

MINISTÉRIO DA CIÊNCIA E TECNOLOGIA  
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

**INPE-8753-TDI/796**

**ESTUDO SOBRE O EFEITO DA UTILIZAÇÃO DE  
PADRÕES TABULEIROS NA PRODUTIVIDADE DO  
EQUIPAMENTO DE CORTES**

Daniel Massaru Katsurayama

Dissertação de Mestrado em Computação Aplicada, orientada pelo Dr. Horácio Hideki  
Yanasse, aprovada em 16 de abril de 2001.

INPE  
São José dos Campos  
2002

519.583

KATSURAYAMA, D. M.

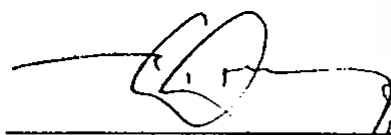
Estudo sobre o efeito da utilização de padrões tabuleiros na produtividade do equipamento de cortes / D. M. Katsurayama - São José dos Campos: INPE, 2001.

121p. – (INPE-8753-TDI/796).

1.Pesquisa operacional. 2.Programação linear. 3.Método simplex . 4.Modelo matemático. 5.Custo de operação. 6. Custo de produção. I.Título.

Aprovado pela Banca Examinadora em cumprimento a requisito exigido para a obtenção do Título de **Mestre em Computação Aplicada**.

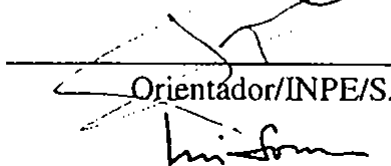
Dr. Luiz Antônio Nogueira Lorena



---

Presidente/INPE/SJCampos-SP

Dr. Horácio Hideki Yanasse



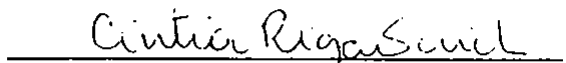
---

Orientador/INPE/SJCampos-SP

Dr. Nei Yoshihiro Soma

Membro da Banca

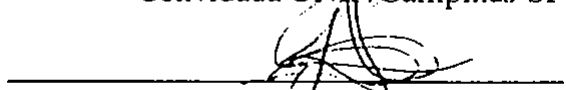
Dr<sup>a</sup> Cintia Rigão Scrich



---

Membro da Banca  
Convidada UNEP/Campinas-SP

Dr. Reinaldo Morabito Neto



---

Membro da Banca  
Convidado UFSCAR/São Carlos-SP

Candidato: Daniel Massaru Katsurayama

A meus Pais,  
**Tadayuki Katsurayama e Saca Nakamura Katsurayama,**  
em memória.

## **AGRADECIMENTOS**

À FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) pelo apoio financeiro.

Ao Instituto Nacional de Pesquisas Espaciais pelas facilidades concedidas à realização deste trabalho.

Ao Dr. Horacio Hideki Yanasse pela valiosa orientação no desenvolvimento desta dissertação.

Ao Professor Reinaldo Morabito pelo importante auxílio no fornecimento de dados e informações sobre os testes computacionais para comparação.

A todos que, direta ou indiretamente, colaboraram para o desenvolvimento desta dissertação.

## RESUMO

Padrões tabuleiro, também conhecidos como padrões 1-grupo, pertencem a uma classe especial de padrões 2-estágios guilhotinados que não necessitam de recortes. Eles podem ser produzidos girando-se a serra de 90 graus, após os cortes do primeiro estágio. As faixas obtidas no primeiro estágio são cortadas todas juntas no segundo estágio. Tais padrões demandam menos tempo de máquina e são de particular interesse em ambientes de grande demanda. Um aumento eventual de perda de material é compensado pelo ganho em produtividade da máquina. Uma formulação não linear para o problema de geração de padrões tabuleiros foi apresentada anteriormente na literatura. Também heurísticas para obtenção de tais padrões foram sugeridas. Apresenta-se aqui um método exato para se encontrar padrões tabuleiros que é baseado em um algoritmo enumerativo para obtenção das K-melhores soluções para o problema da mochila unidimensional. O uso deste algoritmo permite considerar restrições adicionais ao problema básico da mochila sem grande esforço computacional, uma faceta de particular interesse em ambientes de corte onde geralmente, os padrões de corte precisam obedecer a restrições adicionais como número limitado de itens que a máquina de corte pode manipular, restrição de manipulações de materiais, etc. Neste trabalho serão discutidos aspectos computacionais para a implementação deste algoritmo e a viabilidade de sua aplicação prática. Também foi desenvolvida uma função custo que pode ser utilizada no modelo de Gilmore e Gomory (1963, 1965), e que leva em consideração o custo associado à perda de material e ao tempo de corte da chapa. Morabito e Arenales (2000) sugeriram a utilização de uma função simples com a adição de um custo fixo no caso de padrões não tabuleiros. Testes computacionais foram realizados para comparar os resultados obtidos utilizando-se a função de custo fixo de Morabito e Arenales e a nova função proposta.

# **A STUDY ABOUT THE EFFECT OF THE UTILIZATION OF CHECKERBOARD PATTERNS ON THE PRODUCTIVITY OF THE CUTTING MACHINE**

## **ABSTRACT**

Checkerboard patterns, also known as 1-group patterns, belong to a special class of 2-stage guillotine patterns that do not need recuts. They can be produced turning the saw in 90 degrees, after the first stage cuts. The strips obtained in the first stage are all cut together in the second stage. Such patterns demand less machine time, and are of particular interest in high demand settings. An eventual increase in loss of material is compensated by the higher productivity of the machine. A nonlinear formulation of the checkerboard pattern generation problem was presented previously in the literature. Also, heuristics for determining such patterns were suggested. We present here an exact method to determine such patterns, based on an enumerative algorithm for determining the K-best solutions of the unidimensional knapsack problem. The use of this algorithm allow us to consider additional constraints, to the basic knapsack problem, without much effort. This feature is of particular interest in cutting environments where, generally, the patterns must obey additional constraints, such as, limited number of different itens that a cutting machine may handle, material handling constraints, etc. In this work, we focus on some computational aspects of the implementation of this algorithm and the feasibility of its practical use. We also developed a cost function which can be used in the model of Gilmore and Gomory (1963, 1965), that takes into account the cost associated with the waste of material and the board cutting time. Morabito and Arenales (2000) suggested the use of a simple function with the addition of a fixed cost in the case of non-checkerboard patterns. Computational tests were performed to compare the results obtained using Morabito and Arenales fixed cost function and the new proposed function.

## SUMÁRIO

	<u>Pág.</u>
<b>LISTA DE FIGURAS</b>	
<b>LISTA DE TABELAS</b>	
<b>LISTA DE SIGLAS E/OU ABREVIATURAS</b>	
<b>CAPÍTULO 1 - INTRODUÇÃO .....</b>	<b>19</b>
<b>CAPÍTULO 2 - REVISÃO BIBLIOGRÁFICA .....</b>	<b>31</b>
<b>CAPÍTULO 3 - UM ALGORITMO ENUMERATIVO PARA DETERMINAÇÃO DE PADRÕES TABULEIROS EXATOS .....</b>	<b>51</b>
<b>CAPÍTULO 4 - ASPECTOS COMPUTACIONAIS DE IMPLEMENTAÇÃO DO ALGORITMO ENUMERATIVO PARA DETERMINAÇÃO DE PADRÕES TABULEIROS EXATOS.....</b>	<b>63</b>
4.1 - Enumeração das combinações.....	63
4.2 - Combinações dominantes.....	68
4.3 - Escolha da dimensão para se fazer a enumeração.....	69
4.4 - Uso de limitantes para redução do problema da mochila na segunda fase.....	74
4.5 - Recuperação imediata ou postergada de combinações alternativas .....	78
<b>CAPÍTULO 5 - FUNÇÃO CUSTO ASSOCIADA À PERDA DE MATERIAL E AO TEMPO DE CORTE DA CHAPA .....</b>	<b>81</b>
<b>CAPÍTULO 6 - TESTES COMPUTACIONAIS .....</b>	<b>89</b>
<b>CAPÍTULO 7 - CONCLUSÕES .....</b>	<b>99</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>105</b>
<b>APÊNDICE A - TEMPO COMPUTACIONAL DO ALGORITMO ENUMERATIVO PARA DETERMINAÇÃO DE PADRÕES TABULEIROS EXATOS.....</b>	<b>109</b>



<b>APÊNDICE B - ALGORITMO ENUMERATIVO PARA DETERMINAÇÃO DE PADRÕES TABULEIROS NÃO EXATOS .....</b>	<b>111</b>
<b>APÊNDICE C - APLICAÇÃO DA FUNÇÃO CUSTO NO MÉTODO SIMPLEX COM GERAÇÃO DE COLUNAS .....</b>	<b>117</b>
<b>APÊNDICE D - RESULTADOS DOS TESTES COMPUTACIONAIS COMPARANDO O "BALANÇO" ENTRE PADRÕES TABULEIROS EXATOS E PADRÕES 2-ESTÁGIOS EXATOS .....</b>	<b>121</b>

## LISTA DE FIGURAS

	<u>Pág.</u>
1.1 Problema de corte de estoque .....	19
1.2 Um exemplo de padrão de corte. ....	20
1.3 Exemplo de padrões de corte homogêneos.....	20
1.4 Padrão de corte guilhotinado e não guilhotinado.....	25
1.5 Padrão 2-estágios exato e não exato .....	26
1.6 Padrão tabuleiro .....	27
2.1 Esquema da enumeração completa .....	35
2.2 Esquema de ramificação .....	40
2.3 Ramificação para índices maiores ou iguais a $j$ .....	40
3.1 Um exemplo de padrão tabuleiro.....	51
3.2 Faixa gerada na largura.....	55
3.3 Faixas de cortes geradas na largura .....	57
3.4 Padrão tabuleiro exato ótimo obtido para o exemplo .....	60
4.1 Combinações de cortes .....	68
4.2 Combinações dominantes variando-se o tamanho médio das peças em relação às dimensões da chapa.....	70
4.3 Distribuição de frequência acumulada dos tamanhos relativos .....	71
4.4 $\Delta w$ e $\Delta l$ na Tabela de enumeração.....	72
4.5 Faixa (em negrito) no padrão tabuleiro.....	75
5.1 Corte longitudinal .....	82
5.2 Corte transversal .....	82

5.3	Processo de corte de padrões 2-estágios exato .....	83
5.4	Custos para os cortes do padrão exemplificado na Figura 5.3.....	86
5.5	Corte de um padrão do tipo 2-estágios não exato .....	86
6.1	Comparação das soluções obtidas com o uso da função custo e com o uso do custo fixo proposto por Morabito e Arenales (2000).....	97
7.1	Padrão tabuleiro com perdas internas .....	100
7.2	Exemplos de padrões tabuleiros não exatos .....	101
7.3	Convertendo um padrão 2-estágios não exato em um padrão tabuleiro não exato.....	103
B1	Faixas para uma combinação de largura 2 e 6 .....	111
B2	Seleção dos itens presentes nas faixas para se obter um conjunto de faixas com melhor lucratividade .....	112
B3	Padrão tabuleiro não exato obtido para o conjunto de faixas ilustrado na Figura B2 .....	113
B4	Solução com o algoritmo de Morabito e Arenales (2000) (valor 1,36825) .....	114
B5	Solução com uso do algoritmo proposto (valor 1,41667) .....	114
B6	Solução com o algoritmo de Morabito e Arenales (2000) (valor 1,3474) .....	115
B7	Solução com uso do algoritmo proposto (valor 1,37435).....	115

## LISTA DE TABELAS

	<u>Pág.</u>
2.1 Atualização inicial a partir do lado direito 0.....	42
2.3 Atualização a partir do lado direito 2.....	42
2.4 Atualização a partir do lado direito 3 e 4.....	43
2.5 Tabela final de enumeração .....	43
2.6 Tabela com as 5-melhores soluções recuperadas .....	45
3.1 Dados de um exemplo.....	53
3.2 Soluções enumeradas na largura da chapa.....	54
4.1 Combinações enumeradas na largura da chapa .....	64
4.2 Combinações relevantes enumeradas na largura .....	64
4.3 Novas dimensões a serem consideradas de peças de mesmo comprimento .....	66
4.4 Combinações enumeradas na largura da chapa .....	66
4.5 Critérios para a escolha da dimensão para se fazer a enumeração .....	73
4.6 Redução no problema da mochila .....	76
4.7 Redução no problema da mochila (teste com peças de dimensões pouco variadas).....	77
6.1 Itens a serem produzidos.....	90
6.2 Análise do “balanço” de padrões tabuleiros não exatos × padrões 2-estágios não exatos e padrões tabuleiros exatos × padrões 2-estágios não exatos.....	92
6.3 Influência no “balanço” de padrões tabuleiros exatos × padrões 2-estágios exatos com variações nos valores dos parâmetros da função custo.....	94
6.4 Comparação entre a função custo e o custo fixo de Morabito e Arenales (2000).....	97
A1 Tempo computacional: algoritmo proposto × heurística de Morabito e Arenales (2000) .....	109

B1	Exemplo 1: conjunto de peças para determinação de padrões tabuleiros não exatos (chapa: $W=1850, L=3670$ ) .....	114
B2	Exemplo 2: conjunto de peças para determinação de padrões tabuleiros não exatos (chapa: $W=1850, L=3670$ ) .....	115
D1	“Balanço” entre padrões tabuleiros exatos e padrões 2-estágios exatos.....	121

## LISTA DE SIGLAS E/OU ABREVIATURAS

- FFD - Algoritmo “First-Fit Decreasing” para resolução do Problema do Bin-packing
- PD - Algoritmo de Programação Dinâmica
- PM - Problema da Mochila
- PMI - Problema da Mochila Irrestrito
- YSM - Algoritmo de Yanasse, Soma e Maculan (2000) para determinação das K-melhores soluções do problema unidimensional da mochila

## CAPÍTULO 1

### INTRODUÇÃO

*Problemas de corte* aparecem em ambientes de produção de algumas indústrias, tais como de papel, móveis, vidro, metalúrgica, plástica, têxtil, etc. Nestas indústrias depara-se frequentemente com o problema de se cortar peças grandes (objetos) em pedaços menores (itens) requeridos internamente por outros setores da empresa ou por clientes externos. É necessário planejar os cortes para minimizar os efeitos negativos, como o desperdício que onera os custos de produção.

Considere a seguinte situação prática:

“Uma indústria de aço mantém em estoque um número suficientemente grande de barras de comprimento  $L$  que devem ser cortadas em peças menores de comprimentos  $l_i$  para atender uma demanda  $d_i, i=1, \dots, n$ .”

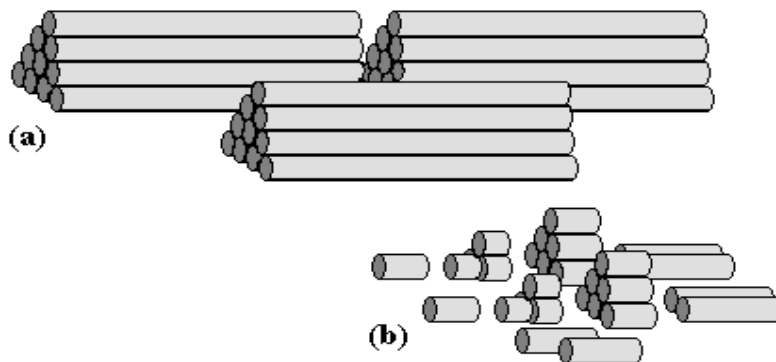


Fig. 1.1 - Problema de corte de estoque: (a) objetos em estoque, (b) itens demandados.

A situação prática descrita anteriormente ilustra um exemplo de problema de corte de estoque. As barras a serem cortadas serão genericamente referenciadas como *objetos* e as peças a serem produzidas serão genericamente referenciadas como *itens*. O problema do corte de estoque consiste basicamente em cortar os objetos para produzir os itens de

maneira a atender a demanda associada a eles. Devido ao tamanho diferenciado dos itens, os objetos podem ser cortados de diferentes maneiras.

**Definição 1.1:** Um *padrão de corte* é a maneira como os objetos em estoque serão cortados para produzir os itens menores.



Fig. 1.2 - Um exemplo de padrão de corte.

**Definição 1.2:** Um *padrão de corte homogêneo* é um padrão de corte formado por peças do mesmo tipo.

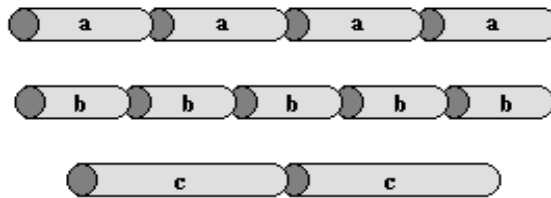


Fig. 1.3 - Exemplo de padrões de corte homogêneos.

O problema de corte de estoque é resolvido determinando-se os padrões de corte a serem utilizados de forma a otimizar uma função objetivo que pode ser, por exemplo, a maximização do lucro ou a minimização da perda de material.

No exemplo da indústria de aço, de corte de barras tem-se um problema de corte onde os objetos a serem cortados são idênticos e em quantidade ilimitada. Apenas uma dimensão é relevante para o problema de corte e os itens a serem produzidos estão especificados em quantidade limitada pela demanda. Particularmente, este problema pode ser tipicamente encontrado na construção civil. Na prática, diversos outros problemas de corte podem ser encontrados nos mais diferentes ambientes industriais. Devido à grande variedade de tipos de problemas de corte que ocorrem no mundo real,



bem como o crescente interesse de pesquisadores pelo assunto, Dyckhoff (1990) sugeriu uma forma de classificá-los. Apresenta-se a seguir as quatro principais características referenciadas no trabalho de Dyckhoff para classificar os problemas de cortes:

#### Dimensionalidade:

Característica relacionada ao número de dimensões relevantes no processo de cortagem. Por exemplo, no corte de barras na indústria de aço, apenas o comprimento é relevante para a determinação do padrão de corte; já no processo de corte de chapas de madeira na indústria de móveis, a largura e o comprimento são relevantes para a determinação do padrão associado ao objeto. Em resumo, um problema de corte em relação à sua dimensão pode ser:

- **1:** o problema é unidimensional, ou seja, quando apenas uma das dimensões é relevante no processo de cortagem. Exemplo: corte de barras de alumínio, aço, etc.
- **2:** o problema é bidimensional, ou seja, quando duas dimensões são relevantes no processo de cortagem. Exemplo: corte de chapas retangulares de madeira, chapas de aço, etc.
- **3:** o problema é tridimensional, ou seja, quando três dimensões são relevantes no processo de cortagem. Exemplo: corte de espumas para fabricação de colchões, travesseiros, etc.
- ***n*:** o problema é *n*-dimensional, ou multidimensional, ou seja, quando *n* dimensões ( $n > 3$ ) são relevantes no processo de cortagem. Exemplo: empacotamento de caixas de comida em fornos para cozimento (neste caso, o tempo de cozimento representa a quarta dimensão).

Além das classes citadas anteriormente, pode-se ainda incluir as referidas como  $n + \frac{1}{2}$  dimensional (notação introduzida por Haessler, 1978) quando  $n+1$  dimensões são relevantes no processo de cortagem e uma delas é “variável”. Por exemplo, no corte de rolos de tecidos, o objeto possui uma largura fixa, porém seu comprimento é variável.

Note que duas dimensões são relevantes no processo de corte, porém ele difere do problema bidimensional (ou do unidimensional). Neste caso, o problema é dito ser do tipo  $1\frac{1}{2}$ -dimensional.

#### Seleção de itens e objetos:

Os itens a serem produzidos são combinados respeitando-se restrições associadas aos objetos. Itens e objetos podem ser selecionados de acordo com as seguintes possibilidades de combinação:

- B: alguns itens são atribuídos a todos os objetos. Exemplos: problema da mochila (clássico), problema da mochila multidimensional, problema do carregamento de pallet, problema do bin-packing dual, problema da troca de moeda, problema do investimento financeiro em vários períodos.
- V: todos os itens são atribuídos a alguns objetos. Exemplos: problema do corte de estoque (clássico), problema do carregamento de veículos, problema do bin-packing (clássico), problema do balanceamento de uma linha de montagem, problema da alocação de tarefas, problema da alocação de memória.

#### Sortimento de objetos:

Atributo relacionado ao tipo e aparência dos objetos. O sortimento dos objetos é representado da seguinte maneira:

- O: apenas um objeto. Exemplos: problema da mochila (clássico), problema do carregamento de pallet, problema da mochila multidimensional, problema do bin-packing dual.
- I: objetos de mesmo formato, tamanho e orientação. Exemplos: problema da alocação de memória, problema do carregamento de veículos, problema do bin-packing (clássico).

- D: objetos de vários formatos, tamanhos e orientações. Exemplos: problema do bin-packing bidimensional.

Sortimento de itens:

Atributo relacionado ao tipo e aparência dos itens. Quanto ao sortimento, os itens podem ser classificados como:

- F: poucos itens de diferentes aparências (formatos, tamanhos e orientações). Exemplo: problema do carregamento de veículos.
- M: muitos itens de muitas aparências diferentes. Exemplos: problema da alocação de memória, problema do bin-packing (clássico), problema do bin packing dual.
- R: muitos itens, porém, pouca variedade de tipos de itens (itens não congruentes). Exemplos: problema da troca de moeda, problema do corte de estoque bidimensional.
- C: itens congruentes. Exemplos: problema do carregamento de pallet, problema da mochila multidimensional.

Os exemplos apresentados foram todos retirados do artigo de Dyckhoff (1990). As características apresentadas formam uma nomenclatura que permite relacionar um problema de corte e empacotamento com outros através da quádrupla: Dimensionalidade/Seleção de itens e objetos/Sortimento de objetos/Sortimento de itens. Em termos da nomenclatura sugerida por Dyckhoff, o exemplo ilustrado no início desta seção pode ser classificado como um problema do tipo 1/V/I/R, onde: 1 significa que se trata de um problema unidimensional, V significa que todos os itens devem ser produzidos para atender a demanda, I significa que se está considerando objetos do mesmo tipo, R indica que a demanda  $d_i$  pode ser grande, mas o número  $n$  de diferentes itens é relativamente pequeno. Mais tarde, Dyckhoff e Finke (1992) apresentaram um estudo mais detalhado desta tipologia classificando uma extensa bibliografia de problemas de corte e empacotamento.

A importância econômica aliada à dificuldade de resolução de problemas de corte (ou empacotamento, se visto sob uma outra ótica, vide Arenales, 1997) tem motivado a comunidade acadêmica a procurar métodos de solução eficientes. No Brasil, vários estudos vêm sendo realizados sobre os problemas de corte e empacotamento (vide <http://www.lac.inpe.br/po/projects/pce>).

O objetivo de minimizar as perdas é um dos mais frequentes em ambientes de produção de cortes. Um outro objetivo normalmente utilizado é o de minimizar o custo de produção. Nem sempre a solução de menor desperdício implica em uma solução mais econômica para a empresa. Na indústria de móveis, por exemplo, pode-se ter chapas retangulares de madeira (tipo aglomerado) que são adquiridas de fornecedores em tamanhos padronizados para serem cortadas em peças menores que irão compor partes de mesas, cadeiras, armários, etc. As chapas podem ser cortadas segundo padrões de corte com a menor perda de material. Apesar da solução de mínimo desperdício ser desejável, elas quase sempre estão associadas 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. Neste caso, 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.

Uma solução que pode ser aplicada por estas empresas para contornar o problema consiste em utilizar padrões de corte denominados tabuleiros, mais fáceis de serem produzidos pela máquina e que permitem aumentar a produtividade do equipamento de corte.

*Padrões tabuleiros* estão intrinsecamente relacionados a alguns problemas de corte bidimensionais. Apresenta-se a seguir, algumas definições pertinentes para a apresentação deste padrão.

**Definição 1.3:** Um corte *guilhotinado* é um corte reto que vai de um lado a outro do objeto, dividindo-o em 2 partes.

**Definição 1.4:** Denomina-se *padrão guilhotinado* o padrão obtido por sucessivos cortes guilhotinados.

**Definição 1.5:** Padrões guilhotinados *ortogonais* de um objeto retangular são aqueles obtidos apenas com cortes guilhotinados paralelos a um dos lados do retângulo. Caso contrário, eles são classificados como *não ortogonais* (vide Figura 1.4).

