeto retangular de dimensão  $W \times L$  a ser cortado de modo a produzir diferentes tipos de itens retangulares menores de dimensão  $w_k \times l_k$  ( $k=1, \dots, n$ ), em determinadas quantidades  $\lambda_k$ ,  $k=1, \dots, n$ . O problema a ser resolvido consiste em gerar um padrão tabuleiro exato que contenha esta combinação  $\lambda = (\lambda_1, \dots, \lambda_n)$  dada destes itens.

No algoritmo proposto tenta-se identificar a cada passo, que faixas horizontais e/ou verticais devem existir no padrão. Numa primeira fase, estas faixas são identificadas a partir dos itens que devem estar contidos no padrão, ou seja, se existe um padrão tabuleiro exato com a combinação  $\lambda = (\lambda_1, \dots, \lambda_n)$ , o padrão a ser gerado deve ser composto por faixas verticais e horizontais de comprimentos e larguras compatíveis com os comprimentos e larguras dos itens que fazem parte desta combinação, ou seja, faixas verticais e horizontais de comprimentos e larguras compatíveis com os itens correspondentes a  $\lambda_k > 0$ ,  $k=1, \dots, n$ . Seja  $\Omega_w$ , o conjunto dos diferentes tipos de larguras  $w_k$  ( $k=1, \dots, n$ ) e,  $\Omega_l$ , o conjunto dos diferentes tipos de comprimentos  $l_k$  ( $k=1, \dots, n$ ), dos itens existentes na combinação  $\lambda$ , com  $\lambda_k > 0$ ,  $k=1, \dots, n$ . O padrão

tabuleiro, caso exista, deve ter no mínimo, uma faixa vertical de dimensão  $W \times c$ , para todo  $c \in \Omega_l$ , e uma faixa horizontal de dimensão  $b \times L$ , para todo  $b \in \Omega_w$ .

A segunda fase que se segue é aplicada recursivamente, cada vez que novas faixas verticais e /ou horizontais são definidas. As novas faixas são arranjadas ao longo da área disponível no objeto. As faixas verticais e horizontais são arranjadas a partir do canto inferior esquerdo, conforme suas especificações. Com a sua inclusão são definidos de imediato os itens resultantes com os cortes das mesmas. Se o comprimento e/ou largura do objeto não for suficiente para se acomodar todas as faixas, então se conclui imediatamente que não será possível gerar o padrão com aquela combinação de itens e o algoritmo é encerrado. Caso contrário, uma vez arranjadas as faixas, determinamos quais itens resultam do corte das faixas e que fazem parte da combinação e quais são perdas de material. As faixas já definidas definem um retângulo de dimensão  $D \times C$ , localizado na parte inferior esquerda do objeto, que já faz parte da solução final do padrão tabuleiro exato, caso ele exista, como ilustrado na Figura 3.5.

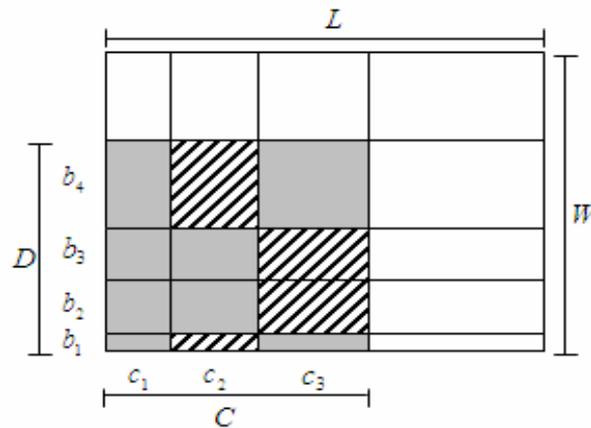


Figura 3.5 – Construção da solução inicial.

Sabemos que, caso seja possível gerar um padrão que atenda às restrições de maneira exata, então a perda estimada para o padrão final pode ser calculada através de

$$\chi = WL - \sum_{k=1}^n \lambda_k w_k l_k .$$

Portanto, se a perda no retângulo  $D \times C$  superar  $\chi$ , concluímos

também de imediato que não será possível gerar o padrão tabuleiro exato desejado e o algoritmo é encerrado. Caso contrário, a combinação  $\lambda$  dos itens ainda a serem

cortados é atualizada subtraindo-se os itens já definidos do retângulo  $D \times C$ . Se não há mais itens na combinação, o padrão tabuleiro exato foi obtido e o algoritmo é encerrado, caso contrário, prosseguimos conforme o que se segue.

A proposta é tentar definir novas faixas verticais e horizontais necessárias para se completar a combinação  $\lambda$  restante. A definição destas novas faixas é feita resolvendo-se o problema do bin-packing unidimensional relativo aos “bins” localizados à direita e acima do retângulo  $D \times C$ . É importante destacar que os problemas de bin-packing não precisam ser resolvidos de maneira exata. Apenas um limitante simples é quase sempre suficiente para estabelecer um número mínimo de “bins” a serem utilizados. Seja  $c$ , o comprimento do item  $r$  (para algum  $r$ ), tal que  $\lambda_r > 0$ . Se, o número de “bins” já existentes de dimensão  $(W - D) \times c$  não for suficiente para acomodar todos os itens restantes com comprimento  $c$  da combinação  $\lambda$ , então é necessário cortar pelo menos mais uma faixa vertical de dimensão  $W \times c$ .

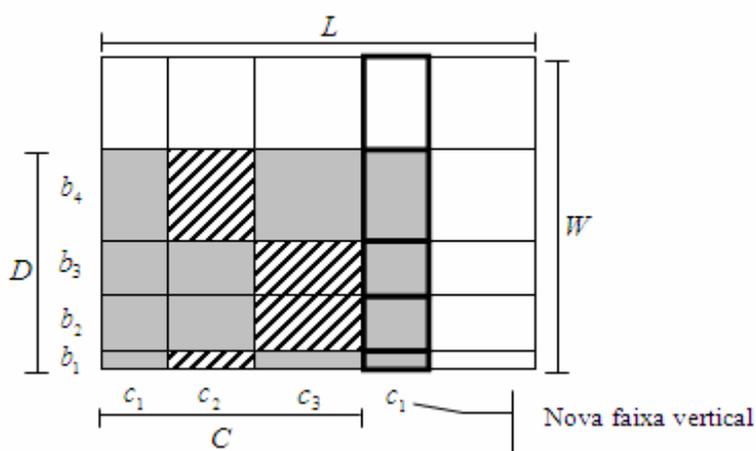


Figura 3.6 – Corte de uma nova faixa vertical.

De forma similar, seja  $b$ , a largura do item  $s$  (para algum  $s$ ), tal que  $\lambda_s > 0$ , então uma faixa horizontal de dimensão  $b \times L$  deverá ser cortada se o número de “bins” de dimensão  $b \times (L - C)$  não for suficiente para acomodar todos os itens da combinação  $\lambda$ , tendo largura  $b$ .

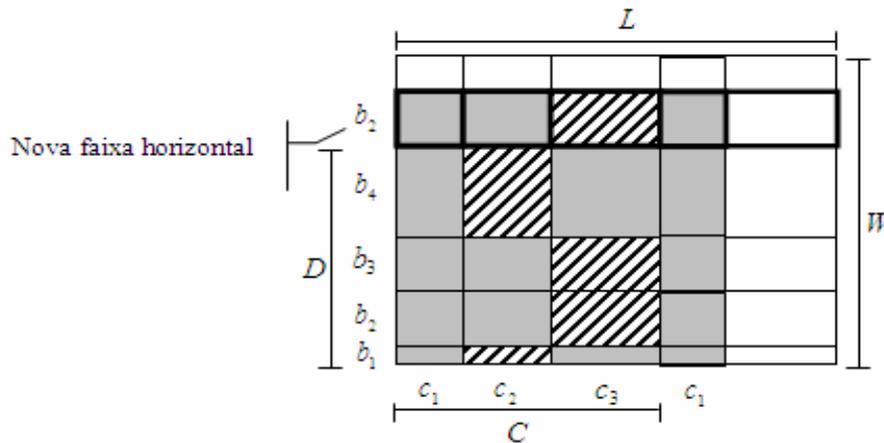


Figura 3.7 – Corte de uma nova faixa horizontal.

Se novas faixas verticais ou horizontais forem definidas, retorna-se a segunda fase do algoritmo. Novos itens serão cortados, novos espaços vazios de perdas serão possivelmente definidos, um novo retângulo  $D \times C$  é definido, a combinação  $\lambda$  é atualizada e verificamos novamente se a geração do padrão tabuleiro ainda é possível.

Este procedimento é aplicado de forma iterativa até que se encontre o padrão tabuleiro, ou se conclua que não é possível obter-se um padrão tabuleiro com aquela combinação ou ainda, chega-se a conclusão de que todos os “bins” de dimensão  $(W - D) \times c$  e de dimensão  $b \times (L - C)$  são, cada um deles, isoladamente, suficientes para acomodar todos os itens restantes da combinação  $\lambda$ . Neste caso, é preciso verificar se eles também são suficientes para cortar todos os itens restantes quando considerados de forma integrada. Entretanto, não se tem agora uma definição se os itens restantes deverão ser cortados de uma faixa vertical ou horizontal. Assim, inicia-se um procedimento de enumeração considerando as duas possibilidades possíveis, a inserção de algum item restante em um dos “bins” disponíveis de mesma largura ou em um “bin” de mesmo comprimento.

Por exemplo, suponha que inicialmente a inserção de um determinado item seja feita num “bin” acima do retângulo  $D \times C$ . Com isto, uma nova faixa horizontal, da largura do item inserido, será definida e podemos retornar à segunda fase do algoritmo. Com isso, o procedimento segue de forma recursiva, até que se encontra uma solução, e finalizamos o algoritmo ou, se não foi possível gerar o padrão através desta chamada recursiva, verificamos a outra possibilidade de inserção do item que é num “bin” à

direita do retângulo  $D \times C$ . Neste caso, uma nova faixa vertical, do comprimento do item inserido, será definida e podemos retornar à segunda fase do algoritmo. O procedimento segue de forma recursiva até que se encontra uma solução, e finalizamos o algoritmo ou, se conclui que não é possível gerar um padrão tabuleiro com a combinação  $\lambda$  e o algoritmo é encerrado.

Para ilustrar o funcionamento do algoritmo, considere um objeto de largura  $W = 155$  e comprimento  $L = 210$ . Tem-se como objetivo determinar se é possível obter um padrão tabuleiro exato contendo 5 itens diferentes, conforme as especificações apresentadas na Tabela 3.7 que se segue.

Tabela 3.7 – Exemplo para geração de um padrão tabuleiro exato.

$k$	$w_k$	$l_k$	$\lambda_k$
1	20	30	4
2	20	35	3
3	30	35	2
4	30	25	2
5	40	20	2

Para este exemplo, temos:

$$\Omega_w = \{20,30,40\},$$

$$\Omega_l = \{20,25,30,35\} \text{ e}$$

$$\lambda = (4,3,2,2,2).$$

Portanto, as faixas horizontais que irão compor o padrão tabuleiro, caso um exista, deverão ter larguras 20, 30, 40 e, as faixas verticais deverão ter larguras 20, 25, 30, 35. Pelo menos uma faixa de cada um desses comprimentos e larguras deverão estar presentes no padrão. Definidas estas faixas, elas são dispostas ao longo do objeto, a partir do canto inferior esquerdo, conforme ilustrado na Figura 6. O corte delas resulta na produção de alguns itens desejados e também de partes já perdidas, que estão também ilustradas na Figura 3.8:

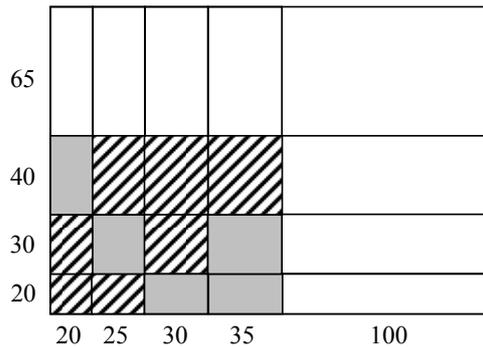


Figura 3.8 – Solução parcial após a primeira fase.

Caso seja possível gerar um padrão tabuleiro  $\lambda$ , então a perda deste padrão deverá ser

$$\chi = WL - \sum_{k=1}^n \lambda_k w_k l_k = 22850 .$$

A perda já definida na solução parcial após a primeira fase é de  $(40 \times 25) + (40 \times 30) + (40 \times 35) + (30 \times 20) + (30 \times 30) + (20 \times 20) + (20 \times 25) = 6000$ . Como  $\chi > 6000$ , não temos evidências de que o padrão tabuleiro não possa ser gerado.

Atualizamos então a combinação  $\lambda$  subtraindo de  $\lambda_k$  ( $k = 1, \dots, n$ ), a quantidade de cada tipo de item produzido no retângulo  $90 \times 110$  da Figura 3.8. A combinação restante  $\lambda$  é  $\lambda = (3, 2, 1, 1, 1)$ , restando portanto, itens a serem cortados. Prosseguimos com a segunda fase do algoritmo, tentando cortar novas faixas verticais e horizontais de modo a completar a demanda restante.

Achamos um limitante inferior para o problema do bin-packing para os “bins” de dimensão  $65 \times 20$ ,  $65 \times 25$ ,  $65 \times 30$ ,  $65 \times 35$ , localizados acima do retângulo  $90 \times 110$  (vide Figura 6). O problema de bin-packing a ser resolvido são para os itens com demanda ainda não atendidos e de comprimentos iguais ao comprimento do “bin”. Verificamos que para os “bins” de dimensão  $65 \times 20$ ,  $65 \times 25$  e  $65 \times 30$ , eles são suficientes para atender a todos os itens de comprimentos 20, 25 e 30, respectivamente. No entanto, o “bin” de dimensão  $65 \times 35$  não é suficiente para atender todos os itens de comprimento 35. Portanto, conclui-se que devemos ter pelo menos mais uma faixa vertical de comprimento 35 para poder cortar todos os itens de comprimento 35. Retornamos então à segunda fase do algoritmo.

O corte de mais esta nova faixa produz o retângulo  $90 \times 145$ , apresentado na Figura 3.9.

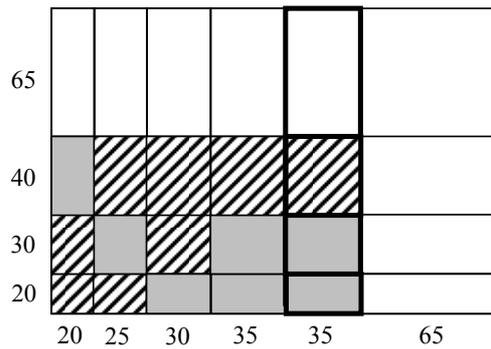


Figura 3.9 – Corte da nova faixa vertical de largura 35.

A perda no retângulo 90x145 com o corte da nova faixa vertical é de  $6000 + (40 \times 35) = 7400$  e continuamos ainda sem evidências de que o padrão tabuleiro não possa ser gerado. Atualizando os valores de  $\lambda$ , obtemos:  $\lambda = (3, 1, 0, 1, 1)$ , ou seja, ainda existem itens a cortar. Agora, todos os “bins” de dimensão 65x20, 65x25, 65x30 e 65x35 são suficientes para acomodar todos os itens de comprimentos 20, 25, 30 e 35. Investiga-se, então, de forma similar, os “bins” à direita do retângulo 90x145.

Determinamos um limitante inferior para o problema do bin-packing para os “bins” de dimensão 20x65, 30x65, 40x65, verificamos que para os “bins” de dimensão 30x65 e 40x65, eles são suficientes para atender a todos os itens de largura 30 e 40, mas o “bin” de dimensão 20x65 é insuficiente para atender a todos os itens de largura 20. Portanto, conclui-se que devemos ter pelo menos mais uma faixa horizontal de largura 20 para poder cortar todos os itens de largura 20. Retornamos então à segunda fase do algoritmo. O corte de mais esta nova faixa produz o retângulo 110x145, conforme apresentado na Figura 3.10.

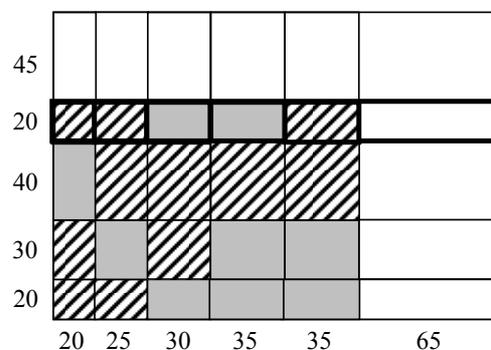


Figura 3.10 – Corte da faixa horizontal de largura 20.

A perda no retângulo 110x145 com o corte da nova faixa horizontal é de  $7400+(20 \times 20)+(20 \times 25)+(20 \times 35)=9000$  e continuamos sem evidências de que o padrão não possa ser gerado. Atualizando  $\lambda$ , obtemos  $\lambda=(2,0,0,1,1)$ , ou seja, restam ainda itens a serem cortados.

Todos os “bins” à direita do retângulo 110x145 são agora suficientes para acomodar todos os itens de respectivas larguras 20, 30 e 40. No entanto, devido ao corte de uma nova faixa horizontal, não podemos garantir que os novos “bins” acima deste retângulo continuam ainda suficientes para acomodar todos os itens a serem cortados. Portanto, a análise destes “bins” deve ser repetida.

Determinamos um limitante inferior para o problema do bin-packing para os “bins” 45x20, 45x25 e 45x30, concluímos que estes “bins” são suficientes para acomodar todos os itens restantes em  $\lambda$ , de comprimento 20, 25 e 45, respectivamente. O procedimento de enumeração é então iniciado. Seleccionemos, por exemplo, o item 5, de tamanho 40x20. Este item deverá ser cortado do objeto através do corte de uma nova faixa horizontal ou vertical. Estas são as duas únicas possibilidades possíveis. Investiguemos inicialmente, a possibilidade de obtê-lo através do corte de uma faixa horizontal. Definida esta faixa, podemos retornar à segunda fase do procedimento. O corte de mais esta nova faixa produz o retângulo 150x145, conforme apresentado na Figura 3.11.

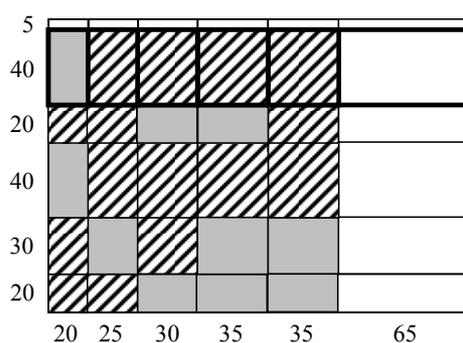


Figura 3.11 – Corte da faixa horizontal de largura 40.

A perda no retângulo 150x145 com o corte da nova faixa horizontal é de  $9000+(40 \times 25)+(40 \times 30)+(2 \times 40 \times 35)=14000$ . Portanto, ainda não temos evidência de que o padrão não possa ser gerado e  $\lambda$  é atualizado para  $\lambda=(2,0,0,1,0)$ .

Como restam itens a cortar, prossegue-se com a segunda fase do algoritmo. Os “bins” de dimensão 5x25 e 5x30 não são suficientes para acomodar os itens restantes, de dimensão de comprimentos 25 e 30, respectivamente. Uma nova faixa vertical de largura 25 é então definida e o corte de mais esta nova faixa produz o retângulo 150x170, conforme apresentado na Figura 3.12.

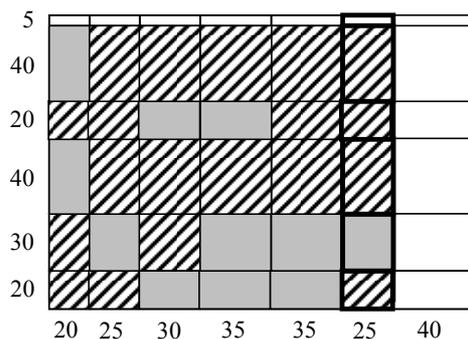


Figura 3.12 – Corte da faixa vertical de largura 25.

A perda no retângulo 150x170 com o corte da nova faixa vertical é de  $14000 + (2 \times 40 \times 25) + (2 \times 20 \times 25) = 17000$ . Portanto, ainda não temos evidência de que um padrão tabuleiro não possa ser gerado e  $\lambda$  é atualizado para  $\lambda = (2, 0, 0, 0, 0)$ .

Como restam itens a cortar, prosseguimos com a segunda fase do procedimento, definindo o corte de mais uma faixa vertical, de comprimento 30. O corte de mais esta nova faixa produz o retângulo 150x200, conforme apresentado na Figura 3.13.

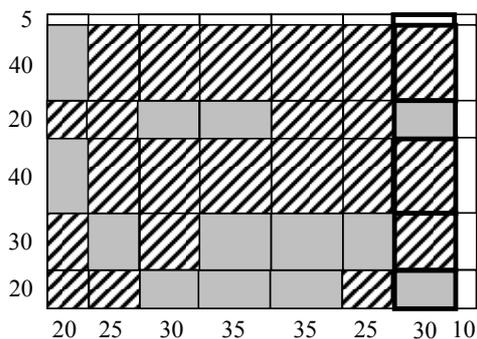


Figura 3.13 – Corte da faixa vertical de largura 30.

A perda no retângulo 150x200 é de  $17000+(2 \times 40 \times 30)+(30 \times 30) = 20300$ . Portanto, sem evidência ainda de que o padrão tabuleiro não possa ser gerado e  $\lambda$  é atualizado para  $\lambda=(0,0,0,0,0)$ .

Não há mais itens a cortar, portanto, um padrão tabuleiro exato foi obtido e o algoritmo pára. O padrão tabuleiro exato final encontrado é o apresentado na Figura 3.14.

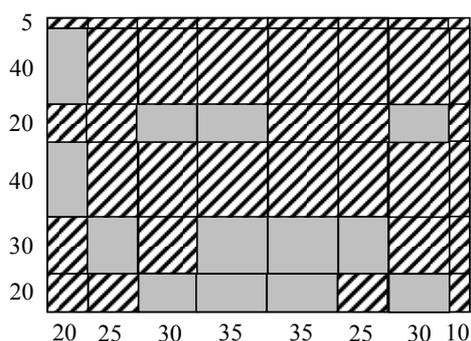


Figura 3.14 – Padrão tabuleiro exato final.

Neste exemplo não foi necessário investigar a outra possibilidade, pois já encontramos uma solução para o problema. Caso este não fosse o caso, passaríamos a investigar a possibilidade de obter o padrão através do corte de uma faixa vertical. Se esta possibilidade não chegasse a uma solução, então concluiríamos que não seria possível obter um padrão tabuleiro exato com a combinação dada.

### 3.3 – Métodos Exatos para Determinação de Padrões Tabuleiros Exatos e Restritos

Os métodos exatos descritos a seguir, utilizam o procedimento apresentado na seção 3.2 para geração de padrões tabuleiros exatos a partir de uma combinação de itens dados. Para uma descrição detalhada destes métodos, considere, como dados de entrada, um objeto retangular de dimensão  $W \times L$  a ser cortado de modo a produzir diferentes tipos de itens retangulares menores de dimensão  $w_i \times l_i$ , lucro (valor de utilidade)  $\pi_i$  e demanda  $d_i$ ,  $i = 1, \dots, n$ .