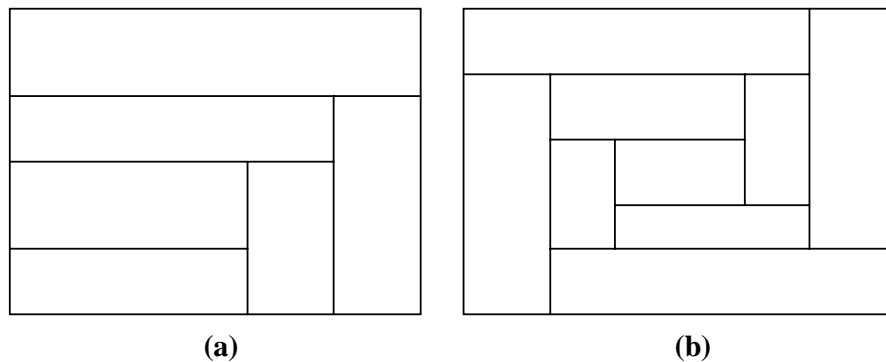


Fig. 1.4 - Padrão de corte guilhotinado (a) e não guilhotinado (b).

Os padrões de corte são definidos com cortes em vários estágios. Cada estágio indica, normalmente, uma mudança na direção do corte.

**Definição 1.6:** 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.

**Definição 1.7:** Um padrão guilhotinado ortogonal 2-estágios é dito ser *exato* se todos os itens são gerados após o segundo estágio de corte. Caso haja a necessidade de um recorte para a eliminação de refiles e obtenção dos itens ele é denominado *não exato* (vide Figura 1.5).

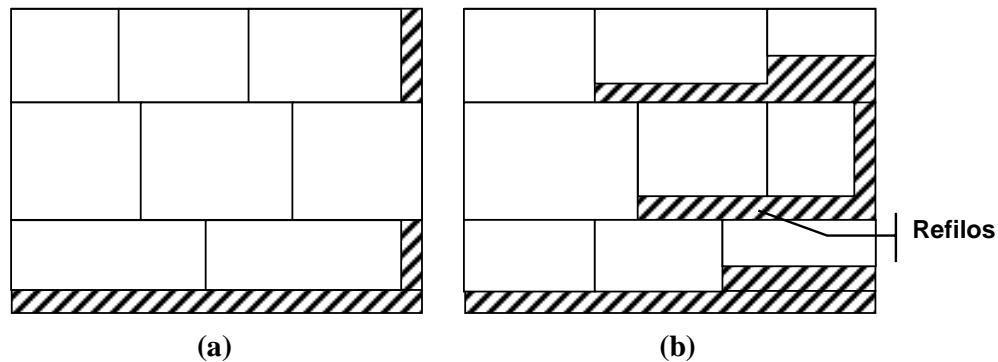


Fig. 1.5 - Padrão 2-estágios exato (a) e não exato (b).

Na Figura 1.5 e nas demais que se seguem, as áreas hachuradas nas Figuras representam refilos ou material desperdiçado.

Rigorosamente falando, o padrão 2-estágios não exato é na realidade 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).

*Padrões tabuleiros*, também conhecidos como *Padrões 1-grupo* (Gilmore e Gomory, 1965) pertencem a uma classe especial de padrões 2-estágios exato, e podem ser produzidos sem a necessidade de cortar separadamente cada uma das faixas obtidas no primeiro estágio de modo que apenas a serra é girada em 90 graus após o corte do primeiro estágio (vide Figura 1.6).

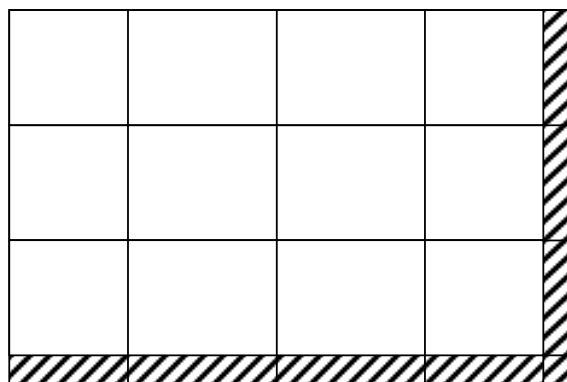


Fig. 1.6 - Padrão tabuleiro.

Tal padrão acelera a produtividade da serra, e é particularmente interessante em ambientes de grande demanda, onde o principal gargalo de produção da indústria está na máquina de corte.

Embora a perda de material em padrões tabuleiros possa aumentar, o ganho em produtividade da máquina pode compensar esta perda.

Existe, portanto, o interesse em avaliar se devemos cortar padrões que minimizem o desperdício de material, mas que reduzem a produtividade do equipamento ou, cortar padrões tabuleiros que melhoram a produtividade da máquina, mas que podem resultar em maiores desperdícios de material.

Neste trabalho focaliza-se o padrão tabuleiro, sua geração e sua influência na produtividade do equipamento de corte de chapas. Pretende-se analisar o “balanço” entre o custo da perda de material e a produtividade do equipamento de corte utilizando tais padrões.

Apesar de haverem artigos desde a década de 60 propondo modelos para o problema de corte em diversas indústrias, tem-se conhecimento de poucos trabalhos que tratam especificamente de padrões tabuleiros. A maioria deles apenas os citam sem entrar em estudos mais detalhados.

Recentemente Morabito e Arenales (2000) formularam o problema de geração de padrões de corte tabuleiros como um problema quadrático inteiro e propuseram um procedimento heurístico para a sua solução. O estudo realizado neste trabalho é similar ao desenvolvido por Morabito e Arenales (2000). Ao contrário do realizado por esses autores, neste trabalho encontra-se uma solução exata para o padrão tipo tabuleiro; para isso, adapta-se o algoritmo para obtenção das K-melhores soluções para o problema da mochila unidimensional sugerido por Yanasse, Soma e Maculan (2000). O algoritmo para determinação de padrões tabuleiros é composto de dois estágios, primeiro,

buscando uma melhor combinação nas faixas e, segundo, buscando uma boa combinação de faixas. Busca-se as K-Melhores soluções ( $K > 1$ ) para o Problema da Mochila, ao invés de se determinar apenas uma única solução ótima, pois nem sempre as melhores combinações nas faixas nos dão um conjunto de faixas que combinadas proporcionam a melhor solução para um padrão tabuleiro.

A opção pela escolha do método proposto por Yanasse, Soma e Maculan é devido ao fato de que a enumeração proposta permite considerar restrições adicionais ao problema básico da mochila sem grande esforço computacional adicional. Isto porque no processo de recuperação de soluções pode-se verificar se elas atendem às restrições impostas pelo problema, ou cuidar para que apenas as soluções que atendam à restrição sejam recuperadas. Esta faceta é particularmente interessante em ambientes de corte, pois muitas vezes os padrões de corte são restritivos, por exemplo, com o número limitado de itens diferentes no padrão devido às limitações da máquina de corte, problemas de manipulação de materiais, etc.

Embora a heurística sugerida por Morabito e Arenales aparenta ser eficiente na maioria dos casos encontrados na indústria de móveis, acredita-se que um método exato, para determinação de padrões tabuleiros, com bom desempenho computacional, tem atrativos ainda maiores para a própria indústria ou para outros tipos de ambientes de produção, pois pode tornar o processo ainda mais lucrativo.

Neste trabalho desenvolve-se também uma função custo que leva em consideração o tempo de corte da chapa expressando mais adequadamente o custo associado aos diferentes tipos de padrões cortados. Esta função é mais geral que a utilizada em Morabito e Arenales (2000) que sugerem a utilização de uma função simples com a adição de um custo fixo no caso de padrões não tabuleiro.

Experimentos computacionais foram realizados com esta nova função custo. Foram também realizadas implementações e testes computacionais do novo algoritmo proposto



com o objetivo principal de observar o comportamento das soluções em função das diferenças relativas de custo de material e de máquina.

O texto da presente dissertação está organizado nos seguintes Capítulos:

- Capítulo 1: introdução;
- Capítulo 2: apresenta-se os trabalhos encontrados na literatura que tratam do tema da presente dissertação ou que estão relacionados com os objetivos propostos no trabalho;
- Capítulo 3: apresenta-se o algoritmo desenvolvido para a determinação de padrões tabuleiros;
- Capítulo 4: apresenta-se alguns aspectos computacionais relativos à implementação deste algoritmo e melhorias introduzidas;
- Capítulo 5: apresenta-se a função custo desenvolvida que leva em consideração o custo de corte do padrão;
- Capítulo 6: apresenta-se os resultados computacionais obtidos com o novo algoritmo proposto comparando-os com os resultados apresentados por Morabito e Arenales (2000);
- Capítulo 7: são apresentadas as conclusões finais sobre os resultados obtidos;

e Apêndices:

- Apêndice A: apresenta os tempos computacionais obtidos pelo algoritmo enumerativo para determinação de padrões tabuleiros exatos apresentado no Capítulo 3.
- Apêndice B: apresenta uma adaptação do algoritmo para determinação de padrões tabuleiros para o caso não exato e alguns exemplos numéricos resolvidos pelo algoritmo adaptado.
- Apêndice C: apresenta em maiores detalhes a aplicação da função custo discutida no Capítulo 5 no método Simplex com geração de colunas.

- Apêndice D: complementa a análise do “balanço” entre padrões tabuleiros e padrões não tabuleiros discutida no Capítulo 6 apresentando uma análise do “balanço” entre padrões tabuleiros exatos e padrões 2-estágios exatos.

## CAPÍTULO 2

### REVISÃO BIBLIOGRÁFICA

Neste Capítulo apresenta-se uma breve revisão dos trabalhos encontrados na literatura relacionados com o tema desta dissertação. Para revisões gerais sobre problemas de corte e empacotamento recomenda-se, por exemplo, os trabalhos de Hinxman (1980), Dowsland e Dowsland (1992), Dyckhoff e Finke (1992), Sweeney e Pasternoster (1992), Arenales (1997), Soma *et al.* (1997).

Os trabalhos de Gilmore e Gomory (1961, 1963, 1965) causaram grande repercussão na década de 60 e são freqüentemente referenciados na literatura devido à introdução de algumas das abordagens mais clássicas para a resolução de problemas de corte de estoque. No trabalho de 1961, Gilmore e Gomory formularam o problema do corte de estoque unidimensional como um problema de programação linear inteira. A proposta de Gilmore e Gomory para a resolução do problema consiste em relaxar as condições de integralidade do problema original e resolvê-lo utilizando o método Simplex com um procedimento de geração de colunas, onde cada coluna representa um padrão de corte. Para ilustrar o método proposto por Gilmore e Gomory, considere o seguinte problema de corte de estoque unidimensional:

“Uma indústria de aço possui em estoque  $N$  barras de comprimento  $L$  e custo unitário  $C$ . As barras devem ser cortadas para produzir peças menores considerando-se  $n$  diferentes itens de comprimento  $l_i$  e demanda  $d_i$  ( $i=1, \dots, n$ ). Sabendo-se que o número de barras em estoque é suficientemente grande para atender toda a demanda, a empresa deseja encontrar uma solução que minimize o custo de produção.”

O problema de corte de estoque apresentado pode ser matematicamente formulado considerando-se as  $m$  possíveis colunas  $A_1, A_2, A_3, \dots, A_m$ , correspondentes aos padrões de corte, onde:

$$A_1 = \begin{pmatrix} a_{11} \\ a_{21} \\ a_{31} \\ \vdots \\ a_{n1} \end{pmatrix}, A_2 = \begin{pmatrix} a_{12} \\ a_{22} \\ a_{32} \\ \vdots \\ a_{n2} \end{pmatrix}, A_3 = \begin{pmatrix} a_{13} \\ a_{23} \\ a_{33} \\ \vdots \\ a_{n3} \end{pmatrix}, \dots, A_m = \begin{pmatrix} a_{1m} \\ a_{2m} \\ a_{3m} \\ \vdots \\ a_{nm} \end{pmatrix} \quad (2.1)$$

e  $a_{ij}$  é a quantidade de itens de comprimento  $l_i$  existentes no padrão  $j$ . Portanto, os  $a_{ij}$ 's devem respeitar à seguinte restrição:

$$l_1 a_{1j} + l_2 a_{2j} + l_3 a_{3j} + \dots + l_n a_{nj} \leq L, \quad (2.2)$$

$j=1, \dots, m$  e  $a_{1j}, a_{2j}, \dots, a_{nj}$  inteiros não negativos.

Com isto o problema pode ser modelado como um problema de programação linear inteira da seguinte forma:

$$\text{Minimizar } z = Cx_1 + Cx_2 + Cx_3 + \dots + Cx_m \quad (2.3.1)$$

$$\text{Sujeito a: } A_1 x_1 + A_2 x_2 + A_3 x_3 + \dots + A_m x_m \geq d \quad (2.3.2)$$

$$x_i \geq 0 \text{ e inteiro } (i=1, \dots, m), \quad d = (d_1, \dots, d_n)^T \quad (2.3.3)$$

onde  $x_i$  é a quantidade de vezes que o padrão  $A_i$  deve ser cortado.

A função (2.3.1) é o custo de produção, a restrição (2.3.2) impõe que o total de itens cortados de cada tipo deve, pelo menos superar a demanda das barras, a restrição (2.3.3) impõe a não negatividade e integralidade do número de vezes que cada padrão deverá ser cortado.

Este problema de programação inteira é computacionalmente complexo para ser resolvido uma vez que existem muitas variáveis inteiras, pois o número de padrões  $A_i$  é muito grande em casos práticos. Uma possível alternativa para simplificar a sua

resolução é relaxar a condição de integralidade do problema. Para problemas com grande demanda esta relaxação da restrição de integralidade das variáveis pode não comprometer muito a qualidade da solução encontrada para o problema.

Observe que o problema relaxado (2.3.1)-(2.3.3) apresenta dificuldades ao ser resolvido pelo método Simplex, pois o número de possíveis padrões em casos práticos pode crescer muito, atingindo valores extremamente elevados. Gilmore e Gomory resolveram este problema introduzindo um procedimento de geração de colunas ao método Simplex. O método consiste basicamente em gerar as colunas do problema (padrões de corte) à medida do necessário ao invés de gerá-las todas a priori.

Como solução básica inicial foram escolhidos apenas padrões de corte unidimensionais homogêneos. No caso unidimensional, a base inicial composta de  $m$  padrões de corte será:

$$B = \begin{pmatrix} \left\lfloor \frac{L}{l_1} \right\rfloor & 0 & \dots & 0 \\ 0 & \left\lfloor \frac{L}{l_2} \right\rfloor & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \left\lfloor \frac{L}{l_m} \right\rfloor \end{pmatrix}$$

No caso bidimensional, ter-se-ia:

$$B = \begin{pmatrix} \left\lfloor \frac{L}{l_1} \right\rfloor \times \left\lfloor \frac{W}{w_1} \right\rfloor & 0 & \dots & 0 \\ 0 & \left\lfloor \frac{L}{l_2} \right\rfloor \times \left\lfloor \frac{W}{w_2} \right\rfloor & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \left\lfloor \frac{L}{l_m} \right\rfloor \times \left\lfloor \frac{W}{w_m} \right\rfloor \end{pmatrix}$$

Seja  $\pi$ , o vetor multiplicador Simplex (ou vetor das variáveis duais). Para a escolha da variável a entrar na base deve-se encontrar a coluna  $A$  que maximiza  $\{\pi A - C\}$ , onde  $A$  é um padrão de corte viável. Seja  $A_j$  esta coluna e ela entra na base se  $\pi A_j - C > 0$ .

Considerando que  $C$  é constante, o problema se reduz a encontrar a coluna  $A$  que maximiza  $\{\pi A\}$ , onde  $A$  é um padrão de corte viável.

No caso unidimensional este problema é o seguinte problema unidimensional da mochila:

$$\text{Maximizar } \sum_{i=1}^n \pi_i a_i \quad (2.4.1)$$

$$\text{Sujeito a: } \sum_{i=1}^n l_i a_i \leq L, \quad (2.4.2)$$

$$a_i \text{ inteiro e } a_i \geq 0, i=1, \dots, m. \quad (2.4.3)$$

Em problemas bidimensionais, a resolução prática do problema de corte torna-se mais difícil e o padrão de corte pode ser gerado através de algoritmos para determinação de padrões bidimensionais (vide Gramani, 1997; Morabito e Garcia, 1998; Pinto, 1999).

Observe que o Problema da Mochila a ser resolvido em (2.4.1)-(2.4.3) não apresenta limitação explícita no número de itens (peças) a serem colocados na mochila. Este tipo de problema é conhecido na literatura como Problema da Mochila Unidimensional Irrestrito (aqui referenciado por PMI).

Uma vez que o PMI tenha sido resolvido, se  $\pi A_T - C = \underset{j}{\text{Máximo}}\{\pi A_j - C\} \leq 0$  então a base corrente é ótima. Caso contrário, a coluna  $T$  entra na base e a coluna a sair da base segue os passos convencionais do método Simplex.

Métodos de enumeração implícita, “branch-and-bound” e programação dinâmica são as técnicas mais comuns para a resolução do PMI. O PMI pode ser resolvido através de um método de enumeração completa, ou seja, enumerando todas as possíveis soluções resultantes da combinação das variáveis e selecionando a melhor dentre elas. Considera-se a mochila inicialmente vazia com espaço disponível igual a  $L$ . As combinações das variáveis podem ser realizadas por um procedimento recursivo que consiste em inserir cada um dos diferentes tipos de itens na mochila e atualizar o espaço restante da inserção.

Num primeiro estágio, inclui-se o item tipo  $i$  ( $i=1,\dots,n$ ) obtendo-se o espaço restante  $L-l_i$ . No segundo estágio, o processo é repetido para o item tipo  $j$  ( $j=1,\dots,n$ ) sobre o espaço restante obtido no primeiro estágio obtendo-se  $L-l_i-l_j$ . O procedimento continua a ser repetido nos estágios seguintes, similarmente ao primeiro e segundo estágios até que o espaço restante na mochila não seja suficiente para acomodar nenhum tipo de peça. O algoritmo termina quando todas as possibilidades de inclusão tiverem sido exploradas. Este processo pode ser melhor visualizado através de uma árvore de enumeração (vide Figura 2.1), onde cada nó representa o espaço disponível na mochila resultante da combinação empregada.

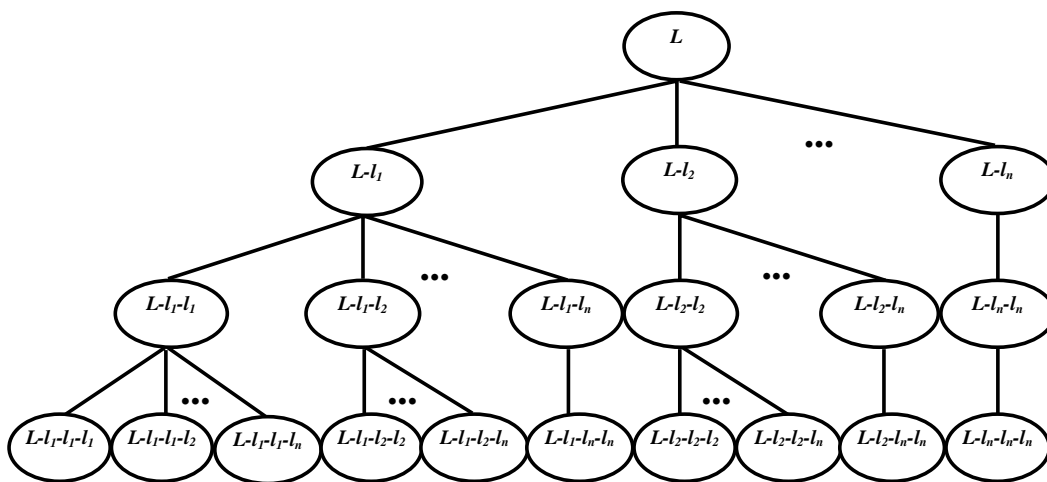


Fig. 2.1 - Esquema da enumeração completa.

Apesar do método da enumeração completa garantir a otimalidade (pois não perde nenhuma das soluções), possui a desvantagem de enumerar um grande número de soluções intermediárias inviabilizando a resolução de muitos problemas práticos devido ao tempo computacional para processamento. Gilmore e Gomory (1961) propuseram a resolução do PMI combinando a utilização de um método guloso (ad-hoc) com um algoritmo de Programação Dinâmica (Bellman, 1957).

O método guloso resolvia o PMI encontrando a melhor solução baseada na inserção consecutiva dos itens de melhor relação  $\pi_i/l_i$  (em ordem não crescente) na mochila. Este método não garantia a otimalidade da solução, porém, servia para acelerar a geração dos padrões para que a Programação Dinâmica (aqui referenciada por PD) fosse aplicada e o problema fosse resolvido otimamente. A PD consiste em encontrar a solução ótima utilizando uma formulação recorrente que implementa uma estratégia de busca em largura para percorrer a árvore de enumeração. Seja  $F_k(\hat{l})$ , o valor máximo da função objetivo para os  $k$  primeiros itens ( $k=1,\dots,n$ ) limitados por uma capacidade  $\hat{l}$  da mochila e seja  $F_0(\hat{l})=0$  ( $\hat{l}=1,\dots,L$ ). Apresenta-se a seguir, a formulação recorrente utilizada por Gilmore e Gomory (1961) para resolver o PMI:

$$F_{k+1}(\hat{l}) = \max_r \{r\pi_{k+1} + F_k(\hat{l} - rl_{k+1})\},$$

onde  $r$  é um inteiro positivo,  $0 \leq r \leq \left\lfloor \frac{\hat{l}}{l_{k+1}} \right\rfloor$ .

Note que todos os valores do domínio da função devem estar resolvidos para que a formulação recorrente seja aplicada. Entretanto, pode-se considerar somente os valores que representam uma combinação linear dos comprimentos  $l_j$  tal que  $\hat{l} \leq L$ , ou seja, ao invés de considerarmos o problema com  $\hat{l} = 1,\dots,L$ , considera-se  $\hat{l} = l_1, l_2, \dots, l_n, l_1+l_1, l_1+l_2, \dots, l_1+l_n, l_1+l_1+l_1, l_1+l_1+l_2, \dots$



Embora a PD tenha sido aplicada com sucesso no trabalho de 1961, Gilmore e Gomory constataram mais tarde, em 1963, que o método não era adequado quando aplicado a uma indústria de papel estudada pelos autores. Os tamanhos dos problemas gerados pela indústria poderiam resultar em tempos computacionais demasiadamente altos e, para contornar esta nova dificuldade Gilmore e Gomory sugeriram um método de enumeração implícita. O método proposto então se baseava em um esquema de busca em profundidade primeiro e aplicava o conceito de limitantes superiores e inferiores para que o processo de enumeração fosse realizado de maneira implícita, sem a necessidade de se percorrer todas as soluções do problema.

Limitantes superiores e inferiores permitem identificar as combinações (buscas) que devem ser evitadas e quais podem gerar uma solução ótima potencial. Para aplicar os limitantes, admite-se, sem perda de generalidade que  $\frac{\pi_1}{l_1} \geq \frac{\pi_2}{l_2} \geq \dots \geq \frac{\pi_n}{l_n}$ . Como limitante inferior pode-se utilizar a melhor solução encontrada até o momento. Soluções inferiores a este limitante são automaticamente descartadas. O limitante superior é calculado tomando-se como base o valor potencial da melhor solução que pode ser recuperada a partir do nó corrente na árvore de enumeração. Para melhor ilustrar o uso destes limitantes, descreve-se a seguir os passos do método da enumeração implícita sugerido por Gilmore e Gomory:

Passo 1: Ordene os itens de maneira que  $\frac{\pi_1}{l_1} \geq \frac{\pi_2}{l_2} \geq \dots \geq \frac{\pi_n}{l_n}$  e considere inicialmente o limitante inferior  $\underline{F} = 0$ .

Passo 2: Obtenha uma solução inicial da forma:

$$\alpha_1 = \left\lfloor \frac{L}{l_1} \right\rfloor, \alpha_2 = \left\lfloor \frac{(L - l_1 \alpha_1)}{l_2} \right\rfloor, \dots, \alpha_n = \left\lfloor \frac{L - \sum_{i=1}^{n-1} l_i \alpha_i}{l_n} \right\rfloor.$$

$\alpha = (\alpha_1, \dots, \alpha_n)$  é uma solução inicial factível para o problema.

Passo 3: Avalie a solução corrente e armazene-a se ela for a mais valiosa:

$$\text{Seja } f(\alpha) = \sum_{i=1}^n \pi_i \alpha_i :$$

Se  $\underline{F} < f(\alpha)$  então faça:  $\underline{F} = f(\alpha)$  e guarde a solução correspondente.

Passo 4: Verifique se a otimalidade foi atingida, ou seja :

Determine o maior índice  $k$ , tal que  $\alpha_k \neq 0$ .

Se  $k$  não existe ( $\alpha = (0, \dots, 0)$ ) então PARE, a melhor solução guardada é uma solução ótima.

Senão, calcule o limitante superior:

$$\bar{F} = \pi_1 \alpha_1 + \pi_2 \alpha_2 + \dots + \pi_k (\alpha_k - 1) + \frac{\pi_{k+1}}{l_{k+1}} (L - l_1 \alpha_1 - l_2 \alpha_2 - \dots - l_k (\alpha_k - 1)),$$

onde ( $\pi_{n+1} = 0$  e  $l_{n+1} = 1$ ) se  $k=n$ .

Passo 5: Backtracking:

Se  $\bar{F} \leq \underline{F}$  faça  $\alpha_k = 0$  e volte ao Passo 4.

Senão, se  $\bar{F} > \underline{F}$  então faça  $\alpha_k = \alpha_k - 1$  e defina a nova solução  $\alpha$  :

$$\alpha_{j+1} = \left\lfloor \frac{L - \sum_{i=1}^n l_i \alpha_i}{l_{j+1}} \right\rfloor, \quad j = k, \dots, n-1.$$

Volte ao Passo 3.

Além da programação dinâmica e do método da enumeração implícita de Gilmore e Gomory (1963) apresentados anteriormente, diversos outros algoritmos para a resolução do PMI podem ser encontrados na literatura (Gilmore e Gomory, 1966; Salkin e Kluyver, 1975). Martello e Toth (1990b) aprimoraram o método da enumeração

implícita sugerindo o uso de um limitante superior mais efetivo e de critérios especiais para reduzir os movimentos de retorno improdutivo na árvore de enumeração e eliminar muitos dos nós que são desnecessariamente gerados no nível inferior da árvore de enumeração. Outros métodos para resolução do PMI se propõem a resolver problemas que envolvem um grande número de variáveis (Martello e Toth, 1990a; Andonov *et al.*, 2000) e não serão abordados por não estarem no escopo deste trabalho.

Os algoritmos apresentados anteriormente obtêm apenas 1 solução ótima para o PMI. Yanasse, Soma e Maculan (2000) desenvolveram um algoritmo exato que utiliza um esquema enumerativo para determinar as K-melhores soluções ( $K > 1$ ) do problema da mochila unidimensional ao invés de encontrar apenas a solução ótima. Segundo os autores, o interesse em se encontrar as K-melhores soluções do problema da mochila pode surgir, por exemplo, quando existem restrições ao problema básico da mochila que são difíceis de considerar explicitamente no modelo matemático e quando considerados, aumentam a complexidade e o tamanho do problema.

Encontrando-se a melhor, a segunda melhor,..., a K-ésima melhor solução, pode-se verificar seqüencialmente, as soluções que melhor atendam às restrições impostas pelo problema. Tal situação pode ser tipicamente encontrada em problemas de corte de estoque, onde normalmente as operações de corte devem obedecer a uma série de restrições devido às limitações da máquina de corte, problemas de manipulação de materiais, espalhamento das ordens de serviços, etc...

No algoritmo admite-se, sem perda de generalidade, que os dados do problema são ordenados de maneira que  $l_1 \leq l_2 \leq l_3 \leq \dots \leq l_n$ . O método proposto por Yanasse, Soma e Maculan (2000) pode ser melhor visualizado através de uma árvore enumerativa onde cada nó da árvore corresponde a uma possível combinação das variáveis e onde as ramificações dos nós é feita adicionando-se +1 a cada uma das variáveis de decisão  $(a_1, a_2, a_3, \dots, a_n)$ , resultando em uma outra combinação. A ilustração deste método é

apresentado na Figura 2.2 e Figura 2.3 (reproduzidas do trabalho de Yanasse, Soma e Maculan (2000)).

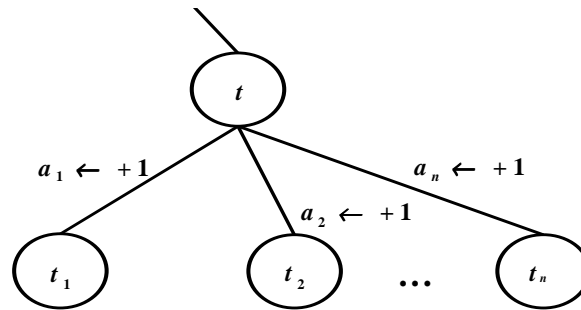


Fig. 2.2 - Esquema de ramificação.  
 FONTE: Yanasse, Soma e Maculan (2000).

Observe que ao se adicionar em seqüência  $a_j \leftarrow +1, a_k \leftarrow +1, a_q \leftarrow +1$ , atinge-se a mesma solução que  $a_j \leftarrow +1, a_q \leftarrow +1, a_k \leftarrow +1$ . Pode-se evitar estas soluções duplicadas, aplicando-se a enumeração a partir das variáveis de índices maiores ou iguais ao índice do nó da ramificação de chegada conforme pode ser observado na Figura 2.3:

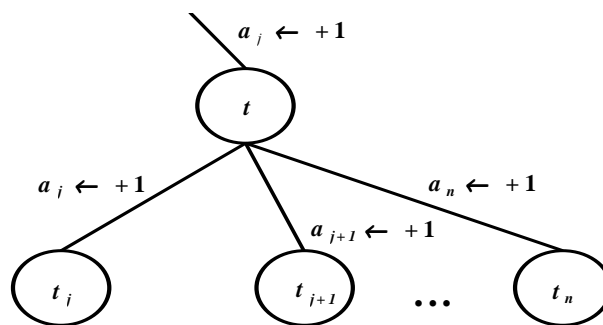


Fig. 2.3 - Ramificação para índices maiores ou iguais a  $j$ .  
 FONTE: Yanasse, Soma e Maculan (2000).

O procedimento sugerido por Yanasse, Soma e Maculan (2000), pode ser implementado computacionalmente utilizando-se uma Tabela que contém uma construção recursiva de soluções ótimas para  $KP^j$ , o problema da mochila com lado direito igual a  $j$  ( $j=0, \dots, L$ ).

O processo de enumeração é realizado combinando-se os lucros associados aos itens e distribuindo-se os valores resultantes das combinações ao longo das colunas da Tabela conforme a utilização dos itens que compõem a solução.

A recuperação das soluções enumeradas é realizada percorrendo-se as colunas da Tabela correspondentes aos itens que foram combinados e computando-se os valores das variáveis de decisão.

Para ilustrar o funcionamento deste algoritmo, considere o seguinte exemplo numérico:

$$\text{Maximizar } z = 3a_1 + 2a_2 + 4a_3$$

$$\text{Sujeito a: } 2a_1 + 3a_2 + 4a_3 \leq 8$$

$$a_i \geq 0 \text{ e inteiro, } i = 1, 2, 3.$$

O algoritmo inicia o processo de enumeração combinando os itens referentes à solução ótima do problema da mochila com o lado direito igual a 0 (espaço ocupado 0). Estas combinações são resultantes da adição entre o valor corrente do lado direito (0) com cada um dos possíveis tamanhos ( $l_1=2, l_2=3, l_3=4$ ) associados aos itens e, na adição entre o lucro correspondente ao lado direito (inicialmente 0) e os lucros ( $\pi_1=3, \pi_2=2, \pi_3=4$ ). Desta forma, obtêm-se as seguintes combinações  $(0+2)=2, (0+3)=3, (0+4)=4$ , sendo que os valores correspondentes da função objetivo que produzem o lucro mais alto possível são iguais a  $(0+3)=3, (\text{Max}\{0,0\}+2)=2, (\text{Max}\{0,0,0\}+4)=4$ . Na Tabela 2.1 ilustra-se esta situação:

TABELA 2.1 – ATUALIZAÇÃO INICIAL A PARTIR DO LADO DIREITO 0

$$l_1 = 2, \pi_1 = 3 \quad l_2 = 3, \pi_2 = 2 \quad l_3 = 4, \pi_3 = 4$$

$a_1$	$a_2$	$a_3$	Lado direito
0	0	0	0
			1
3			2
	2		3
		4	4
			5
			6
			7
			8

As próximas combinações consideram as soluções referentes ao lado direito do problema da mochila igual a 2. Os itens combinados são  $(2+2)=4$ ,  $(2+3)=5$ ,  $(2+4)=6$  e os valores correspondentes da função objetivo são dados por  $(3+3)=6$ ,  $(\text{Max}\{3,0\}+2)=5$ ,  $(\text{Max}\{3,0,0\}+4)=7$ .

TABELA 2.2 – ATUALIZAÇÃO A PARTIR LADO DIREITO 2

$$l_1 = 2, \pi_1 = 3 \quad l_2 = 3, \pi_2 = 2 \quad l_3 = 4, \pi_3 = 4$$

$a_1$	$a_2$	$a_3$	Lado direito
0	0	0	0
			1
3			2
	2		3
6		4	4
	5		5
		7	6
			7
			8

Prossegue-se considerando o lado direito igual a 3, obtendo  $(3+3)=6$ ,  $(3+4)=7$ , com valor correspondente da função objetivo  $(2+2)=4$ ,  $(\text{Max}\{2,0\}+4)=6$ . Para o lado direito igual a 4, obtêm-se  $(4+2)=6$ ,  $(4+3)=7$ ,  $(4+4)=8$ , com os respectivos valores da função objetivo  $(6+3)=9$ ,  $(\text{Max}\{6,0\}+2)=8$ ,  $(\text{Max}\{6,4\}+4)=10$ .

TABELA 2.3 – ATUALIZAÇÃO A PARTIR DO LADO DIREITO 3 E 4

$$l_1 = 2, \pi_1 = 3$$

$$l_2 = 3, \pi_2 = 2$$

$$l_3 = 4, \pi_3 = 4$$

$a_1$	$a_2$	$a_3$	Lado direito
0	0	0	0
			1
3			2
	2		3
6		4	4
	5		5
9	4	7	6
	8	6	7
		10	8

Repetindo o esquema de enumeração para o lado direito igual a 5, 6, 7 e 8, obtêm-se, ao final de enumeração, a Tabela 2.4:

TABELA 2.4 – TABELA FINAL DE ENUMERAÇÃO

$$l_1 = 2, \pi_1 = 3$$

$$l_2 = 3, \pi_2 = 2$$

$$l_3 = 4, \pi_3 = 4$$

$a_1$	$a_2$	$a_3$	Lado direito
0	0	0	0
			1
3			2
	2		3
6		4	4
	5		5
9	4	7	6
	8	6	7
12	7	10	8

Para recuperar as melhores soluções, define-se uma lista inicial contendo as K-soluções de maior valor extraídas da Tabela 2.4 em ordem não crescente. No exemplo, admitindo-se que se deseja encontrar as 5-melhores soluções (K=5), a lista inicial definida é 12, 10, 9, 8, 7, 7, onde o limitante inferior para o problema corresponde à última solução da lista (de valor 7).

As soluções são recuperadas analisando-se as combinações correspondentes à cada uma das soluções presentes na lista, da melhor à K-ésima melhor solução. Para cada solução, retrocede-se nas linhas e colunas da Tabela a fim de se recuperar a combinação resultante do processo de enumeração. Isto é feito computando-se +1 às variáveis de decisão de índice referenciado pelas colunas.

É importante ressaltar que algumas das melhores soluções do problema podem não estar explicitamente indicados na Tabela por estarem dominadas pelas soluções principais encontradas. Estas soluções, denominadas de soluções alternativas, consistem de outras possíveis combinações dominadas pela solução principal a ser recuperada. Soluções alternativas são identificadas durante o próprio processo de recuperação das soluções.

Na lista de soluções anteriormente definida, a melhor solução possui valor 12, lado direito 8 e se encontra na coluna 1 (correspondente ao item 1) da Tabela 2.4. A recuperação desta solução é feita aplicando-se o seguinte processo retroativo:

solução de valor  $(12-3)=9$ , lado direito  $(8-2)=6$ , coluna 1,  $a_1 = (0+1) = 1$  ;

solução de valor  $(9-3)=6$ , lado direito  $(6-2)=4$ , coluna 1,  $a_1 = (1+1) = 2$  ;

solução de valor  $(6-3)=3$ , lado direito  $(4-2)=2$ , coluna 1,  $a_1 = (2+1) = 3$  ;

solução de valor  $(3-3)=0$ , lado direito  $(2-2)=0$ , coluna 1.  $a_1 = (3+1) = 4$  ;

solução recuperada:  $a_1 = 4$  .

A segunda melhor solução da lista possui valor 10, lado direito 8 e se encontra na coluna 3 da Tabela 2.4. O processo retroativo é novamente aplicado, obtendo-se num primeiro estágio, a solução de valor  $(10-4)=6$ , lado direito  $(8-4)=4$ , coluna 3,  $a_3 = (0+1) = 1$  . Note que a linha correspondente ao lado direito 4, indica que existem 2 possíveis combinações que podem ser recuperadas. A combinação relacionada à solução principal corresponde a:



solução de valor  $(6-3)=3$ , lado direito  $(4-2)=2$ , coluna 1,  $a_1 = (0 + 1) = 1$ ;

solução de valor  $(3-3)=0$ , lado direito  $(2-2)=0$ , coluna 1,  $a_1 = (1 + 1) = 2$ ;

solução recuperada:  $a_1 = 2, a_3 = 1$ .

A outra possível combinação, corresponde à recuperação da seguinte solução alternativa:

solução de valor  $(4-4)=0$ , lado direito  $(4-4)=2$ , coluna 3,  $a_3 = (1 + 1) = 2$ .

A solução recuperada é dada por  $a_3 = 2$ , cujo valor corresponde a  $10-(6-4)=8$ . Como o valor da solução alternativa é superior ao limitante inferior (de valor 7), então a solução é inserida na lista das K-melhores soluções que é atualizada para 12\*, 10\*, 9, \*8, 8, onde (\*) indica soluções que já foram recuperadas.

A terceira e quinta melhores soluções são recuperadas aplicando-se o processo retroativo de forma similar ao apresentado anteriormente, obtendo-se:

solução de valor 9:  $a_1 = 3$ ;

solução de valor 8:  $a_1 = 2, a_2 = 1$ .

Portanto, obtém-se finalmente a Tabela 2.5 com as 5-melhores soluções recuperadas para o exemplo:

TABELA 2.5 – TABELA COM AS 5-MELHORES SOLUÇÕES RECUPERADAS

$a_1$	$a_2$	$a_3$	Valor da Função Objetivo
4	0	0	12
2	0	1	10
3	0	0	9
0	0	2	8
2	1	0	8

Em 1965, Gilmore e Gomory analisaram o problema do corte de estoque de duas ou mais dimensões. Neste trabalho, os autores descreveram pela primeira vez, o uso de padrões tabuleiros na indústria de vidro e aço e referiram-se a eles como *padrões 1-grupo*. A utilização do termo padrão tabuleiro foi observado em algumas empresas de corte de chapas de madeira e deve-se possivelmente à forma do padrão.

Gilmore e Gomory não entraram em muitos detalhes, apenas descreveram padrões tabuleiros como um tipo de padrão de corte que pertence a uma dentre outras subclasses especiais de padrões 2-estágios guilhotinados descritos no trabalho. Foram citadas as dificuldades de se trabalhar com este tipo de padrão devido às restrições da escolha dos itens para gerá-lo.

Gilmore e Gomory trataram do problema do corte bidimensional de padrões 2-estágios guilhotinado não exato e irrestrito apresentando o seguinte método para resolvê-lo. Sejam  $W$  e  $L$ , respectivamente, a largura e o comprimento da chapa e sejam  $w_j$  e  $l_j$ , respectivamente a largura e comprimento associados à  $j$ -ésima peça,  $j=1, \dots, n$ . A determinação do padrão é feita em 2 fases. Na primeira fase busca-se uma melhor combinação nas faixas  $L \times w_j$  para cada uma das  $m$  diferentes larguras  $w_j$  do problema:

$$V_j = \text{Maximizar } \sum_{i \in W_j} \pi_i \beta_{ij}$$

$$\text{Sujeito a: } \sum_{i \in W_j} l_i \beta_{ij} \leq L$$

$$\beta_{ij} \geq 0 \quad (j=1, \dots, m) \text{ e } \beta_{ij} \text{ inteiro,}$$

onde  $W_j = \{i \mid w_i \leq w_j\}$ .

Na segunda fase determina-se uma boa combinação de faixas para as faixas definidas na primeira fase resolvendo-se mais um problema unidimensional da mochila:

$$\begin{aligned} &\text{Maximizar } \sum_{j=1}^m V_j \phi_j \\ &\text{Sujeito a: } \sum_{j=1}^m w_j \phi_j \leq W \\ &\phi_j \geq 0 \text{ e } \phi_j \text{ inteiro.} \end{aligned}$$

Estes padrões gerados permitem refilos (recortes). Padrões com um número maior de refilos ou de estágios diminuem a produtividade da máquina devido ao tempo requerido para o seu processamento.

A literatura que trata de problemas de corte principalmente nos últimos trinta anos é extensa. Entretanto, poucos trabalhos foram encontrados que abordam questões sobre a produtividade do equipamento de corte e o uso de padrões tabuleiros. Destaca-se o trabalho de Antonio *et al.* (1996) que trata efetivamente de problemas de corte de estoque e a produtividade do equipamento de cortes. Antonio *et al.* (1996), resolveram um problema de corte de estoque unidimensional que consiste não somente em minimizar a perda de material como nos tradicionais problemas de corte de estoque, mas também o tempo de corte. No trabalho, barras de comprimento  $L_i$  e disponibilidade  $M_i$  ( $i=1, \dots, N$ ) são cortadas para produzir peças menores de comprimento  $l_j$  e demanda  $m_j$ , ( $j=1, \dots, n$ ). Para acelerar a produtividade da máquina de corte, as barras de mesmo tipo, aparência e pedido são separadas e empilhadas uma sobre as outras de maneira a formar um conjunto denominado pacote. Considera-se que o tempo de corte de um pacote é menor que a soma do tempo de corte individual de cada um dos objetos. O problema é resolvido aplicando-se uma heurística FFD (First Fit Decreasing) tendo o custo associado aos objetos calculado de acordo com o tamanho dos pacotes.

Um outro destaque é o trabalho recente de Morabito e Arenales (2000). Morabito e Arenales utilizaram a abordagem clássica de Gilmore e Gomory (1961, 1963, 1965) para resolver o problema de corte de estoque de uma indústria de móveis. Nesta empresa, o processo de cortes de chapas de madeiras deve ser feito com alto grau de

padronização e em grande escala de produção. Segundo Morabito e Arenales, se o custo dos objetos (chapas) forem diferenciados, nem sempre as soluções de perda mínima são as mais econômicas. Portanto, um critério de custo mínimo de material deve ser empregado para avaliar as melhores soluções. Para resolver o problema, os autores construíram um modelo baseado na abordagem clássica de Gilmore e Gomory (1965) combinando um programa linear com um procedimento de geração de colunas.

Outro importante aspecto estudado por Morabito e Arenales foi a análise do “balanço” da utilização de padrões tabuleiros para melhorar a produtividade da máquina e o desperdício de material. Apesar destes padrões acelerarem a produção na indústria de móveis eles podem acarretar um grande desperdício de material. Conseqüentemente, existe um “balanço” entre cortar padrões tabuleiros que melhoram a produtividade da máquina, mas que resultam em um desperdício maior de matéria-prima ou cortar padrões que resultam em menores perdas, mas que reduzem a produtividade da serra. Para analisar este problema, Morabito e Arenales sugeriram a utilização de um custo fixo  $\delta$  para os padrões não tabuleiros devido ao tempo médio extra de processamento de corte. Eles 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:

$$\text{Maximizar } \sum_{i=1}^n \sum_{k=1}^n \pi_{ik} \lambda_i \mu_k$$

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

$$\sum_{k=1}^n w_k \mu_k \leq W$$

$$\lambda_i \geq 0, \mu_k \geq 0 \quad (i, k=1, \dots, n),$$

onde  $\lambda_i$  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;  $\pi_{ik}$  é uma matriz  $n \times n$  que representa a utilidade/valor da peça demandada de dimensão  $l_i \times w_k$ , sendo:

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

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):

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

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:

$$V_j = \text{Maximizar } \sum_{i \in W_j} \pi_i \lambda_i$$

$$\text{Sujeito a: } \sum_{i \in W_j} l_i \lambda_i \leq L$$

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

onde  $W_j = \{i \mid w_i = w_j\}$ .

Passo 2:

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

onde  $D_w$  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.

Não se tem conhecimento de algoritmos exatos para determinação de padrões tabuleiros propostos na literatura. O algoritmo sugerido nesta dissertação é, portanto, uma nova contribuição para esta área. O algoritmo utiliza adaptações de algoritmos exatos para a resolução do problema da mochila. Em particular, o algoritmo para determinação das K-melhores soluções do problema da mochila unidimensional proposto por Yanasse, Soma e Maculan (2000) apresentado anteriormente é fundamental no algoritmo exato proposto.

### CAPÍTULO 3

## UM ALGORITMO ENUMERATIVO PARA DETERMINAÇÃO DE PADRÕES TABULEIROS EXATOS

Neste Capítulo apresenta-se um algoritmo exato para a geração de padrões tipo tabuleiro. O algoritmo proposto funciona em duas fases, primeiro, buscando uma melhor combinação nas faixas e, depois, buscando uma boa combinação de faixas.

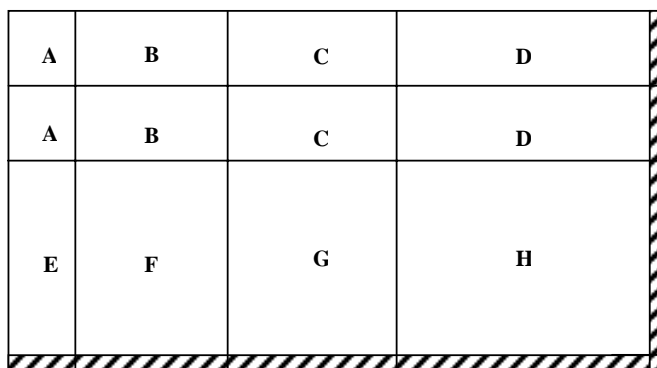


Fig. 3.1 – Um exemplo de padrão tabuleiro.

Por exemplo, para obtenção do padrão apresentado na Figura 3.1, 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.

Deseja-se determinar um padrão de corte tabuleiro exato que maximize o lucro das peças cortadas. Existem algumas características de ordem prática que precisam ser consideradas, por exemplo, as lâminas de corte geram desperdícios causados pelas espessuras das lâminas que podem ser diferentes na largura e comprimento da chapa.

Estes podem ser considerados simplesmente adicionando-se as respectivas espessuras da lâmina de corte, na largura e comprimento da chapa e, na largura e comprimento de cada peça (vide Morabito e Arenales, 2000). Pode ocorrer também que a rotação de peças para o corte sejam permitidas. Para considerar isto basta adicionar mais uma peça ao problema tendo as dimensões da peça original rotacionada.

Considera-se a seguir a enumeração de combinações na primeira fase do algoritmo. Sem perda de generalidade, admite-se que a enumeração nesta primeira fase esteja sendo feita na largura da chapa. A enumeração, portanto, deverá ser feita para cada grupo de peças de mesmo comprimento. Se existirem  $n$  peças no problema, podem existir  $n$  comprimentos diferentes e, portanto, são resolvidos até  $n$  problemas da mochila nesta fase. Ao se enumerar todas as faixas possíveis, é possível construir todos os padrões tabuleiros com o potencial de atingir o lucro máximo possível dentre todos os padrões tabuleiros que podem ser gerados.