### 3.3.1 – Algoritmo Enumerativo

O procedimento a ser apresentado se baseia no Algoritmo Enumerativo de Yanasse, Soma e Maculan (2000), aqui denominado YSM, para determinação das K-Melhores soluções para o problema da mochila unidimensional. A opção pela escolha do Algoritmo YSM se deve ao fato de que a enumeração proposta permite considerar restrições adicionais ao problema básico da mochila sem grande esforço computacional adicional. Esta faceta é particularmente interessante em ambientes de corte, pois muitas vezes os padrões de corte são restritivos. No problema estudado, esta restrição equivale ao número limitado de itens no padrão a ser determinado que não deve exceder à sua demanda. A área ocupada por cada um destes itens no padrão é dada por  $\tilde{a}_i = w_i l_i$ . Sabe-se que uma condição necessária que deve ser satisfeita por quaisquer combinações de itens é que a soma total das áreas de todos os itens não deve exceder a área do objeto, o que corresponde a resolver o seguinte problema relaxado:

$$\text{Maximizar } \sum_{i=1}^n \pi_i \lambda_i$$

$$\text{Sujeito a: } \sum_{i=1}^n \tilde{a}_i \lambda_i \leq WL,$$

$$\lambda_i \leq d_i, \quad i = 1, \dots, n,$$

$$\lambda_i \geq 0, \quad i = 1, \dots, n \text{ e inteiro.}$$

No algoritmo proposto, encontra-se a melhor solução para o problema relaxado e verifica-se a possibilidade de gerar um padrão tabuleiro exato para a combinação  $\lambda = (\lambda_1, \dots, \lambda_n)$  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  $\lambda$ , até que o padrão tabuleiro exato seja obtido. Um resumo deste procedimento é apresentado adiante:

**Passo 1:**

*{ Construção da lista com as K-melhores combinações  $\lambda$  }*

Usando o algoritmo YSM, construa uma lista com as K-melhores soluções (combinações) para o problema relaxado;

**Passo 2:**

*{ Escolha da melhor combinação }*

Pegue a melhor combinação da lista construída;

**Passo 3:**

*{ Geração do padrão associado à melhor combinação }*

Usando o Algoritmo Construtivo, 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.

O procedimento citado anteriormente descreve os passos básicos para obtenção do melhor padrão tabuleiro exato e restrito usando o algoritmo enumerativo de Yanasse, Soma e Maculan (2000) que se aplica ao caso irrestrito da resolução do problema unidimensional da mochila. Na prática, a execução destes passos requer algumas modificações no algoritmo YSM para lidar com o caso restrito.

Numa primeira etapa, constrói-se a tabela de enumeração do algoritmo YSM. Na segunda etapa, é feita a recuperação das combinações viáveis desta tabela de enumeração, em ordem decrescente pelo valor. Cada combinação obtida é verificada quanto à possibilidade de geração do padrão tabuleiro exato e restrito correspondente. Se a combinação não atende à restrição da demanda dos itens, então ela é imediatamente descartada e pega-se a próxima melhor. Caso contrário, verifica-se se é possível gerar um padrão tabuleiro exato para esta combinação. Se o padrão tabuleiro for gerado, então o melhor padrão tabuleiro exato e restrito foi encontrado e o algoritmo enumerativo é encerrado. Caso contrário, avalia-se a próxima melhor combinação da tabela. Para ilustrar o funcionamento do algoritmo enumerativo, considere o corte de um objeto de largura  $W=4$  e comprimento  $L=4$ . Deseja-se obter, através do algoritmo

enumerativo, o melhor padrão tabuleiro exato e restrito que considere a produção de 3 tipos de itens, cujos dados estão listados na Tabela 3.8:

Tabela 3.8 – Itens a serem produzidos (exemplo numérico para determinação de padrões tabuleiros exatos e restritos).

$i$	$w_i$	$l_i$	$\pi_i$	$d_i$
1	1	3	9	2
2	2	1	3	3
3	2	3	5	2

Os itens são colocados em ordem crescente pelas suas áreas e o objetivo será resolver o seguinte problema relaxado:

$$\text{Maximizar } 3\lambda_1 + 9\lambda_2 + 5\lambda_3$$

$$\text{Sujeito a: } 2\lambda_1 + 3\lambda_2 + 6\lambda_3 \leq 16$$

$$\lambda_1 \leq 3$$

$$\lambda_2 \leq 2$$

$$\lambda_3 \leq 2$$

$$\lambda_1, \lambda_2, \lambda_3, \geq 0, \text{ e inteiros.}$$

Aplicando o algoritmo enumerativo YSM, obtemos a tabela de enumeração apresentada a seguir:

Tabela 3.9 – Tabela de enumeração gerada pelo algoritmo YSM para o exemplo numérico da Tabela 3.8.

Índice			Lado direito ( $\bar{b}$ )
1	2	3	
0	0	0	0
-1	-1	-1	1
3	-1	-1	2

(Continua)

Tabela 3.9 – Conclusão.

Índice			Lado direito ( $\bar{b}$ )
1	2	3	
-1	9	-1	3
6	-1	-1	4
-1	12	-1	5
9	18	5	6
-1	15	-1	7
12	21	8	8
-1	27	14	9
15	24	11	10
-1	30	17	11
18	36	23	12
-1	33	20	13
21	39	26	14
-1	45	32	15
24	42	29	16

Parte das soluções a serem recuperadas da Tabela 3.9, estão listadas na Tabela 3.10, em ordem não crescente de valor:

Tabela 3.10 – Soluções a serem recuperadas da Tabela 3.9 (em ordem não crescente de valor).

Solução (valor)	Lado direito ( $\bar{b}$ )	Índice
45	15	2
42	16	2
39	14	2
36	12	2
33	13	2
32	15	3
30	11	2
29	16	3
27	9	2
26	14	3
24	10	2
24	16	1
23	12	3
21	8	2

Avaliando-se as soluções 45, 42 e 39 representadas na Tabela 3.10, nota-se que apesar destas soluções estarem presentes na tabela de enumeração (Tabela 3.9), nenhuma delas

(incluindo soluções alternativas obteníveis no processo de recuperação destas soluções) atende à restrição de demanda e, portanto, elas são descartadas. A próxima solução da Tabela 3.10, apresenta valor 36 e ela também não atende à restrição de demanda e será descartada. No entanto, uma de suas soluções alternativas de valor 27 atende à restrição de demanda. Esta solução alternativa apresenta 3 itens 1x3 e 2 itens 2x1 e não gera um padrão tabuleiro viável. Portanto esta solução alternativa também é descartada.

A próxima melhor solução da Tabela 3.10, possui valor 33 e não apresenta nenhuma solução que atenda à restrição de demanda e é descartada. A solução seguinte da Tabela 3.10, de valor 32, apresenta duas soluções alternativas de valores 19 e 23 que atendem às restrições de demanda, mas nenhuma delas gera um padrão tabuleiro exato e, portanto, as duas soluções são descartadas. A próxima solução, de valor 30 não apresenta nenhuma solução (incluindo as soluções alternativas) que atenda às restrições e é descartada. A solução seguinte, de valor 29, atende à restrição de demanda, mas não gera um padrão tabuleiro exato. No entanto, ela apresenta uma solução alternativa de valor 16 que atende à restrição de demanda e que gera um padrão tabuleiro exato. Portanto, esta solução de valor 16 será guardada e corresponde à melhor solução encontrada até o momento.

Apesar de termos obtido uma solução para o problema, não podemos encerrar o procedimento, pois não obedecemos a ordem correta de verificação das combinações. Existem ainda algumas combinações que tem valores superiores aos da solução corrente. Assim, continuando o procedimento de recuperação das soluções, a próxima solução da Tabela 3.10 apresenta valor 27, superior ao valor da melhor solução armazenada (16) e, portanto, deve-se tentar recuperá-la. A solução de valor 27 não atende à restrição de demanda e será descartada, mas uma solução alternativa, de valor 18 atende à esta restrição. Infelizmente esta solução alternativa de valor 18 não resulta em um padrão tabuleiro viável e é descartada. A próxima solução da tabela, de valor 26 atende à restrição de demanda e gera um padrão tabuleiro viável. A melhor solução encontrada até o momento será atualizada para 26 e avalia-se a próxima solução da tabela. A solução seguinte da tabela apresenta valor 24, inferior ao valor da solução armazenada. Portanto, o algoritmo enumerativo é encerrado e a melhor solução

encontrada corresponde à solução armazenada, de valor 26. Na Tabela 3.11 apresenta-se um resumo dos resultados obtidos pelo procedimento de recuperação das soluções:

Tabela 3.11 – Resultado obtido pelo procedimento de recuperação das soluções.

<b>Solução (valor)</b>	<b>Soluções Recuperadas que Atendam à Restrição de Demanda</b>
45	Nenhuma
42	Nenhuma
39	Nenhuma
36	Solução alternativa de valor 27 (solução não gera um padrão tabuleiro)
33	Nenhuma
32	Solução alternativa de valor 19 (solução não gera um padrão tabuleiro exato) Solução alternativa de valor 23 (solução não gera um padrão tabuleiro exato)
30	Nenhuma
29	Solução alternativa de valor 16 (solução <b>GERA</b> um padrão tabuleiro exato) Solução alternativa de valor 29 (solução não gera um padrão tabuleiro exato) Solução de valor 16 é armazenada.
27	Solução alternativa de valor 18 (solução não gera um padrão tabuleiro exato)
26	Solução 26 <b>GERA</b> um padrão tabuleiro exato Solução armazenada é atualizada para 26
24	-

Note que a tabela de enumeração apresentada (Tabela 3.9) foi construída a partir da resolução do problema irrestrito. Uma grande parte das soluções da tabela, avaliadas na segunda etapa do algoritmo é descartada por elas não atenderem à demanda dos itens. Pode-se reduzir o número de soluções a serem descartadas restringindo-se o processo de enumeração da primeira etapa do algoritmo levando em conta que a demanda dos itens precisa ser satisfeita.

Foram realizados testes computacionais para avaliar a eficiência do algoritmo proposto e compará-lo com o modelo de Yanasse e Morabito (2003). O modelo de Yanasse e

Morabito (2003) é baseado na formulação não linear de Morabito e Arenales (2000) para padrões tabuleiros irrestritos. Segundo Yanasse e Morabito (2003), este modelo foi o que apresentou o melhor desempenho dentre os modelos lineares testados pelos autores na determinação de padrões tabuleiros restritos.

O algoritmo enumerativo foi implementado em linguagem C++ e os testes foram executados 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 pelos autores. Na Tabela 3.12 apresentam-se os resultados obtidos para algumas instâncias extraídas de Yanasse e Morabito (2003). Os tempos computacionais estão expressos em segundos:

Tabela 3.12 – Algoritmo enumerativo x modelo de Yanasse e Morabito (2003) (Instâncias extraídas de Yanasse e Morabito, 2003).

<b>Instância</b>	<b><i>n</i></b>	<b>Valor da Solução Ótima</b>	<b>Tempo Algoritmo Enumerativo</b>	<b>Tempo Modelo de Yanasse e Morabito (2003)</b>
1	5	7800*	6.89	0.32
2	5	8280*	6.99	0.35
3	5	7000*	3.66	0.29
4	5	7566	1.48	0.39
5	5	7936*	7.50	0.35
6	5	9306	0.39	0.29
7	5	4770	9.67	0.55
8	5	7904*	6.74	0.35
9	5	7800*	1.13	0.26
10	5	5810	5.47	0.35

\* Solução homogênea

Conforme pode ser observado na Tabela 3.12, o tempo computacional do algoritmo enumerativo foi bastante elevado, comparado com o tempo computacional utilizado na resolução do modelo de Yanasse e Morabito (2003).

Refinamentos foram introduzidos no algoritmo enumerativo visando melhorar seu tempo computacional. No algoritmo proposto, é construída uma matriz de dimensão  $n \times WL$  para enumerar as possíveis combinações de itens que devem ser recuperadas a

partir da própria tabela. O algoritmo, portanto, apresenta um requisito de memória de ordem de  $O(WLn)$ . Em problemas práticos, onde as dimensões dos objetos podem 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 é 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.

Nos testes computacionais que se seguem, foram utilizados dois conjuntos de instâncias extraídas da literatura para avaliar novamente o algoritmo enumerativo. Os testes já levam em consideração a mudança de escala nos dados de entrada do problema para redução do tempo computacional.

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). Na Tabela 3.13 apresentam-se os resultados obtidos para o primeiro conjunto de instâncias. Os tempos computacionais estão expressos em segundos:

Tabela 3.13 – Algoritmo enumerativo 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).

<b>Instância</b>	<b><i>n</i></b>	<b>Valor da Solução Ótima</b>	<b>Tempo Algoritmo Enumerativo</b>	<b>Tempo Modelo de Yanasse e Morabito (2003)</b>
1	5	7800*	0.40	0.32
2	5	8280*	0.17	0.35
3	5	7000*	0.14	0.29
4	5	7566	0.08	0.39
5	5	7936*	0.07	0.35
6	5	9306	0.03	0.29
7	5	4770	0.04	0.55
8	5	7904*	0.03	0.35
9	5	7800*	0.03	0.26
10	5	5810	0.01	0.35

(Continua)

Tabela 3.13 - Conclusão.

<b>Instância</b>	<b><i>n</i></b>	<b>Valor da Solução Ótima</b>	<b>Tempo Algoritmo Enumerativo</b>	<b>Tempo Modelo de Yanasse e Morabito (2003)</b>
11	5	63*	0.002	0.25
12	5	90	0.005	0.26
13	5	70	0.001	0.29
14	5	80	0.001	0.23
15	5	90	0.012	0.26
16	5	78	0.0006	0.29
17	5	100*	0.0002	0.14
18	5	82	0.002	0.28
19	5	100*	0.003	0.16
20	5	70	0.002	0.28
VAG	5	6450	0.04	0.34
MA	13	6715500	**	9.99
CW1	7	240	0.009	0.19
CW2	10	1687	1.28	0.38
CW3	20	1080*	1.44	1.16
W	20	2277*	1.45	1.96
OF1	10	1769	0.33	0.47
OF2	10	1684	0.21	0.43

\* Solução homogênea

\*\* não encontrou uma solução no limite de tempo fixado em 3600 segundos

Note que as dez primeiras instâncias da Tabela 3.13 correspondem às mesmas instâncias avaliadas nos testes da Tabela 3.12. Observe que houve uma grande redução do tempo computacional destas instâncias já avaliadas. De todas as instâncias avaliadas nos testes da Tabela 3.13, apenas 1 instância (instância MA, de Morabito e Arenales, 2000) não teve a solução obtida pelo algoritmo enumerativo no limite de tempo fixado em 3600 segundos. No entanto, o algoritmo enumerativo foi mais rápido do que o modelo de Yanasse e Morabito (2003) em 13 das 18 instâncias avaliadas.

O segundo conjunto de instâncias foi extraído de Hifi (2005). Na Tabela 3.14 apresentam-se os resultados obtidos para o segundo conjunto de instâncias testadas:

Tabela 3.14 – Algoritmo enumerativo: soluções obtidas (instâncias extraídas de Hifi, 2005).

<b>Instância</b>	<b><i>n</i></b>	<b>Valor da Solução Ótima</b>	<b>Tempo (em segundos) Algoritmo Enumerativo</b>
HH	5	7020*	0.02
2	10	1687	1.28
3	20	1080*	1.44
A1	20	1040*	2.20
A2	20	1560	11.88
STS2	30	3800	**
CHL1	30	6579	**
CHL2	10	1226	0.28
CW1	25	5358	922.45
CW2	35	2964*	526.98
CW3	40	4428*	**
Hchl2	35	7408	**
Hchl9	35	4250	**
2s	10	1689	1.26
3s	20	2277*	1.37
A1s	20	1750*	2.20
A2s	20	2714	11.85
STS2s	30	3952	**
STS4s	20	7614	**
CHL1s	30	12276	**
CHL2s	10	1922	0.27
A3	20	4524	115.26
A4	20	3180	195.24
A5	20	12276	**
CHL5	10	220	0.099
CHL6	30	13113	**
CHL7	35	12958	**
CU1	25	12200	381.349
CU2	35	20250	1205.92
Hchl3s	10	7388	352.01
Hchl4s	10	7388	13.37
Hchl6s	22	40835	**
Hchl7s	40	41869	**
Hchl8s	10	416	0.68

\* Solução homogênea

\*\* não encontrou uma solução no limite de tempo fixado em 3600 segundos

Conforme pode ser observado na Tabela 3.14, o algoritmo enumerativo falhou na determinação do padrão tabuleiro restrito em 13 das 34 instâncias testadas. A eficiência do algoritmo enumerativo proposto na determinação dos padrões parece estar

relacionada com os dados de entrada do problema. Em Katsurayama (2002), foi realizado um estudo para analisar os diversos fatores que podem influenciar no tempo computacional de um algoritmo para determinação de padrões tabuleiros exatos baseado no algoritmo enumerativo de Yanasse, Soma e Maculan (2000). Embora o estudo realizado por Katsurayama (2002) tenha sido focado apenas na determinação de padrões tabuleiros irrestritos, o algoritmo enumerativo para determinação de padrões tabuleiros restritos proposto nesta tese de doutorado utiliza um procedimento de enumeração similar ao algoritmo apresentado pelo autor. Segundo Katsurayama (2002), o tamanho relativo, a frequência e o número de diferentes tipos de itens são alguns dos fatores que podem influenciar no aumento ou redução do tempo de processamento do algoritmo enumerativo. Ainda assim, o algoritmo enumerativo conseguiu obter soluções em boa parte do segundo conjunto de instâncias avaliado.

### **3.3.2 – Algoritmo de Enumeração Implícita**

O próximo algoritmo desenvolvido baseia-se no método da enumeração implícita de Gilmore e Gomory (1963). A vantagem deste algoritmo de enumeração implícita sobre o algoritmo enumerativo apresentado anteriormente é a possibilidade de interromper sua execução em qualquer instante e ser possível recuperar uma solução para o problema. No algoritmo enumerativo, a solução só é obtida no final de sua execução.

Gilmore e Gomory (1963) aplicaram um método de enumeração implícita para resolver 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 foi embutido no novo algoritmo. O algoritmo de enumeração implícita proposto neste trabalho mantém as mesmas idéias introduzidas por Gilmore e Gomory (1963), ou seja, o método proposto se baseia num esquema de 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 percorrer todas as soluções do problema.

De forma similar ao método da enumeração implícita de Gilmore e Gomory (1963), o algoritmo proposto também pode ser representado através de uma árvore de

enumeração. As combinações  $\lambda$  de itens são enumeradas nesta árvore de enumeração, de modo que cada nó da árvore representa uma combinação de itens associada a um padrão tabuleiro viável, gerado pelo algoritmo construtivo.

Inicia-se o algoritmo colocando-se os itens em ordem decrescente pelo seu valor de utilidade por unidade de área e, com base nesta seqüência ordenada de itens, é feita uma busca em profundidade primeiro na árvore de enumeração adicionando-se os diferentes tipos de itens aos padrões tabuleiros das combinações representadas em cada um dos nós. O procedimento obtém uma solução sempre melhorada para os nós seguintes, de modo que, para cada um destes nós, uma nova combinação de itens seja construída com base na inclusão de um novo tipo de item no espaço restante (sobra) do padrão tabuleiro associado à combinação representada no nó precedente.

Seja  $X$ , o espaço restante (sobra) do padrão, calculado através de  $X = X - \sum_{j=1}^i w_j l_j d_j$ ,

onde  $d_i$  é a demanda associada ao item  $i$  ( $i = 1, \dots, n$ ) e, inicialmente,  $X = WL$  e  $i = 1$ , a frequência  $\lambda_i$  associada a cada item  $i$  no padrão é calculada por

$$\lambda_i = \min \left\{ \left\lfloor \frac{X}{w_i l_i} \right\rfloor, d_i, u_i \right\}.$$

Em que  $u_i$  corresponde ao número de itens do tipo  $i$  que podem ser colocados no padrão e, ainda assim, o padrão puder ser gerado. Um valor para  $u_i$  pode ser determinado através de um procedimento de busca por biseção, que visa encontrar um valor inteiro  $f$

que gere um padrão tabuleiro exato, tal que  $f = \max(j)$ ,  $j = 0, 1, 2, \dots, \min \left\{ \left\lfloor \frac{X}{w_i l_i} \right\rfloor, d_i \right\}$ .

Seja  $\theta = \min \left\{ \left\lfloor \frac{X}{w_i l_i} \right\rfloor, d_i \right\}$ . A busca por biseção verifica, inicialmente, se um padrão

tabuleiro exato pode ser gerado considerando a inserção de  $\theta$  itens no espaço restante  $X$  do padrão determinado anteriormente. Se a geração do padrão não for possível, então o

procedimento verifica se esta geração é possível inserindo-se  $\frac{\theta}{2}$  itens em  $X$ . Se sim, então é verificado se a geração ainda é possível inserindo-se uma quantidade de itens equivalente à metade entre  $\frac{\theta}{2}$  e  $\theta$  itens. Em caso negativo, a verificação é feita na metade entre 0 e  $\frac{\theta}{2}$ . A busca por biseção prossegue dividindo os intervalos e repetindo esta mesma verificação até que o intervalo se reduz a um único valor.

Através deste procedimento para cálculo da frequência  $\lambda_i$ , busca-se inserir o número máximo de itens do tipo  $i$  no espaço restante  $X$ , partindo sempre do item que possui o maior valor de utilidade por unidade de área. Os itens são inseridos até que não haja mais espaço suficiente para acomodá-los ou, até que todos os tipos de itens tenham sido considerados.

Conforme já descrito por Gilmore e Gomory (1963), a solução encontrada na busca em profundidade primeiro é uma boa solução, mas não necessariamente ótima do problema de corte. A otimalidade deve ser garantida verificando-se todos os outros possíveis caminhos da árvore de enumeração, considerando a remoção de um tipo de item por vez e a inclusão de outros tipos de itens que ainda não foram considerados. Infelizmente, dependendo das características do problema de corte, o número de combinações a serem investigadas pode ser muito grande, demandando elevado tempo computacional. Alternativas devem ser estudadas para reduzir este tempo computacional e, ao mesmo tempo, garantir a otimalidade na resolução do problema.