Observe que para todo padrão tabuleiro é possível reordenar as faixas que o compõem de modo que a faixa mais lucrativa seja a primeira delas (faixa base de referência). Assim, enumerando-se todas as faixas bases possíveis em ordem não crescente de lucro pode-se gerar todos os padrões tabuleiros exatos possíveis. Muitas destas faixas podem ser descartadas comparando-se o lucro potencial que o padrão gerado, a partir destas, poderá vir a atingir.

Para a comparação de faixas, utiliza-se o lucro, por unidade de comprimento associado às peças, ou seja, associado ao grupo de peças de mesmo comprimento  $l_i$ , a  $m$ -ésima peça tem lucro associado  $c_m = p_m / l_i$ , onde  $p_m$  é o lucro associado a  $m$ -ésima peça. Note que se  $c_m$  for o lucro por unidade de comprimento da faixa, o lucro máximo de um padrão tabuleiro gerado por esta faixa é dado por  $c_m L$ .

A enumeração de todas as faixas bases possíveis pode ser feita resolvendo-se o problema de se achar todas as possíveis soluções de um problema da mochila



unidimensional. Para isso, foi escolhido o algoritmo de Yanasse, Soma e Maculan (2000) denotado por YSM, que gera as K-melhores soluções do problema da mochila unidimensional (vide Capítulo 2). O algoritmo YSM para obtenção das K-melhores soluções do problema da mochila foi desenvolvido para resolver especificamente problemas onde a restrição da mochila tem apenas coeficientes naturais. Na prática, os dados são sempre inteiros ou racionais. Neste último caso, eles podem ser convertidos em inteiros, se necessário. Considere uma chapa de dimensões  $W=15$  e  $L=23$  e peças a serem cortadas, conforme apresentadas na Tabela 3.1:

TABELA 3.1– DADOS DE UM EXEMPLO

Peça $i$	$(w_i \times l_i)$		Rotação (S/N) S=sim, N=não	Lucro $p_i$
1	2	3	S	6
2	2	6	S	10
3	3	4	N	2
4	4	2	N	3
5	4	4	N	8
6	5	2	N	9
7	5	3	S	5
8	6	3	N	1
9	6	4	S	40

Inicialmente, agrupa-se as peças de mesmo comprimento ou de mesma largura. Desta forma são determinadas as dimensões para os quais as combinações podem ser realizadas. No caso do exemplo, as dimensões dos itens de mesmo comprimento são:

Comprimento 2:  $\{w_1'=3, w_4'=4, w_6'=5, w_2'=6\}$  com lucros respectivos de  $\{c_1=3, c_4=1.5, c_6=4.5, c_2=5\}$ ; Comprimento 3:  $\{w_1'=2, w_7'=5, w_8'=6\}$  com lucros respectivos de  $\{c_1=2, c_7=1.67, c_8=0.33\}$ ; Comprimento 4:  $\{w_3'=3, w_5'=4, w_9'=6\}$  com lucros respectivos de  $\{c_3=0.5, c_5=2, c_9=10\}$ ; Comprimento 5:  $\{w_7'=3\}$  com lucros respectivos de  $\{c_7=1\}$ ; Comprimento 6:  $\{w_2'=2, w_9'=4\}$  com lucros respectivos de  $\{c_2=1.67, c_9=6.67\}$ , onde  $w_i'$  corresponde à dimensão relevante a ser considerada na peça  $i$ .

Sendo que os itens de mesma largura são:

Largura 2:  $\{l_1'=3, l_2'=6\}$  com lucros respectivos de  $\{c_1=3, c_2=5\}$ ; Largura 3:  $\{l_1'=2, l_3'=4, l_7'=5\}$  com lucros respectivos de  $\{c_1=2, c_3=0.67, c_7=1.67\}$ ; Largura 4:  $\{l_4'=2, l_5'=4, l_9'=6\}$  com lucros respectivos de  $\{c_4=0.75, c_5=2, c_9=10\}$ ; Largura 5:  $\{l_6'=2, l_7'=3\}$  com lucros respectivos de  $\{c_6=4.5, c_7=1\}$ ; Largura 6:  $\{l_2'=2, l_8'=3, l_9'=4\}$  com lucros respectivos de  $\{c_2=1.67, c_8=0.17, c_9=6.67\}$ , onde  $l_i'$  corresponde à dimensão relevante a ser considerada na peça  $i$ .

Utilizando o algoritmo YSM enumerando as soluções na largura da chapa, obtêm-se a Tabela 3.2, que apresenta as possíveis combinações de soluções (de 0 a  $W$ ) para cada comprimento de faixa.

TABELA 3.2 – SOLUÇÕES ENUMERADAS NA LARGURA DA CHAPA

$w_1'=3$ $c_1=3$	$w_4'=4$ $c_4=1.5$	$w_6'=5$ $c_6=4.5$	$w_2'=6$ $c_2=5$	$w_1'=2$ $c_1=2$	$w_7'=5$ $c_7=1.67$	$w_8'=6$ $c_8=0.33$	$w_3'=3$ $c_3=0.5$	$w_5'=4$ $c_5=2$	$w_9'=6$ $c_9=10$	$w_7'=3$ $c_7=1$	$w_2'=2$ $c_2=1.67$	$w_9'=4$ $c_9=6.67$	
$y_1$	$y_2$	$y_3$	$y_4$	$y_1$	$y_2$	$y_3$	$y_1$	$y_2$	$y_3$	$y_1$	$y_1$	$y_2$	$W$
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1
-1	-1	-1	-1	2	-1	-1	-1	-1	-1	-1	1.67	-1	2
3	-1	-1	-1	-1	-1	-1	0.5	-1	-1	1	-1	-1	3
-1	1.5	-1	-1	4	-1	-1	-1	2	-1	-1	3.33	6.67	4
-1	-1	4.5	-1	-1	1.67	-1	-1	-1	-1	-1	-1	-1	5
6	-1	-1	5	6	-1	0.33	1	-1	10	2	5	8.33	6
-1	4.5	-1	-1	-1	3.67	-1	-1	2.5	-1	-1	-1	-1	7
-1	3	7.5	-1	8	-1	2.33	-1	4	-1	-1	6.67	13.33	8
9	-1	6	8	-1	5.67	-1	1.5	-1	10.5	3	-1	-1	9
-1	7.5	9	6.5	10	3.33	4.33	-1	3	12	-1	8.33	15	10
-1	6	10.5	9.5	-1	7.67	2	-1	4.5	-1	-1	-1	-1	11
12	4.5	9	11	12	5.33	6.33	2	6	20	4	10	20	12
-1	10.5	12	9.5	-1	9.67	4	-1	3.5	12.5	-1	-1	-1	13
-1	9	13.5	12.5	14	7.33	8.33	-1	5	14	-1	11.67	21.67	14
15	7.5	13.5	14	-1	11.67	6	2.5	6.5	20.5	5	-1	-1	15

A solução será encontrada avaliando-se as melhores combinações apresentadas na Tabela 3.2. Note que o número de combinações a serem analisadas pode ser extenso. O uso de limitantes para o problema permite restringir a busca de faixas descartando combinações de itens que resultem em lucros inferiores ao limitante já em mãos. Um primeiro limitante pode ser calculado utilizando, por exemplo, uma solução baseada em padrões homogêneos (vide Capítulo 1).

Seleciona-se o melhor padrão tabuleiro homogêneo e atribui-se ao limitante inferior para as faixas o valor da relação (Lucro Total)/L resultante do padrão escolhido. Para o exemplo numérico, o valor do limitante inicial calculado é 17.39 que corresponde a 10 peças 6×4.

Portanto, na Tabela 3.2 as combinações de itens de lucro superior ou igual ao limitante calculado cujas combinações devem ser investigadas são as que apresentam os seguintes valores (em ordem decrescente): 21.67, 20.5, 20, 20. Na tentativa de se obter o melhor tabuleiro avalia-se estas combinações em ordem não crescente de lucratividade. A solução de maior lucro por unidade de comprimento é 21.67. Aplicando-se o processo de recuperação do algoritmo YSM obtêm-se a combinação de 3 peças 4×6 e 1 peça 2×6.

O lucro da faixa pode ser obtido multiplicando-se o valor (21.67) pela largura da faixa (6), ou seja, lucro da faixa = 21.67 (6) = 130. Na Figura 3.2 ilustra-se a faixa obtida:

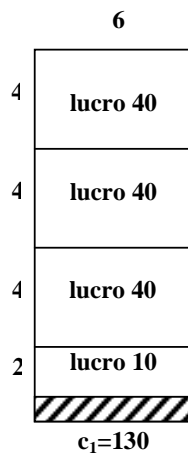


Fig. 3.2 – Faixa gerada na largura.

Deve-se agora selecionar itens de mesmo comprimento que possam ser combinados e formar um padrão tabuleiro. A escolha destes itens é feita selecionando-se grupos de comprimentos iguais relacionados a cada item de largura da solução da faixa encontrada. Verificando-se os itens que foram recuperados, constata-se que os itens de largura 2 possuem comprimentos {3, 6} e os itens de largura 4 possuem comprimentos

{2, 4, 6}. A intersecção desses dois comprimentos dá apenas o comprimento 6, que é a faixa de largura já obtida.

Nosso próximo passo será obter a combinação de faixas para o padrão tabuleiro. Como apenas uma faixa foi encontrada para ser combinada e produzir um padrão tabuleiro, resolve-se então o seguinte problema unidimensional da mochila:

$$\begin{aligned} & \text{Maximizar } 130y'_1 \\ & \text{Sujeito a: } 6y'_1 \leq 23, \quad y'_1 \text{ inteiro não negativo.} \end{aligned}$$

Obtém-se a solução  $y'_1=3$  com valor total de 390, sendo que o padrão tabuleiro correspondente produz 9 peças de tamanho  $4 \times 6$  e 3 peças de tamanho  $2 \times 6$ . A relação Lucro/L ( $390/23=16.96$ ) correspondente a este padrão é pior que o valor do limitante corrente (17.39), portanto, esta solução é descartada e o valor do limitante permanece inalterado.

Uma série de combinações alternativas podem ser geradas no processo de recuperação desta combinação de maior lucro. Apesar de ter-se utilizado a melhor solução na largura da chapa, é possível que uma combinação alternativa de itens, de retorno inferior resulte numa melhor solução global para o problema. Deve-se recuperar e/ou verificar todas as soluções alternativas para assegurar que a melhor solução tenha realmente sido encontrada. O procedimento de recuperação de soluções que foi descrito anteriormente resulta nas seguintes combinações alternativas: Valor 18.33, combinação de 3 peças  $2 \times 6$  e 2 peças  $4 \times 6$ ; Valor 15, combinação de 5 peças  $2 \times 6$  e 1 peça  $4 \times 6$ . As peças que compõem as faixas encontradas para as soluções alternativas de valor 18.33 e 15 são as mesmas da solução 21.67, apenas as quantidades são diferentes.

O lucro total da faixa correspondente à solução alternativa 18.33 é  $18.33(6)=110$ . Como no caso anterior, não existem outras faixas que possam, juntamente com esta, produzir um padrão tabuleiro.

Resolvendo-se o seguinte problema unidimensional da mochila:

$$\begin{aligned} & \text{Maximizar } 110y'_1 \\ & \text{Sujeito a: } 6y'_1 \leq 23, \quad y'_1 \text{ inteiro não negativo,} \end{aligned}$$

obtém-se a solução  $y'_1=3$  com valor total de 330, sendo que o padrão tabuleiro correspondente produz 6 peças de tamanho  $4 \times 6$  e 9 peças de tamanho  $2 \times 6$ . A relação  $\text{Lucro}/L=330/23=14.35$  correspondente a este padrão é pior que o valor do limitante corrente (17.39). Portanto, o valor do limitante permanece inalterado e a solução é descartada. A solução alternativa de valor 15 é automaticamente descartada, pois seu valor já é inferior ao limitante corrente (17.39).

O segundo maior valor selecionado da Tabela 3.2 é 20.5 que é resultante de uma combinação de 1 peça  $3 \times 4$  e 2 peças  $6 \times 4$ . Os itens de largura 3 possuem comprimentos  $\{2, 4, 5\}$  e os itens de largura 6 possuem comprimentos  $\{2, 3, 4\}$ . Efetuando-se a intersecção obtém-se as faixas de comprimentos 2 e 4, este último, já disponível. Os lucros associados a cada faixa são: o lucro da faixa 4 é  $20.5(4)=82$  e o lucro da faixa 2 é  $1(6)+2(10)=26$ . Na Figura 3.3 apresentam-se as faixas que foram obtidas.

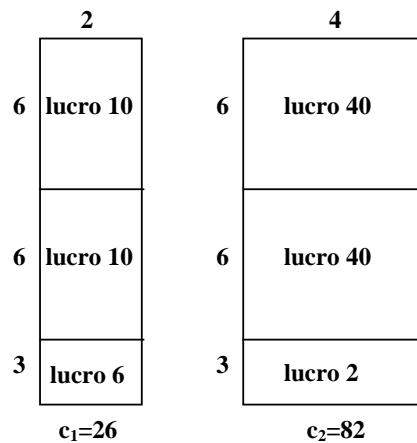


Fig. 3.3 - Faixas de cortes geradas na largura.

Para se determinar a melhor combinação de faixas no comprimento, resolve-se o seguinte problema unidimensional da mochila:

Maximizar  $26y'_1 + 82y'_2$

Sujeito a:  $2y'_1 + 4y'_2 \leq 23$ ,  $y'_1$  e  $y'_2$  inteiros não negativos.

O problema anterior pode ser resolvido, por exemplo, aplicando-se o algoritmo YSM ou outros métodos propostos na literatura para resolução do problema da mochila (vide Capítulo 2). No caso, a solução é dada por  $y'_1=1$  e  $y'_2=5$  com valor total de 436 e o padrão tabuleiro correspondente produz: 10 peças de tamanho  $6 \times 4$ , 5 peças de tamanho  $3 \times 4$ , 2 peças de tamanho  $6 \times 2$  e 1 peça de tamanho  $3 \times 2$ . A relação  $\text{Lucro}/L=436/23=18.96$  correspondente a este padrão é melhor que o valor do limitante corrente (17.39). Portanto, esta solução é armazenada e o valor do limitante é atualizado para 18.96.

Enumerando-se as soluções alternativas no processo de recuperação para a solução de 20.5 temos 3 itens de largura 3 e 1 item de largura 6 com valor por unidade de comprimento igual a 11.5. Como este valor é menor que o limitante corrente ele é automaticamente descartado.

Não se encontra nenhuma solução alternativa com relação  $\text{Lucro}/L$  melhor que o limitante corrente, desta forma prossegue-se a análise das outras combinações de itens da Tabela 3.2, cujos valores sejam superiores ao do limitante corrente.

Resumidamente, a próxima melhor combinação da Tabela tem valor igual a 20 e é composta de 2 itens de largura 6. As faixas possíveis tem comprimento 2, 3, 4 e lucros respectivos de 20, 2, 80. O valor do melhor padrão tabuleiro é 420 fornecendo um  $\text{Lucro}/L=18.26$  que é inferior ao limitante corrente (18.96) e, portanto, a solução é descartada. A partir desta solução recupera-se apenas uma única solução alternativa, composta de 2 itens de largura 3 e 1 item de largura 6. Esta combinação é imediatamente descartada, pois tem valor (11) inferior ao limitante corrente.

A próxima (e última) solução selecionada da Tabela, também possui valor 20 e as combinações recuperadas são descritas a seguir:

**Melhor Combinação:**

Combinação: 3 itens de largura 4

Valor desta combinação:  $20 > \text{limitante corrente} = 18.96$

Faixas de possíveis comprimentos: 2, 4, 6

Lucros associados às faixas: 9, 24, 120

Valor do melhor padrão tabuleiro: 384 ou seja,

$\text{Lucro}/L = 16.6957 < \text{limitante} = 18.96$ . Portanto este padrão é descartado.

**Combinações Alternativas:**

Combinação 1: 2 itens de largura 2, 2 itens de largura 4

Valor desta combinação:  $16.67 < \text{limitante} = 18.96$  (combinação descartada)

Combinação 2: 4 itens de largura 2, 1 item de largura 4

Valor desta combinação:  $13.33 < \text{limitante} = 18.96$  (combinação descartada)

Não havendo mais combinações a serem verificadas que possam gerar um padrão tabuleiro melhor que o corrente, a solução ótima foi encontrada. Tem-se, portanto a seguinte solução ótima para o problema:

Padrão:

na largura: 1 faixa de largura 3, 2 faixas de largura 6.

no comprimento: 1 faixa de comprimento 2, 5 faixas de comprimento 4.

Peças Produzidas:

10 peças de tamanho  $6 \times 4$ , 5 peças de tamanho  $3 \times 4$ , 2 peças de tamanho  $6 \times 2$  e 1 peça de tamanho  $3 \times 2$ .

Valor da solução: 436

O padrão obtido está ilustrado na Figura 3.4:

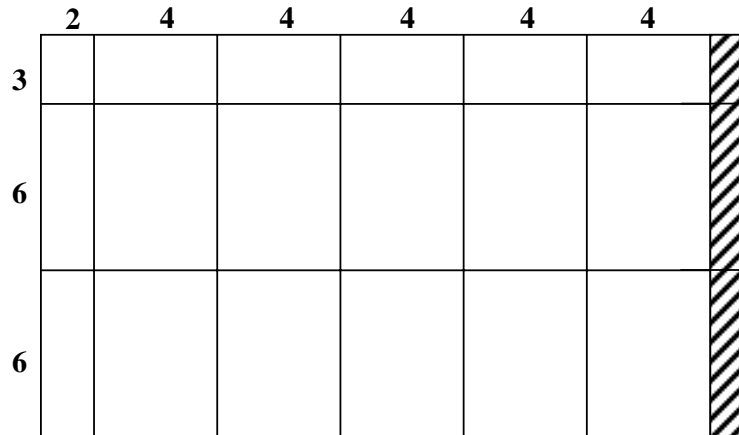


Fig. 3.4 – Padrão tabuleiro exato ótimo obtido para o exemplo.

Um resumo dos passos deste algoritmo para geração de padrões tabuleiros exatos é apresentado a seguir:

Passo 1: Enumeração das Combinações (admitindo serem feitas na largura do objeto)

Separe as peças em grupos de mesmo comprimento e de mesma largura e aplique o algoritmo YSM para cada comprimento de faixa considerando as possíveis combinações (de 0 a  $W$ ).

Passo 2: Cálculo do limitante inicial e seleção das combinações a serem investigadas

Faça  $\text{limitante} = \text{Lucro do Melhor padrão homogêneo} / L$ ;

Guarde no vetor  $\text{lista}[]$  o valor das combinações enumeradas tal que:

$\text{lista}[0] \geq \text{lista}[1] \geq \dots \geq \text{lista}[m] > \text{limitante}$

Passo 3: Recuperação do melhor padrão tabuleiro

Faça  $k=0$ ;



```

Enquanto ( $k < m$ ) faça:
  Enquanto houverem combinações alternativas relativas à lista[k] faça:
    Recupere a combinação;
    Se (valor da combinação recuperada > limitante) faça
      Determine o padrão tabuleiro associado à combinação recuperada;
      Se (lucro do padrão determinado/L > limitante) faça
        limitante = lucro do padrão determinado/L;
        Guarde a solução associada ao limitante calculado;
        Remova soluções de lista[] | valor da solução ≤ limitante;
        Para cada solução removida de lista[] faça  $m = m - 1$ ;
      Fim-Se;
    Fim-Se;
  Fim-Enquanto;
   $k = k + 1$ ;
Fim-Enquanto.

```



## CAPÍTULO 4

### ASPECTOS COMPUTACIONAIS DE IMPLEMENTAÇÃO DO ALGORITMO ENUMERATIVO PARA DETERMINAÇÃO DE PADRÕES TABULEIROS EXATOS

Neste Capítulo serão discutidos alguns aspectos computacionais para a implementação do algoritmo sugerido no Capítulo 3 e a viabilidade de sua aplicação prática. Alguns resultados computacionais são apresentados.

#### ***4.1 - Enumeração das combinações***

Como pode ser observado o algoritmo YSM utiliza um espaço de memória da ordem de  $nW$ , para armazenar todos os dados desejados.

No caso de exemplos numéricos como o ilustrado no Capítulo 3 isto não constitui um problema. Entretanto, o fato de que os requisitos de memória para este algoritmo dependem dos dados do problema precisa estar claro para o usuário que o utiliza. Embora isto possa ser um fator limitante no uso do algoritmo, observa-se que existem outros mecanismos alternativos que podem ser utilizados.

Considere novamente o exemplo numérico do Capítulo 3 e suponha que as dimensões da chapa e das peças estão em metros. Considere, agora, os mesmos dados do exemplo multiplicados por um fator de 1000, ou seja, as dimensões estariam sendo fornecidas em milímetros ao invés de metros. A nova Tabela correspondente do algoritmo YSM (vide Tabela 4.1), para este exemplo, teria agora 15000 linhas. Entretanto, as combinações relevantes a serem consideradas são as mesmas apresentadas na Tabela 4.2, ou seja, o número de linhas na matriz que carregam informação relevante continua o mesmo.

TABELA 4.1- COMBINAÇÕES ENUMERADAS NA LARGURA DA CHAPA

$w_1'=3000$ $c_1=3$	$w_4'=4000$ $c_4=1.5$	$w_6'=5000$ $c_6=4.5$	$w_2'=6000$ $c_2=5$	$w_1'=2000$ $c_1=2$	$w_7'=5000$ $c_7=1.67$	$w_8'=6000$ $c_8=0.33$	$w_3'=3000$ $c_3=0.5$	$w_5'=4000$ $c_5=2$	$w_9'=6000$ $c_9=10$	$w_7'=3000$ $c_7=1$	$w_2'=2000$ $c_2=1.67$	$w_9'=4000$ $c_9=6.67$	W
$y_1$	$y_2$	$y_3$	$y_4$	$y_1$	$y_2$	$y_3$	$y_1$	$y_2$	$y_3$	$y_1$	$y_1$	$y_2$	
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0
:	:	:	:	:	:	:	:	:	:	:	:	:	:
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1000
:	:	:	:	:	:	:	:	:	:	:	:	:	:
-1	-1	-1	-1	2	-1	-1	-1	-1	-1	-1	1.67	-1	2000
:	:	:	:	:	:	:	:	:	:	:	:	:	:
3	-1	-1	-1	-1	-1	-1	0.5	-1	-1	1	-1	-1	3000
:	:	:	:	:	:	:	:	:	:	:	:	:	:
-1	1.5	-1	-1	4	-1	-1	-1	2	-1	-1	3.33	6.67	4000
:	:	:	:	:	:	:	:	:	:	:	:	:	:
-1	-1	4.5	-1	-1	1.67	-1	-1	-1	-1	-1	-1	-1	5000
:	:	:	:	:	:	:	:	:	:	:	:	:	:
6	-1	-1	5	6	-1	0.33	1	-1	10	2	5	8.33	6000
:	:	:	:	:	:	:	:	:	:	:	:	:	:
-1	4.5	-1	-1	-1	3.67	-1	-1	2.5	-1	-1	-1	-1	7000
:	:	:	:	:	:	:	:	:	:	:	:	:	:
-1	3	7.5	-1	8	-1	2.33	-1	4	-1	-1	6.67	13.33	8000
:	:	:	:	:	:	:	:	:	:	:	:	:	:
9	-1	6	8	-1	5.67	-1	1.5	-1	10.5	3	-1	-1	9000
:	:	:	:	:	:	:	:	:	:	:	:	:	:
-1	7.5	9	6.5	10	3.33	4.33	-1	3	12	-1	8.33	15	10000
:	:	:	:	:	:	:	:	:	:	:	:	:	:
-1	6	10.5	9.5	-1	7.67	2	-1	4.5	-1	-1	-1	-1	11000
:	:	:	:	:	:	:	:	:	:	:	:	:	:
12	4.5	9	11	12	5.33	6.33	2	6	20	4	10	20	12000
:	:	:	:	:	:	:	:	:	:	:	:	:	:
-1	10.5	12	9.5	-1	9.67	4	-1	3.5	12.5	-1	-1	-1	13000
:	:	:	:	:	:	:	:	:	:	:	:	:	:
-1	9	13.5	12.5	14	7.33	8.33	-1	5	14	-1	11.67	21.67	14000
:	:	:	:	:	:	:	:	:	:	:	:	:	:
15	7.5	13.5	14	-1	11.67	6	2.5	6.5	20.5	5	-1	-1	15000

Pelo exposto conclui-se que na nova matriz de 15000 linhas, existe um grande número de linhas sem informação, ou seja, os dados relevantes estão presentes de maneira esparsa na matriz. Baseado nesta observação, uma implementação alternativa seria utilizar-se de listas encadeadas que ligam apenas os dados relevantes de interesse. Assim, no caso do exemplo considerado, apenas os dados não negativos e em negrito presentes na Tabela 4.2 estariam listados e encadeados de alguma forma, e caso os dados estivessem multiplicados por 1000, os mesmos dados da Tabela 4.2, estariam listados e encadeados.