Note que, se no nó  $k$ , a inclusão do item  $r$  já foi considerada, então é suficiente considerar apenas a inclusão dos  $n - r$  itens subsequentes ao nó  $k$ , ou seja, soluções duplicadas são evitadas considerando a inclusão apenas dos itens  $r + 1, r + 2, \dots, n$ . Na Figura 3.15 ilustra-se a representação das possíveis combinações para 3 tipos de itens numa árvore de enumeração:

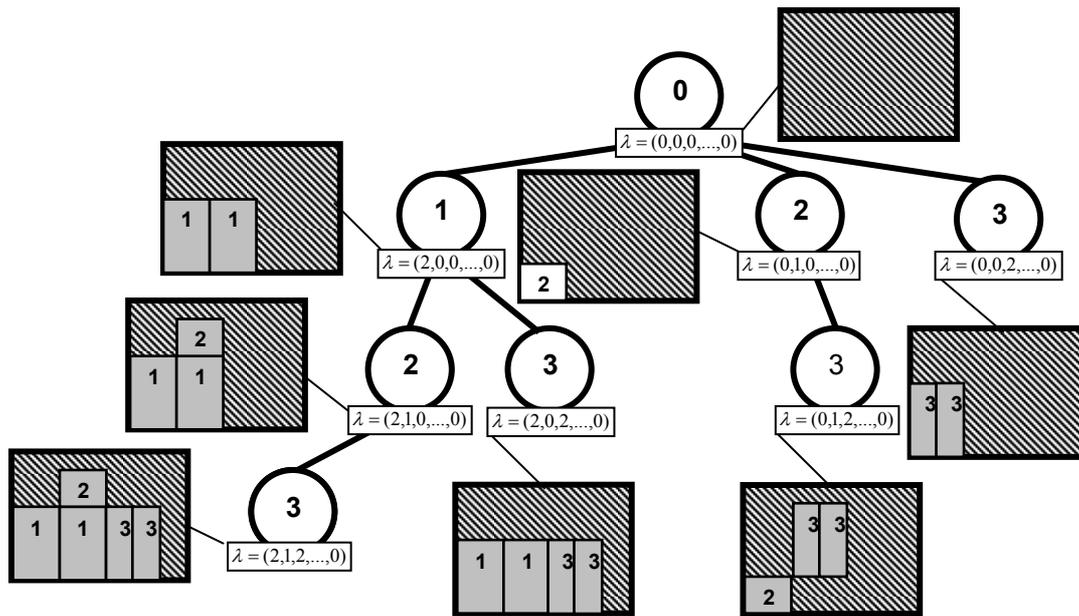


Figura 3.15 – Representação das combinações  $\lambda$  para 3 tipos de itens numa árvore de enumeração.

Uma outra maneira de se reduzir o número de combinações a serem investigadas é utilizando-se limitantes inferiores e superiores. Neste caso, a partir de uma dada combinação obtida no nó, calcula-se o valor máximo  $\bar{G}$  estimado para as combinações dos nós seguintes, com a remoção de 1 item do tipo  $k$  e inclusão dos próximos tipos de itens. Caso o valor do limitante superior  $\bar{G}$  calculado seja inferior ou equivalente ao valor da melhor solução  $\underline{G}$  obtida até o momento, então, certamente a remoção deste item não irá resultar numa solução melhor do que a solução  $\underline{G}$ . Dependendo da maneira como o limitante  $\bar{G}$  for calculado, a remoção de 1 ou mais itens  $k$  não resulta em melhores soluções. Neste caso, faz-se  $\lambda_k = 0$  e uma boa parte das combinações que partem do nó  $k$  poderá ser descartada.

Estes passos do algoritmo anteriormente descrito são apresentados a seguir:

**Passo 1:**

*{ Inicialização dos dados }*

Para cada tipo de item  $i$ , defina  $v_i = \frac{\pi_i}{w_i l_i}$ ,  $i = 1, 2, \dots, n$  ;

Reordene os itens, tal que  $v_1 \geq v_2 \geq \dots \geq v_n$ .

**Passo 2:**

*{ Determinação da solução inicial }*

Determine  $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_n)$ , tal que:

$$\lambda_i = \min \left\{ \left\lfloor \frac{X}{w_i l_i} \right\rfloor, d_i, u_i \right\}, i = 1, \dots, n, \text{ em que:}$$

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

$u_i$  = número de itens do tipo  $i$  que podem ser colocados no padrão e, ainda assim, o padrão puder ser gerado.

**Passo 3:**

*{ Avaliação da solução corrente e armazenamento da solução mais valiosa }*

Determine  $g(\lambda) = \sum_{i=1}^n \pi_i \lambda_i$ ;

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

**Passo 4:**

*{ Teste de otimalidade e cálculo do limitante superior }*

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

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

Senão faça:

Início:

$$\bar{X} = WL - w_1 l_1 \lambda_1 - \dots - w_k l_k (\lambda_k - 1);$$

Determine o item crítico  $j$ , tal que:

$$\begin{cases} \text{(a)} & \sum_{i=k+1}^{j-1} w_i l_i d_i \leq \bar{X} \\ \text{(b)} & \sum_{i=k+1}^j w_i l_i d_i > \bar{X} \end{cases};$$

Se o item crítico  $j$  for indeterminado ( $j$  não atende à restrição (a), ou (b)), então faça:

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

$$\bar{G} = \pi_1 \lambda_1 + \dots + \pi_k (\lambda_k - 1) + \pi_{k+1} \left( \frac{\bar{X}}{w_{k+1} l_{k+1}} \right);$$

Senão faça  $\bar{G} = \pi_1 \lambda_1 + \dots + \pi_k (\lambda_k - 1) + \pi_{k+1} d_{k+1} + \dots + \pi_n d_n$ ;

Senão faça:

**Início:**

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

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

$$\bar{G} = \pi_1 \lambda_1 + \dots + \pi_k (\lambda_k - 1) + \pi_{k+1} \lambda_{k+1} + \dots + \pi_j \lambda_j;$$

**Fim-Senão;**

**Fim-Senão.**

**Passo 5:**

*{ Backtracking }*

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

**Início:**

$$\lambda_k = 0;$$

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

**$X$ =espaço restante da mochila (inicialmente,  $X = WL - w_1 l_1 \lambda_1 - \dots - w_{k-1} l_{k-1} \lambda_{k-1}$ );**

**$u_{i+1}$ =número de itens do tipo  $i+1$  ( $i = k, \dots, n - 1$ ) que podem ser colocados no padrão e, ainda assim, o padrão puder ser gerado;**

**Volte para o Passo 3;**

**Fim-Se;**

**Senão faça:**

**Início:**

$$\lambda_k = \lambda_k - 1;$$

**Se ( $\lambda_k = 0$ ), então volte para o Passo 4;**

$$\lambda_{i+1} = \min \left( \left\lfloor \frac{X}{w_{i+1} l_{i+1}} \right\rfloor, d_{i+1}, u_{i+1} \right), i = k, \dots, n - 1, \text{ em que:}$$

**$X$ =espaço restante da mochila (inicialmente,  $X = WL - w_1 l_1 \lambda_1 - \dots - w_k l_k \lambda_k$ );**

**$u_{i+1}$  representa o número de itens do tipo  $i+1$  ( $i = k, \dots, n - 1$ ) que podem ser colocados no padrão e, ainda assim, o padrão puder ser gerado;**

**Fim-Senão e volte para o Passo 3.**

Observe que no Passo 5 podemos fazer  $\lambda_k = 0$ , pois o limitante superior  $\bar{G}$  ao se retirar 2, 3, ...,  $\lambda_k$  itens  $k$  é menor ou igual ao obtido quando se retira apenas 1 item  $k$  devido ao ordenamento dos itens feito anteriormente.

Para ilustrar o funcionamento deste algoritmo de enumeração implícita, considere novamente o exemplo numérico (Tabela 3.8) que foi utilizado anteriormente para ilustrar o funcionamento do algoritmo enumerativo. O limitante inferior usado pelo algoritmo de enumeração implícita é dado por  $\underline{G} = 0$ . Os itens são colocados em ordem não crescente pela relação  $\pi_i / \tilde{a}_i$ ,  $\tilde{a}_i = w_i l_i$ , o que corresponde à mesma seqüência de itens ordenados já apresentado na Tabela 3.8. Calcula-se a primeira solução baseada na busca em profundidade primeiro. O primeiro item a ser incluído no padrão é o item 1, de dimensão 1x3, valor de utilidade 9, demanda 2. A área inicial do padrão, disponível para a inclusão do item 1 é dada por  $X = WL = 16$ . Tenta-se incluir o número máximo de itens do tipo 1 na área disponível  $X$  do padrão, de modo que um padrão tabuleiro exato seja gerado e que a freqüência do item 1 no padrão não exceda sua demanda. Um padrão tabuleiro foi gerado com a inclusão de 2 itens. Portanto, têm-se  $\lambda_1 = 2$ . A área restante do padrão para inclusão do item seguinte é atualizada para  $X = 16 - 2(1 \times 3) = 10$ . O próximo item a ser incluído no padrão é o item 2, de dimensão 2x1, valor de utilidade 3, demanda 3. Não foi possível gerar um padrão com a inclusão de 3 itens do tipo 2 na área restante  $X$ , nem com a inclusão de 2 itens, mas foi possível gerar um padrão usando 1 item do tipo 2. Faz-se, portanto,  $\lambda_2 = 1$  e atualiza-se  $X = 10 - 1(2 \times 1) = 8$ . O item seguinte a ser incluído na área restante  $X$  é o item 3, de dimensão 2x3, valor de utilidade 5, demanda 2. Não foi possível gerar um padrão usando 2 itens do tipo 3, mas foi possível gerá-lo com 1 item. Portanto,  $\lambda_3 = 1$ . Assim, a primeira solução obtida pela busca em profundidade primeiro é dada por  $\lambda = (2,1,1)$ . Esta solução inicial apresenta um padrão tabuleiro exato de valor 26, superior ao valor do limitante inferior  $\underline{G}$ . O limitante inferior passa a ser atualizado para  $\underline{G} = 26$  e o padrão tabuleiro é armazenado. O item 3 foi o último tipo de item avaliado, então é efetuado o “backtracking”. Remove-se 1 item do tipo 2 para verificar a possibilidade de geração de um padrão tabuleiro melhor com a inclusão do item seguinte (item 3). Têm-se  $\lambda_2 = \lambda_2 - 1 = 1 - 1 = 0$ . Através desta remoção obtém-se um padrão tabuleiro exato com a inclusão de 1 item do tipo 3 ( $\lambda_3 = 1$ ). O padrão obtido apresenta valor 23, inferior ao valor do padrão tabuleiro já obtido pela primeira busca em profundidade. Esta solução é descartada e

avalia-se a solução que potencialmente será obtida a partir da remoção de mais 1 item do tipo 2. Como  $\lambda_2 = 0$ , pega-se o próximo item, anterior ao item 2, tal que  $\lambda_i > 0$ . O item a ser avaliado é o item 1, com  $\lambda_1 = 2$ . O limitante superior  $\bar{G}$  passa a ser calculado com base na remoção de 1 item do tipo 1:

$$\bar{G} = \pi_1(\lambda_1 - 1) + \pi_2 d_2 + \left\lfloor \frac{WL}{w_2 l_2} \right\rfloor \pi_3 = 9(2 - 1) + 3(3) + \left\lfloor \frac{7}{6} \right\rfloor 5 = 23.$$

O valor obtido pelo limitante superior é pior do que o valor armazenado no limitante inferior  $\underline{G}$ . Portanto não compensa remover 1 ou mais itens do tipo 1 e faz-se  $\lambda_1 = 0$ . Tenta-se incluir os itens 2 e 3 para gerar o padrão tabuleiro. A solução obtida foi  $\lambda = (0,3,0)$ , de valor 9, inferior a  $\underline{G}$  e, portanto, ela será descartada. Calcula-se o valor do limitante superior levando em conta a remoção de 1 item do tipo 2:

$$\bar{G} = \pi_2(\lambda_2 - 1) + \pi_3 d_3 = 3(3 - 1) + 5(2) = 16, \text{ valor inferior a } \underline{G} = 26.$$

Novamente não irá compensar remover 1 ou mais itens do tipo 2 para inclusão do item seguinte, pois o valor do limitante superior é pior do que o valor do limitante inferior  $\underline{G}$ . Faz-se  $\lambda_2 = 0$  e tenta-se inserir o item tipo 3. A solução obtida com esta inclusão é  $\lambda = (0,0,2)$ , o padrão possui valor 10, inferior ao valor de  $\underline{G}$ . A solução também é descartada. Todas as possibilidades da árvore de enumeração já foram avaliadas. Portanto, uma solução ótima foi encontrada. A solução ótima é o padrão tabuleiro obtido na primeira busca em profundidade primeiro. O padrão apresenta valor 26 e é composto por 2 itens 1x3, 1 item 2x1 e 1 item 2x3.

Este algoritmo de enumeração implícita para determinação de padrões tabuleiros exatos e restritos foi implementado em linguagem C++. Testes computacionais foram realizados para comparar os tempos de processamento (em segundos) do algoritmo de enumeração implícita com os tempos de processamento do algoritmo enumerativo proposto anteriormente. Os testes utilizaram o mesmo equipamento e os mesmos conjuntos de instâncias utilizados na avaliação do algoritmo enumerativo. Na Tabela

3.15 apresentam-se os resultados que foram obtidos para o primeiro conjunto de instâncias:

Tabela 3.15 – 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).

<b>Instância</b>	<b><i>n</i></b>	<b>Valor da Solução Ótima</b>	<b>Tempo (em segundos) Algoritmo Enumerativo</b>	<b>Tempo (em segundos) Algoritmo de Enumeração Implícita</b>	<b>Tempo (em segundos) Modelo de Yanasse e Morabito (2003)</b>
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	5	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

Na Tabela 3.15 verifica-se que o algoritmo de enumeração implícita foi o mais eficiente dentre os procedimentos avaliados, exceto para a instância MA Para esta instância, o

algoritmo enumerativo não encontrou solução e o algoritmo de enumeração implícita apresentou um tempo de execução bem elevado comparado com o observado em Yanasse e Morabito (2003). A instância MA corresponde a um exemplo prático extraído da indústria de móveis. A Tabela 3.16 apresenta os dados de entrada da instância MA, adaptados do trabalho de Yanasse e Morabito (2003):

Tabela 3.16 – Exemplo da fábrica de móveis (instância MA).

$i$	$w_i$	$l_i$	$\pi_i$	$d_i$
1	609	273	166866	100
2	380	274	104120	100
3	425	330	140250	100
4	650	361	234650	100
5	348	270	93960	100
6	705	270	190350	100
7	718	328	235504	100
8	705	300	211500	100
9	465	330	153450	100
10	480	330	158400	100
11	674	302	203548	100
12	674	270	181980	100
13	636	270	171720	100

Fonte: adaptado de Yanasse e Morabito (2003)

Conforme pode ser observado na Tabela 3.16, a instância MA apresenta itens com grande demanda, portanto, na realidade, não há limitação de itens no padrão, ou seja, o exemplo é um caso irrestrito do problema. Nesta instância, a existência de vários itens relativamente pequenos em comparação ao tamanho do objeto também contribui para dificultar a sua resolução pelo método da enumeração implícita. 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.

Na Tabela 3.17, apresentam-se os resultados dos testes computacionais para o segundo conjunto de instâncias:

Tabela 3.17 – Algoritmo enumerativo x algoritmo de enumeração implícita (instâncias extraídas de Hifi, 2005).

<b>Instância</b>	<b><i>n</i></b>	<b>Valor da Solução Ótima</b>	<b>Tempo (em segundos) Algoritmo Enumerativo</b>	<b>Tempo (em segundos) Algoritmo de Enumeração Implícita</b>
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

Os resultados apresentados na Tabela 3.17 comprovam a eficiência do algoritmo de enumeração implícita na determinação de padrões tabuleiros exatos e restritos. O algoritmo de enumeração implícita conseguiu determinar todos os padrões no limite de tempo fixado em 3600 segundos. O tempo médio para determinação do padrão tabuleiro usando o algoritmo proposto ficou abaixo de 1 minuto.

O algoritmo de enumeração implícita pode ser facilmente adaptado para a determinação de padrões tabuleiros irrestritos bastando apenas especificar uma demanda suficientemente grande para os itens a serem produzidos. Na Tabela 3.18 apresenta-se os resultados obtidos pelo algoritmo de enumeração implícita na determinação de padrões tabuleiros exatos e irrestritos para algumas das instâncias extraídas de Beasley (1985). Para efeito de comparação, na Tabela 3.18 apresenta-se também os tempos de processamento obtidos pelo algoritmo de Katsurayama e Yanasse (1999, 2000) para determinação de padrões tabuleiros exatos irrestritos. Os testes com o algoritmo de enumeração implícita foram realizados numa estação de trabalho Sun UltraSPARC-II, 296Mhz, rodando o sistema operacional SunOS 5.6, mesmo equipamento utilizado por Katsurayama e Yanasse (1999, 2000) em seus testes computacionais.

Tabela 3.18 – Algoritmo de enumeração implícita x algoritmo de Katsurayama e Yanasse (1999, 2000) (instâncias OR-Library).

Instância	$n$	Algoritmo de Katsurayama e Yanasse (1999, 2000)	Algoritmo de Enumeração Implícita	Tempo Algoritmo de Katsurayama e Yanasse (1999, 2000)	Tempo Algoritmo de Enumeração Implícita
gcut1	10	53808	53808	0.002082	0.0134521
gcut2	20	45114	45114	0.003364	0.324928
gcut3	30	52392	52392	0.004466	0.540531
gcut4	50	61008	61008	0.006518	4.59732
gcut5	10	246000	246000	0.003774	0.092603
gcut6	20	207855	207855	0.006182	0.220302
gcut7	30	216372	216372	0.008148	0.324064
gcut8	50	226000	226000	0.012656	3.83822
gcut9	10	971100	971100	0.007392	0.0339129
gcut10	20	894222	894222	0.011501	0.043664
gcut11	30	810540	810540	0.0157	0.79294
gcut12	50	933120	933120	0.024767	1.59905
gcut13	32	8.806e+06	8.4799e+06(*)	0.051716	-

\* Solução obtida no limite de tempo fixado em 3600 segundos

Como já era esperado, o tempo de processamento do algoritmo de enumeração implícita foi pior do que o tempo de processamento do algoritmo de Katsurayama e Yanasse (1999, 2000). Note que algumas das instâncias resolvidas através do algoritmo de enumeração implícita apresentaram um tempo de processamento demasiadamente elevado sendo que uma delas (instância gcut13) não foi resolvida no limite de tempo fixado em 1 hora.

Em virtude destas observações, uma alternativa para melhorar o tempo de processamento deste algoritmo de enumeração implícita é utilizar a solução obtida pelo algoritmo de Katsurayama e Yanasse (1999, 2000) como potencial limitante superior. Inicia-se determinando o melhor padrão tabuleiro exato e irrestrito através do algoritmo de Katsurayama e Yanasse (1999, 2000). Em seguida, verifica-se se o padrão obtido atende ou não à limitação dos itens do problema restrito. Em caso positivo, então o problema foi resolvido com sucesso. Caso contrário prossegue-se com o algoritmo de enumeração implícita utilizando a solução do algoritmo de Katsurayama e Yanasse (1999, 2000) como limitante superior.

Refinamentos no cálculo dos limitantes podem ser utilizados para se tentar reduzir o tempo computacional do algoritmo de enumeração implícita. Por exemplo, 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. As seguintes modificações nos Passos 1 e 4 são sugeridas:

**Modificação no Passo 1:**

*{ Itens com o mesmo retorno por unidade de área são ordenados em função de suas áreas }*

Para cada tipo de item  $i$ , defina  $v_i = \frac{\pi_i}{w_i l_i}$  ( $i = 1, \dots, n$ );

Reordene os itens, tal que  $v_1 \geq v_2 \geq \dots \geq v_n$  e, tal que  $w_i l_i \geq w_{i+1} l_{i+1} \geq K \geq w_{i+s} l_{i+s}$ , 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 para o índice  $j$  }*

**Se o item crítico  $j$  for indeterminado então faça:**

**Se  $(w_{k+1}l_{k+1}d_{k+1} > \bar{X})$ , então faça:**

**Início:**

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

**Se  $r$  for indeterminado, então faça  $\bar{G} = \pi_1\lambda_1 + \dots + \pi_k(\lambda_k - 1)$ ;**

**Senão faça:**

$$\bar{G} = \pi_1\lambda_1 + \dots + \pi_k(\lambda_k - 1) + \pi_{r+1} \left\lfloor \frac{\bar{X}}{w_{r+1}l_{r+1}} \right\rfloor + \pi_{r+2} \left( \frac{\bar{X} - \left\lfloor \frac{\bar{X}}{w_{r+1}l_{r+1}} \right\rfloor w_{r+1}l_{r+1}}{w_{r+2}l_{r+2}} \right);$$

**Fim-Se;**

**Senão faça  $\bar{G} = \pi_1\lambda_1 + \dots + \pi_k(\lambda_k - 1) + \pi_{k+1}d_{k+1} + \dots + \pi_n d_n$ ;**

**Senão faça:**

**Início:**

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

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

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

**Se  $r$  for indeterminado, então faça:**

$$\bar{G} = \pi_1\lambda_1 + \dots + \pi_k(\lambda_k - 1) + \pi_{k+1}\lambda_{k+1} + \dots + \pi_j\lambda_j;$$

**Senão faça:**

$$\begin{aligned} \bar{G} = & \pi_1 \lambda_1 + \dots + \pi_k (\lambda_k - 1) + \pi_{k+1} \lambda_{k+1} + \dots + \pi_j \lambda_j + \\ & + \pi_{r+1} \left[ \frac{\bar{X} - \sum_{i=k+1}^j w_i l_i \lambda_i}{w_{r+1} l_{r+1}} \right] + \pi_{r+2} \left[ \frac{\bar{X} - \sum_{i=k+1}^j w_i l_i \lambda_i - \left[ \frac{\bar{X} - \sum_{i=k+1}^j w_i l_i \lambda_i}{w_{r+1} l_{r+1}} \right] w_{r+1} l_{r+1}}{w_{r+2} l_{r+2}} \right]; \end{aligned}$$

**Fim-Senão.**

Com esta modificação no cálculo do limitante superior  $\bar{G}$ , a frequência  $\lambda_k$  do item no padrão não poderá ser agora reduzida à zero diretamente quando  $\bar{G} \leq \underline{G}$ . Isto ocorre em virtude do limitante ter sido calculado com um valor inteiro dos itens que podem ser alocados na solução e, reduzindo-se  $\lambda_k$  em 1, 2, 3,... unidades ( $\lambda_k \neq 0$ ), é possível obter uma solução inteira melhor do que a calculada. Portanto, no Passo 5 do algoritmo, se  $\bar{G} \leq \underline{G}$ , faz-se  $\lambda_k = \lambda_k - 1$ , ao invés de  $\lambda_k = 0$ ;

Os tempos computacionais obtidos com esta versão modificada do algoritmo de enumeração implícita para o primeiro conjunto de instâncias estão apresentados na Tabela 3.19:

Tabela 3.19 – Versão modificada 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	$n$	Valor da Solução Ótima	Tempo Algoritmo de Enumeração Implícita	Tempo Versão Modificada do Algoritmo de Enumeração Implícita	Tempo Modelo de Yanasse e Morabito (2003)
1	5	7800*	0.15	0.13	0.32
2	5	8280*	0.07	0.05	0.35
3	5	7000*	0.06	0.04	0.29
4	5	7566	0.04	0.03	0.39
5	5	7936*	0.05	0.04	0.35
6	5	9306	0.01	0.01	0.29
7	5	4770	0.03	0.02	0.55

(Continua)

Tabela 3.19 – Conclusão.

Instância	$n$	Valor da solução ótima	Tempo Algoritmo de Enumeração Implícita	Tempo Versão Modificada do Algoritmo de Enumeração Implícita	Tempo Modelo de Yanasse e Morabito (2003)
8	5	7904*	0.01	0.01	0.35
9	5	7800*	0.01	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	0.004	0.003	0.26
13	5	70	0.001	0.001	0.29
14	5	80	0.002	0.003	0.23
15	5	90	0.022	0.016	0.26
16	5	78	0.001	0.001	0.29
17	5	100*	0.004	0.003	0.14
18	5	82	0.002	0.002	0.28
19	5	100*	0.008	0.007	0.16
20	5	70	0.001	0.001	0.28
VAG	5	6450	0.015	0.017	0.34
MA	13	6715500	147.52	146	9.99
CW1	7	240	0.009	0.03	0.19
CW2	10	1687	0.041	0.05	0.38
CW3	20	1080*	0.034	0.034	1.16
W	20	2277*	0.021	0.018	1.96
OF1	10	1769	0.013	0.01	0.47
OF2	10	1684	0.016	0.01	0.43

\* Solução homogênea

Da Tabela 3.19, observa-se uma redução pouco significativa no tempo de processamento das instâncias testadas com o refinamento sugerido. Verifica-se também que parte das instâncias avaliadas apresentaram um aumento no tempo de processamento com uso da versão refinada. Certamente, o aumento deste tempo de processamento se deve à impossibilidade de redução imediata da frequência  $\lambda_k$  a zero na avaliação do limitante superior. Observa-se também que a instância MA ainda apresenta um tempo computacional bastante elevado.

Na Tabela 3.20 mostra-se os resultados obtidos pela versão modificada para o segundo conjunto de instâncias:

Tabela 3.20 – Versão modificada do algoritmo de enumeração implícita (instâncias extraídas de Hifi, 2005).

Instância	$n$	Valor da Solução Ótima	Tempo Algoritmo de Enumeração Implícita	Tempo Versão Modificada do Algoritmo de Enumeração Implícita
HH	5	7020*	0.02	0.02
2	10	1687	0.04	0.06
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	0.67
CHL1	30	6579	2.43	2.54
CHL2	10	1226	0.02	0.02
CW1	25	5358	0.23	0.34
CW2	35	2964*	0.19	0.20
CW3	40	4428*	0.28	0.38
Hchl2	35	7408	21.45	21.8
Hchl9	35	4250	2.06	2.28
2s	10	1689	0.07	0.05
3s	20	2277*	0.03	0.02
A1s	20	1750*	0.05	0.03
A2s	20	2714	0.07	0.05
STS2s	30	3952	0.62	0.56
STS4s	20	7614	1.83	1.45
CHL1s	30	12276	2.43	1.9
CHL2s	10	1922	0.02	0.02
A3	20	4524	0.29	0.17
A4	20	3180	0.32	0.19
A5	20	12276	0.88	0.54
CHL5	10	220	0.04	0.01
CHL6	30	13113	7.26	5.37
CHL7	35	12958	25.61	18.7
CU1	25	12200	0.25	0.19
CU2	35	20250	0.36	0.25
Hchl3s	10	7388	3.10	2.2
Hchl4s	10	7388	1.06	0.79
Hchl6s	22	40835	4.93	4.08
Hchl7s	40	41869	37.40	22.6
Hchl8s	10	416	0.06	0.06

\* Solução homogênea

Pela Tabela 3.20, verifica-se novamente um aumento no tempo computacional de algumas instâncias devido à impossibilidade de redução imediata da frequência  $\lambda_k$  a

zero na avaliação do limitante superior. Apesar desta impossibilidade, observa-se uma redução do tempo de processamento em boa parte das instâncias testadas com uso da versão modificada (vide, por exemplo, instâncias STS2s, STS4s, CHL1s, A3, A4, A5, CHL6, CU1, CU2, Hchl3s, Hchl4s, Hchl6s, Hchl7s).

### 3.3.3 – Um Algoritmo Melhorado de Enumeração Implícita

A aplicação do algoritmo de enumeração implícita anterior é razoável em problemas que apresentem itens de demanda não muito elevada e de tamanho relativo não muito pequeno. Problemas que apresentem itens pequenos e com demanda elevada podem resultar em elevado tempo computacional. Uma das razões deste elevado tempo computacional se deve ao limitante superior que parece ser não muito restritivo, pois é baseado apenas nas áreas dos itens. Desenvolvemos um algoritmo melhorado com limitantes superiores mais restritivos que consideram também as dimensões dos itens.

Consideramos que os itens estejam ordenados em ordem decrescente de área quando apresentam a mesma relação valor de utilidade por unidade de área. Introduzimos um novo componente, denominado retângulo cerne, em cada nó da árvore de enumeração. Um retângulo cerne é um subpadrão que certamente estará presente em qualquer padrão tabuleiro gerado naquele nó e nos nós que o sucedem. O retângulo cerne é representado por uma combinação  $\beta$  de itens, tal que  $\beta \subset \lambda$ . O algoritmo para determinação de um retângulo cerne utiliza, essencialmente, as mesmas idéias do algoritmo construtivo para geração de um padrão tabuleiro exato a partir de uma dada combinação de itens. A diferença básica entre os dois procedimentos ocorre quando se chega à conclusão de que todos os “bins” de dimensão  $(W - D) \times c$  e de dimensão  $b \times (L - C)$  são, cada um deles, isoladamente, suficientes para acomodar todos os itens restantes do tipo  $i$ . O algoritmo para determinação do retângulo cerne é encerrado neste ponto. O subpadrão formado no retângulo  $D \times C$  é o retângulo cerne. Para ilustrar a geração de um retângulo cerne, considere, por exemplo, o padrão tabuleiro exato apresentado na Figura 3.16. As dimensões do objeto e dos itens produzidos pelo padrão estão especificadas na figura:

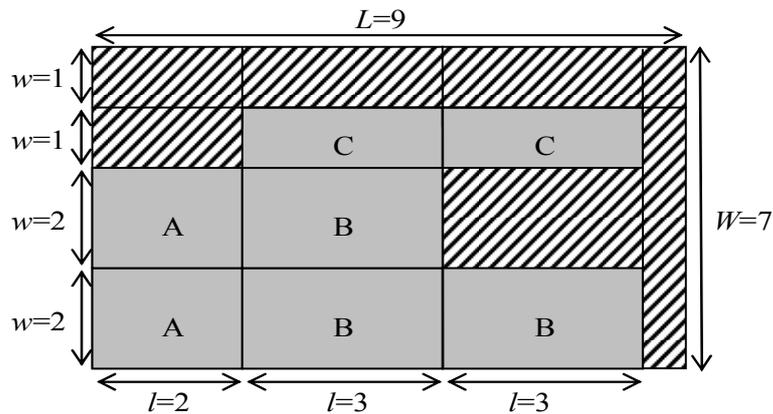


Figura 3.16 – Padrão tabuleiro exato para o exemplo da geração do retângulo cerne.

Considere também que os valores de utilidade por unidade de área dos itens sejam dados por  $v_A > v_B > v_C$ . Portanto, os itens serão inseridos na seqüência **A**, **B**, **C** na área disponível do objeto.

O primeiro item a ser inserido é o item tipo **A**, de dimensão 2x2. A demanda do item **A** é dada por  $\tilde{u}_A = 2$ . O primeiro item do tipo **A** é colocado no canto inferior esquerdo da área do objeto, conforme mostra a Figura 3.17:



Figura 3.17 – Inclusão do primeiro item **A** na área do objeto.

Com a inclusão do primeiro item tipo **A**, sua demanda é atualizada para  $\tilde{u}_A = 2 - 1 = 1$ , restando, portanto, apenas 1 item para completá-la. O segundo item do tipo **A** poderá ser colocado no bin acima ou à direita do item inserido. Ambos os bins apresentam área suficiente para acomodar o segundo item tipo **A**. Devido à indefinição da localização deste segundo item **A**, prossegue-se avaliando o item seguinte. O próximo item é o item

tipo **B**, de demanda  $\tilde{u}_B = 3$ . Note que o item **B** apresenta a mesma largura do item **A**, mas comprimento diferente. Assim, devemos ter uma faixa vertical com este comprimento. O item **A** cabe na área disponível à sua direita, assim, o primeiro item tipo **B** é inserido neste local, conforme mostrado na Figura 3.18:

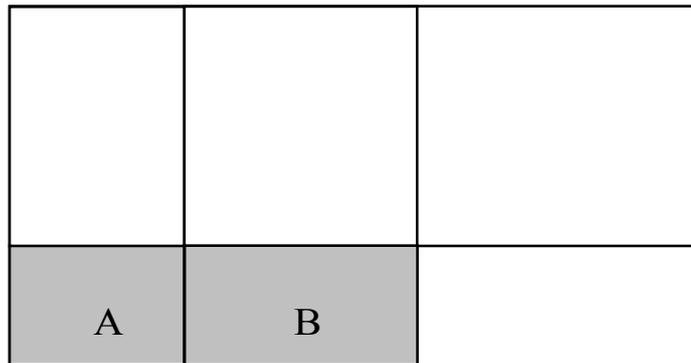


Figura 3.18 – Inclusão do primeiro item tipo **B**.

A demanda do item **B** é atualizada para  $\tilde{u}_B = 3 - 1 = 2$ . O bin disponível à direita do item **B** não é suficiente para acomodar os 2 itens restantes do tipo **B**. Portanto, uma nova faixa horizontal para acomodar o item **B** deve existir e é definida conforme mostra a Figura 3.19. Note que com a geração da nova faixa horizontal para inclusão do item **B**, surgiu uma área interna no cerne (área mais escura), compatível com a área do item tipo **A**, cuja demanda ainda não foi totalmente atendida. Esta área é preenchida com o item tipo **A** e sua demanda é atualizada para  $\tilde{u}_A = 1 - 1 = 0$ .

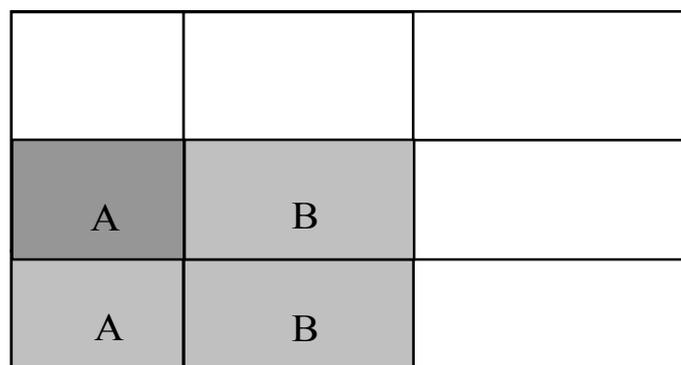


Figura 3.19 – Inclusão do segundo item tipo **B** e preenchimento da área correspondente ao item **A**.

Com a nova faixa horizontal definida, a demanda do item **B** é atualizada para  $\tilde{u}_B = 2 - 1 = 1$ , restando, portanto, apenas 1 item para completá-la que poderia ser inserido tanto no bin definido acima como nos bins definidos à direita dos itens **B**. A área nos bins é suficiente para acomodar o item. Dada a indefinição, analisa-se a inclusão do item seguinte.

O próximo e, último item a ser incluído no retângulo cerne é o item tipo **C**, de demanda  $\tilde{u}_C = 2$ . O item **C** apresenta o mesmo comprimento do item **B**, incluído anteriormente, na largura diferente dos itens anteriores. Assim, devemos ter uma faixa horizontal com esta largura que é definida conforme ilustrado na Figura 3.20. A área do bin, disponível acima do item **B** é suficiente para acomodar o item **C** e, portanto, o primeiro item **C** será incluído neste bin.

	C	
A	B	
A	B	

Figura 3.20 – Inclusão do primeiro item tipo **C**.

Com a inclusão do item tipo **C**, sua demanda é atualizada para  $\tilde{u}_C = 2 - 1 = 1$ , restando apenas 1 item para completá-la. O segundo item do tipo **C** poderá ser incluído tanto no bin acima, como no bin à direita do item tipo **C** e existe área suficiente em ambos os bins para acomodá-lo.

Com a inclusão do item **C**, não é mais possível incluir o item **B** no bin definido acima deste item. Assim, deverá existir mais uma faixa vertical de comprimento igual ao do item **B**, conforme a Figura 3.16. Portanto, o retângulo cerne obtido para o padrão tabuleiro exato ilustrado na Figura 3.16 é o próprio padrão tabuleiro.

O interesse em se utilizar um retângulo cerne associado a cada padrão tabuleiro gerado está em determinar de forma mais precisa o espaço restante  $X$  a ser utilizado na inserção do próximo tipo de item. Considere, por exemplo, o retângulo cerne representado no canto inferior esquerdo da Figura 3.21.

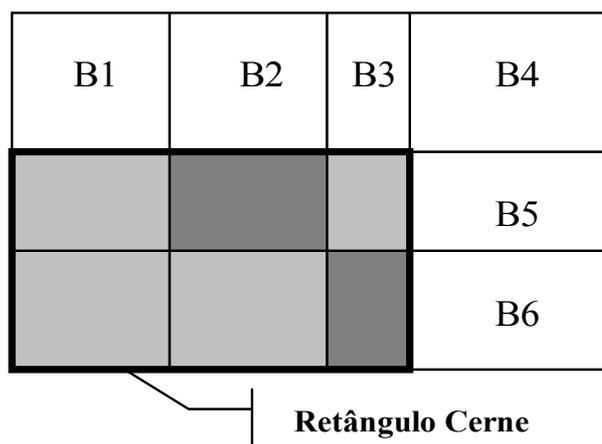


Figura 3.21 – Exemplo de retângulo cerne.

Uma vez definida a combinação  $\beta$  associada aos itens  $1, \dots, k$  que compõem o retângulo cerne, têm-se os bins localizados acima (áreas **B1**, **B2**, **B3** da Figura 3.21), à direita (áreas **B5** e **B6** da Figura 3.21) e à nordeste do retângulo (área **B4** da Figura 3.21), disponíveis para a inclusão dos próximos itens (itens  $k+1, \dots, n$ ). Têm-se também as áreas internas vazias do retângulo cerne (área mais escura da Figura 3.21) disponíveis para a inclusão destes itens. O item  $k+1$  pode ser inserido em bins que tenham o mesmo comprimento ou largura que este item. Também pode ser inserido em **B4** e nas áreas internas vazias do retângulo cerne, se estas áreas forem do exato tamanho do item. Com base nesta observação, obtêm-se um limitante superior  $\theta$  de itens tipo  $k+1$  que podem ser incluídos.

A frequência  $f$  do item  $i$  no padrão é determinada de maneira similar ao procedimento de busca por biseção, utilizado pelo algoritmo de enumeração implícita, caso haja itens adicionais indefinidos que não foram incluídos no retângulo cerne, ou seja,  $\lambda_s \neq \beta_s$ , para algum  $s$ . Caso contrário, tem-se  $f = \theta$ .

O cálculo do novo limitante superior  $\bar{G}$  passa a levar em consideração o valor máximo correspondente à inserção dos itens que possam, eventualmente, ser alocados nos bins definidos acima, à direita e à nordeste do retângulo cerne. Neste caso, dado que o item  $k$  foi inserido no retângulo cerne associado ao padrão tabuleiro exato do nó corrente, considera-se a inserção do próximo item  $(k+1, \dots, n)$  que apresente a melhor relação valor de utilidade por unidade de área e que tenha dimensões compatíveis com as dimensões do bin considerado. Para os bins localizados acima do retângulo cerne, isto corresponde a selecionar, para cada um destes bins, o item  $r$  ( $r = k+1, \dots, n$ ) que maximize

$$\left\{ \frac{\pi_r}{w_r l_r} \right\}, \text{ tal que } l_r = W^T,$$

em que  $W^T$  corresponde à largura do bin localizado acima do retângulo cerne (vide Figura 3.22a). Para os bins localizados à direita do retângulo cerne, seleciona-se, para cada um destes bins, o item  $r$  ( $r = k+1, \dots, n$ ) que maximize

$$\left\{ \frac{\pi_r}{w_r l_r} \right\}, \text{ tal que } w_r = W^R,$$

em que  $W^R$  corresponde à largura do bin localizado à direita do retângulo cerne (vide Figura 3.22b). Para o bin localizado à nordeste do retângulo cerne, basta selecionar o item mais lucrativo que caiba em sua área, sem necessariamente preenche-la de maneira exata, ou seja, seleciona-se o item  $r$  ( $r = k+1, \dots, n$ ) que maximize

$$\left\{ \frac{\pi_r}{w_r l_r} \right\}, \text{ tal que } w_r \leq W^N \text{ e } l_r \leq L^N,$$

em que  $W^N$  e  $L^N$  correspondem, respectivamente, a largura e o comprimento do bin localizado à nordeste do retângulo cerne (vide Figura 3.22c).

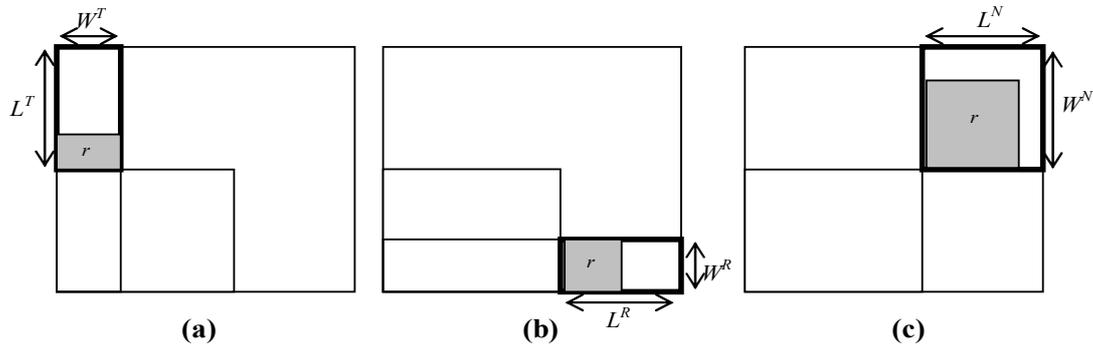


Figura 3.22 – Escolha do item  $r$  para inclusão nos bins acima, à direita e à nordeste do retângulo cerne para cálculo do limitante superior  $\bar{G}$ : (a) item  $r$  escolhido para o bin acima do retângulo cerne; (b) item  $r$  escolhido para o bin à direita do retângulo cerne; (c) item  $r$  escolhido para o bin à nordeste do retângulo cerne.

Uma vez que o item  $r$  tenha sido selecionado para cada um dos bins acima, à direita e à nordeste do retângulo cerne, o valor máximo obtido com a inserção deste item em cada

um destes bins poderá ser calculado através de  $v_r \tilde{B}$ , onde  $v_r = \frac{\pi_r}{w_r l_r}$  e  $\tilde{B}$  é a área do

bin considerado. Este limitante superior é válido somente nos casos onde a demanda dos itens tipo  $s$  ( $s = 1, \dots, k$ ) do padrão tabuleiro tenha sido completamente satisfeita pelo retângulo cerne, ou seja, nos casos onde  $(\lambda_s - \beta_s) = 0$ . Para os casos onde  $(\lambda_s - \beta_s) \neq 0$  (para algum  $s$  inteiro, não negativo), é necessário reduzir a área de alguns dos bins para levar em consideração os itens  $s$  que ainda não foram incluídos no retângulo cerne. Nestes casos, o item  $s$  pode ser incluído tanto em um bin acima como em um bin à direita do retângulo cerne, pois estes bins apresentam larguras compatíveis com a largura (ou comprimento) do item  $s$ . O item  $s$  é incluído no bin que resulte no maior retorno lucrativo global. Sejam  $v^T$  e  $v^R$ , os valores de utilidade por unidade de área correspondentes aos itens  $k+1, \dots, n$  mais lucrativos considerados, respectivamente, nos bins acima e à direita do retângulo cerne. O valor máximo obtível com a inserção do item  $s$  no bin acima do retângulo cerne é:

$$v^R \tilde{B}^R + v^T (\tilde{B}^T - \tilde{a}_s), \quad (3.1)$$

em que  $\tilde{B}^T$  e  $\tilde{B}^R$  correspondem, respectivamente, à área total do bin acima e à direita do retângulo cerne e,  $\tilde{a}_s$  é a área total ocupada pelo item  $s$ .

Similarmente, o valor máximo obtível com a inserção do item  $s$  no bin à direita do retângulo cerne é:

$$v^R (\tilde{B}^R - \tilde{a}_s) + v^T \tilde{B}^T. \quad (3.2)$$

Das equações (3.1) e (3.2), deduz-se o maior retorno resultante se adotarmos o seguinte:

Se  $v^T < v^R$ , então o item  $s$  deve ser inserido no bin acima do retângulo cerne, caso contrário, ele deve ser inserido no bin à direita do retângulo cerne. Considerando que o item  $s$  tenha sido inserido no bin acima do retângulo cerne, o valor máximo obtido com o preenchimento desta faixa é dado por:

$$\frac{[\tilde{B}^T - (\lambda_s - \beta_s)\tilde{a}_s]}{\tilde{a}^T} \pi^T + (\lambda_s - \beta_s)\pi_s, \quad (3.3)$$

em que  $\tilde{a}^T$  e  $\pi^T$  representam, respectivamente, a área e o valor de utilidade do item mais lucrativo (item  $r$ ) que preenche o bin acima do retângulo cerne.

Isolando os termos da equação (3.3), obtém-se:

$$\frac{\pi^T}{\tilde{a}^T} \tilde{B}^T - \gamma^T, \quad (3.4)$$

em que  $\gamma^T$  corresponde à parcela a ser descontada do limitante superior, dada por

$$\gamma^T = \left( \frac{\pi^T}{\tilde{a}^T} \tilde{a}_s - \pi_s \right) (\lambda_s - \beta_s).$$

No caso do item  $s$  ter sido inserido no bin à direita do retângulo cerne, a equação (3.4) fica:

$$\frac{\pi^R}{\tilde{a}^R} \tilde{B}^R - \gamma^R, \quad (3.5)$$

em que  $\tilde{a}^R$  e  $\pi^R$  representam, respectivamente, a área e o valor de utilidade do item mais lucrativo (item  $r$ ) que preenche o bin à direita do retângulo cerne e,  $\gamma^R$  corresponde à parcela a ser descontada do limitante superior, dada por

$$\gamma^R = \left( \frac{\pi^R}{\tilde{a}^R} \tilde{a}_s - \pi_s \right) (\lambda_s - \beta_s).$$

Caso não haja um item  $s$  que preencha o bin acima (ou à direita) do retângulo cerne, então  $\pi^T$  (ou  $\pi^R$ ) é igual a zero. Para que as equações de  $\gamma^T$  e  $\gamma^R$  possam ser utilizadas também para estes casos, faz-se  $\frac{\pi^T}{\tilde{a}^T} = 0$  (ou  $\frac{\pi^R}{\tilde{a}^R} = 0$ ).

Portanto, o limitante superior  $\bar{G}$  a ser utilizado pelo Algoritmo Melhorado de Enumeração Implícita poderá ser calculado através de:

$$\bar{G} = \sum_{i=1}^k \pi_i \beta_i + \sum_p \frac{\pi^T}{\tilde{a}^T} \tilde{B}_p^T + \sum_q \frac{\pi^R}{\tilde{a}^R} \tilde{B}_q^R + \frac{\pi^N}{\tilde{a}^N} \tilde{B}^N - \sum_p \gamma_p^T - \sum_q \gamma_q^R, \quad (3.6)$$

em que  $p$  e  $q$  indicam, respectivamente, os bins localizados acima e à direita do retângulo cerne e a somatória deve ser feita para todos os tipos.