TABELA 4.2 – COMBINAÇÕES RELEVANTES ENUMERADAS NA LARGURA

$w_1'=3000$ $c_1=3$	$w_4'=4000$ $c_4=1.5$	$w_6'=5000$ $c_6=4.5$	$w_2'=6000$ $c_2=5$	$w_1'=2000$ $c_1=2$	$w_7'=5000$ $c_7=1.67$	$w_8'=6000$ $c_8=0.33$	$w_3'=3000$ $c_3=0.5$	$w_5'=4000$ $c_5=2$	$w_9'=6000$ $c_9=10$	$w_7'=3000$ $c_7=1$	$w_2'=2000$ $c_2=1.67$	$w_9'=4000$ $c_9=6.67$	W
$y_1$	$y_2$	$y_3$	$y_4$	$y_1$	$y_2$	$y_3$	$y_1$	$y_2$	$y_3$	$y_1$	$y_1$	$y_2$	
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1000
-1	-1	-1	-1	2	-1	-1	-1	-1	-1	-1	<b>1.67</b>	-1	2000
3	-1	-1	-1	-1	-1	-1	<b>0.5</b>	-1	-1	<b>1</b>	-1	-1	3000
-1	<b>1.5</b>	-1	-1	4	-1	-1	-1	<b>2</b>	-1	-1	<b>3.33</b>	<b>6.67</b>	4000
-1	-1	<b>4.5</b>	-1	-1	<b>1.67</b>	-1	-1	-1	-1	-1	-1	-1	5000
6	-1	-1	5	6	-1	<b>0.33</b>	1	-1	10	2	5	<b>8.33</b>	6000
-1	<b>4.5</b>	-1	-1	-1	<b>3.67</b>	-1	-1	<b>2.5</b>	-1	-1	-1	-1	7000
-1	3	7.5	-1	8	-1	2.33	-1	4	-1	-1	<b>6.67</b>	<b>13.33</b>	8000
9	-1	6	8	-1	5.67	-1	1.5	-1	10.5	3	-1	-1	9000
-1	7.5	9	6.5	10	3.33	4.33	-1	3	12	-1	<b>8.33</b>	15	10000
-1	6	10.5	9.5	-1	7.67	2	-1	4.5	-1	-1	-1	-1	11000
12	4.5	9	11	12	5.33	6.33	2	6	20	4	10	20	12000
-1	10.5	12	9.5	-1	9.67	4	-1	3.5	12.5	-1	-1	-1	13000
-1	9	13.5	12.5	14	7.33	8.33	-1	5	14	-1	11.67	21.67	14000
15	7.5	13.5	14	-1	11.67	6	2.5	6.5	20.5	5	-1	-1	15000

Observe que matrizes esparsas ocorrerão sempre que as dimensões das peças não forem significativamente menores que as dimensões das chapas (ou seja, o número total de peças que podem ser cortadas de uma chapa não é muito grande).

É preciso lembrar que a utilização de listas encadeadas apresenta um custo adicional de manipulação de listas que precisa ser considerado.

Uma outra possibilidade a ser considerada, e que pode limitar o tamanho da matriz a ser enumerada em troca de um acréscimo no número de soluções a considerar, utiliza uma mudança de escala nos dados do problema, conforme sugerido anteriormente em Yanasse *et al.* (1991).

Considere o exemplo que vem sendo utilizado e imagine que os dados estivessem sido dados em milímetros. Ter-se-ia, pois uma matriz de 15000 linhas a ser considerada. Se a escala dos dados do problema fosse mudada e o metro fosse considerado como a unidade de medida, no novo problema ter-se-ia que enumerar somente a matriz apresentada na Tabela 4.4, ao invés da matriz com 15000 linhas. Este caso ilustra bem o que se pretende realizar com a mudança de escala.

A mudança de escala funciona sem nenhum problema quando a escala utilizada é um divisor comum dos dados do problema. No caso do exemplo, todas as dimensões consideradas eram múltiplas de 1000. Para se diminuir o tamanho da matriz a ser enumerada, sem que se tenha nenhum problema, deve-se dividir as dimensões das peças e chapas pelo máximo divisor comum (MDC). Mesmo após a realização desta divisão pelo MDC os valores resultantes a serem manipulados podem ser grandes implicando na enumeração de uma grande matriz.

A sugestão seria a escolha de um fator de escala adequado que resultasse em valores “manipuláveis”. Entretanto, nesta nova escala, os valores resultantes não mais são

inteiros e, desta forma, para que o algoritmo de YSM possa ser utilizado, arredondamentos terão que ser realizados. Sugere-se a adoção do seguinte procedimento: os dados não inteiros das dimensões da peças são arredondados para os maiores inteiros menores ou iguais às dimensões das peças; os dados não inteiros das dimensões da chapa são arredondados para os menores inteiros maiores ou iguais às dimensões da chapa.

Por exemplo, no caso dos dados do problema da Tabela 3.2, suponha que se divida todos os dados por 2. Assim, para a enumeração, os seguintes dados seriam considerados:

TABELA 4.3 – NOVAS DIMENSÕES A SEREM CONSIDERADAS DE PEÇAS DE MESMO COMPRIMENTO

<i>i</i>	Comprimento <i>i</i>	Larguras correspondentes ajustadas na nova escala
1	2	1, 2, 2, 3
2	3	1, 2, 3
3	4	1, 2, 3
4	5	1
5	6	1, 2

A enumeração feita levaria à Tabela 4.4 cujo tamanho é praticamente a metade (número de linhas anterior dividido pelo fator de escala utilizado) da Tabela 4.2.

TABELA 4.4 – COMBINAÇÕES ENUMERADAS NA LARGURA DA CHAPA

$w_1'=1$ $c_1=3$	$w_4'=2$ $c_4=1.5$	$w_6'=2$ $c_6=4.5$	$w_2'=3$ $c_2=5$	$w_1'=1$ $c_1=2$	$w_7'=2$ $c_7=1.67$	$w_8'=3$ $c_8=0.33$	$w_3'=1$ $c_3=0.5$	$w_5'=2$ $c_5=2$	$w_9'=3$ $c_9=10$	$w_7'=1$ $c_7=1$	$w_2'=1$ $c_2=1.67$	$w_9'=2$ $c_9=6.67$	
$y_1$	$y_2$	$y_3$	$y_4$	$y_1$	$y_2$	$y_3$	$y_1$	$y_2$	$y_3$	$y_1$	$y_1$	$y_2$	<i>W</i>
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0
3	-1	-1	-1	2	-1	-1	0.5	-1	-1	1	1.67	-1	1
6	1.5	4.5	-1	4	1.67	-1	1	2	-1	2	3.33	6.67	2
9	4.5	7.5	5	6	3.67	0.33	1.5	2.5	10	3	5	8.33	3
12	7.5	10.5	8	8	5.67	2.33	2	4	10.5	4	6.67	13.33	4
15	10.5	13.5	11	10	7.67	4.33	2.5	4.5	12	5	8.33	15	5
18	13.5	16.5	14	12	9.67	6.33	3	6	20	6	10	20	6
21	16.5	19.5	17	14	11.67	8.33	3.5	6.5	20.5	7	11.67	21.67	7
24	19.5	22.5	20	16	13.67	10.33	4	8	22	8	13.33	26.67	8

Devido aos arredondamentos efetuados, nem todas as soluções geradas são pertinentes ao problema original. Assim, uma verificação da pertinência da solução gerada deverá ser realizada.

Considere, por exemplo, a combinação de maior lucro por unidade de comprimento obtida da Tabela 4.4. O valor desta solução é 26.67 e é obtida com a combinação de 4 peças  $4 \times 6$ . Facilmente se verifica que esta combinação é inviável, pois ultrapassa a largura da chapa, portanto, ela é descartada. Uma série de combinações alternativas (de valor inferior) também podem ser obtidas. Por exemplo, algumas combinações que poderiam ser geradas seriam:

Valor 21.68, combinação de 1 peça  $2 \times 6$  e 3 peças  $4 \times 6$  (viável);

Valor 16.68, combinação de 2 peças  $2 \times 6$  e 2 peças  $4 \times 6$  (viável);

Valor 11.68, combinação de 3 peças  $2 \times 6$  e 1 peça  $4 \times 6$  (viável);

O segundo maior valor é 24 que é resultante de uma combinação de 8 peças  $3 \times 2$ . A combinação é descartada, pois ultrapassa (e muito) a largura. Não existe combinação alternativa com exatamente o mesmo lucro ou inferior, desta forma, prossegue-se verificando outras combinações de lucro elevado. O terceiro maior valor é 22.5 que é proveniente de uma combinação de 6 peças  $3 \times 2$  e 1 peça  $5 \times 2$ . Novamente, é facilmente verificado que esta combinação não é viável, pois ultrapassa a largura da chapa. Neste caso, uma série de combinações alternativas (de valor inferior) também podem ser obtidas. Por exemplo, algumas combinações que poderiam ser geradas seriam:

Valor 21, combinação de 4 peças  $3 \times 2$  e 2 peças  $5 \times 2$ ;

Valor 19.5, combinação de 2 peças  $3 \times 2$  e 3 peças  $5 \times 2$ ;

Valor 18.0, combinação de 4 peças  $5 \times 2$ ;

Valor 18.0, combinação de 4 peças  $3 \times 2$ , 1 peça  $4 \times 2$  e 1 peça  $5 \times 2$ ;

Valor 16.5, combinação de 2 peças  $3 \times 2$ , 1 peça  $4 \times 2$  e 2 peças  $5 \times 2$ ;

e assim, sucessivamente. Observe que todas estas soluções anteriores geradas são inviáveis.

O que precisa ser notado é que o procedimento é capaz de gerar todas as combinações desejadas do problema geral, embora com um esforço computacional adicional de cálculo, para recuperação e verificação da viabilidade das combinações encontradas. A vantagem deste procedimento é ter sempre o controle do tamanho da matriz sendo enumerada, o que limita o tempo computacional para sua montagem.

#### 4.2 - Combinações dominantes

Os padrões tabuleiros que são gerados pelo procedimento proposto no Capítulo 3 são particulares, no sentido de que “perdas internas” não são permitidas. Por exemplo, no padrão ilustrado na Figura 3.1, se qualquer uma das peças A, B, C, D, E, F ou G fosse na verdade um desperdício de material, o procedimento sugerido não seria capaz de gerar este padrão. Padrões tabuleiros, sem perdas internas, permitem que se considere apenas *combinações dominantes*. Considere, por exemplo, as combinações de cortes para uma mesma largura de faixa, apresentadas na Figura 4.1:

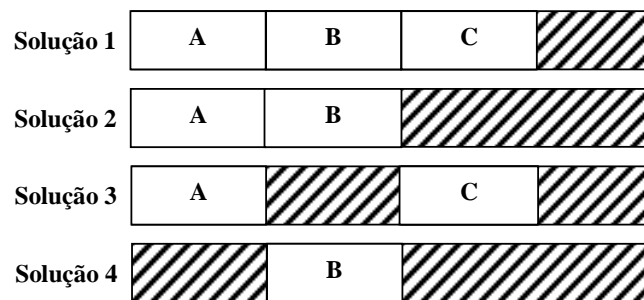


Fig. 4.1- Combinações de cortes.

A solução 1 domina as demais (é mais lucrativa que as demais), portanto, como se está interessado em gerar o padrão tabuleiro mais lucrativo sem buracos internos, pode-se



considerar apenas ela, não havendo a necessidade de recuperar todas as demais combinações dominadas.

Combinações dominantes são facilmente identificáveis no processo enumerativo realizado. Estas combinações são todas aquelas para as quais nenhuma ramificação é realizada na enumeração progressiva do algoritmo de YSM.

#### **4.3 - Escolha da dimensão para se fazer a enumeração**

Classificam-se inicialmente as peças em grupos de mesmo comprimento e/ou mesma largura. Um processo de enumeração é feito apenas com os itens de um mesmo grupo. Devido à particularidade dos padrões tabuleiros, faixas podem ser definidas na largura ou comprimento da chapa. Por exemplo, para o padrão apresentado na Figura 3.1, poder-se-ia ter gerado na primeira fase a faixa (E, F, G, H) e gerado a faixa (A, B, C, D) na segunda fase do algoritmo. Alternativamente, poder-se-ia ter gerado a faixa (A, A, E) na primeira fase e, gerado as faixas (B, B, F), (C, C, G) e (D, D, H) na segunda fase do algoritmo.

Uma primeira decisão a ser tomada é escolher se a enumeração das combinações será feita na largura ou no comprimento da chapa. Se as peças puderem ser rotacionadas, a melhor escolha seria realizar a enumeração na largura da chapa (admitindo-se  $W < L$ ), pois o número de combinações possíveis é menor (ou igual) do que na outra dimensão. Por outro lado, caso boa parte das peças não puder ser rotacionada, pode-se pensar que o tamanho relativo das dimensões das peças com relação às dimensões da chapas poderá ditar esta escolha.

Alguns testes computacionais foram realizados visando determinar algumas características nos dados do problema que possam indicar qual enumeração é menos custosa. Na Figura 4.2 são apresentados os resultados de testes computacionais considerando-se uma chapa de dimensão 1858×2031. O total de diferentes tipos de

peças ( $n$ ) foi fixado em 28, sendo que os itens não podem ser rotacionados. As dimensões de cada peça foram geradas aleatoriamente fixando-se os comprimentos das peças entre 25% e 30% do comprimento da chapa e variando-se suas larguras entre 10% a 14%, 15% a 19%, 20% a 24%, 30% a 34%, 40 a 44%, 50% a 54% e 60% a 64% da largura da chapa. Embora a escolha destas dimensões poderia ser feita em outros intervalos de tamanhos relativos, a faixa de comprimento escolhida procurou apenas garantir um número médio de 3 a 4 itens por faixa.

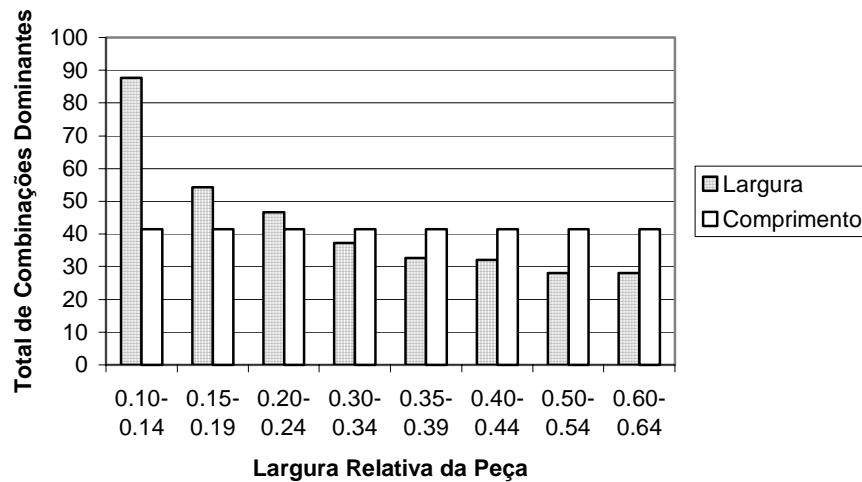


Fig. 4.2 – Combinações dominantes variando-se o tamanho médio das peças em relação às dimensões da chapa.

O tamanho relativo da peça em relação ao da chapa poderia ser utilizado na escolha da dimensão a ser enumerada. Uma média de tamanhos relativos na primeira dimensão (comprimento) superior à média da segunda dimensão (largura) indica que, potencialmente, a primeira possui menos combinações que a segunda e, portanto, a primeira dimensão deveria ser escolhida. Entretanto, nem sempre esta média é suficiente para se determinar a melhor escolha. Apesar da enumeração utilizar menos esforço computacional, o número de combinações dominantes pode ser maior no caso da média relativa ser maior.

Nos testes computacionais ilustrados na Figura 4.2 o tamanho médio relativo como critério de escolha funciona bem, talvez, devido aos desvios padrões dos tamanhos

relativos serem pequenos nos dados gerados. Na Figura 4.3 apresenta-se os resultados de testes computacionais realizados quando os tamanhos relativos tem uma maior dispersão. As médias relativas e os totais de combinações dominantes obtidas em cada dimensão estão indicados na Figura 4.3a e 4.3b.

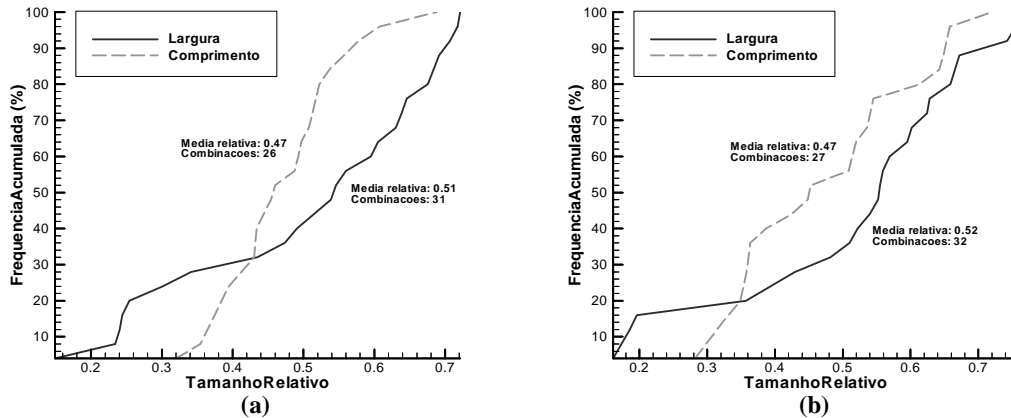


Fig. 4.3 - Distribuição de freqüência acumulada dos tamanhos relativos: (a) largura min. 0.14, largura max. 0.76, comprimento min. 0.30, comprimento max. 0.70; (b) largura min. 0.14, largura max. 0.76, comprimento min. 0.26, comprimento max. 0.76.

Um critério que poderia funcionar bem em boa parte dos casos em que o critério de média relativa falha consiste em escolher a dimensão a ser enumerada pelo menor dentre os intervalos  $\Delta w$  e  $\Delta l$  relativos, ou seja:

$$\min\left(\frac{\Delta w}{W}, \frac{\Delta l}{L}\right),$$

onde  $\Delta w = w_{\max} - w_{\min}$  e  $\Delta l = l_{\max} - l_{\min}$  ( $w_{\max}$ ,  $w_{\min}$ ,  $l_{\max}$  e  $l_{\min}$  são respectivamente a maior e menor largura relativa e o maior e menor comprimento relativo).

Conforme pode ser observado na esquematização apresentada na Figura 4.4, este valor é proporcional (estimativa grosseira) ao número de combinações dominantes presentes na Tabela de enumeração (representados por  $x$  na Tabela).

Fig. 4.4 -  $\Delta w$  e  $\Delta l$  na Tabela de enumeração.

Este critério pode ser melhorado considerando-se também a média relativa dos tamanhos das peças para se determinar o número de combinações baseado numa estimativa dos tamanhos dos itens presentes nos intervalos  $\Delta w$  e  $\Delta l$ , ou seja:

$$\min \left( \frac{\frac{\Delta w}{W}}{\frac{w}{W}}, \frac{\frac{\Delta l}{L}}{\frac{l}{L}} \right) = \min \left( \frac{\Delta w}{w}, \frac{\Delta l}{l} \right),$$

onde  $\bar{w}$  é a média dos tamanhos das peças com relação à largura da chapa e  $\bar{l}$  é a média dos tamanhos das peças com relação ao comprimento da chapa.

Um outro critério consiste em escolher a dimensão que apresente menor variedade de diferentes itens, ou seja, escolher a dimensão que apresente o menor número de tamanhos diferentes de itens (larguras ou comprimentos das peças) a serem combinados.

Na Tabela 4.5 apresenta-se os resultados de testes computacionais realizados onde os diversos critérios,  $\min\left(\frac{\Delta w}{W}, \frac{\Delta l}{L}\right)$ , média relativa,  $\min\left(\frac{\Delta w}{w}, \frac{\Delta l}{l}\right)$  e menor variedade, foram aplicados. Basicamente os mesmos dados de entrada do teste apresentado na Figura 4.2 foram considerados nos testes realizados: chapa de dimensões 1858×2031, comprimentos das peças fixados entre 25% e 30% do tamanho da chapa, total de

peças=28. As larguras relativas das peças (com relação à largura da chapa) foram geradas aleatoriamente em diferentes intervalos. Para cada intervalo foram considerados 20 exemplos. Nas colunas 2 e 3 da Tabela apresenta-se os valores mínimos e máximos dos intervalos de larguras que os dados foram gerados e as 8 últimas colunas mostram o número de exemplos em que os critérios funcionaram e falharam. Os valores em negrito destacam os casos onde o critério funcionou em boa parte dos exemplos testados:

TABELA 4.5 - CRITÉRIOS PARA A ESCOLHA DA DIMENSÃO PARA SE FAZER A ENUMERAÇÃO

Intervalo	Largura mínima da peça	Largura máxima da peça	$\min\left(\frac{\Delta w}{W}, \frac{\Delta l}{L}\right)$		Média Relativa $\max(\bar{w}, \bar{l})$		$\min\left(\frac{\Delta w}{w}, \frac{\Delta l}{l}\right)$		Menor Variedade	
			OK	FALHA	OK	FALHA	OK	FALHA	OK	FALHA
1	0,14	0,20	9	11	<b>13</b>	<b>7</b>	9	11	<b>19</b>	<b>1</b>
2	0,14	0,21	4	16	<b>17</b>	<b>3</b>	<b>17</b>	<b>3</b>	<b>16</b>	<b>4</b>
3	0,14	0,22	8	12	<b>12</b>	<b>8</b>	<b>12</b>	<b>8</b>	<b>14</b>	<b>6</b>
4	0,14	0,23	2	18	<b>18</b>	<b>2</b>	<b>18</b>	<b>2</b>	<b>16</b>	<b>4</b>
5	0,14	0,24	3	17	<b>17</b>	<b>3</b>	<b>17</b>	<b>3</b>	<b>16</b>	<b>4</b>
6	0,20	0,21	<b>20</b>	<b>0</b>	0	20	<b>20</b>	<b>0</b>	<b>19</b>	<b>1</b>
7	0,20	0,22	<b>19</b>	<b>1</b>	1	19	<b>19</b>	<b>1</b>	<b>19</b>	<b>1</b>
8	0,20	0,23	<b>19</b>	<b>1</b>	1	19	<b>19</b>	<b>1</b>	<b>18</b>	<b>2</b>
9	0,20	0,24	<b>20</b>	<b>0</b>	0	20	<b>20</b>	<b>0</b>	<b>19</b>	<b>1</b>
10	0,20	0,25	<b>16</b>	<b>4</b>	5	15	<b>16</b>	<b>4</b>	<b>17</b>	<b>3</b>
11	0,20	0,26	<b>14</b>	<b>6</b>	6	14	<b>14</b>	<b>6</b>	<b>18</b>	<b>2</b>
12	0,20	0,27	<b>15</b>	<b>5</b>	6	14	<b>15</b>	<b>5</b>	<b>16</b>	<b>4</b>
13	0,20	0,28	<b>16</b>	<b>4</b>	4	16	<b>16</b>	<b>4</b>	<b>18</b>	<b>2</b>
14	0,20	0,29	<b>14</b>	<b>6</b>	7	13	<b>12</b>	<b>8</b>	<b>17</b>	<b>3</b>
15	0,28	0,48	5	15	<b>15</b>	<b>5</b>	5	15	<b>16</b>	<b>4</b>
16	0,30	0,48	4	16	<b>17</b>	<b>2</b>	<b>13</b>	<b>7</b>	<b>19</b>	<b>1</b>
17	0,34	0,49	3	17	<b>18</b>	<b>1</b>	4	16	<b>17</b>	<b>3</b>
18	0,34	0,50	1	19	<b>19</b>	<b>1</b>	4	16	<b>18</b>	<b>2</b>
19	0,34	0,51	2	18	<b>19</b>	<b>1</b>	2	18	<b>14</b>	<b>6</b>
20	0,28	0,38	<b>15</b>	<b>5</b>	<b>15</b>	<b>5</b>	<b>15</b>	<b>5</b>	<b>16</b>	<b>4</b>
21	0,30	0,38	<b>19</b>	<b>1</b>	<b>19</b>	<b>1</b>	<b>19</b>	<b>1</b>	<b>16</b>	<b>4</b>
22	0,32	0,38	<b>20</b>	<b>0</b>	<b>20</b>	<b>0</b>	<b>20</b>	<b>0</b>	<b>16</b>	<b>4</b>
23	0,34	0,38	<b>20</b>	<b>0</b>	<b>20</b>	<b>0</b>	<b>20</b>	<b>0</b>	<b>19</b>	<b>1</b>

O critério “menor variedade” parece ser aquele que produz o menor número de erros dentre os testados. Através dos diversos testes computacionais realizados não foram identificadas características gerais que permitissem predizer quando a enumeração no comprimento ou na largura é mais vantajosa.