De forma similar ao algoritmo de enumeração implícita, comparam-se os valores obtidos pelos limitantes superior e inferior. Neste novo algoritmo, esta comparação é feita toda vez que um novo item é inserido no padrão. Se  $\bar{G} > \underline{G}$ , então a solução atual, que utiliza  $f_k$  itens do tipo  $k$  é guardada e tenta-se incluir o próximo item. Caso contrário, se  $\bar{G} \leq \underline{G}$ , então esta solução não resultará numa solução melhor do que a encontrada até o momento. Neste caso, considera-se a remoção de 1 ou mais itens do tipo  $k$  para inclusão dos itens seguintes. O valor de  $f_k$  agora não poderá ser imediatamente reduzido a 0, tal como no algoritmo de enumeração implícita, uma vez que o retângulo cerne pode ser modificado, mudando o cálculo do limitante dado pela

equação 3.6. Para garantir que a solução ótima seja obtida, considera-se a remoção de 1 item do tipo  $k$  por vez. Se não houver nenhuma alteração no retângulo cerne em relação ao retângulo cerne anterior, então, não há alterações no valor do limitante superior e, portanto, nenhuma solução melhor poderá ser obtida. Remove-se então, mais 1 item do tipo  $k$  e repete-se esta verificação. Caso a remoção do item tipo  $k$  não resulte em um retângulo cerne diferente, então remove-se mais outro item do tipo  $k$  e prossegue-se com esta remoção até que um retângulo cerne diferente seja obtido, ou até que a frequência do item  $k$  seja igual a zero. Se a frequência do item  $k$  tornar-se nula, prossegue-se com o algoritmo tentando incluir os itens seguintes,  $k+1$ ,  $k+2$ , ...,  $n$  e atualiza-se o limitante inferior  $\underline{G}$ , no caso de se obter uma solução melhor. Caso contrário, se a frequência do item  $k$  for não nula, calcula-se o limitante superior  $\overline{G}$  e repete-se este mesmo procedimento de remoção do item  $k$ , se  $\overline{G} \leq \underline{G}$ , ou tenta-se incluir os itens seguintes, se  $\overline{G} > \underline{G}$ .

Para fins de implementação é importante notar que o novo algoritmo necessita de uma série de informações para o cálculo do limitante superior. Para garantir um acesso rápido a estas informações, é conveniente que elas estejam armazenadas em cada um dos nós da árvore de enumeração. Caso contrário é necessário recuperá-las. Na Tabela 3.21 apresenta-se uma lista das informações convenientes de serem guardadas nos nós da árvore de enumeração.

Tabela 3.21 – Informações utilizadas pelo algoritmo melhorado de enumeração implícita e que é conveniente armazená-las em cada um dos nós da árvore de enumeração.

<b>Informação</b>	<b>Descrição</b>
$nold$	Identificação do nó;
$index$	Tipo de item considerado no nó;
$P^{nold}$	Padrão tabuleiro exato determinado com a inclusão do item $index$ , sem levar em conta o preenchimento dos espaços livres compatíveis com os itens viáveis que surgiram no interior do padrão;
$\lambda^{nold}$	Combinação de itens associado ao padrão tabuleiro exato $P^{nold}$ determinado;
$Q^{nold}$	Retângulo cerne associado ao padrão tabuleiro exato $P^{nold}$ ;
$\beta^{nold}$	Combinação de itens associado ao retângulo cerne $Q^{nold}$ ;

(Continua)

Tabela 3.21 – Conclusão.

Informação	Descrição
$R^{nold}$	Retângulo cerne que leva em conta o preenchimento dos espaços livres compatíveis com os itens viáveis;
$\varphi^{nold}$	Combinação de itens associado ao retângulo cerne $R^{nold}$ ;
$H^{nold}$	Padrão tabuleiro exato que leva em conta o preenchimento dos espaços livres referentes aos itens que foram utilizados no preenchimento do retângulo cerne $R^{nold}$ ;
$\alpha^{nold}$	Combinação de itens associado ao padrão tabuleiro exato $H^{nold}$ ;
$\tilde{u}^{nold}$	Demanda atualizada (restante) para os itens $index+1, \dots, n$ que foram usados para preencher o retângulo cerne $R^{nold}$ .

Observe que o procedimento de busca em profundidade primeiro utilizado pelo algoritmo de enumeração implícita pode ser representado através de uma estrutura de pilha de dados, onde os dados correspondem aos nós já percorridos na árvore de enumeração, ou seja, o caminho atualmente percorrido. Para cada tipo de item, cria-se um novo nó que apresenta informações a respeito da inclusão do item na área restante do padrão. O nó é adicionado na pilha, desde que a frequência calculada para o novo item seja não nula. Se em algum momento, a frequência do item tornar-se nula, então este nó é removido da pilha. Os passos deste novo algoritmo de enumeração implícita estão resumidos adiante:

**Passo 1:**

*{ Inicialização }*

Para cada tipo de item  $i$ , defina  $v_i = \frac{\pi_i}{w_i l_i}$  ( $i = 1, \dots, n$ ) e reordene os itens, tal

que  $v_1 \geq v_2 \geq \dots \geq v_n$  e, tal que  $w_i l_i \geq w_{i+1} l_{i+1} \geq K \geq w_{i+s} l_{i+s}$ , se  $v_i = v_{i+1} = \dots = v_{i+s}$  (para algum  $i$  ou  $s$ , inteiros não negativos);

Faça  $\underline{h}$  =melhor padrão homogêneo restrito e  $\underline{G}$  =valor ( $\underline{h}$ );

Faça  $noId=0$ ,  $\lambda^{nold} = 0$ ,  $\beta^{nold} = 0$ ,  $\varphi^{nold} = 0$ ,  $\alpha^{nold} = 0$  e  $\tilde{u}^{nold} = d$ ,  $index = 0$ .

**Passo 2:**

*{ Busca em profundidade primeiro }*

Se ( $index < n$ ) então faça  $index = index + 1$ ;

Senão vá para o Passo 4;

Calcule  $f_{index}$  e gere o padrão tabuleiro exato  $P^{nold+1}$  correspondente à inclusão do item  $index$ ;

Se ( $f_{index} = 0$ ), então faça Se ( $index < n$ ), então vá para o Passo 2;

Senão vá para o Passo 4;

$$\lambda^{nold+1} = \lambda^{nold} + f_{index} e_{index},$$

em que  $e_{index}$  é um vetor, cujos elementos são iguais a zero, exceto o elemento da posição  $index$  que é igual a 1;

Determine o retângulo cerne  $Q^{nold+1}$  e sua combinação  $\beta^{nold+1}$  correspondente;

Determine o retângulo cerne preenchido  $R^{nold+1}$  e sua combinação  $\varphi^{nold+1}$  correspondente;

Determine o padrão tabuleiro exato  $H^{nold+1}$  ( $H^{nold+1}$  é equivalente ao padrão  $P^{nold+1}$ , mas com os espaços livres ocupados pelos itens viáveis que preenchem o retângulo cerne  $R^{nold+1}$ );

$$\alpha^{nold+1} = \alpha^{nold} + f_{index} e_{index} + (\text{"itens viáveis que preenchem o retângulo cerne } R^{nold+1}\text{"}),$$

em que  $e_{index}$  é um vetor, cujos elementos são iguais a zero, exceto o elemento da posição  $index$  que é igual a 1;

Se ( $\text{valor}(\alpha^{nold+1}) > \underline{G}$ ), então faça  $\underline{b} = H^{nold+1}$  e  $\underline{G} = \text{valor}(\alpha^{nold+1})$ .

### **Passo 3:**

*{ Cálculo do limitante superior }*

Calcule o valor do limitante superior  $\overline{G}$  para a combinação  $\alpha^{nold+1}$ ;

Se ( $\overline{G}(\alpha^{nold+1}) > \underline{G}$ ) ou ( $nold=0$ ), então faça:

Início:

$$nold = nold + 1;$$

$$idx(nold)=index;$$

$$\text{Armazene } \lambda^{nold}, P^{nold}, \beta^{nold}, Q^{nold}, \varphi^{nold}, R^{nold}, \alpha^{nold}, H^{nold}, \tilde{u}^{nold};$$

Vá para o Passo 2;

Fim-Se;

Senão faça:

Início:

Determine um valor  $y$  que reduza  $f_{index}$  a um valor que defina um retângulo cerne diferente de  $R^{nold+1}$ ;

**Se ( $\gamma > 0$ ), então faça:**

**Início:**

$$\lambda^{nold+1} = \lambda^{nold} + \gamma e_{index},$$

**em que  $e_{index}$  é um vetor, cujos elementos são iguais a zero, exceto o elemento da posição  $index$  que é igual a 1;**

**Atualize  $P^{nold+1}$ ,  $Q^{nold+1}$ ,  $\beta^{nold+1}$ ,  $R^{nold+1}$ ,  $\varphi^{nold+1}$ ,  $H^{nold+1}$ ,  $\tilde{u}^{nold+1}$ ;**

**Vá para o Passo 3;**

**Fim-Se;**

**Senão vá para o Passo 2;**

**Fim-Senão.**

**Passo 4:**

**{ Backtracking }**

**$nold=0$ ;**

**Se (*Pilha vazia!*), então Pare;**

**$index=idx(nold)$ ;**

**$nold=nold-1$ ;**

**Determine um valor  $\gamma$  que reduza  $f_{index}$  a um valor que defina um retângulo cerne diferente de  $R^{nold+1}$ ;**

**Se ( $\gamma > 0$ ), então faça:**

**Início:**

$$\lambda^{nold+1} = \lambda^{nold} + \gamma e_{index},$$

**em que  $e_{index}$  é um vetor, cujos elementos são iguais a zero, exceto o elemento da posição  $index$  que é igual a 1;**

**Atualize  $P^{nold+1}$ ,  $Q^{nold+1}$ ,  $\beta^{nold+1}$ ,  $R^{nold+1}$ ,  $\varphi^{nold+1}$ ,  $H^{nold+1}$ ,  $\tilde{u}^{nold+1}$ ;**

**Vá para o Passo 3;**

**Fim-Se;**

**Senão vá para o Passo 2.**

Para ilustrar o funcionamento do Algoritmo Melhorado de Enumeração Implícita, um novo exemplo numérico é apresentado. Considere o corte de um objeto de dimensão 10x10, cujos itens a serem produzidos estão apresentados na Tabela 3.22:

Tabela 3.22 – Itens a serem produzidos (exemplo numérico para o algoritmo melhorado de enumeração implícita).

$i$	$w_i$	$l_i$	$\pi_i$	$d_i$
1	4	3	24	2
2	6	3	19	2
3	2	4	22	2

O algoritmo melhorado inicia no nó 0 da árvore de enumeração e tenta inserir os itens na área restante do padrão, seguindo a ordem não crescente dada pela relação valor de utilidade por unidade de área destes itens (vide Tabela 3.23).

Tabela 3.23 – Itens a serem produzidos (em ordem não crescente pela relação valor de utilidade por unidade de área).

$i$	$w_i$	$l_i$	$a_i$	$\pi_i$	$\frac{\pi_i}{\tilde{a}_i}$	$d_i$
1	2	4	8	22	2.75	2
2	4	3	12	24	2	2
3	6	3	18	19	1.05	2

Inicialmente, têm-se as seguintes informações armazenadas no nó 0:

$index=0$  (nenhum item foi incluído)

$\lambda^0 = (0,0,0)$  (nenhum padrão tabuleiro exato foi determinado)

$\beta^0 = (0,0,0)$  (nenhum retângulo cerne associado ao padrão foi determinado)

$\varphi^0 = (0,0,0)$  (nenhum retângulo cerne preenchido foi determinado)

$\alpha^0 = (0,0,0)$  (nenhum padrão tabuleiro preenchido foi determinado)

$\tilde{u}^0 = d = (2,2,2)$

O limitante inferior inicial  $\underline{G}$  para o exemplo numérico corresponde ao valor do melhor padrão homogêneo restrito, composto de 2 itens 4x3. Para este padrão homogêneo, obtêm-se  $\underline{G} = 2(24) = 48$ .

O primeiro item a ser incluído é o item  $index=1$  (item  $2 \times 4$ ). A área disponível  $\theta$  para inclusão deste item corresponde à toda área do objeto. Assim, nenhum retângulo cerne foi gerado. A frequência do item 1 no padrão é calculada através de:

$$f_1 = \min \left\{ d_1, \left\lfloor \frac{W}{w_1} \right\rfloor \left\lfloor \frac{L}{l_1} \right\rfloor \right\} = \min \{ 2, 5 \times 2 \} = 2 .$$

O item 1 foi incluído resultando no padrão tabuleiro exato  $P^1$ , apresentado na Figura 3.23:

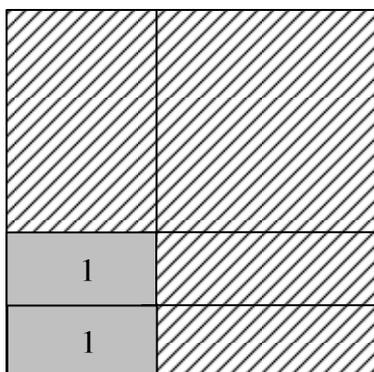


Figura 3.23 – Padrão tabuleiro exato  $P^1$  obtido com a inclusão do item 1.

A combinação de itens associada ao padrão tabuleiro da Figura 3.23 é  $\lambda^1 = (2,0,0)$ . Esta combinação de itens apresenta valor total  $Z = 2(22) + 0(24) + 0(19) = 44$ , inferior ao valor do limitante inferior  $\underline{G}$  calculado. A melhor solução encontrada, associada a  $\underline{G}$  não será atualizada.

O passo seguinte é calcular o limitante superior  $\overline{G}$  que considera a inclusão dos itens 2 e 3. Para calcular este limitante superior, gera-se o retângulo cerne associado ao padrão tabuleiro da Figura 3.23. O retângulo cerne  $Q^1$  gerado é apresentado na Figura 3.24 e sua combinação de itens, não preenchida, é  $\beta^1 = (1,0,0)$ :

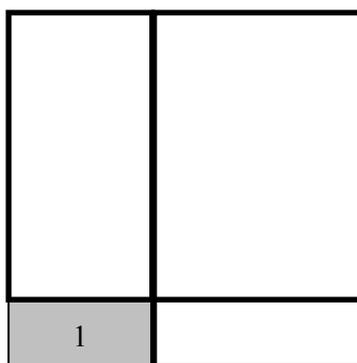


Figura 3.24 – Retângulo cerne  $Q^1$  associado ao padrão tabuleiro exato da Figura 3.23.

O retângulo cerne  $R^1$  e o padrão tabuleiro  $H^1$  são obtidos a partir do preenchimento dos espaços livres do retângulo cerne  $Q^1$ . Como não há possibilidade de preenchimento deste retângulo cerne, têm-se  $R^1 = Q^1$  e  $H^1 = P^1$ . A combinação  $\varphi^1$  associada ao retângulo cerne preenchido  $R^1$  é  $\varphi^1 = \beta^1$  e, a combinação  $\alpha^1$  associada ao padrão tabuleiro preenchido  $H^1$  é  $\alpha^1 = \lambda^1$ . O limitante atualizado para os itens seguintes é  $\tilde{u}^1 = (0,2,2)$ . Caso houvesse possibilidade de preenchimento de  $Q^1$ , um padrão tabuleiro  $H^1$  teria sido obtido com a inclusão dos itens que preenchem  $Q^1$  e seria necessário verificar novamente se o padrão obtido corresponde ou não a uma solução melhor que a solução armazenada no limitante inferior  $\underline{G}$ .

Prossegue-se avaliando o limitante superior  $\overline{G}$ . O cálculo de  $\overline{G}$  leva em consideração as seguintes informações que devem ser calculadas a partir do retângulo cerne preenchido  $R^1$ :

- $G_0$ =valor total dos itens presentes no cerne;
- $G_1$ =valor estimado para os itens acima do cerne;
- $G_2$ =valor estimado para os itens à direita do cerne;
- $G_3$ =desconto sobre os valores  $G_1$  e  $G_2$ , relativo aos itens não incluídos no retângulo cerne e presentes no padrão, ou seja, nos casos onde  $(\lambda_s - \beta_s) \neq 0$ ;

- $G_4$ =valor estimado para os itens à nordeste do retângulo cerne.

Neste caso,  $R^1$  corresponde ao mesmo retângulo cerne  $Q^1$  apresentado na Figura 3.24.

Para este retângulo cerne, têm-se:

$$G_0=1(22)=22$$

$G_1=0$  (nenhum item do tipo 2 ou 3 cabe no bin acima do cerne)

$G_2=0$  (nenhum item do tipo 2 ou 3 cabe no bin à direita do cerne)

$$G_3=-22(2-1)=-22$$

$$G_4=\frac{24}{4 \times 3} 8 \times 6 = 96$$

O valor calculado para o limitante superior  $\underline{G}$  é:

$$\bar{G} = G_0 + G_1 + G_2 - G_3 + G_4 = 140.$$

O valor calculado para o limitante inferior é  $\underline{G} = 48$ , Portanto, o limitante superior  $\bar{G}$  é melhor. O nó 1 é criado e suas informações são armazenadas para uso nos próximos nós.

O próximo item a ser incluído é o item  $index=2$  (item  $4 \times 3$ ). O item 2 só pode ser inserido no retângulo à nordeste do retângulo cerne do nó anterior (nó 1, vide Figura 3.24). Este retângulo cerne tem dimensão  $(10-2) \times (10-4)$ . Portanto, o número máximo

possível para inclusão do item 2 é  $f_2 = \min \left\{ d_2, \left\lfloor \frac{6}{3} \right\rfloor \left\lfloor \frac{8}{4} \right\rfloor \right\} = 2$ . Verifica-se realmente

que é possível obter-se um padrão tabuleiro exato com esta frequência. O padrão tabuleiro exato  $P^2$ , obtido com a inclusão do item 2 é apresentado na Figura 3.25:

	2	2
1		
1		

Figura 3.25 – Padrão tabuleiro exato  $P^2$  obtido com a inclusão do item 2 no nó 2.

A combinação de itens associada ao padrão tabuleiro da Figura 3.25 é  $\lambda^2 = (2,2,0)$ . Esta combinação de itens apresenta valor total  $Z = 2(22) + 2(24) + 0(19) = 92$ , superior ao valor do limitante inferior  $\underline{G}$  calculado. Portanto, a melhor solução encontrada é atualizada para o padrão tabuleiro determinado neste nó e o limitante inferior é atualizado para  $\underline{G} = 92$ . O retângulo cerne  $Q^2$ , não preenchido, associado a este padrão tabuleiro é definido pela combinação  $\beta^2 = (2,2,0)$  é apresentado na Figura 3.26:

	2	2
1		
1		

Figura 3.26 – Retângulo cerne  $Q^2$  associado ao padrão tabuleiro exato da Figura 3.25.

Não há possibilidade de preenchimento do retângulo cerne  $Q^2$ . Portanto têm-se  $R^2 = Q^2$ ,  $H^2 = P^2$ ,  $\varphi^2 = \beta^2$ ,  $\alpha^2 = \lambda^2$  e  $\tilde{u}^2 = (0,0,2)$ . O retângulo cerne preenchido  $R^2$  corresponde, portanto, ao mesmo retângulo cerne apresentado na Figura 3.26. O limitante superior calculado a partir deste retângulo cerne é:

$$\bar{G} = G0 + G1 + G2 - G3 + G4 = 92 + 0 + 0 - (0) + 0 = 92$$

O valor de  $\bar{G}$  é equivalente ao valor armazenado no limitante inferior. Certamente não será obtida uma solução melhor considerando a frequência calculada para o item 2. Portanto, faz-se  $f_2 = f_2 - 1 = 2 - 1 = 1$ . Com esta remoção do item 2, determina-se o novo padrão tabuleiro exato  $P^2$  e o correspondente retângulo cerne  $Q^2$ , apresentados na Figura 3.27:

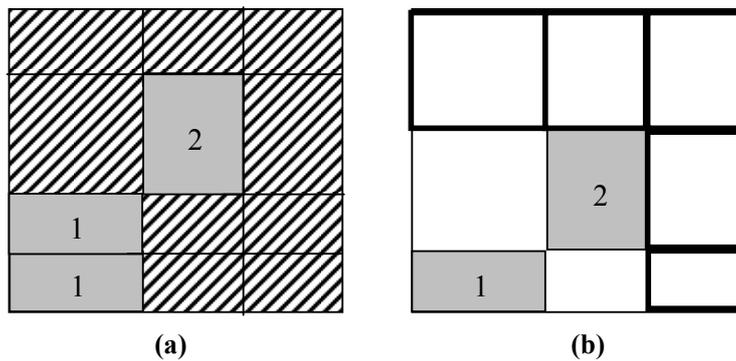


Figura 3.27 – Padrão tabuleiro exato  $P^2$  e seu retângulo cerne  $Q^2$  correspondente, obtidos com a remoção de 1 item do tipo 2 no nó 2: (a) padrão tabuleiro exato  $P^2$ , (b) retângulo cerne  $Q^2$  associado.

Novamente, não há possibilidade de preenchimento do retângulo cerne  $Q^2$  da Figura 3.27b. Portanto têm-se  $R^2 = Q^2$ ,  $H^2 = P^2$ ,  $\varphi^2 = \beta^2$ ,  $\alpha^2 = \lambda^2$  e  $\tilde{u}^2 = (0,0,2)$ . Note que o retângulo cerne  $Q^2$  foi modificado com relação ao retângulo cerne  $Q^2$  gerado antes da remoção do item 2 (vide Figura 3.26). Portanto os valores de  $P^2$ ,  $Q^2$ ,  $H^2$ ,  $\beta^2$ ,  $R^2$ ,  $\varphi^2$ ,  $H^2$  e  $\tilde{u}^2$  são atualizados no nó e o limitante superior  $\bar{G}$  é recalculado, levando em consideração as informações do novo retângulo cerne  $R^2$  (ou retângulo cerne  $Q^2$ , vide Figura 3.27b). O valor do limitante superior  $\bar{G}$  passa a ser:

$$\bar{G} = G0 + G1 + G2 - G3 + G4 = 46 + 0 + 0 - (-22) + 0 = 68$$

O valor de  $\bar{G}$  é inferior ao valor armazenado no limitante inferior. Certamente não será obtida uma solução melhor considerando a frequência calculada para o item 2. Repete-

se a remoção do item 2 fazendo  $f_2 = f_2 - 1 = 1 - 1 = 0$ . Com esta remoção, o padrão tabuleiro  $P^2$  determinado passa a ser o mesmo padrão apresentado na Figura 3.23. O novo retângulo cerne  $Q^2$  obtido é alterado para o retângulo cerne apresentado na Figura 3.24. Têm-se  $f_2 = 0$  e, tenta-se incluir o próximo tipo de item.

O próximo item a incluir é o item  $index=3$  (item 6x3). A frequência calculada para o item 3 é  $f_3 = 2$  e o padrão tabuleiro exato  $P^3$  determinado é o da Figura 3.28:

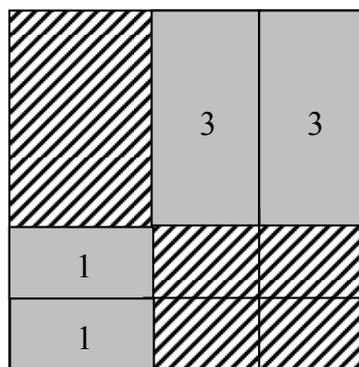


Figura 3.28 – Padrão tabuleiro exato  $P^3$  obtido a partir da remoção do segundo item do tipo 2 no nó 1 e inclusão do item tipo 3 no nó 3.