Os comportamentos observados são muito variados parecendo ser fortemente dependentes dos dados do problema.

#### ***4.4 - Uso de limitantes para redução do problema da mochila na segunda fase***

No Capítulo anterior foi proposto o uso de padrões homogêneos para se determinar um limitante inferior inicial para o lucro por unidade de comprimento de uma faixa na geração de padrões tabuleiros exatos. Este valor, que é atualizado sempre que um padrão tabuleiro melhor é obtido, permite restringir a busca de faixas, descartando combinações de itens que resultem em lucros inferiores ao limitante já obtido.

Faixas candidatas potenciais a fornecer as melhores soluções globais são recuperadas para o problema, ou seja, combinações que possuam uma boa relação lucro por unidade de comprimento são recuperadas. Estas são fortes candidatas a gerar os melhores padrões tabuleiros. Apenas aquelas faixas que fornecem um lucro superior ao limitante corrente poderão resultar em um padrão tabuleiro mais lucrativo. Assim, percorre-se o conjunto de combinações dominantes em busca daquelas com uma relação lucro por unidade de comprimento melhor que o limitante corrente.

Recuperada uma melhor combinação com lucro superior ao do limitante corrente, passa-se para a segunda etapa da determinação de padrões tabuleiros onde se determina as demais faixas que combinadas com a solução na largura formam um padrão tabuleiro. O lucro associado a cada faixa determinada é calculado.

Um teste simples pode ser realizado para saber se a faixa determinada necessita ser considerada para compor o padrão tabuleiro. Admita que esta faixa estará contida no padrão tabuleiro (vide Figura 4.5). Com isso tem-se uma redução do tamanho disponível na chapa para acomodar as demais faixas do padrão. Considerando a melhor hipótese possível que seria o preenchimento completo do restante com a faixa de melhor lucratividade, pode-se estimar o lucro máximo que seria atingido. Se este lucro não ultrapassa o limitante corrente, a faixa definida certamente não deverá estar em um padrão tabuleiro com lucratividade maior do que o valor corrente e, portanto, pode ser descartada.

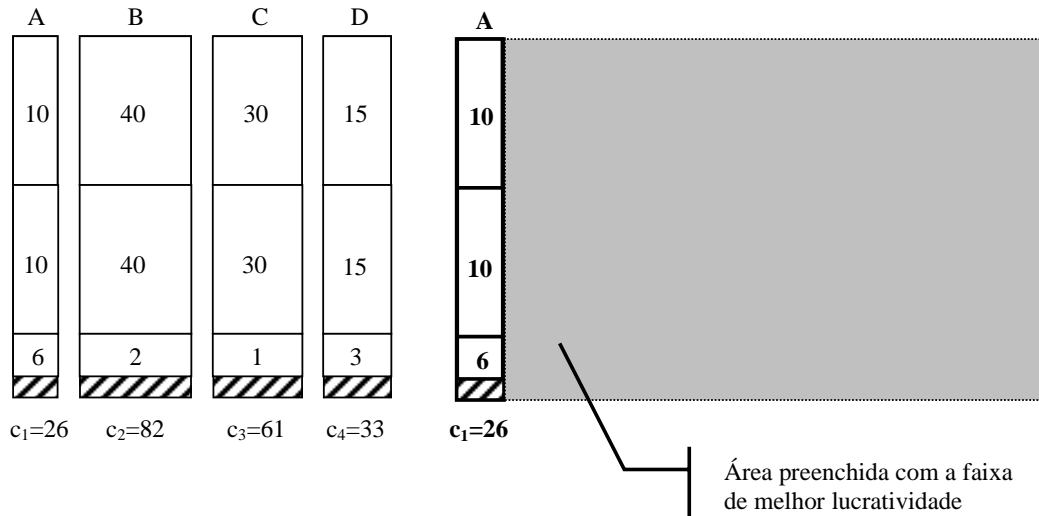


Fig. 4.5 – Faixa (em negrito) no padrão tabuleiro.

Com este teste simples, é possível reduzir-se o número de faixas a serem combinadas nesta segunda fase, ou seja, pode-se reduzir a dimensão do problema da mochila a ser resolvido.

Nas Tabelas 4.6 e 4.7 apresentam-se os resultados de experimentos computacionais que foram realizados para averiguar a eficácia deste teste considerando-se uma chapa de dimensões  $1251 \times 1587$  a ser cortada para produzir peças menores geradas aleatoriamente com dimensões variando de 15% a 63% das dimensões da chapa.

Embora o lucro associado às peças seja um fator determinante no número de soluções a serem processadas, eles foram gerados de forma aleatória variando no intervalo de 1 a 10 (inteiros positivos).

Na Tabela 4.6, estão os resultados de testes aplicados a problemas com número de peças variando-se de 10 a 100. Para cada número fixo de peças foram testados 20 exemplos. Considera-se que as peças não podem ser produzidas em quaisquer orientações.

O número de exemplos onde houveram e não houveram redução do tamanho do problema da mochila estão indicados na Tabela. Está indicado, também, o número de problemas compostos de uma única faixa não havendo, portanto, o que reduzir no problema da mochila.

TABELA 4.6 – REDUÇÃO NO PROBLEMA DA MOCHILA

<b>Exemplar (Nº peças)</b>	<b>Total de PM Resolvidos</b>	<b>PM compostos de uma única faixa</b>	<b>PM reduzidos</b>	<b>PM não reduzidos</b>
10	41	20	15	6
20	40	21	16	3
30	41	24	15	2
40	46	26	20	0
50	48	21	23	4
60	44	20	22	2
70	41	14	26	1
80	44	23	19	2
90	38	16	20	2
100	41	18	21	2

Da Tabela 4.6 pode-se observar que foram poucos os problemas da mochila onde não houve nenhuma redução. Observe também que uma boa parte dos problemas resolvidos eram compostos de uma única faixa, talvez, devido ao fato das dimensões das peças serem muito diferenciadas.

Desta forma foi realizado um experimento complementar onde as dimensões das peças eram pouco variadas. Foram aplicados basicamente os mesmos dados de entrada do



teste anterior (chapa de dimensões 1251×1587, peças geradas aleatoriamente com dimensões variando de 15% a 63% das dimensões da chapa). Entretanto, para a determinação das dimensões das peças, realizou-se a combinação entre os itens de um conjunto de  $n$  números inteiros gerados aleatoriamente. Por exemplo, dado  $n=3$  e os números aleatórios 2, 3, 4, obtém-se as 2×2, 2×3, 2×4, 3×2, 3×3, 3×4, 4×2, 4×3, 4×4. Desta forma, o número de itens do Problema da Mochila a ser resolvido na segunda fase corresponde a exatamente  $n$ .

Na Tabela 4.7 apresenta-se o resultado obtido. A coluna 1 daquela Tabela apresenta o número de itens do PM para cada exemplar (20 exemplos gerados aleatoriamente).

TABELA 4.7 – REDUÇÃO NO PROBLEMA DA MOCHILA (TESTE COM PEÇAS DE DIMENSÕES POUCO VARIADAS)

<b>Nº de itens do PM</b>	<b>Total de PM Resolvidos</b>	<b>PM compostos de uma única faixa</b>	<b>PM reduzidos</b>	<b>PM não reduzidos</b>
10	51	2	48	1
11	51	9	42	0
12	53	1	52	0
13	54	5	49	0
14	72	1	71	0
15	50	3	47	0
16	63	1	61	1
17	48	6	42	0
18	60	0	60	0
19	97	4	93	0
20	74	0	74	0
21	57	0	57	0
22	96	2	94	0
23	106	0	106	0
24	54	0	54	0
25	61	0	61	0
26	94	3	91	0
27	145	5	140	0
28	93	3	90	0

Pela Tabela 4.7 observa-se que a inclusão de peças de dimensões pouco variadas fez com que houvesse uma redução no número de Problemas da Mochila compostos de uma única faixa e ocorreu um aumento no número de Problemas da Mochila que puderam ser reduzidos.

Os resultados dos testes computacionais realizados podem explicar também a razão pela qual a heurística de Morabito e Arenales (2000) para determinação de padrões tabuleiros, demonstrou ser bastante eficiente para os casos extraídos da indústria de móveis encontrando soluções ótimas na maioria dos casos. A grande maioria das soluções é composta de apenas uma única faixa repetida.

Os problemas unidimensionais da mochila resolvidos na segunda fase do algoritmo enumerativo apresentaram sempre dimensões relativamente pequenas. O número  $n$  de itens variou de 1 a 5, sendo que na sua grande maioria eram constituídos de apenas 1 item. Para resolver estes problemas, um algoritmo de programação dinâmica ou um método de enumeração implícita tal como o apresentado por Gilmore e Gomory (1963) são bastante adequados. Por esta razão não houve uma preocupação, nesta dissertação pela procura e aplicação de métodos mais eficientes.

#### ***4.5 - Recuperação imediata ou postergada de combinações alternativas***

Na análise de faixas da primeira fase, uma combinação recuperada melhor que o limitante corrente não assegura que o padrão tabuleiro correspondente gerado seja melhor que o corrente. O padrão precisa ser recuperado para se verificar se o seu lucro é, de fato, melhor que o encontrado até o momento.

Na recuperação de uma combinação potencialmente melhor que o limitante corrente também são identificadas combinações alternativas de valores potencialmente melhores que o limitante corrente. Combinações alternativas são recuperadas e avaliadas sendo armazenadas caso gerem soluções melhores que a corrente.

Uma modificação no algoritmo enumerativo poderia postergar a recuperação do padrão correspondente às combinações alternativas guardando-as na lista de combinações com valores maiores do que o limitante corrente. Isto porque podem existir muitas outras combinações potencialmente melhores do que elas e que podem gerar padrões melhores eliminando, portanto a necessidade de examiná-las.

O algoritmo pode recuperar o padrão tabuleiro apenas no momento oportuno e isto pode resultar em algum ganho no tempo computacional.

Testes computacionais foram realizados para estimar se o uso deste procedimento é realmente vantajoso. Os resultados mostraram que não houve diferenças significativas de tempo computacional entre os dois procedimentos.

Aparentemente, o tempo computacional consumido pelas rotinas de inserção e remoção de elementos da lista ordenada de combinações potenciais é da ordem de grandeza do tempo economizado na geração dos padrões. Cabe observar que o algoritmo implementado para inserção ordenada de itens na lista consome um tempo computacional de ordem  $n$ . Um algoritmo mais eficiente poderia ser implementado e testes computacionais poderiam ser realizados, mas isto fica como sugestão para trabalhos futuros.



## **CAPÍTULO 5**

### **FUNÇÃO CUSTO ASSOCIADA À PERDA DE MATERIAL E AO TEMPO DE CORTE DA CHAPA**

Apesar dos padrões tabuleiros acelerarem a produção, eles podem resultar em grande desperdício de material. Conseqüentemente, existe um “balanço” entre cortar padrões tabuleiros que melhoram a produtividade da máquina, mas que resultam em maiores desperdícios de matéria-prima ou cortar padrões com menores perdas, mas que reduzem a produtividade da serra.

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, se o padrão não for tabuleiro então o seu custo associado é  $C + \delta$ , caso contrário, o seu custo é simplesmente  $C$ .

Sugere-se a seguir uma função com custo variável, associado ao tempo extra de corte da chapa utilizando padrões 2-estágios não exatos e padrões tabuleiros gerados pelo algoritmo apresentado no Capítulo 3.

Admite-se que no ambiente de corte de chapas de madeira focalizado, os cortes são efetuados por uma máquina automática composta de uma única serra circular, capaz de produzir cortes longitudinais e transversais sobre a superfície da chapa. No caso de padrões 2-estágios, o corte é feito em 2 etapas.

Na primeira etapa (vide Figura 5.1) a serra realiza cortes paralelos à lateral da chapa na direção longitudinal (comprimento).

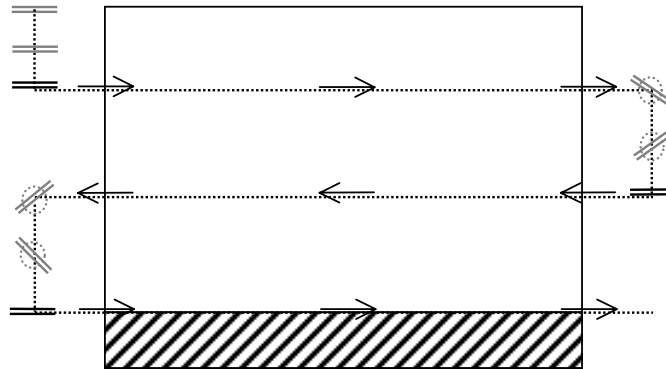


Fig. 5.1 – Corte longitudinal.

Na segunda etapa (vide Figura 5.2), se o padrão a ser cortado corresponde a um padrão do tipo tabuleiro, a serra é rotacionada em 90 graus e em seguida a máquina efetua cortes na direção transversal da chapa produzindo o padrão desejado.

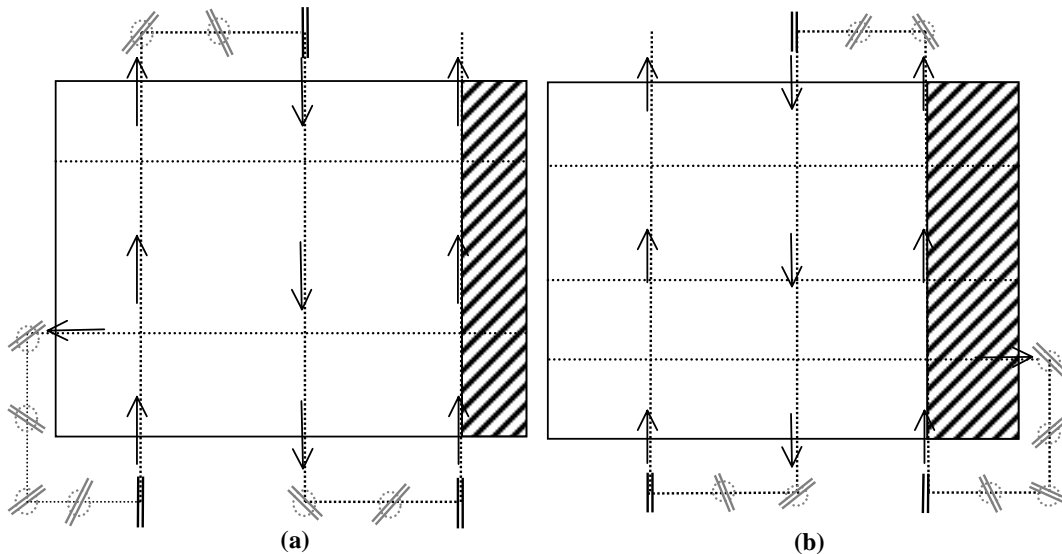


Fig. 5.2 – Corte transversal: (a) posição inicial da serra à esquerda da chapa, (b) posição inicial da serra à direita da chapa.

Caso o padrão não corresponda a um padrão tabuleiro, as faixas obtidas na primeira etapa são separadas em grupos de faixas e o processo de corte baseado em padrões

tabuleiros é repetido para cada um dos grupos formados. Na Figura 5.3 ilustra-se o processo de corte para padrões não tabuleiros do tipo 2-estágios exato:

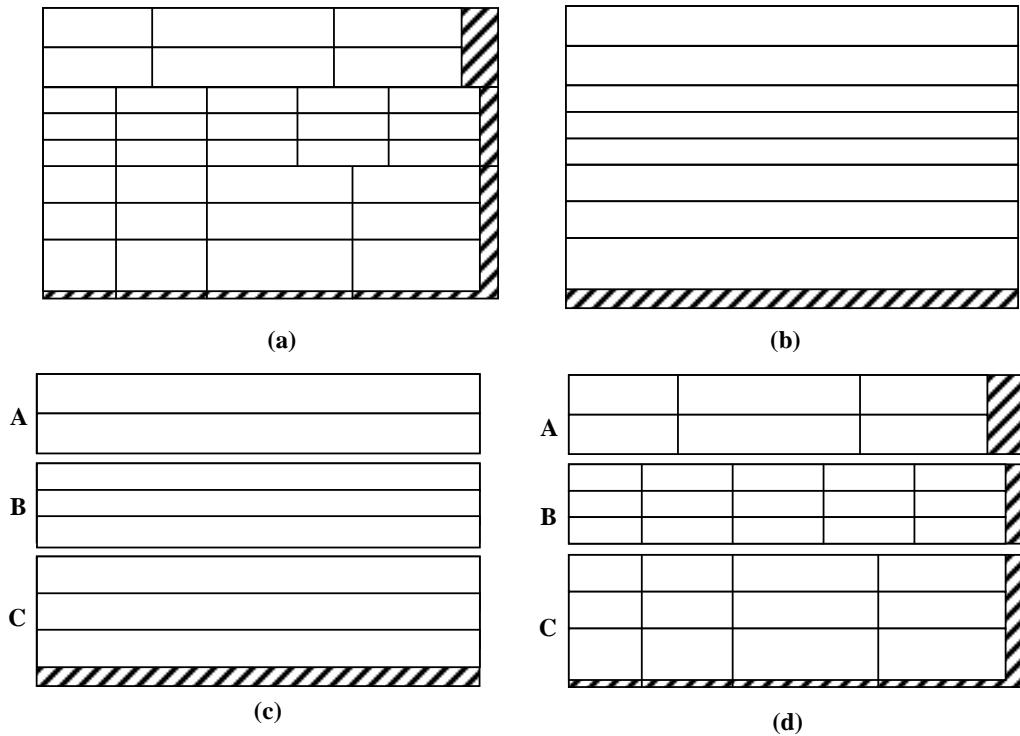


Fig. 5.3 - Processo de corte de padrões 2-estágios exato: (a) padrão, (b) corte na primeira etapa, (c) faixas são agrupadas, (d) grupos de faixas são cortados separadamente.

Uma função associada ao custo do corte deve levar em consideração várias peculiaridades existentes na prática como, por exemplo, o tempo para rotação da serra ou a velocidade de corte do equipamento nas direções longitudinal e transversal.

Admita, por exemplo, que o corte longitudinal inicia-se no topo esquerdo da chapa. Após realizados os cortes longitudinais, o corte transversal pode iniciar-se da esquerda para a direita ou direita para a esquerda da chapa, dependendo da posição da serra após o corte da última faixa longitudinal. Se a serra estiver posicionada no canto inferior

esquerdo, então o sentido do corte será da esquerda para a direita, caso contrário, o corte será efetuado da direita para a esquerda.

O custo do tempo consumido pelos cortes efetuados nas direções longitudinais e transversais da chapa, no caso de padrões tabuleiros, são proporcionais respectivamente, a  $L/v_1$ , onde  $L$  é a distância a ser percorrida pela serra na direção longitudinal (correspondente ao comprimento da chapa) e  $v_1$  é a velocidade da serra na direção longitudinal e  $W/v_2$ , onde  $W$  é a distância a ser percorrida pela serra na direção transversal (correspondente à largura da chapa) e  $v_2$  é a velocidade da serra na direção transversal.

O custo deve também levar em consideração o tempo para a rotação e deslocamento da serra ao ponto inicial do corte de cada faixa. Admitindo-se que a serra é rotacionada ao mesmo tempo em que se desloca para o ponto inicial de corte, este tempo é dado, no corte longitudinal por  $\max\left(\frac{w_k}{v_2}, \gamma\right)$ , onde  $\frac{w_k}{v_2}$  representa o tempo necessário para a serra percorrer verticalmente uma distância equivalente à largura  $w_k$  da  $k$ -ésima faixa e  $\gamma$  é o tempo para a serra ser rotacionada em 180 graus. De forma similar, o tempo associado ao deslocamento e rotação da serra no corte transversal é dado pelo  $\max\left(\frac{l_k}{v_1}, \gamma\right)$ , onde  $\frac{l_k}{v_1}$  representa o tempo necessário para a serra deslocar-se horizontalmente de uma distância  $l_k$  até o ponto de corte de cada faixa. A distância  $l_k$  percorrida é o comprimento da faixa  $k$ .

O tempo de um corte longitudinal ao seguinte tem uma duração de  $\max\left(\frac{w_k}{v_2}, \gamma\right) + \frac{L}{v_1}$ .

Se o tempo  $\gamma$  para a rotação da serra for superior ao tempo de deslocamento da serra até o ponto do início do próximo corte, então todos os cortes longitudinais duram o mesmo tempo.



Mesmo que  $\gamma$  não seja um termo dominante, se as faixas não tiverem larguras muito diferenciadas então todos os cortes longitudinais duram praticamente o mesmo tempo.

Admitindo-se que os cortes longitudinais duram praticamente o mesmo tempo, se  $s$  é o total de faixas cortadas na largura, o custo do corte das faixas obtidas na primeira etapa pode ser simplificado para:

$$\rho(s - 1),$$

onde  $\rho$  é o custo médio para um corte longitudinal.

De forma similar, pode-se simplificar o cálculo do custo do corte na segunda etapa (cortes transversais). Seja  $\sigma$ , o custo médio associado ao corte transversal em cada faixa (ou grupo de faixas) para produzir os itens requeridos. Se  $h_i$  é o total de cortes transversais na faixa  $i$  então o custo dos cortes transversais desta faixa é dado por:

$$\sigma(h_i)$$

e o custo total para o corte do padrão é dado por:

$$c(\rho, \sigma) = \rho(s - 1) + \sigma \left( \sum_{i=1}^s h_i \right).$$

O valor de  $\sigma$  pode variar ligeiramente conforme a faixa sendo cortada mas não deve flutuar muito entre as faixas se estas tiverem larguras de dimensões aproximadas. No exemplo ilustrado na Figura 5.3, o custo de corte é dado por  $c(\rho, \sigma) = 8\rho + 12\sigma$ .

Na Figura 5.4 ilustra-se os custos incorridos devido aos cortes efetuados:

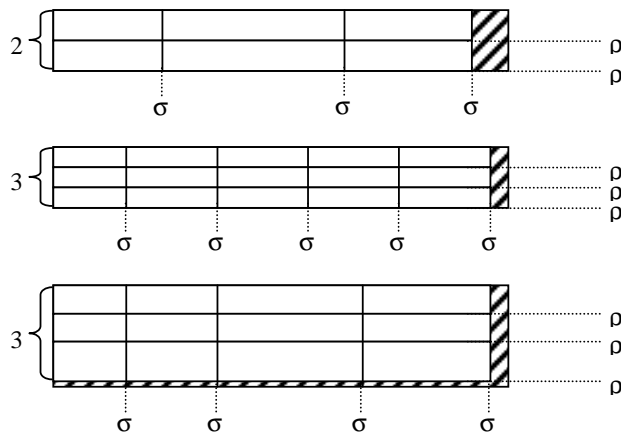


Fig. 5.4 – Custos para os cortes do padrão exemplificado na Figura 5.3.

No caso de padrões tabuleiros os cortes transversais são feitos para um único grupo de faixas. Desta forma, a função custo associada ao corte de padrões tabuleiros é dada por:

$$t(\rho, \sigma) = \rho(s - 1) + \eta(h),$$

onde  $\eta$  é o custo médio de um corte transversal do objeto.

No caso de padrões 2-estágios não exatos tem-se ainda operações extras da máquina de corte devido à presença de refilos. Na Figura 5.5 ilustra-se o corte de um padrão 2-estágios não exato.

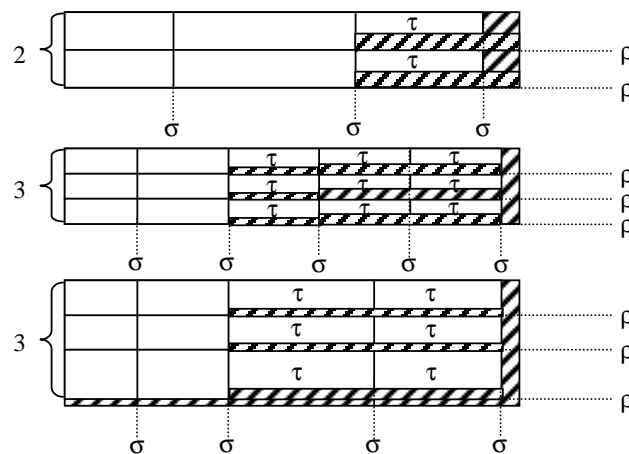


Fig. 5.5 – Corte de um padrão do tipo 2-estágios não exato.

Observe que este tipo de padrão tem um custo adicional do corte das peças com refilos. Este custo adicional é dado por  $\tau(v)$ , onde  $v$  é o número de peças com refilos e  $\tau$  é o custo médio associado a este corte. Definidas estas funções custo associadas ao tempo de corte de padrões tabuleiros e não tabuleiros, a utilização delas no método Simplex com geração de colunas é difícil, pois a função objetivo do subproblema para a geração do padrão a entrar na base passa a depender do padrão de corte a ser gerado, tornando sua resolução muito complexa.

Para contornar esta dificuldade, pode-se resolver o subproblema de geração maximizando o “lucro” dos itens cortados no padrão e apenas posteriormente o custo referente ao corte do padrão é considerado para verificar se o padrão deve ou não entrar na base.

O problema relaxado é, portanto resolvido de maneira subótima. Alguns testes computacionais realizados adotando este procedimento foram realizados e os resultados obtidos estão apresentados no próximo Capítulo.

Embora a função apresentada seja simplificada ela tenta incorporar a essência dos custos associados ao corte do padrão e acredita-se que ela reflita com maior fidedignidade a situação prática do que aquela de custo fixo sugerida por Morabito e Arenales (2000).

Os detalhes da implementação da função custo no método Simplex com geração de colunas podem ser consultados no Apêndice C.



## **CAPÍTULO 6**

### **TESTES COMPUTACIONAIS**

Neste Capítulo apresentam-se os resultados de alguns testes computacionais realizados com a função custo apresentada no Capítulo 5.

Um microcomputador Pentium III de 450 MHz com 128 Mb de memória RAM com o sistema operacional Linux foi utilizado para a implementação da função custo e dos algoritmos para determinação dos padrões de corte.

As implementações foram feitas em linguagem orientada a objetos utilizando o compilador GNU C++. Posteriormente, todo o código fonte foi portado para a plataforma RISC em uma estação de trabalho Sun Ultra-1 200 Mhz com 256 Mb de memória RAM. Na estação de trabalho, o método Simplex com geração de colunas foi implementado com o suporte da biblioteca CPLEX versão 6.5. Esta biblioteca forneceu grande parte dos recursos de programação para resolver o modelo de programação linear proposto. Portanto, duas diferentes plataformas foram utilizadas para a realização dos testes computacionais.

Para efeito de comparação dos resultados computacionais com a literatura, os dados utilizados na realização dos testes foram os mesmos utilizados por Morabito e Arenales (2000). Os dados utilizados nos testes computacionais são apresentados a seguir.

#### Dados de entrada:

Chapas retangulares de dimensões 1850×3670 mm que devem ser cortadas para produzir todos os itens da seguinte Tabela:

TABELA 6.1 – ITENS A SEREM PRODUZIDOS

$i$	$w_i$	$l_i$	$d_i$
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).

Admite-se que os itens podem ser produzidos em quaisquer orientações e considera-se uma espessura de 4 mm da lâmina de corte que acarreta desperdícios na largura e comprimento da chapa.

As chapas possuem um custo unitário de \$1 e considera-se que existe um número suficiente de objetos em estoque para produzir todos os itens.

De forma similar a Morabito e Arenales (2000), utiliza-se a solução relaxada do problema para comparação para evitar variações locais devido ao arredondamento das variáveis.

Critério de parada (cutoff):

O critério de parada foi o mesmo estabelecido por Gilmore e Gomory (1963) para interromper o método Simplex, ou seja, parar se após algumas iterações, o valor da função objetivo não for significativamente melhorado.

Nos testes computacionais realizados, a execução do algoritmo foi interrompida se nas últimas 10 iterações, o valor da função objetivo não foi reduzido em pelo menos 0.1%.

#### Erros em ponto flutuante:

Considera-se uma tolerância de  $\pm 10^{-6}$  para erros em ponto flutuante.

Na análise do trade-off padrões tabuleiros  $\times$  padrões não tabuleiros sugerida por Morabito e Arenales (2000), os autores consideraram padrões exatos e não exatos, i.e., quaisquer itens podem ser selecionados desde que a largura (ou comprimento) das peças seja inferior ou igual a cada uma das faixas de largura (ou comprimento) a serem cortadas. Também os padrões não tabuleiros considerados eram do tipo 2-estágios exatos e não exatos.

Conforme pode ser observado no Capítulo 3, o algoritmo enumerativo para determinação de padrões tabuleiros descrito no presente trabalho foi desenvolvido para determinar apenas padrões tabuleiros exatos, ou seja, os itens devem ser selecionados de forma que o padrão seja composto de peças de mesma largura e/ou comprimento.

A heurística desenvolvida por Morabito e Arenales (2000) foi adaptada para a determinação somente de padrões tabuleiros exatos de modo a poder compará-la com os padrões tabuleiros exatos gerados pelo algoritmo enumerativo apresentado no Capítulo 3.

Na Tabela 6.2 apresenta-se a análise do “balanço” entre padrões tabuleiros não exatos  $\times$  padrões 2-estágios não exatos realizada por Morabito e Arenales (2000) e também a análise do “balanço” no caso de padrões tabuleiros exatos  $\times$  padrões 2-estágios não exatos usando-se aquela heurística adaptada para o caso exato.

TABELA 6.2 – ANÁLISE DO “BALANÇO” DE PADRÕES TABULEIROS NÃO EXATOS × PADRÕES 2-ESTÁGIOS NÃO EXATOS E PADRÕES TABULEIROS EXATOS × PADRÕES 2-ESTÁGIOS NÃO EXATOS

Custo Fixo	Custo Mínimo Total		Total de Chapas Cortadas		Padrões não Tabuleiros (%)		Desperdício (%)	
	tabuleiro não exato	tabuleiro exato	tabuleiro não exato	tabuleiro exato	tabuleiro não exato	tabuleiro exato	tabuleiro não exato	tabuleiro exato
	x 2-estágios não exato	x 2-estágios não exato	x 2-estágios não exato	x 2-estágios não exato	x 2-estágios não exato	x 2-estágios não exato	x 2-estágios não exato	x 2-estágios não exato
<b>0,00</b>	348,7	348,7	348,7	348,7	93,3	93,3	2,67	2,67
<b>0,01</b>	351,7	351,7	348,8	348,8	83,7	83,7	2,70	2,70
<b>0,02</b>	354,6	354,6	349,1	348,9	77,8	82,5	2,79	2,72
<b>0,03</b>	357,2	357,5	349,9	348,9	69,5	82,4	3,00	2,72
<b>0,05</b>	361,3	363,2	353,5	349,1	43,8	80,8	4,01	2,77
<b>0,10</b>	363,6	376,8	362,0	351,5	4,4	71,8	6,25	3,46
<b>0,15</b>	364,3	386,0	364,3	365,9	0,0	36,5	6,84	7,26
<b>0,18</b>	364,3	389,8	364,3	367,5	0,0	33,8	6,84	7,65
<b>0,20</b>	364,3	391,5	364,3	388,3	0,0	4,1	6,84	12,59
<b>0,22</b>	364,3	391,8	364,3	388,3	0,0	4,1	6,84	12,59
<b>0,24</b>	364,3	392,1	364,3	388,3	0,0	4,1	6,84	12,59
<b>0,26</b>	364,3	392,4	364,3	388,3	0,0	4,1	6,84	12,59
<b>0,28</b>	364,3	392,7	364,3	388,3	0,0	4,1	6,84	12,59
<b>0,30</b>	364,3	393,1	364,3	388,3	0,0	4,1	6,84	12,59
<b>0,32</b>	364,3	393,4	364,3	388,3	0,0	4,1	6,84	12,59
<b>0,34</b>	364,3	393,7	364,3	388,3	0,0	4,1	6,84	12,59
<b>0,36</b>	364,3	394,0	364,3	388,3	0,0	4,1	6,84	12,59
<b>0,38</b>	364,3	394,3	364,3	388,3	0,0	4,1	6,84	12,59
<b>0,40</b>	364,3	394,7	364,3	388,3	0,0	4,1	6,84	12,59
<b>0,42</b>	364,3	395,0	364,3	388,3	0,0	4,1	6,84	12,59
<b>0,44</b>	364,3	395,3	364,3	388,3	0,0	4,1	6,84	12,59
<b>0,46</b>	364,3	395,6	364,3	388,3	0,0	4,1	6,84	12,59
<b>0,48</b>	364,3	395,9	364,3	388,3	0,0	4,1	6,84	12,59
<b>0,50</b>	364,3	396,3	364,3	388,3	0,0	4,1	6,84	12,59
<b>0,52</b>	364,3	396,5	364,3	388,3	0,0	4,1	6,84	12,59
<b>0,54</b>	364,3	396,6	364,3	396,6	0,0	0,0	6,84	14,44

É importante notar que em ambas as análises, à medida que o custo fixo é incrementado, a proporção de padrões não tabuleiros tende a diminuir e a percentagem de desperdício aumenta. O aumento do desperdício ocorre devido ao crescimento da proporção de padrões tabuleiros que possuem maiores perdas de material comparados aos padrões 2-estágios gerais. O menor número de itens por padrão de corte decorrente do uso de padrões tabuleiros, implica em um aumento do número de chapas cortadas e, conseqüentemente, um aumento no valor de custo mínimo.



Das colunas da Tabela 6.2 observa-se que para o custo fixo variando de 0.20 a 0.52, o total de placas cortadas, a proporção de padrões não tabuleiros e a porcentagem de desperdício permanecem constantes, ao mesmo tempo em que o valor do custo mínimo aumenta. Isto ocorre porque padrões tabuleiros exatos não apresentam soluções melhores para o problema por serem muito restritos. Assim, a inclusão de mais padrões tabuleiros só é vantajosa quando o valor do custo fixo cresce significativamente. Testes computacionais para analisar o “balanço” entre padrões tabuleiros exatos e 2-estágios exatos também foram realizados e os resultados podem ser analisados no Apêndice D.

Testes computacionais realizados com o algoritmo enumerativo exato descrito no Capítulo 3 apresentaram os mesmos resultados que os da heurística de Morabito e Arenales (2000) para determinação de padrões tabuleiros exatos. Isto ocorre devido ao fato dos dados utilizados apresentarem peças de dimensões muito diferenciadas, causando a geração de padrões tabuleiros formados por peças do mesmo tipo (solução homogênea) ou com repetição de uma mesma faixa. A heurística sugerida por Morabito e Arenales, portanto é bastante eficiente neste caso, pois tem como principal característica a repetição das faixas do mesmo tipo (eventualmente compostas de peças do mesmo tipo) para a determinação do padrão tabuleiro exato.

Na Tabela 6.2, observa-se que a proporção de padrões tabuleiros atinge 100% para um custo fixo de 0.15; no caso de padrões tabuleiros exatos, o valor do custo fixo para que esta mesma proporção seja atingida é de 0.54. É importante ressaltar que os padrões tabuleiros utilizados por Morabito e Arenales (2000) podem requerer recortes devido a presença de refilos e o custo do corte deste tipo de padrão é certamente maior comparado com o custo do corte de padrões tabuleiros exatos o que não foi considerado em seu trabalho. Possivelmente os autores consideraram que o corte destes refilos seja realizado manualmente e consideraram apenas as operações efetuadas na serra automática, responsável pelo corte do padrão, sem o recorte dos refilos. Uma outra razão se deve ao fato destes autores terem feito sua comparação com padrões 2-estágios não exatos.

Considere a nova função custo descrita no Capítulo 5, associada ao tempo de corte dos padrões. Seja  $\eta$ , o custo adicional associado ao corte transversal em padrões tabuleiros e seja  $\sigma$ , o custo associado ao corte transversal dos itens das faixas geradas por padrões 2-estágios. Testes computacionais foram realizados variando-se os valores de  $\eta$  e  $\sigma$  e os resultados obtidos estão apresentados na Tabela 6.3. Embora exista um custo associado ao corte longitudinal em padrões tabuleiros e não tabuleiros, este foi fixado em 0. Isto não deve ter influência nos resultados computacionais uma vez que os custos associados ao corte longitudinal devem ter, em média, os mesmos valores para ambos os tipos de padrões.

TABELA 6.3 – INFLUÊNCIA NO “BALANÇO” DE PADRÕES TABULEIROS EXATOS  $\times$  PADRÕES 2-ESTÁGIOS EXATOS COM VARIAÇÕES NOS VALORES DOS PARÂMETROS DA FUNÇÃO CUSTO

$\eta$	$\sigma$	Custo Mínimo	Placas Cortadas	Proporção de Padrões não Tabuleiros (%)	Desperdício (%)
0,00	0,000	349,603	349,603	100	2,92278
0,01	0,000	349,603	349,603	100	2,92278
0,01	0,002	366,79	353,189	100	3,90833
0,01	0,004	380,904	356,117	80,0905	4,69837
0,01	0,006	393,593	372,244	42,6014	8,82716
0,01	0,008	399,307	376,241	34,6473	9,79572
0,01	0,010	404,401	381,365	20,7151	11,0077
0,02	0,000	349,603	349,603	100	2,92278
0,02	0,002	366,118	354,717	100	4,32219
0,02	0,004	376,405	355,833	100	4,62232
0,02	0,006	391,925	362,169	71,5003	6,29084
0,02	0,008	406,948	373,665	42,6998	9,17403
0,02	0,010	410,801	373,7	42,6182	9,18236
0,02	0,012	415,437	375,663	39,1537	9,65692
0,02	0,014	422,122	381,365	20,7151	11,0077
0,02	0,016	428,034	388,565	4,11772	12,6567
0,02	0,018	428,546	388,565	4,11772	12,6567
0,02	0,020	429,058	388,565	4,11772	12,6567
0,03	0,000	349,603	349,603	100	2,92278
0,03	0,002	368,354	357,283	100	5,00942
0,03	0,004	376,438	356,275	100	4,74079

(continua)

TABELA 6.3 – CONTINUAÇÃO

0,03	0,006	390,427	361,491	98,8163	6,11511
0,03	0,008	405,537	365,887	76,8715	7,24326
0,03	0,010	416,189	360,995	79,6453	5,98616
0,03	0,012	423,589	373,665	42,6998	9,17403
0,03	0,014	428,718	375,051	39,97	9,50949
0,03	0,016	432,795	376,539	36,7938	9,86714
0,03	0,018	444,685	387,927	8,36068	12,5131
0,03	0,020	446,745	388,565	4,11772	12,6567
0,03	0,022	447,257	388,565	4,11772	12,6567
0,03	0,024	447,769	388,565	4,11772	12,6567
0,03	0,026	448,281	388,565	4,11772	12,6567
0,03	0,028	448,793	388,565	4,11772	12,6567
0,03	0,030	449,305	388,565	4,11772	12,6567
0,04	0,000	349,603	349,603	100	2,92278
0,04	0,002	365,677	353,703	100	4,04786
0,04	0,004	379,763	358,456	100	5,32013
0,04	0,006	389,797	361,094	100	6,01192
0,04	0,008	396,344	363,69	100	6,68279
0,04	0,010	413,592	372,353	76,9242	8,85398
0,04	0,012	423,276	365,771	71,4751	7,21383
0,04	0,014	436,306	373,848	44,5528	9,21836
0,04	0,016	440,362	374,071	42,8616	9,2725
0,04	0,018	445,892	374,338	40,6407	9,33732
0,04	0,020	449,472	376,539	36,7938	9,86714
0,04	0,022	462,731	387,927	8,36068	12,5131
0,04	0,024	467,89	395,314	4,04742	9,07916
0,04	0,026	468,402	395,314	4,04742	9,07916
0,04	0,028	468,914	395,314	4,04742	9,07916
0,04	0,030	469,426	395,314	4,04742	9,07916
0,04	0,032	469,938	395,314	4,04742	9,07916
0,04	0,034	470,45	395,314	4,04742	9,07916
0,08	0,000	349,603	349,603	100	2,92278
0,08	0,002	366,089	353,49	100	3,9902
0,08	0,004	379,02	355,324	100	4,48572
0,08	0,006	389,736	355,025	100	4,40535
0,08	0,008	397,748	368,473	100	7,89423
0,08	0,010	406,164	372,108	100	8,7939
0,08	0,012	410,639	373,461	100	9,12426
0,08	0,014	416,11	373,819	100	9,21133
0,08	0,016	421,66	374,806	100	9,45032
0,08	0,018	427,522	372,486	100	8,8865

(continua)

TABELA 6.3 – CONCLUSÃO.

0,08	0,020	452,73	372,076	80,5549	8,78614
0,08	0,022	486,405	367,546	76,5551	7,66177
0,08	0,024	491,114	373,848	44,5528	9,21836
0,08	0,026	496,909	373,686	42,2194	9,17901
0,08	0,028	500,619	373,967	41,7488	9,24734
0,08	0,030	506,835	376,539	36,7938	9,86714
0,08	0,032	509,949	376,539	36,7938	9,86714
0,08	0,034	519,308	382,896	24,9241	9,11072

Como esperado o valor relativo de  $\eta$  em comparação a  $\sigma$  é que influencia a quantidade de padrões tabuleiros na solução. A medida que se incrementa o valor de  $\eta$ , a proporção de padrões não tabuleiros tende a atingir 100 % para quaisquer valores de  $\sigma$  testados.

Testes computacionais também foram realizados com padrões com refilos (padrões tabuleiros não exatos e padrões 2-estágios não exatos). Os testes foram realizados para comparar as soluções obtidas com a nova função custo proposta e a função com custo fixo proposto por Morabito e Arenales (2000). Para poder comparar as soluções obtidas, adotou-se o seguinte procedimento:

Passo 1: Encontre as soluções aplicando a função custo desenvolvida para os valores de  $\rho, \eta, \sigma, \tau$ .

Passo 2: Encontre um valor médio para o custo fixo correspondente às soluções obtidas no Passo 1. O valor do custo fixo pode ser calculado com base na diferença entre o valor médio da função custo associada à padrões não tabuleiros (2-estágios exatos ou não exatos) e o valor médio da função custo associada à padrões tabuleiros (exatos ou não exatos).

Passo 3: Encontre a solução aplicando o custo fixo calculado no Passo 2.

Passo 4: Encontre o valor da solução obtida no Passo 3, utilizando a função custo desenvolvida.

Na Tabela 6.4 apresenta-se os resultados obtidos fixando-se  $\rho = 0,002$ ,  $\eta = 0,0009$ ,  $\sigma = 0,0006$  e variando-se o custo  $\tau$  associado ao tempo de corte dos refilos.

TABELA 6.4 – COMPARAÇÃO ENTRE A FUNÇÃO CUSTO E O CUSTO FIXO DE MORABITO E ARENALES (2000)

$\tau$	Função custo	$\delta$	Custo fixo	Função custo associada à solução de custo fixo
0,0000	358,00	0,0102754	351,812	357,74
0,0001	360,21	0,0112064	352,084	360,14
0,0002	362,71	0,0139009	352,87	362,37
0,0003	365,43	0,0173976	353,863	365,02
0,0004	367,25	0,0147703	353,124	367,11
0,0005	369,41	0,0166329	353,649	369,66
0,0008	374,19	0,0215869	355,005	376,35
0,0010	378,19	0,0215420	354,993	381,00
0,0012	380,61	0,0149921	353,188	384,98
0,0014	384,66	0,0142598	352,975	391,20
0,0016	386,61	0,0092490	351,513	395,52
0,0018	387,70	0,0154000	353,304	398,81

Era de se esperar que os valores das soluções obtidas com o uso da função custo desenvolvida fossem sempre melhores que os valores das soluções obtidas com a função de custo fixo, sugerida por Morabito e Arenales (2000). No entanto, conforme pode ser observado na Figura 6.1, que ilustra visualmente os dados da Tabela 6.4, não foi exatamente isto o que aconteceu:

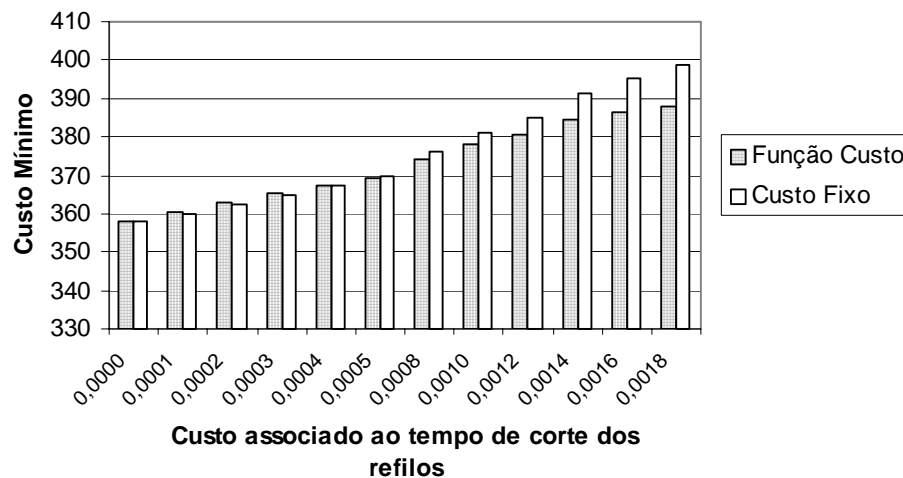


Fig. 6.1 – Comparação das soluções obtidas com o uso da função custo e com o uso do custo fixo proposto por Morabito e Arenales (2000).

Note que para  $0,0000 \leq \tau \leq 0,0004$  as soluções com o uso da função custo foram ligeiramente superiores aos valores correspondentes utilizando o custo fixo. Isto ocorre porque no subproblema para a geração de coluna não se utiliza diretamente a função custo desenvolvida tendo apenas o custo associado ao tempo de corte adicionado ao custo da chapa após obtido o padrão de retorno máximo. Portanto, o subproblema não é resolvido de maneira ótima e, conseqüentemente não há garantia de que o problema relaxado de corte está sendo resolvido também de maneira ótima. No caso do custo associado ao tempo de corte ser fixo, este procedimento funciona muito bem, pois os valores adicionados aos custos dos objetos independem das particularidades envolvidas no corte de cada padrão (corte longitudinal, transversal, dos refilos, etc). No caso da utilização da nova função custo, a função é não linear (varia de acordo com o padrão). A nova função custo, entretanto apresenta soluções melhores que as obtidas com custo fixo no caso quando o custo de corte associado aos refilos torna-se maior, pelo menos pelos resultados observados no exemplo implementado. Para conclusões mais definitivas seria necessário um volume maior de testes com muito mais exemplares.

## **CAPÍTULO 7**

### **CONCLUSÕES**

No trabalho desenvolvido, destacam-se as seguintes contribuições:

- Um algoritmo enumerativo exato para determinação de padrões tabuleiros sem perdas internas;
- Proposição de uma função custo que considera também o custo associado ao tempo utilizado pela máquina para o corte do padrão.