A combinação de itens associada ao padrão tabuleiro  $P^3$  da Figura 3.28 é  $\lambda^3 = (2,0,2)$ . Esta combinação de itens apresenta valor total  $Z = 2(22) + 0(24) + 2(19) = 82$ , pior do que o valor do limitante inferior  $\underline{G}$  calculado. Portanto, a melhor solução encontrada não será atualizada e prossegue-se com o cálculo do limitante superior para a inclusão do próximo item.

O último tipo de item já foi considerado e, portanto, efetua-se o “backtracking”. Retorna-se ao nó 1 com  $index=1$ . Faz-se  $f_{index} = f_1 - 1 = 2 - 1 = 1$  e  $noId=0$ . Têm-se  $f_{index} > 0$  e, portanto, o padrão tabuleiro  $P^1$  e o seu retângulo cerne  $Q^1$  correspondente são atualizados, conforme mostra a Figura 3.29:

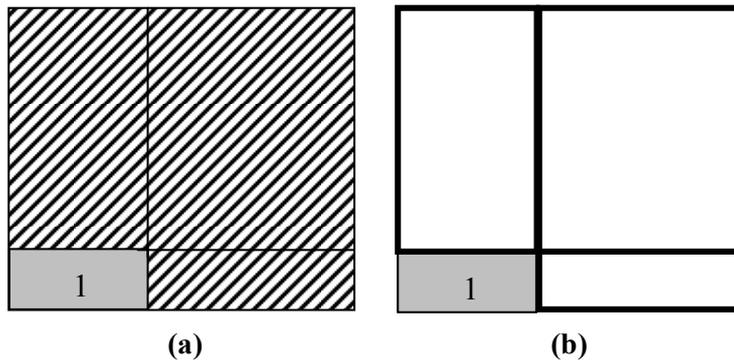


Figura 3.29 – Padrão tabuleiro exato  $P^1$  e retângulo cerne  $Q^1$  correspondente, obtidos a partir da remoção de 1 item do tipo 1, no nó 1: (a) padrão tabuleiro exato, (b) retângulo cerne associado.

Não há possibilidade de preenchimento do retângulo cerne  $Q^1$  da Figura 3.29b e têm-se  $R^1 = Q^1$ ,  $H^1 = P^1$ ,  $\varphi^1 = \beta^1$ ,  $\alpha^1 = \lambda^1$  e  $\tilde{u}^1 = (0,2,2)$ .

O novo valor calculado para o limitante superior neste nó utiliza o retângulo cerne preenchido  $R^1$ , equivalente ao retângulo cerne  $Q^1$  da Figura 3.29b. Para este retângulo cerne, obtém-se  $\bar{G} = G_0 + G_1 + G_2 - G_3 + G_4 = 140$ , superior ao valor do limitante inferior  $\underline{G}$ . Prossegue-se tentando-se incluir o item  $index=2$  na área disponível. A frequência calculada para o item 2 no padrão é  $f_2 = 2$ , o que determina o padrão tabuleiro exato  $P^2$  apresentado na Figura 3.30.

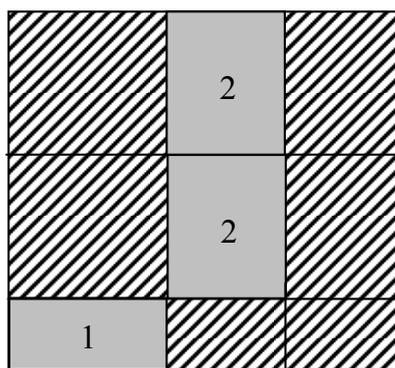


Figura 3.30 – Padrão tabuleiro exato  $P^2$  obtido a partir da remoção de 1 item do tipo 1 no nó 1 e inclusão do item 2 no nó 2.

A combinação de itens associada ao padrão tabuleiro da Figura 3.30 é  $\lambda^2 = (1,2,0)$ . Esta combinação de itens apresenta valor total  $Z = 1(22) + 2(24) + 0(19) = 70$ , pior do que o valor do limitante inferior  $\underline{G}$  calculado. A melhor solução encontrada não será atualizada e calcula-se o limitante superior para a inclusão do item 3.

O retângulo cerne associado ao padrão da Figura 3.30 apresenta combinação  $\beta^2 = (1,1,0)$  e corresponde ao mesmo retângulo cerne já apresentado na Figura 3.27b. O limitante superior calculado a partir deste retângulo cerne é  $\bar{G} = 92$ . O valor de  $\bar{G}$  é igual ao valor armazenado no limitante inferior e, portanto, faz-se  $f_2 = f_2 - 1 = 2 - 1 = 1$ . Com esta remoção do item 2, determina-se o padrão tabuleiro exato  $P^2$  e seu retângulo cerne  $Q^2$  correspondente, apresentados na Figura 3.31:

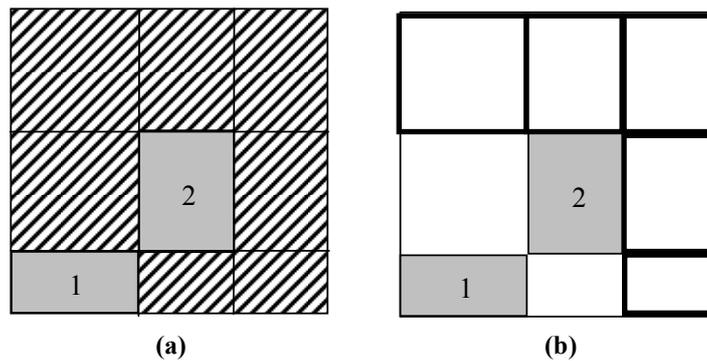


Figura 3.31 – Padrão tabuleiro exato  $P^2$  e seu retângulo cerne  $Q^2$  correspondente, obtidos a partir da remoção de 1 item do tipo 2, no nó 2: (a) padrão tabuleiro exato, (b) retângulo cerne associado.

Não houve alterações no retângulo cerne da Figura 3.31, em relação ao retângulo cerne anterior à remoção do item 2 (vide Figura 3.27b). Portanto, faz-se  $f_2 = f_2 - 1 = 1 - 1 = 0$ . Têm-se  $f_2 = 0$  e, neste caso, considera-se a inclusão do próximo item que é o item index=3 (item 6x3). A frequência calculada para o item 3 é  $f_3 = 2$ , o que determina o padrão tabuleiro exato  $P^3$  apresentado na Figura 3.32:

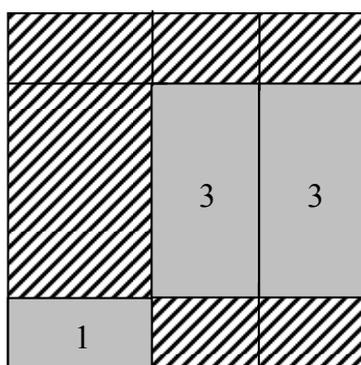


Figura 3.32 – Padrão tabuleiro exato  $P^3$  obtido com a remoção do segundo item do tipo 2 no nó 4 e inclusão do item 3 no nó 3.

A combinação de itens associada ao padrão tabuleiro da Figura 3.32 é  $\lambda^3 = (1,0,2)$ . Esta combinação de itens apresenta valor total  $Z = 1(22) + 0(24) + 2(19) = 60$ , pior do que o valor do limitante inferior  $\underline{G}$  calculado. Portanto, a melhor solução encontrada não será atualizada e prossegue-se com o cálculo do limitante superior para a inclusão do próximo item. O último tipo de item já foi considerado, portanto, efetua-se o “backtracking”. Retorna-se ao nó 1 e faz-se  $f_{index} = f_1 - 1 = 1 - 1 = 0$ .

Inclui-se o próximo item  $index=2$  (item  $4 \times 3$ ). A frequência calculada para este item no padrão é  $f_2 = 2$ , o que determina uma combinação  $\lambda^2 = (0,2,0)$ , cujo padrão  $P^2$  está apresentado na Figura 3.33:

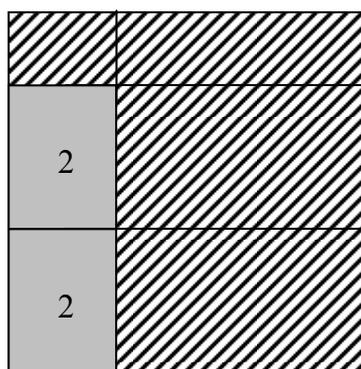


Figura 3.33 - Padrão tabuleiro exato  $P^2$  obtido com a inclusão do item 2 no nó 2.

O valor da combinação  $\lambda^2$  do padrão tabuleiro exato da Figura 3.33 é  $Z=48$ , inferior ao valor do limitante inferior e, portanto, a melhor solução não é atualizada. O retângulo

cerne não preenchido  $Q^2$  associado ao padrão da Figura 3.33 é apresentado na Figura 3.34:

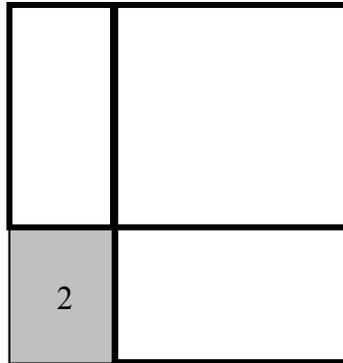


Figura 3.34 – Retângulo cerne  $Q^2$  associado ao padrão tabuleiro  $P^2$  da Figura 3.33.

Não há possibilidade de preenchimento de  $Q^2$  e o limitante superior obtido a partir deste retângulo cerne (vide Figura 3.34) é:

$$\bar{G} = G_0 + G_1 + G_2 - G_3 + G_4 = 24 + 19 + 0 - (-24) + 44,3333 = 111,3333$$

Tem-se  $\bar{G} > \underline{G}$  e, portanto, prossegue-se com a inclusão do próximo item. O próximo item a incluir é o item  $index=3$  (item 6x3). A frequência calculada para este item no padrão é  $f_3 = 2$ , o que determina uma combinação  $\lambda^3 = (0,2,2)$ , de valor  $Z=86$ . O valor do padrão é inferior ao valor do limitante inferior. Portanto, a melhor solução não é atualizada.

Retorna-se ao nó 2 e faz-se  $index=2$ . Faz-se  $f_{index} = f_2 - 1 = 2 - 1 = 1$  e  $noId=0$ . O padrão tabuleiro  $P^2$  e o seu retângulo cerne  $Q^2$  são atualizados conforme apresentado na Figura 3.35:

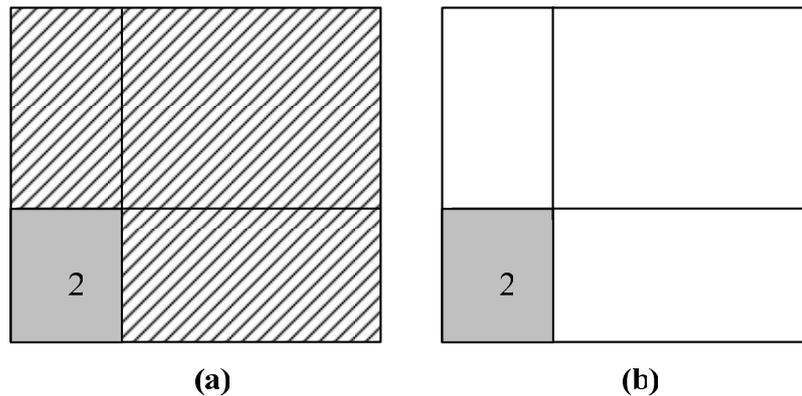


Figura 3.35 – Padrão tabuleiro exato  $P^2$  e seu retângulo cerne  $Q^2$  correspondente, obtidos a partir da remoção de 1 item do tipo 2, no nó 2: (a) padrão tabuleiro exato  $P^2$ , (b) retângulo cerne  $Q^2$  associado.

Não houve alteração no retângulo cerne  $Q^2$  em relação ao retângulo cerne anteriormente obtido (vide Figura 3.34). Faz-se  $f_{index} = f_2 - 1 = 1 - 1 = 0$ . A frequência tornou-se nula e, portanto, considera-se a inclusão do item  $index=3$ . A combinação obtida com a inclusão deste item é  $\lambda^3 = (0,0,2)$ , de valor  $Z=38$ , pior do que o valor do limitante inferior.

O item 3 foi o último item considerado e todos os outros caminhos da árvore de enumeração já foram explorados. Portanto, a solução ótima de valor  $\underline{G}=92$  foi encontrada e o padrão tabuleiro exato obtido é apresentado na Figura 3.36:

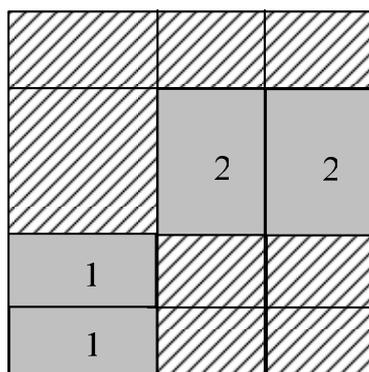


Figura 3.36 – Solução final: padrão tabuleiro exato obtido.

O Algoritmo Melhorado de Enumeração Implícita foi implementado em linguagem C++. Foram realizados testes computacionais para avaliar o tempo de execução desta implementação. As instâncias utilizadas nos testes computacionais foram extraídas de Yanasse e Morabito (2005). As instâncias de Yanasse e Morabito (2005) consistem de problemas restritos, gerados aleatoriamente por estes autores para 5, 10, 20, 50 e 100 diferentes tipos de itens. Estas instâncias foram utilizadas por Yanasse e Morabito (2005) para avaliação dos modelos matemáticos desenvolvidos por eles para padrões tabuleiros exatos e não exatos restritos.

Os testes apresentados a seguir, comparam os tempos computacionais que foram obtidos pelo Algoritmo Enumerativo (CECA), pelo Algoritmo de Enumeração Implícita (GGCECA), pelo Algoritmo Melhorado de Enumeração Implícita (NGGCECA) e pelo modelo de Yanasse e Morabito (2005) (YM2005). O modelo de Yanasse e Morabito (2005) é equivalente ao modelo linearizado para determinação de padrões tabuleiros exatos e restritos, apresentado anteriormente em Yanasse e Morabito (2003). Nos testes realizados, os tempos computacionais estão expressos em segundos. Os algoritmos CECA, GGCECA e NGCECA foram executados num PC Pentium 4, 2.2GHz, com 256MB RAM, rodando o sistema operacional Linux. Os resultados obtidos pelo modelo YM2005 foram executados neste mesmo equipamento, mas com auxílio da linguagem de modelagem GAMS Rev. 118 e da biblioteca de otimização CPLEX 7.0 utilizando o sistema operacional Windows XP Professional (SP2).

Na Tabela 3.24 apresenta-se os resultados obtidos para as instâncias com 5 tipos de itens e objeto de dimensão 100x100:

Tabela 3.24 – Resultados obtidos para o conjunto 1 de instâncias extraídas de Yanasse e Morabito (2005) com 5 tipos de itens diferentes e objeto de dimensão 100x100.

<b>Instância</b>	<b>Solução Obtida</b>	<b>CECA</b>	<b>GGCECA</b>	<b>NGGCECA</b>	<b>YM2005</b>
1	7800	0.98	0.1	0.03	0.12
2	8280	0.45	0.03	0.005	0.12

(Continua)

Tabela 3.24 – Conclusão.

<b>Instância</b>	<b>Solução Obtida</b>	<b>CECA</b>	<b>GGCECA</b>	<b>NGGCECA</b>	<b>YM2005</b>
3	7000	0.35	0.02	0.01	0.13
4	7566	0.22	0.01	0.01	0.14
5	7936	0.17	0.01	0.002	0.12
6	9306	0.10	0.01	0.003	0.11
7	4770	0.10	0.02	0.01	0.13
8	7904	0.10	0.01	0.002	0.12
9	7800	0.10	0.01	0.003	0.1
10	5810	0.04	0.01	0.005	0.22

Observa-se, pela Tabela 3.24 que o Algoritmo Melhorado de Enumeração Implícita apresentou os melhores tempos computacionais dentre os métodos testados. Os tempos do novo algoritmo ficaram sempre abaixo de décimos de segundos e, em alguns casos, atingiu milésimos de segundos. Nota-se também um tempo computacional bastante satisfatório obtido pelo algoritmo de enumeração implícita que ficou com o segundo melhor tempo computacional. O Algoritmo Enumerativo obteve tempos computacionais bastante elevados e foi o pior avaliado. Em alguns casos, o tempo computacional do Algoritmo Enumerativo chegou a quase 1 segundo de execução (instância 1).

Na Tabela 3.25 apresenta-se os resultados obtidos para as instâncias com 5 tipos de itens, e com um objeto de dimensão 10x10:

Tabela 3.25 – Resultados obtidos para o conjunto 2 de instâncias extraídas de Yanasse e Morabito (2005) com 5 tipos de itens diferentes e objeto de dimensão 10x10.

<b>Instância</b>	<b>Solução Obtida</b>	<b>CECA</b>	<b>GGCECA</b>	<b>NGGCECA</b>	<b>YM2005</b>
1	100	0.01	0.05	0.03	0.09
2	70	0.04	0.01	0.004	0.07
3	80	0.30	0.06	0.02	0.11
4	84	0.80	0.04	0.03	0.11
5	100	0.22	0.02	0.02	0.08
6	90	0.02	0.005	0.003	0.13

(Continua)

Tabela 3.25 – Conclusão.

<b>Instância</b>	<b>Solução Obtida</b>	<b>CECA</b>	<b>GGCECA</b>	<b>NGGCECA</b>	<b>YM2005</b>
7	90	0.22	0.02	0.004	0.08
8	100	0.03	0.01	0.01	0.06
9	80	0.12	0.02	0.01	0.12
10	100	0.01	0.002	0.01	0.10

Conforme apresentado na Tabela 3.25, a mudança na escala resultou na redução do tempo computacional do Algoritmo Enumerativo que se apresentava bastante elevado. No entanto, esta melhoria no tempo computacional ainda não foi suficiente para tornar o Algoritmo Enumerativo competitivo com relação aos demais métodos testados. O tempo computacional do Algoritmo Melhorado de Enumeração Implícita continuou sendo o melhor dentre os métodos avaliados, seguido do tempo computacional obtido pelo Algoritmo de Enumeração Implícita e pelo tempo computacional do modelo de Yanasse e Morabito (2005).

Na Tabela 3.26 apresenta-se os resultados obtidos para as instâncias com 10 tipos de itens e objeto de dimensão 100x100:

Tabela 3.26 – Resultados obtidos para o conjunto 3 de instâncias extraídas de Yanasse e Morabito (2005) com 10 tipos de itens diferentes e objeto de dimensão 100x100.

<b>Instância</b>	<b>Solução Obtida</b>	<b>CECA</b>	<b>GGCECA</b>	<b>NGGCECA</b>	<b>YM2005</b>
1	8730	450.18	0.6	0.1	0.35
2	6615	13.85	0.06	0.02	0.51
3	9200	47.04	0.1	0.02	0.29
4	6880	5.34	0.03	0.02	0.28
5	8470	1475.13	1.86	0.04	0.45
6	9100	2014.66	1.11	0.1	0.39
7	7200	283.18	0.32	0.03	0.58
8	9216	1646.7	2.75	0.2	0.31
9	8178	13.63	0.05	0.01	0.30
10	9300	375.37	0.83	0.1	0.31

Pode-se observar pela Tabela 3.26, que o tempo computacional do Algoritmo Enumerativo foi novamente o pior dentre os métodos avaliados, sendo extremamente elevado e, em alguns casos, ultrapassando o limite de meia hora de execução (instância 6). Isto se deve, em parte, à dimensão dos itens e do objeto, que não sofreram uma mudança de escala, tal como o ocorrido nos dados das instâncias da Tabela 3.25. Observa-se também pela Tabela 3.26, um aumento no tempo computacional do Algoritmo de Enumeração Implícita (terceiro melhor tempo) que, em alguns casos, ficou acima de 1 segundo de execução (instâncias 5, 6 e 8). O Algoritmo Melhorado de Enumeração Implícita (melhor tempo) e o Modelo de Yanasse e Morabito (2005) (segundo melhor tempo) obtiveram tempos computacionais sempre abaixo de 1 segundo, sendo que o Algoritmo Melhorado foi melhor do que o Modelo de Yanasse e Morabito (2005) em todas as instâncias testadas..

Na Tabela 3.27 apresenta-se os resultados obtidos para as instâncias com 20 tipos de itens e objeto de dimensão 100x100:

Tabela 3.27 – Resultados obtidos para o conjunto 4 de instâncias extraídas de Yanasse e Morabito (2005) com 20 tipos de itens diferentes e objeto de dimensão 100x100.

<b>Instância</b>	<b>Solução Obtida</b>	<b>CECA</b>	<b>GGCECA</b>	<b>NGGCECA</b>	<b>YM2005</b>
1	9900	-	25.09	0.7	1.09
2	9300	-	2.38	0.1	2.50
3	8544	-	1.20	0.1	1.21
4	9216	-	2.40	0.3	1.60
5	9072	-	10.69	0.2	1.78
6	8600	-	1.01	0.1	2.23
7	8960	-	1.45	0.1	1.11
8	9460	-	6.79	0.1	1.40
9	9120	-	1.15	0.05	1.51
10	9250	-	0.71	0.1	1.23

Para os testes da Tabela 3.27, os tempos computacionais do Algoritmo Enumerativo foram omitidos por serem demasiadamente elevados e não serem competitivos em relação aos demais métodos. O Algoritmo Melhorado de Enumeração Implícita foi novamente o mais rápido dentre os métodos avaliados, sendo mais rápido do que o

Modelo de Yanasse e Morabito (2005) (segundo melhor tempo) nas 10 instâncias avaliadas. O Algoritmo de Enumeração Implícita foi o terceiro mais rápido, com tempo computacional acima de 1 segundo de execução na maior parte das instâncias testadas.

Na Tabela 3.28 apresenta-se os resultados obtidos para as instâncias com 50 tipos de itens e objeto de dimensão 100x100:

Tabela 3.28 – Resultados obtidos para o conjunto 5 de instâncias extraídas de Yanasse e Morabito (2005) com 50 tipos de itens diferentes e objeto de dimensão 100x100.

<b>Instância</b>	<b>Solução Obtida</b>	<b>CECA</b>	<b>GGCECA</b>	<b>NGGCECA</b>	<b>YM2005</b>
1	10000	-	1514.2	6.43	29.53
2	10000	-	97.06	3.87	13.31
3	10000	-	92.13	1.98	19.62
4	9800	-	53.56	1.13	14.36
5	10000	-	68.61	1.18	15.46
6	9540	-	1539.45	17.67	25.87
7	9900	-	17.71	1.48	11.06
8	9604	-	166.68	3.05	14.27
9	9600	-	31.08	0.99	18.56
10	9702	-	198.11	2.07	21.96

Os resultados dos testes apresentados na Tabela 3.28 omitem novamente os tempos do Algoritmo Enumerativo por não serem competitivos aos demais métodos. Pela Tabela 3.28, observa-se que o Algoritmo Melhorado de Enumeração Implícita, continua sendo o mais rápido dentre todos os métodos avaliados. Verifica-se novamente que este algoritmo foi mais rápido do que o Modelo de Yanasse e Morabito (2005) (segundo melhor tempo) em todas as instâncias avaliadas. Observa-se também na Tabela 3.28 uma grande diferença de tempo computacional favorável ao Algoritmo Melhorado de Enumeração Implícita, em relação à sua versão anterior. Os tempos computacionais do Algoritmo Melhorado ficaram sempre abaixo de 1 minuto de execução, enquanto que os tempos computacionais do Algoritmo de Enumeração Implícita ultrapassaram 25 minutos de execução, em algumas instâncias (instâncias 1 e 6).

Na Tabela 3.29 apresenta-se os resultados obtidos para as instâncias com 100 tipos de itens e objeto de dimensão 100x100:

Tabela 3.29 – Resultados obtidos para o conjunto 6 de instâncias extraídas de Yanasse e Morabito (2005) com 100 tipos de itens diferentes e objeto de dimensão 100x100.

<b>Instância</b>	<b>Solução Obtida</b>	<b>CECA</b>	<b>GGCECA</b>	<b>NGGCECA</b>	<b>YM2005</b>
1	10000	-	-	41.79	254.95
2	10000	-	-	6.79	234.70
3	10000	-	-	55.28	240.76
4	10000	-	-	10.86	252.09
5	9900	-	-	36.52	265.70
6	9800	-	-	54.34	213.32
7	9800	-	-	39.38	214.63
8	10000	-	-	19.97	225.36
9	10000	-	-	3.44	192.92
10	9800	-	-	28.08	239.47

Os resultados apresentados na Tabela 3.29 referem-se apenas ao tempo computacional do Algoritmo Melhorado de Enumeração Implícita e ao tempo computacional do Modelo de Yanasse e Morabito (2005). Os tempos computacionais dos demais métodos foram omitidos, por falta de competitividade por serem demasiadamente elevados (Algoritmo Enumerativo e Algoritmo de Enumeração Implícita). Os resultados apresentados na Tabela 3.29 mostram que o Algoritmo Melhorado de Enumeração Implícita também é mais rápido do que o Modelo de Yanasse e Morabito (2005) em instâncias com grande variedade de tipos de itens. Pela Tabela 3.29, o Algoritmo Melhorado de Enumeração Implícita foi mais rápido do que o Modelo de Yanasse e Morabito (2005) em todas as instâncias avaliadas com grande diferença de tempo computacional a seu favor.

Na Tabela 3.30, apresenta-se os tempos médios obtidos com os métodos avaliados:

Tabela 3.30 – Tempos computacionais médios obtidos para o conjunto de instâncias extraídas de Yanasse e Morabito (2005) com 5, 10, 20, 50 e 100 tipos de itens diferentes.

Conj. Instâncias	<i>n</i>	CECA	GGCECA	NGGCECA	YM2005
Conj. 1	5	0.26	0.02	0.01	0.13
Conj. 2	5	0.18	0.02	0.01	0.09
Conj. 3	10	632.50	0.8	0.1	0.38
Conj. 4	20	-	5.29	0.2	1.57
Conj. 5	50	-	377.86	3.98	18.4
Conj. 6	100	-	-	29.64	233.39

Conforme apresentado na Tabela 3.30, o Algoritmo Melhorado de Enumeração Implícita obteve os melhores tempos computacionais médios em todos os conjuntos de instâncias avaliados. Para os conjuntos de instâncias com grande variedade de tipos de itens (conjuntos 5 e 6), o tempo computacional médio do Algoritmo Melhorado permaneceu abaixo de 30 segundos. O Algoritmo Enumerativo e o Algoritmo de Enumeração Implícita apresentaram tempos computacionais competitivos apenas em instâncias com, no máximo 10 tipos de itens diferentes. Nas instâncias com mais de 10 itens, o tempo computacional ficou demasiadamente elevado para estes algoritmos, inviabilizando sua aplicação.

Na Tabela 3.31 apresenta-se os resultados obtidos pelo Algoritmo Melhorado de Enumeração Implícita para o conjunto de instâncias da Tabela 3.19:

Tabela 3.31 – Versão melhorada 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	<i>n</i>	Valor da Solução Ótima	Tempo Algoritmo de Enumeração Implícita	Tempo Versão Melhorada do Algoritmo de Enumeração Implícita	Tempo Modelo de Yanasse e Morabito (2005)
1	5	7800*	0.08	0.03	0.12
2	5	8280*	0.03	0.005	0.12

(Continua)

Tabela 3.31 - Conclusão.

<b>Instância</b>	<b><i>n</i></b>	<b>Valor da Solução Ótima</b>	<b>Tempo Algoritmo de Enumeração Implícita</b>	<b>Tempo Versão Melhorada do Algoritmo de Enumeração Implícita</b>	<b>Tempo Modelo de Yanasse e Morabito (2005)</b>
3	5	7000*	0.02	0.01	0.13
4	5	7566	0.01	0.004	0.14
5	5	7936*	0.01	0.002	0.12
6	5	9306	0.01	0.003	0.11
7	5	4770	0.01	0.01	0.13
8	5	7904*	0.01	0.001	0.12
9	5	7800*	0.005	0.001	0.1
10	5	5810	0.01	0.005	0.22
11	5	63*	0.001	0.001	0.11
12	5	90	0.002	0.002	0.09
13	5	70	0.001	0.001	0.07
14	5	80	0.001	0.001	0.09
15	5	90	0.01	0.01	0.10
16	5	78	0.0005	0.0001	0.10
17	5	100*	0.001	0.001	0.11
18	5	82	0.001	0.001	0.13
19	5	100*	0.001	0.001	0.09
20	5	70	0.001	0.001	0.11
VAG	5	6450	0.01	0.01	0.14
MA	13	6715500	87.7	2.72	0.26
CW1	7	240	0.005	0.004	0.13
CW2	10	1687	0.02	0.01	0.2
CW3	20	1080*	0.02	0.02	0.54
W	20	2277*	0.01	0.005	0.63
OF1	10	1769	0.005	0.004	0.2
OF2	10	1684	0.01	0.01	0.2

\* Solução homogênea

Conforme apresentado na Tabela 3.31, obteve-se uma redução mais significativa do tempo computacional na instância MA (de Yanasse e Morabito, 2005) com o uso do Algoritmo Melhorado de Enumeração Implícita. Para esta instância, extraída de Yanasse e Morabito (2005), o Algoritmo de Enumeração Implícita obteve um tempo computacional acima de 1 minuto de execução. Com o uso do Algoritmo Melhorado de Enumeração Implícita, este tempo computacional foi reduzido para cerca de 2.72 segundos de execução, mas ainda não foi o suficiente para ser mais rápido do que o

tempo computacional do Modelo de Yanasse e Morabito (2005), executado abaixo em 0.26 segundos. O exemplo extraído de Yanasse e Morabito (2005) é, na realidade, um problema irrestrito. A resolução desta instância pelo algoritmo enumerativo de Katsurayama e Yanasse (1999, 2000) demora 0.002 segundos. Portanto, como já mencionado anteriormente, pode-se utilizar o algoritmo enumerativo de Katsurayama e Yanasse (1999, 2000) para determinar uma solução inicial para o problema. Caso a solução seja viável, o algoritmo pára com esta solução ótima. Se inviável, prossegue-se com o Algoritmo Melhorado de Enumeração Implícita. Conforme observado na Tabela 3.31, a redução do tempo computacional foi pouco significativa nas demais instâncias avaliadas. Na Tabela 3.32, apresenta-se os resultados obtidos pelo Algoritmo Melhorado de Enumeração Implícita para as instâncias da Tabela 3.20:

Tabela 3.32 – Versão melhorada do algoritmo de enumeração implícita (instâncias extraídas de Hifi, 2005).

<b>Instância</b>	<b><i>n</i></b>	<b>Valor da Solução Ótima</b>	<b>Tempo Algoritmo de Enumeração Implícita</b>	<b>Tempo Versão Melhorada do Algoritmo de Enumeração Implícita</b>
HH	5	7020*	0.01	0.01
2	10	1687	0.02	0.01
3	20	1080*	0.02	0.02
A1	20	1040*	0.04	0.03
A2	20	1560	0.05	0.04
STS2	30	3800	0.39	0.05
CHL1	30	6579	1.5	0.65
CHL2	10	1226	0.01	0.01
CW1	25	5358	0.13	0.07
CW2	35	2964*	0.11	0.11
CW3	40	4428*	0.15	0.01
Hchl2	35	7408	13.5	4.01
Hchl9	35	4250	1.29	0.18
2s	10	1689	0.02	0.01
3s	20	2277*	0.01	0.005
A1s	20	1750*	0.02	0.02
A2s	20	2714	0.03	0.01
STS2s	30	3952	0.3	0.1
STS4s	20	7614	0.54	0.1
CHL1s	30	12276	0.75	0.05
CHL2s	10	1922	0.01	0.01
A3	20	4524	0.10	0.02

(Continua)

Tabela 3.32 – Conclusão.

<b>Instância</b>	<b><math>n</math></b>	<b>Valor da Solução Ótima</b>	<b>Tempo Algoritmo de Enumeração Implícita</b>	<b>Tempo Versão Melhorada do Algoritmo de Enumeração Implícita</b>
A4	20	3180	0.10	0.10
A5	20	12276	0.16	0.03
CHL5	10	220	0.002	0.002
CHL6	30	13113	2.41	0.15
CHL7	35	12958	9.22	1.38
CU1	25	12200	0.10	0.03
CU2	35	20250	0.16	0.04
Hchl3s	10	7388	0.68	0.21
Hchl4s	10	7388	0.13	0.08
Hchl6s	22	40835	2.18	0.43
Hchl7s	40	41869	11.7	1.46
Hchl8s	10	416	0.02	0.02

\* Solução homogênea

Pela Tabela 3.32, verifica-se uma redução do tempo computacional em grande parte das instâncias avaliadas com uso da versão melhorada do Algoritmo de Enumeração Implícita. Observa-se que esta redução do tempo computacional é ainda mais significativa nas instâncias com maior variedade de tipos de itens (por exemplo, instâncias Hchl7s, CU2, CU1, CHL7, CHL6, Hchl6s, Hchl9, Hchl2, STS2, CHL1).

As instâncias da Tabela 3.32 apresentam itens de dimensões bastante diferenciadas, o que favorece a obtenção de soluções ótimas bastante triviais, compostas de itens ou faixas do mesmo tipo. Também foram realizados testes computacionais para avaliar o tempo computacional do Algoritmo Melhorado de Enumeração Implícita usando instâncias que apresentam itens de dimensões pouco diferenciadas. Na Tabela 3.33 apresenta-se os testes computacionais realizados sobre as instâncias da biblioteca C1GCUT. As instâncias da biblioteca C1GCUT foram geradas aleatoriamente, de forma similar ao conjunto de instâncias 1GCUT apresentado anteriormente para avaliar a heurística para determinação de padrões tabuleiros irrestritos desenvolvida neste trabalho de tese. Considera-se um objeto retangular de dimensão 100x100. Para compor as dimensões dos itens retangulares menores, efetuou-se a combinação um a um entre  $k$  números inteiros gerados aleatoriamente no intervalo entre 20 e 70. Por exemplo, sejam

a, b, c, os  $k=3$  números inteiros gerados neste intervalo, obtêm-se  $n=k \times k=9$  tipos de itens retangulares menores de dimensões  $axa$ ,  $axb$ ,  $axc$ ,  $bxa$ ,  $bxb$ ,  $bxc$ ,  $cxa$ ,  $cxb$ ,  $cxc$ . As demandas associadas a estes itens foram geradas aleatoriamente no intervalo inteiro entre 1 e 10. Considera-se que o valor de utilidade (lucro) associado a cada item seja proporcional ao valor de sua área, com variação de  $\pm 10\%$ .

Tabela 3.33 – Versão melhorada do algoritmo de enumeração implícita (instâncias C1GCUT).

<b>Instância</b>	<b><math>n</math></b>	<b>Valor da Solução Ótima</b>	<b>Tempo Algoritmo de Enumeração Implícita</b>	<b>Tempo Versão Melhorada do Algoritmo de Enumeração Implícita</b>	<b>Tempo Modelo de Yanasse e Morabito (2005)</b>
c1gcut1	9	9494	0.01	0.005	0.08
c1gcut2	9	9497	0.15	0.07	0.22
c1gcut3	9	9019	0.01	0.004	0.12
c1gcut4	16	10154	0.10	0.04	0.26
c1gcut5	16	8737	0.004	0.004	0.09
c1gcut6	16	9813	0.01	0.004	0.08
c1gcut7	25	10286	0.01	0.005	0.17
c1gcut8	25	10357	0.21	0.05	0.26
c1gcut9	25	10030	0.20	0.08	0.24
c1gcut10	36	8629	1.84	0.75	0.77
c1gcut11	36	10202	0.23	0.08	0.41
c1gcut12	49	9773	0.14	0.09	0.5
c1gcut13	49	10190	0.25	0.08	0.66
c1gcut14	64	10217	2.19	0.48	1.06
c1gcut15	64	10014	0.13	0.08	0.83

Os resultados apresentados na Tabela 3.33 mostram que o Algoritmo Melhorado de Enumeração Implícita também é eficiente na resolução de problemas que apresentam itens de dimensões pouco diferenciadas. Observa-se, pela Tabela 3.33, que o Algoritmo Melhorado de Enumeração Implícita foi mais rápido do que sua versão original e o Modelo de Yanasse e Morabito (2005) em todas as instâncias avaliadas.

### 3.3.4 – Uma Modificação para o Algoritmo Melhorado de Enumeração Implícita

Modificações ainda podem ser realizadas no limitante superior visando reduzir ainda mais o tempo computacional do Algoritmo Melhorado de Enumeração Implícita. Vale lembrar que no Algoritmo Melhorado de Enumeração Implícita, o cálculo dos valores estimados  $G1$ ,  $G2$  e  $G4$  para os itens a serem incluídos nos bins acima, à direita e a nordeste do retângulo cerne, é feito de forma relaxada, ou seja, baseia-se no preenchimento das áreas dos bins usando o primeiro item  $(k+1, \dots, n)$  que cabe nestes bins. Na modificação proposta a seguir, utiliza-se um outro limitante superior para  $G1$ ,  $G2$  e  $G4$ .

A idéia da modificação proposta é calcular o limitante superior com base na definição de uma nova faixa horizontal (ou vertical) a ser incluída no retângulo cerne quando existem itens no padrão ainda não considerados no retângulo cerne. A definição da faixa é feita a partir da inclusão de um item no padrão, ainda não considerado no retângulo cerne. A rigor, esta modificação é similar ao cálculo do limitante dando um passo adiante na árvore de enumeração. A diferença é que na modificação proposta, a nova faixa é definida a partir de um item já presente no padrão, mas não considerado no retângulo cerne e não a partir de um item que ainda não foi considerado no processo de enumeração.

Caso exista, selecionam-se os itens presentes no padrão, ainda não considerados no retângulo cerne, ou seja, itens em que  $(\lambda_s - \beta_s) \neq 0$  para algum  $s \in \{1, \dots, k\}$ . O item  $s$  deve estar em algum bin acima ou à direita. Considerando que o item  $s$  esteja incluído no bin acima, define-se uma nova faixa horizontal na largura do item  $s$ . De forma similar, considerando que o item  $s$  esteja incluído no bin à direita, define-se uma nova faixa vertical no comprimento do item  $s$ . Na modificação proposta, calculam-se os valores de  $G1$ ,  $G2$  e  $G4$  considerando-se estas duas possibilidades (vide Figura 3.37).

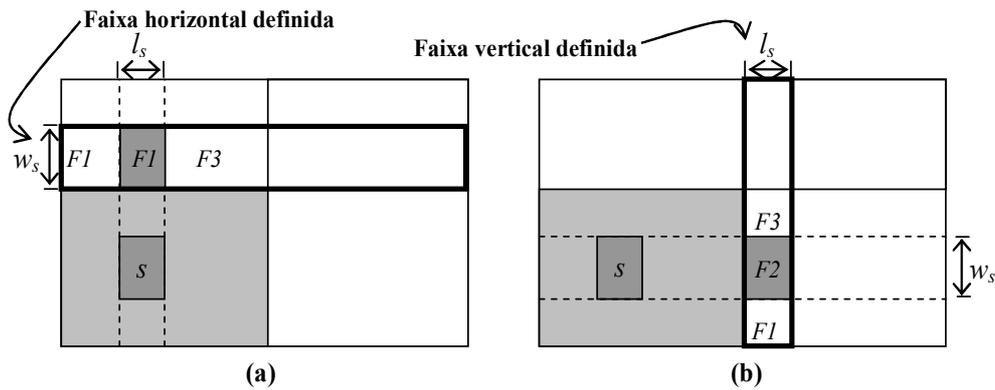


Figura 3.37 – Definição de uma nova faixa horizontal (ou vertical) a partir da inclusão do item  $s$  acima (ou à direita) do retângulo cerne: (a) inclusão do item  $s$  acima do retângulo cerne e definição de uma nova faixa horizontal; (b) inclusão do item  $s$  à direita do retângulo cerne e definição de uma nova faixa vertical.

Note que, uma vez definida a nova faixa horizontal (ou vertical), é possível preenchê-la usando os itens que cabem nos espaços disponíveis na faixa. Admitindo que este preenchimento seja feito nos espaços situados logo acima, ou à direita do retângulo cerne da Figura 3.37 (espaços  $F1$ ,  $F2$  e  $F3$ ), obtêm-se os retângulos cernes aumentados, apresentados na Figura 3.38a e 3.38b.

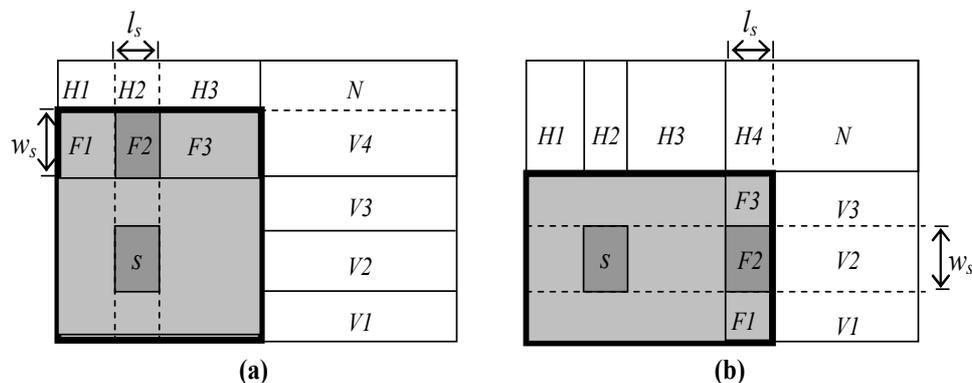


Figura 3.38 – Retângulo cerne aumentado, obtido a partir da definição de uma nova faixa horizontal (ou vertical): (a) retângulo cerne aumentado, obtido a partir da definição de uma nova faixa horizontal; (b) retângulo cerne aumentado, obtido a partir da definição de uma nova faixa vertical.

Os limitantes superiores para os dois casos possíveis são calculados da mesma forma do algoritmo melhorado de enumeração implícita, considerando os itens que podem ser colocados nos bins acima e à direita do retângulo cerne e no canto nordeste, corrigindo

os valores com os itens no padrão ainda não considerados nos retângulos cernes. O limitante superior para cada tipo de item no padrão e ainda não considerado no retângulo cerne, é obtido pegando-se o melhor dos 2 limitantes calculados. Estes limitantes valem para quaisquer itens do padrão ainda não considerados no retângulo cerne. Assim, podemos obter o novo limitante  $\bar{G}$  pegando-se o mínimo dentre os limitantes calculados para cada tipo de item.

Foram realizados testes computacionais utilizando os conjuntos de instâncias extraídas de Yanasse e Morabito (2005) e a modificação proposta. Na Tabela 3.34 apresenta-se a média dos tempos computacionais obtidos pelo Algoritmo Melhorado de Enumeração Implícita e os tempos computacionais médios obtidos com sua versão modificada:

Tabela 3.34 – Tempos computacionais médios obtidos pelo algoritmo melhorado e pela versão modificada para o conjunto de instâncias extraídas de Yanasse e Morabito (2005) com 5, 10, 20, 50 e 100 tipos de itens diferentes.

<b>Conj. Instâncias</b>	<b><i>n</i></b>	<b>Algoritmo Melhorado de Enumeração Implícita</b>	<b>Versão Modificada do Algoritmo Melhorado de Enumeração Implícita</b>
Conj. 1	5	0.01	0.004
Conj. 2	5	0.01	0.01
Conj. 3	10	0.1	0.05
Conj. 4	20	0.2	0.2
Conj. 5	50	3.98	3.96
Conj. 6	100	29.64	29.70

Pela Tabela 3.34, verifica-se, em média, uma redução adicional no tempo computacional médio com uso da versão modificada. É importante notar que o cálculo do novo limitante superior requer um tempo computacional complementar. Os testes computacionais da Tabela 3.34 indicam que este tempo computacional complementar nem sempre compensa.

## CAPÍTULO 4

### CONCLUSÃO

Fatores que podem influenciar os custos de produção e a produtividade em ambientes de corte foram identificados em alguns trabalhos encontrados na literatura. Estes trabalhos foram classificados em dois grupos.

No primeiro grupo de fatores estão os trabalhos que tentam reduzir o tempo de corte com foco na programação da produção. Por exemplo, reduzir o número de diferentes tipos de padrões de corte pode resultar num menor tempo de preparação (setup) do equipamento, aumentando-se a produtividade da serra. Neste mesmo grupo, podem ser incluídos os trabalhos relacionados com o problema de redução do número de ciclos de máquina. Minimizar o número de ciclos de máquina resulta numa redução do tempo geral de produção.

No segundo grupo estão os trabalhos que tentam reduzir o tempo de corte com foco no corte dos padrões. Os padrões de corte podem ser mais simples ou mais complexos de serem produzidos pela máquina. Padrões mais simples exigem um número menor de operações no processo de corte reduzindo o tempo de corte dos objetos. Também estão presentes neste mesmo grupo, trabalhos que exploram a determinação de padrões com redução no comprimento dos cortes. A determinação de padrões que apresentam comprimento de cortes reduzidos, além de reduzir o tempo de corte, permite também reduzir o desgaste da serra.

As principais contribuições desta tese de doutorado estão focadas no segundo grupo de fatores. Foram desenvolvidos novos métodos exatos e heurísticas para determinação de padrões de corte simples, denominados tabuleiros. Destacam-se as seguintes contribuições realizadas:

- 1 - Desenvolvimento de uma nova heurística para determinação de padrões tabuleiros exatos e não exatos irrestritos:**

Foi desenvolvida uma nova heurística para determinação de padrões tabuleiros irrestritos. A heurística proposta combina uma maior variedade de faixas em comparação à heurística de Morabito e Arenales (2000). A heurística foi implementada em linguagem C++. Testes computacionais mostraram que a nova heurística para determinação de padrões tabuleiros irrestritos é bastante eficiente tanto na determinação de padrões tabuleiros tipo exatos como na determinação de padrões tabuleiros tipo não exatos. O tempo de processamento da nova heurística ficou, em média, pouco acima do tempo da heurística de Morabito e Arenales (2000) e bem abaixo do tempo de processamento do algoritmo enumerativo de Katsurayama e Yanasse (1999, 2000). Testes computacionais realizados com instâncias geradas exclusivamente para a determinação de padrões tabuleiros comprovaram a eficiência da heurística proposta mostrando que ela consegue obter melhores soluções em termos de tempo de processamento e qualidade da solução, se comparado com outros procedimentos apresentados na literatura.

## **2 - Desenvolvimento de um algoritmo construtivo para geração de padrões tabuleiros exatos a partir de uma combinação dada de itens:**

Um algoritmo construtivo foi desenvolvido para a geração de padrões tabuleiros exatos a partir de uma combinação dada de itens. O algoritmo pressupõe uma combinação de itens que poderiam estar no padrão, e o algoritmo proposto fornece um padrão tabuleiro exato com estes itens, caso um exista ou conclui que não existe tal padrão. O algoritmo foi implementado em linguagem C++ e os testes computacionais realizados mostraram que o algoritmo é bastante rápido tanto na geração de padrões como no descarte de padrões inviáveis com uso dos limitantes apresentados.

## **3 - Desenvolvimento de um algoritmo enumerativo para determinação de padrões tabuleiros exatos e restritos:**

Foi proposto um algoritmo enumerativo para determinação de padrões tabuleiros exatos, em que o número de itens de cada tipo a ser produzido no padrão é limitado. O algoritmo se baseia no algoritmo enumerativo de Yanasse, Soma e Maculan (2000) para a determinação das K-melhores soluções para o Problema da Mochila unidimensional.

Inicialmente, obtêm-se combinações restritas de itens para a resolução do problema de corte. Em seguida, utiliza-se o algoritmo construtivo desenvolvido neste trabalho, para verificar a possibilidade de gerar padrões tabuleiros com as combinações obtidas. O algoritmo foi implementado em C++. Testes computacionais realizados com algumas instâncias baseadas em dados de problemas reais mostraram que apesar da determinação do padrão tabuleiro ser feita de maneira eficiente, o número de combinações viáveis possíveis a serem investigadas pode ser muito grande até que se consiga assegurar a otimalidade do padrão gerado. Nestes casos, a introdução de refinamentos mostrou-se fundamental na melhoria do tempo computacional. Neste trabalho um refinamento sugerido foi a utilização de fatores de escala. Testes realizados com exemplos extraídos da literatura mostraram que o uso deste algoritmo enumerativo proposto, com a introdução deste refinamento pode resultar num ganho significativo de tempo computacional, sem perda de qualidade da solução, mas os tempos ainda podem ser elevados.

#### **4 - Desenvolvimento de um algoritmo de enumeração implícita para determinação de padrões tabuleiros exatos e restritos:**

Foi desenvolvido um algoritmo de enumeração implícita para determinação de padrões tabuleiros exatos e restritos baseado no Método de enumeração implícita de Gilmore e Gomory (1963) para a resolução do Problema da Mochila unidimensional. A vantagem do algoritmo de enumeração implícita sobre o algoritmo enumerativo apresentado anteriormente é a possibilidade de interromper sua execução em qualquer instante e recuperar a solução corrente que é viável. No algoritmo enumerativo, obtêm-se uma solução viável apenas no final de sua execução. O algoritmo de enumeração implícita proposto utiliza limitantes superiores para verificar combinações de itens que, potencialmente, irão resultar na geração de padrões viáveis. O algoritmo construtivo para geração de padrões é aplicado para confirmar ou não esta possibilidade de geração do padrão. Limitantes inferiores são utilizados para descartar ramos da árvore de enumeração que forneçam limitantes superiores menores ou iguais ao limitante inferior corrente. O algoritmo foi implementado em C++ e os testes computacionais mostraram que este algoritmo de enumeração implícita consegue ser mais rápido do que o

algoritmo enumerativo proposto anteriormente. Este algoritmo de enumeração implícita também foi mais rápido do que a resolução dos modelos matemáticos para determinação de padrões tabuleiros exatos e restritos apresentados por Yanasse e Morabito (2003) em grande parte das instâncias avaliadas. No entanto, constatou-se que este algoritmo ainda apresenta tempos computacionais elevados, se a demanda associada aos itens for elevada e os itens tiverem dimensões pequenas em relação às dimensões dos objeto.

## **5 - Desenvolvimento de um algoritmo melhorado de enumeração implícita para determinação de padrões tabuleiros exatos e restritos:**

Um refinamento foi proposto para se tentar reduzir o tempo de execução do algoritmo de enumeração implícita. O novo algoritmo conserva as mesmas idéias do Método de Enumeração Implícita de Gilmore e Gomory (1963), mas utiliza um limitante superior mais apertado do que o algoritmo de enumeração implícita proposto anteriormente. O limitante superior do novo algoritmo leva em consideração, além das áreas dos itens, suas dimensões e as dimensões das faixas geradas durante a execução. Através do novo algoritmo de enumeração implícita, foi possível reduzir significativamente o tempo computacional das instâncias testadas.

Os algoritmos para determinação de padrões tabuleiros restritos 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 computacionais 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). Em ambientes reais de produção é conveniente analisar o tipo de padrão tabuleiro utilizado para a escolha do algoritmo de melhor desempenho computacional a ser aplicado.

No caso da determinação de padrões tabuleiros irrestritos exatos e não exatos, sugere-se aplicar a nova heurística desenvolvida, pois ela consegue obter soluções num tempo computacional reduzido, se comparado ao tempo computacional do Algoritmo Exato de

Katsurayama e Yanasse (1999, 2000). Além disto, a nova heurística obtém soluções de melhor qualidade do que as soluções obtidas pela heurística de Morabito e Arenales (2000), principalmente nos casos em que existam muitos itens de dimensão comum (por exemplo, fábrica de móveis).

O Algoritmo Melhorado de Enumeração Implícita desenvolvido nesta tese é uma boa alternativa a ser empregada no caso da obtenção de soluções ótimas para o problema da determinação de padrões tabuleiros do tipo exato e restrito. No caso do problema irrestrito, o tempo de execução piora, mas nestes casos, sugere-se o uso do Algoritmo de Katsurayama e Yanasse (1999, 2000) que é mais rápido. Na prática, podemos utilizar estes dois algoritmos de maneira combinada. Inicialmente o Algoritmo de Katsurayama e Yanasse (1999, 2000) 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 Melhorado de Enumeração Implícita para a geração do padrão restrito. Observe que, com isso, para todas as instâncias testadas, esta combinação de algoritmos, resulta no melhor desempenho computacional em termos de tempo de execução.

Acredita-se que outros refinamentos ainda possam ser introduzidos para se tentar reduzir ainda mais o tempo computacional do Algoritmo Melhorado de Enumeração Implícita para determinação de padrões tabuleiros exatos e restritos. Por exemplo, é possível resolver o problema relaxado restrito da mochila para os bins acima, à direita e a nordeste do cerne. Pode-se também testar a combinação de 2 ou mais faixas.

Nesta tese de doutorado, focalizou-se o problema da determinação de padrões mais simples dentre os fatores que influenciam na produtividade em ambientes de corte. Os demais fatores que podem influenciar na produtividade em ambientes de corte também merecem ser investigados mais profundamente, em especial, o problema da redução do número de diferentes tipos de padrões de corte. Este problema é de grande importância em ambientes de produção onde os custos operacionais do equipamento e da perda de produção com a programação do equipamento são significativos em comparação com o custo do material cortado. Embora o problema da redução de padrões seja de grande

aplicação prática, a lista de trabalhos que tratam de sua resolução ainda é pequena. A título de experimentação, por exemplo, realizou-se uma linearização do modelo quadrático inteiro de Vanderbeck (2000) para o problema da redução do número de diferentes tipos de padrões de corte unidimensional, usando o procedimento apresentado por Harjunkoski *et al* (1997). O modelo linearizado foi testado com as mesmas instâncias de Vanderbeck (2000). Esta linearização teve pouco êxito. Nos testes realizados num PC Athlon 64 3200+ (2 GHz), com 1GB RAM executados com auxílio da biblioteca CPLEX versão 7, apenas 2 instâncias das 16 testadas em Vanderbeck (2000) foram resolvidas em 1 hora de execução. Este experimento nos indica que esta linearização testada parece ser pouco promissora. Talvez uma abordagem diferente deva ser pesquisada. Esta pesquisa e outras permanecem para futuros desenvolvimentos.

## REFERÊNCIAS BIBLIOGRÁFICAS

- Antonio, J.; Chu, C.; Sauer, N.; Woff, P. POEM Project and solutions for one-dimensional multi-criteria cutting stock problems. In: Symposium on Operations Research, 1995, Passau (Alemanha). **Proceedings...** Passau: University of Passau, 1996, p. 42-47.
- Arenales, M. N. Mini-curso: o problema de corte e empacotamento e aplicações industriais. In: Congresso Nacional de Matemática Aplicada e Computacional, 20., e Oficina Nacional de Problemas de Corte e Empacotamento, 2., 1997, Gramado (RS). **Anais...** São Carlos (SP): ICMC-USP, 1997. Cap. 1, p. 1-23.
- Arenales, M. N.; Morabito, R. An AND/OR-graph approach to the solution of two-dimensional non-guillotine cutting problems. **European Journal of Operational Research**. v. 84, n. 3, p. 599-617, 1995.
- Beasley, J.E. . Algorithms for unconstrained two-dimensional guillotine cutting. **Journal of the Operational Research Society** v. 36, n. 4, p. 297-306, 1985.
- Belluzzo, L. **Otimização nos planos de corte de chapas de fibra de madeira reconstituída: um estudo de caso**. São Carlos. 2002. 144 p. Dissertação (Mestrado em Engenharia de Produção). Universidade Federal de São Carlos – São Carlos (SP), 2002.
- Belov, G.; Scheithauer, G. **The number of setups (different patterns) in one-dimensional stock cutting**. Dresden University of Technology, Department of Mathematics, 2003. Preprint MATH-NM-15-2003.
- Bhadury, J.; Chandrasekaran, R. Stock cutting to minimize cutting length. **European Journal of Operational Research**, v. 88, n. 1, p. 69-87, 1996.
- Chandrasekaran, R.; Daescu, O.; Luo, J. Cutting out polygons. In: Canadian Conference on Computational Geometry, 17., 2005, Ontario (Canadá). **Proceedings...** Ontario: University of Windsor, 2005. p. 180-183.

- Chu, C.; Antonio, J. Approximation algorithms to solve real-life multicriteria cutting stock problems. **Operations Research**, v. 47, n. 4, p. 495-508, 1999.
- Christofides, N.; Whitlock, C. An algorithm for two-dimensional cutting problems. **Operations Research**, v. 25, n. 1, p. 30-44, 1977.
- Cui, Y. An exact algorithm for generating homogenous T-shape cutting patterns. **Computers and Operations Research**, v. 34, n. 7, p. 143-152, 2007.
- Cui, Y.; He, Dongli; Song, X. Generating optimal two-section cutting patterns for rectangular blanks. **Computers and Operations Research**, v. 33, n. 6, p. 1505-1520, 2006.
- Daescu, O.; Luo, J. Cutting out polygons with lines and rays. In: Annual International Symposium on Algorithms and Computation, 15., 2004, Hong Kong (China). **Proceedings...** Hong Kong: Rudolf Fleischer, Gerhard Trippen, 2004. LNCS 3341, p. 669-680. ISBN3-540-24131-0.
- Diegel, A.; Chetty, M.; Van Schalkwyck, S.; Naidoo, S. **Setup combining in the trim loss problem 3-to-2 & 2-to-1**. Durban: University of Natal, 1993. First Draft.
- Diegel, A.; Miller, G.; Montocchio, E.; Schalkwyk, S. V.; Diegel, O. Enforcing minimum run length in the cutting stock problem. **European Journal of Operational Research**, v. 171, n. 2, p. 708-721, 2006.
- Dumitrescu, A. An approximation algorithm for cutting out convex polygons. **Computational Geometry**, v. 29, n. 3, p. 223-231, 2004.
- Farley, A. A.; Richardson, K. V. Fixed charge with identical fixed charges. **European Journal of Operational Research**, v. 18, n. 2, p. 245-249, 1984.
- Fayard, D.; Zissimopoulos. An approximation algorithm for solving unconstrained two-dimensional knapsack problems. **European Journal of Operational Research**, v. 84, n. 3, p. 618-632, 1995.

- Figueiredo, A. G.; Rangel, S. Uma heurística para aplicação de padrões tabuleiros compostos no corte da matéria prima na indústria de móveis. In: Simpósio Brasileiro de Pesquisa Operacional, 37., 2005, Gramado (RS). **Anais...** Gramado: UFSM/UNISINOS/UFRGS, 2005. p. 131-131. 1 CD-ROM.
- Foerster, H.; Wäscher, G. Pattern reduction in one-dimensional cutting stock problem. **International Journal of Production Research**, v. 38, n. 7, p. 1657-1676, 2000.
- Gilmore, P.; Gomory, R. A linear programming approach to the cutting-stock problem. **Operations Research**, v. 9, n. 6, p. 849-859, 1961.
- 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.
- Gramani, M. C. N. **Problema de corte bidimensional guilhotinado e restrito em 2-estágios**. 1997. 100 p. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas de São Carlos - Universidade de São Paulo, São Carlos, 1997.
- Haessler, R. W. Controlling cutting pattern changes in one-dimensional trim problems. **Operations Research**, v. 23, n. 3, p. 483-493, 1975.
- Haessler, R. W. A procedure for solving the 1.5 dimensional coil slitting problem. **AIEE Transactions**, v. 10, n. 1, p. 70-75, 1978.
- Harjunkoski, I.; Pörn, R.; Westerlund, T.; Skifvars, H. Different strategies for solving bilinear integer non-linear programming problems with convex transformations. **Computers and Chemical Engineering**, v. 21, n. 1, p. 487-492, 1997.
- Hifi, M. Library of instances: 2D cutting test data sets. 2005. Disponível em: <<ftp://cermse.univ-paris1.fr/pub/CERMSEM/hifi/2Dcutting>>. Acesso em: 10 de set. 2005.

- Hifi, M.; Roucairol, C. Approximate and exact algorithms for constrained (un)weighted two-dimensional two-staged cutting stock problems. **Journal of Combinatorial Optimization**, v. 5, n. 4, p. 465-494, 2001.
- Hinxman, A. I. The trim-loss and assortment problems: a survey. **European Journal of Operational Research**. v. 5, n. 1, p. 8-18, 1980.
- Katsurayama, D. M. **Estudo sobre o efeito da utilização de padrões tabuleiros na produtividade do equipamento de cortes**. 2001. 121 p. (INPE-8753-TDI/796). Dissertação (Mestrado em Computação Aplicada). Instituto Nacional de Pesquisas Espaciais, São José dos Campos (SP), 2002.
- Katsurayama, D. M.; Yanasse, H. H. Um algoritmo enumerativo para determinação de padrões tabuleiros. In: Simpósio Brasileiro de Pesquisa Operacional, 31., 1999, Juiz de Fora (MG). **Anais...** Juiz de Fora: Tec Art Editora, 1999. Seção de Corte e Empacotamento. 1 CD-ROM.
- Katsurayama, D. M.; Yanasse, H. H. Um algoritmo enumerativo para determinação de padrões tabuleiros: aspectos computacionais de implementação. In: Simpósio Brasileiro de Pesquisa Operacional, 32., 2000, Viçosa (MG). **Anais...** Viçosa: Tec Art Editora, 2000. p. 370-382. 1 CD-ROM
- Katsurayama, D. M.; Yanasse, H. H. Uma análise da produtividade do equipamento de cortes utilizando-se padrões tabuleiros. In: Encontro Nacional de Engenharia de Produção, 21., 2001, Salvador (BA). **Anais...** Salvador: ABEPRO, 2001. p. 86-86. 1 CD-ROM.
- Limeira, M. S. **Redução do número de padrões em problemas de corte de estoque**. 2003. 176p. (INPE-11896-TDI/966). Tese (Doutorado em Computação Aplicada). Instituto Nacional de Pesquisas Espaciais, São José dos Campos (SP), 2003.
- Limeira, M. S.; Yanasse, H. H. Uma heurística para o problema de redução de padrões de corte. In: Worcap, 2., 2002, São José dos Campos (SP). **Anais...** São José dos Campos: INPE/CAP, 2002. On-line. Disponível na biblioteca digital *URLib*:

<<http://hermes2.dpi.inpe.br:1905/rep-/lac.inpe.br/lucio/2002/11.11.16.05>>. Acesso em: 22 jan. 2007.

- Lodi, A.; Monaci, M. Integer programming models for 2-staged two-dimensional knapsack problems. **Mathematical Programming**, v. 94, n. 2-3, p. 257-278, 2003.
- Martinez, M. **Minimization of discontinuous cost functions by smoothing**. Campinas (SP): UNICAMP, 2001.
- Morabito, R.; Arenales, M. N. Staged and constrained guillotine two-dimensional cutting problems: an AND/OR-graph approach. **European Journal of Operational Research**, v. 94, n. 3, p. 548-560, 1996.
- Morabito, R.; Garcia, V. The cutting stock problem in a hardboard industry: a case study. **Computers & Operations Research**, v. 25, n. 6, p. 469-485, 1998.
- 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.
- Morabito, R. **Formulação matemática para padrões tabuleiros não exatos**. (Instituto Nacional de Pesquisas Espaciais, São José dos Campos – SP, 2001). Comunicação pessoal.
- 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, n. 2, p. 256-266, 1990.
- OR-Library. Unconstrained guillotine cutting. 2003. Disponível em: <<http://mscmga.ms.ic.ac.uk/jeb/orlib/gcutinfo.html>>. Acesso em: 11 de jun. 2003.
- Overmars, M. H.; Welzl, E. The complexity of cutting paper. In: Proceedings of the ACM Annual Symposium on Computational Geometry, 1., 1985, Baltimore (Estados Unidos). **Proceedings...** Nova Iorque (Estados Unidos): ACM Press, 1985. p. 316-321. ISBN: 0-89791-163-6.

- Perin, C.; Rangel, S. O problema do corte bidimensional. In: Congresso Nacional de Matemática Aplicada e Computacional, 12., 1989, São José do Rio Preto (SP). **Anais...** São José do Rio Preto, 1989. Caderno de Resumos. v. 1.
- Pinto, M. J. **O problema de corte de estoque inteiro**. 1999. 98p. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas de São Carlos - Universidade de São Paulo, São Carlos (SP), 1999.
- Poldi, K. C. **Algumas extensões do problema de corte de estoque**. 2003. 141 p. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas de São Carlos – Universidade de São Paulo, São Carlos (SP), 2003.
- Salles Neto, L. L.; Moretti, A. C. Modelo não-linear para minimizar perdas e o setup num problema de corte unidimensional. In: Oficina Nacional de Problemas de Corte e Empacotamento, 7., 2003, Bauru (SP). **Anais...** Bauru: UNESP, 2003. Caderno de Resumos.
- Scheithauer, G. On a two-dimensional guilhotine cutting problem. In: IFORS Triennial Conference, 16., 2002, Eidinburgh (Escócia). **Anais...** Eidinburgh: IFORS, 2002. Cutting and Packing Workshop.
- Umetani, S.; Yagiura, M., Ibaraki, T. A local search approach for one dimensional cutting stock problem. In: Metaheuristics International Conference, 4., 2001, Porto (Portugal). **Anais...** Porto: University of Porto, 2001. p. 69-73.
- Umetani, S.; Yagiura, M., Ibaraki, T. One-dimensional cutting stock problem to minimize the number of different patterns. **European Journal of Operational Research**, v 146, n. 2, p. 388-402, 2003.
- Vanderbeck, F. Exact algorithm for minimizing the number of setups in the one-dimensional cutting stock problem. **Operations Research**, v. 48, n. 6, p. 915-926, 2000.

- Vianna, A. C.; Arenales, M.; Gramani, M. C. Two-stage and constrained two-dimensional guillotine cutting problems. In: Oficina Nacional de Problemas de Corte e Empacotamento, 6., 2002, Campinas (SP). **Anais...** Campinas: UNICAMP, 2002. Livro de resumos.
- Yanasse, H. H.; Harris, R. G.; Zinober, A. S. I. Uma heurística para redução do número de ciclos da serra no corte de chapas. In: Encontro Nacional de Engenharia de Produção, 13., 1993, Florianópolis (SC). **Anais...** Florianópolis: ABEPRO, 1993. v. 2, p. 879-885.
- 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**. São José dos Campos: INPE - UFSCAR, 2003. 22 p.
- Yanasse, H. H.; Morabito, R. **Linear models for one-group two-dimensional guillotine cutting problems**. Aceito pela revista International Journal of Production Research. Acesso em: 12 de fev. 2005.
- Wagner, B. J. A genetic algorithm solution for one-dimensional bundled stock cutting. **European Journal of Operational Research**, v. 117, n. 2, p. 368-381, 1999.
- Walker, W. E. A heuristic adjacent extreme point algorithm for the fixed charge problem. **Management Science**, v. 22, n. 5, p. 587-596, 1976.
- Wang, P. Two algorithms for constrained two-dimensional cutting stock problems. **Operations Research**, v. 31, n. 3, p. 573-586, 1983.
- Westerlund, T.; Isaksson, J.; Harjunkoski, I. Solving a two-dimensional trim-loss problem with MILP. **European Journal of Operational Research**, v. 104, n. 3, p. 572-581, 1998.





## **PUBLICAÇÕES TÉCNICO-CIENTÍFICAS EDITADAS PELO INPE**

### **Teses e Dissertações (TDI)**

Teses e Dissertações apresentadas nos Cursos de Pós-Graduação do INPE.

### **Manuais Técnicos (MAN)**

São publicações de caráter técnico que incluem normas, procedimentos, instruções e orientações.

### **Notas Técnico-Científicas (NTC)**

Incluem resultados preliminares de pesquisa, descrição de equipamentos, descrição e ou documentação de programa de computador, descrição de sistemas e experimentos, apresentação de testes, dados, atlas, e documentação de projetos de engenharia.

### **Relatórios de Pesquisa (RPQ)**

Os Relatórios de Pesquisa reportam resultados ou progressos de pesquisas tanto de natureza técnica quanto científica, cujo nível seja compatível com o de uma publicação em periódico nacional ou internacional.

### **Propostas e Relatórios de Projetos (PRP)**

São propostas de projetos técnico-científicos e relatórios de acompanhamento de projetos, atividades e convênios.

### **Publicações Didáticas (PUD)**

As Publicações Didáticas incluem apostilas, notas de aula e manuais didáticos.

### **Publicações Seriadas**

São os seriados técnico-científicos: boletins, periódicos, anuários e anais de eventos (simpósios e congressos). Constam destas publicações o Internacional Standard Serial Number (ISSN), que é um código único e definitivo para identificação de títulos de seriados.

### **Programas de Computador (PDC)**

São a seqüência de instruções ou códigos, expressos em uma linguagem de programação compilada ou interpretada, a ser executada por um computador para alcançar um determinado objetivo. São aceitos tanto programas fonte quanto executáveis.

### **Pré-publicações (PRE)**

Todos os artigos publicados em periódicos, anais e como capítulos de livros.