Como observação e conclusões, pode-se observar que o algoritmo para determinação de padrões tabuleiros exatos apresenta limitações para o seu uso, pois necessita de recursos computacionais de memória proporcionais aos valores dos dados do problema.

Entretanto para uma boa parte de problemas práticos com a introdução dos melhoramentos sugeridos, seu uso parece ser viável com os recursos computacionais atualmente existentes.

Testes computacionais aplicados em dados reais comprovaram a viabilidade da aplicação prática do algoritmo gerando resultados em centésimos de segundo (vide Apêndice A).

O algoritmo desenvolvido também não considera o caso de perdas internas. Padrões tabuleiros com “perdas internas” podem fornecer retorno maior que os padrões tabuleiros exatos, pois um número maior de faixas podem ser combinadas para a sua formação.

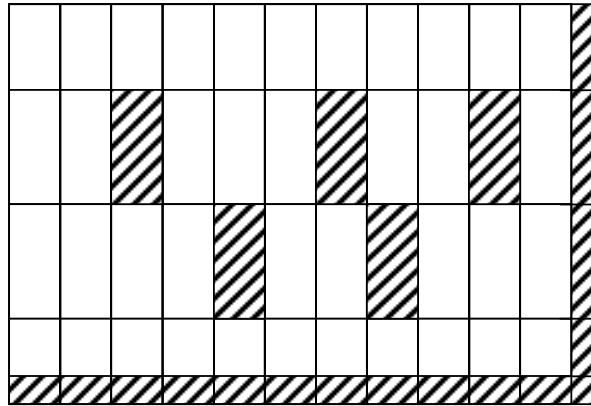


Fig. 7.1 - Padrão tabuleiro com perdas internas.

Utilizando-se alguns artifícios, pode-se determinar padrões tabuleiros com “perdas internas” do tipo esquematizado na Figura 7.1 utilizando-se o algoritmo enumerativo descrito no Capítulo 3. São introduzidos itens fictícios de larguras e/ou comprimentos dos itens existentes no problema, em lucros correspondentes iguais a zero.

O algoritmo foi desenvolvido para resolver problemas irrestritos, sem levar em consideração restrições de demanda de peças e chapas. Para se adaptar o algoritmo para o caso de padrões tabuleiros restritos é preciso resolver problemas da mochila restritos na primeira e segunda fases do algoritmo. A seleção de combinações dominantes descrita no Capítulo 4 não poderá ser usada por este tipo de padrão, pois os desperdícios são inevitáveis devido às restrições dos itens.

Testes computacionais foram realizados para se comparar o algoritmo enumerativo para determinação de padrões tabuleiros exatos com a heurística proposta por Morabito e Arenales (2000). Nos testes realizados, o algoritmo enumerativo obteve sempre a mesma solução que a apresentada pela heurística de Morabito e Arenales. Como as dimensões das peças no exemplo utilizado eram pouco diferenciadas e o algoritmo foi desenvolvido para determinar somente padrões tabuleiros do tipo exato, as soluções encontradas eram quase formadas por peças ou faixas do mesmo tipo. Neste caso, a



heurística proposta por Morabito e Arenales, por ser baseada na repetição de faixas do mesmo tipo, produziu sempre soluções ótimas.

Nos testes realizados para a análise do “balanço” entre padrões tabuleiros e padrões não tabuleiros, constatou-se que em boa parte dos padrões obtidos pelo algoritmo enumerativo os desperdícios de material eram bastante elevados devido à utilização de padrões tabuleiros exatos. Na análise proposta por Morabito e Arenales, os autores consideraram também padrões tabuleiros não exatos. Padrões tabuleiros não exatos apresentam refilos (Figura 7.2a), possibilitando a obtenção de melhores soluções que o caso exato, pois permitem uma maior variedade de combinações. O padrão tabuleiro não exato considerado por Morabito e Arenales apresenta refilos em apenas uma das orientações do corte (Figura 7.2b).

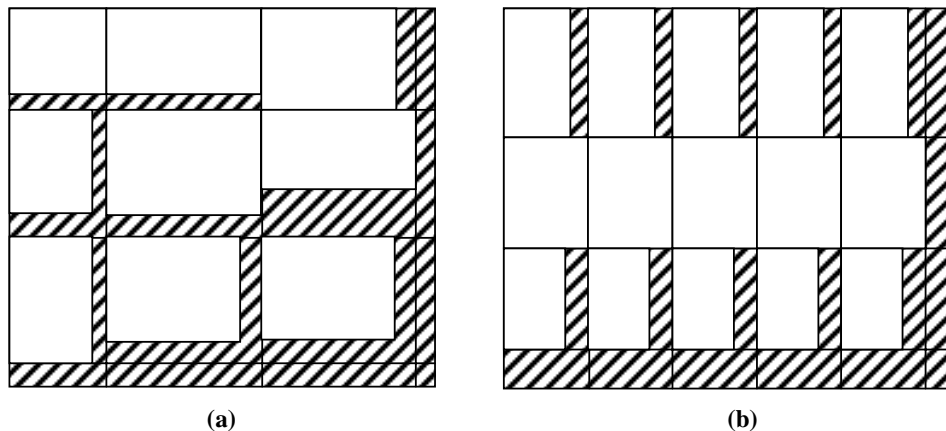


Fig. 7.2 – Exemplos de padrões tabuleiros não exatos: (a) padrão tabuleiro não exato com refilos em ambas as orientações do corte, (b) padrão tabuleiro não exato gerado pela heurística de Morabito e Arenales.

Para uma comparação mais adequada dos resultados computacionais do algoritmo enumerativo proposto com a heurística de Morabito e Arenales é preciso adaptá-lo para explorar também padrões tabuleiros não exatos. Foi realizada uma adaptação do algoritmo enumerativo para o caso não exato e as principais modificações em relação ao algoritmo apresentado no Capítulo 3 podem ser consultadas no Apêndice B.

Testes computacionais realizados para a determinação de padrões tabuleiros não exatos com o algoritmo enumerativo e a heurística de Morabito e Arenales revelaram que o algoritmo proposto obtém soluções melhores que as apresentadas pela heurística. Alguns dos exemplos testados e os resultados que foram obtidos estão apresentados no Apêndice B.

O tempo computacional para processamento dos exemplos, no entanto, foi demasiadamente alto tornando a aplicação do método bastante restrita na resolução de problemas de cortes mais gerais. O tempo para a determinação de cada padrão de corte variou de 45 a 240 minutos. Em razão deste tempo não foram realizados testes computacionais para a análise do “balanço” entre padrões tabuleiros (exatos e não exatos) e padrões não tabuleiros conforme apresentado por Morabito e Arenales (2000). Conclui-se que a aplicação de heurísticas, tais como a sugerida por estes autores ainda representam uma excelente alternativa para a resolução deste tipo problema, apresentando soluções eficientes em termos de tempo de processamento e qualidade de solução.

Deve-se ressaltar que o algoritmo proposto no presente trabalho permite a determinação de padrões tabuleiros gerais (exatos e não exatos) com perda “interna” e, pode ser estendido para a determinação de padrões tabuleiro restrito.

Observa-se também que visando a otimização do tempo de corte da chapa, é possível “converter” ou “aproximar” padrões 2-estágios não exatos para padrões tabuleiros não exatos conforme ilustrado na Figura 7.3. Esta conversão poderia ser feita através de um procedimento de rearranjo das peças que compõem o padrão 2-estágios não exato. Isto pode acelerar o corte de chapas, e se for o caso, pode-se até pensar em se realizar o corte dos refilos ou o corte de “subchapas” separadamente em uma outra serra.

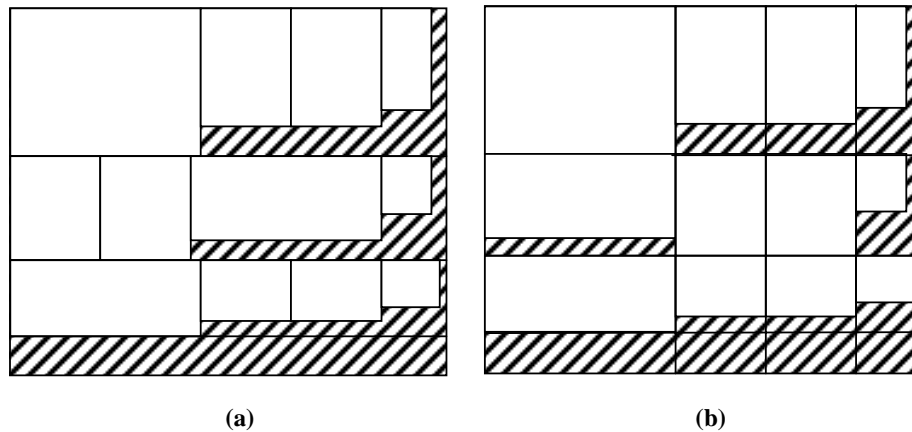


Fig. 7.3 – Convertendo um padrão 2-estágios não exato em um padrão tabuleiro não exato: (a) padrão 2-estágios não exato, (b) padrão tabuleiro não exato equivalente.

Acredita-se que a função custo descrita no Capítulo 5, embora simplificada, capta com maior riqueza o custo dos padrões tabuleiros e não tabuleiros, comparado com o custo fixo sugerido em Morabito e Arenales (2000), apesar de que as soluções obtidas com a aplicação desta função serem “sub-ótimas” pelas dificuldades de se resolver o subproblema de geração de colunas. Testes computacionais realizados mostraram que a função desenvolvida obtém melhores soluções quando o custo de corte de refilos ultrapassa um certo patamar.

Cabe salientar que na geração de colunas, os subproblemas foram resolvidos obtendo-se apenas o padrão de maior lucratividade a cada iteração. Porém, devido ao custo variável do tempo de corte dos padrões, nem sempre o padrão de melhor lucratividade corresponde ao padrão a ser inserido na base. Portanto é necessário verificar a possibilidade de inserção de padrões de menor lucratividade. O algoritmo desenvolvida para a geração de padrões tabuleiros enumera uma série de outros padrões tabuleiros antes de determinar o melhor deles em termos de lucratividade. É possível avaliar a função custo desenvolvida para cada um dos padrões determinados no processo de enumeração. Isto, entretanto não foi feito e fica como sugestão para trabalhos futuros. Uma outra análise que merece ser realizada é a inclusão de mais de um padrão tabuleiro

(mais de uma coluna) a cada iteração do método Simplex de Gilmore e Gomory. Em relação à função custo, acredita-se que na prática, uma boa parte do tempo de corte de padrões tabuleiros e não tabuleiros é gasto em operações de deslocamento ou manuseio das faixas cortadas. Este tempo pode ser implicitamente considerado no custo associado ao tempo de um corte longitudinal da função custo associada aos padrões de corte.

Finalmente, cabe observar que os detalhes sobre a implementação do algoritmo e a realização dos testes computacionais apresentados nesta dissertação também podem ser consultados em Katsurayama e Yanasse (1999) e Katsurayama e Yanasse (2000).

## REFERÊNCIAS BIBLIOGRÁFICAS

- Andonov, R.; Poirriez, V.; Rajopadhye, S. Unbounded knapsack problem: Dynamic programming revised. **European Journal of Operations Research**, v.123, n.2, p.394-407, 2000.
- 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, Passau, 1995. **Proceedings**. Passau. Allemagne, 1996, p.42-47.
- Arenales, M. N. Modelos e métodos básicos. In: Arenales M. N. e Morabito, R. ed. **Mini-curso: o problema de corte e empacotamento e aplicações industriais**. XX Congresso Nacional de Matemática Aplicada e Computacional e II Oficina Nacional de Problemas de Corte e Empacotamento, Gramado: ICMC-USP, 1997. Cap. 1, p.1-23.
- Bellman, R. **Dynamic Programming**. Princeton, N. J.: Princeton University Press, 1957, 339p.
- Dowland, K.; Dowland, W. Packing problems. **European Journal of Operations Research**, v.56, n.1, p.2-14, 1992.
- Dyckhoff, H. A typology of cutting and packing problems. **European Journal of Operations Research**, v.44, n.2, p.145-159, 1990.
- Dyckhoff, H.; Finke, U. **Cutting and packing in production and distribution: typology and bibliography**. Heidelberg: Springer Verlag Co., 1992, 248p.

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.

Gilmore, P.; Gomory, R. The theory and computation of knapsack functions. **Operations Research**, v.14, n.6, p.1045-1074, 1966.

Gramani, M. C. N. **Problema de corte bidimensional guilhotinado e restrito em 2-estágios**. São Carlos. 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. A procedure for solving the 1.5 dimensional coil slitting problem. **AIIE Transactions**, v.10, n.1, p.70-75, 1978.

Hinxman, A. The trim-loss and assortment problems: a survey. **European Journal of Operations Research**, v.5, n.1, p.8-18, 1980.

Katsurayama, D. M.; Yanasse, H. H. Um algoritmo enumerativo para determinação de padrões tabuleiros. [CDROM]. In: Simpósio Brasileiro de Pesquisa Operacional, 31., Juiz de Fora – MG, 1999. **Anais**. Tec Art Editora, 2000. Seção de Corte e Empacotamento.

Katsurayama, D. M.; Yanasse, H. H. Um algoritmo enumerativo para determinação de padrões tabuleiros: aspectos computacionais de implementação. [CDROM]. In:

- Simpósio Brasileiro de Pesquisa Operacional, 32., Viçosa – MG, 2000. **Anais**. Tec Art Editora, 2000. Seção de Corte e Empacotamento.
- Martello, S.; Toth, P. An exact algorithm for large unbounded knapsack problems. **Operations Research Letters**, v.9, n.1, p.15-20, 1990a.
- Martello, S.; Toth, P. **Knapsack problems – Algorithms and computer implementations**. Chichester: John Wiley & Sons, 1990b, 308p.
- Morabito, R. **Formulação matemática para padrões tabuleiros não exatos**. (Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2001). Comunicação pessoal.
- 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.; Garcia, V. The cutting stock problem in a hardboard industry: a case study. **Computers & Operations Research**, v.25, n.6, p.469-485, 1998.
- Pinto, M. J. **O problema de corte de estoque inteiro**. São Carlos. 98p. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas de São Carlos - Universidade de São Paulo – São Carlos, 1999.
- Salkin, H. M.; Kluyver, C.A. The knapsack problem: a survey. **Naval Research Logistics Quartely**, v.22, n.1, p.127-144, 1975.
- Soma, N.; Yanasse, H. H., Maculan, N. O problema da mochila. In: Arenales M.N. e Morabito, R. ed. **Mini-curso: o problema de corte e empacotamento e aplicações**

**industriais.** XX Congresso Nacional de Matemática Aplicada e Computacional e II Oficina Nacional de Problemas de Corte e Empacotamento, Gramado: ICMC-USP, 1997. Cap. 2, p.24-57.

Sweeney, P.; Patternoster, E. Cutting and packing problems: a categorised, application-oriented research bibliography. **Journal of the Operational Research Society**, v.43, n.7, p.691-706, 1992.

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.; Zinober, A. S. I.; Harris, R. G. Two dimensional cutting stock with multiple stock sizes. **Journal of the Operational Research Society**, v.42, n.8, p.673-683, 1991.



**APÊNDICE A**  
**TEMPO COMPUTACIONAL DO ALGORITMO ENUMERATIVO PARA**  
**DETERMINAÇÃO DE PADRÕES TABULEIROS EXATOS**

**Dados de Entrada:**

Dimensões da chapa: W=1251, L=1587;

Dimensões das peças: entre 15% e 63% das dimensões da chapa;

Lucro associado aos itens: gerados aleatoriamente entre 1 e 10.

**Equipamento Utilizado:**

Microcomputador Pentium III 450 MHz, 128 Mb RAM.

TABELA A1 – TEMPO COMPUTACIONAL: ALGORITMO EXATO PROPOSTO x  
HEURÍSTICA DE MORABITO E ARENALES (2000)

Peças	Algoritmo Exato Proposto *	Heurística de Morabito e Arenales (2000) *
10	0,004203	0,00138
20	0,007072	0,001872
30	0,010355	0,002406
40	0,014454	0,002978
50	0,029064	0,006813
60	0,040733	0,004669
70	0,23103	0,004384
80	0,04426	0,006285
90	0,048189	0,006495

(\*) Tempo em segundos



## APÊNDICE B

### ALGORITMO ENUMERATIVO PARA DETERMINAÇÃO DE PADRÕES TABULEIROS NÃO EXATOS

Neste Apêndice são discutidas as principais modificações a serem feitas no algoritmo enumerativo apresentado no Capítulo 3 para que o procedimento considere a geração de padrões tabuleiros não exatos.

Uma possível modificação poderia ser feita na fase de enumeração das combinações e na fase de seleção das faixas que irão compor o padrão tabuleiro. A enumeração será feita sobre um único grupo de larguras. O grupo deve apresentar todas as possíveis faixas a serem combinadas.

Para ilustrar o algoritmo, considere novamente o exemplo prático apresentado no Capítulo 3. No exemplo, as faixas selecionadas e seus respectivos lucros correspondem a:

$$\{w_1'=2, w_2'=2, w_1'=3, w_3'=3, w_7'=3, w_4'=4, w_5'=4, w_9'=4, w_6'=5, w_7'=5, w_2'=6, w_8'=6, w_9'=6\}$$
$$\{c_1=2, c_2=1.67, c_1=3, c_3=0.5, c_7=1, c_4=1.5, c_5=2, c_9=6.67, c_6=4.5, c_7=1.67, c_2=5, c_8=0.33, c_9=10\}$$

A recuperação das combinações será feita aplicando-se a segunda fase do algoritmo YSM de forma similar ao apresentado no Capítulo 3. Tendo sido recuperada uma combinação, prossegue-se com o processo de determinação do padrão tabuleiro não exato realizando-se uma seleção “provisória” dos itens de comprimento que podem estar associados à combinação recuperada. Suponha que no exemplo do Capítulo 3, a combinação recuperada seja dada por 1 item de largura 2 e 2 itens de largura 6. Os itens selecionados para as combinações de largura 2 (comprimentos 3, 6) e largura 6 (comprimentos 2, 3, 4) portanto, correspondem aos comprimentos {2, 3, 4, 6}. Na Figura B1 as faixas obtidas são ilustradas:

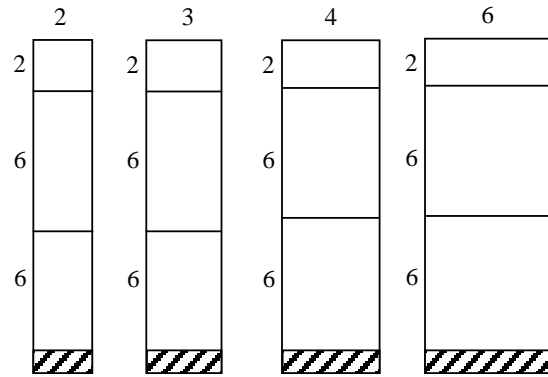


Fig. B1 – Faixas para uma combinação de largura 2 e 6.

O passo seguinte é seleccionar os itens que irão compor cada uma das faixas obtidas. As faixas serão compostas pelos itens de melhor lucratividade considerando-se as larguras das peças fixadas nas larguras das combinações que foram recuperadas e verificando-se os comprimentos inferiores ou iguais a cada um dos comprimentos que estão representados nas faixas seleccionadas. Para o dado exemplo, isto equivale a seleccionar o melhor conjunto de itens viáveis para cada uma das faixas fixando a largura das peças em 2 e 6 e verificando-se seus comprimentos correspondentes inferiores ou iguais respectivamente a 2, 3, 4 e 6. Na Figura B2 apresenta-se as melhores faixas obtidas para o exemplo ilustrado na Figura B1:

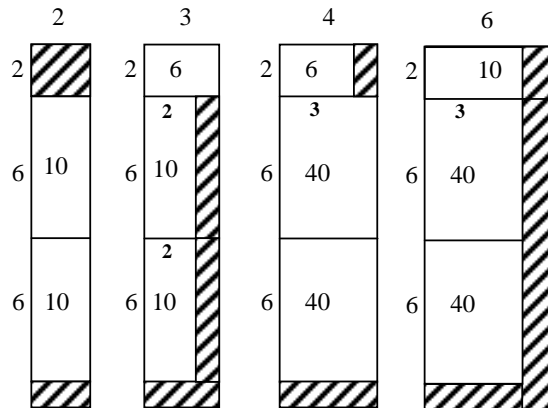


Fig. B2 – Seleção dos itens presentes nas faixas para se obter um conjunto de faixas com melhor lucratividade.

A partir das faixas obtidas, resolve-se o seguinte problema da mochila unidimensional:

$$\text{Maximizar } 20y'_1 + 26y'_2 + 86y'_3 + 90y'_4$$

$$\text{Sujeito à: } 2y'_1 + 3y'_2 + 4y'_3 + 6y'_4 \leq 23$$

$$y'_i \geq 0 \text{ inteiro, } i=1, \dots, 4.$$

O resultado é o padrão tabuleiro não exato ilustrado na Figura B3:

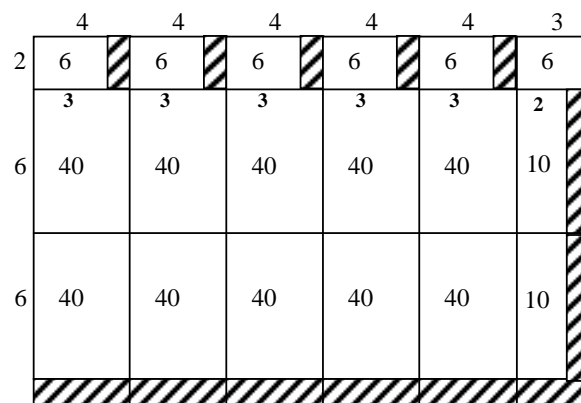


Fig. B3 – Padrão tabuleiro não exato obtido para o conjunto de faixas ilustradas na Figura B2.

Os passos seguintes do algoritmo permanecem inalterados em relação ao algoritmo apresentado no Capítulo 3. Entretanto, todo o procedimento descrito anteriormente deve ser repetido, desta vez, considerando o comprimento da chapa como dimensão inicial escolhida para a realização da enumeração das combinações. O melhor padrão tabuleiro encontrado é dado pela melhor solução obtida pelo algoritmo aplicado nas duas dimensões.

Os exemplos seguintes apresentam alguns dos casos onde o algoritmo adaptado obteve melhores soluções que a heurística de Morabito e Arenales:

TABELA B1 – EXEMPLO 1: CONJUNTO DE PEÇAS PARA DETERMINAÇÃO DE PADRÕES TABULEIROS NÃO EXATOS (CHAPA:  $W=1850, L=3670$ )

item $i$	$w_i$	$l_i$	$p_i$
1	274	609	0,0277778
2	274	380	0,0185185
3	330	425	0,00519481
4	361	650	0,04
5	270	348	0,0166667
6	270	705	0,0333333
7	328	718	0,04
8	300	705	0,0333333
9	330	465	0,0285714
10	330	480	0,0285714
11	250	1956	0,142857
12	302	674	0,0333333
13	270	674	0,0333333
14	270	636	0,0333333

11	14	2	5	5
11	14	2	5	5
11	14	2	5	5
11	14	2	5	5
11	14	2	5	5
11	14	2	5	5
11	14	2	5	5

Fig. B4 - Solução com o algoritmo de Morabito e Arenales (2000) (valor 1,36825).

11	14	5	5	5
11	14	5	5	5
11	14	5	5	5
11	14	5	5	5
11	14	5	5	5
11				
11				

Fig. B5 - Solução com uso do algoritmo proposto (valor 1,41667).

TABELA B2 – EXEMPLO 2 : CONJUNTO DE PEÇAS PARA DETERMINAÇÃO DE PADRÕES TABULEIROS NÃO EXATOS (CHAPA:  $W=1850$   $L=3670$ )

item $i$	$w_i$	$l_i$	$p_i$
1	274	609	0,025641
2	274	380	0,017094
3	330	425	0,00519481
4	361	650	0,04
5	270	348	0,0102564
6	270	705	0,0333333
7	328	718	0,04
8	300	705	0,0333333
9	330	465	0,0285714
10	330	480	0,0285714
11	250	1956	0,142857
12	302	674	0,0333333
13	270	674	0,0312821
14	270	636	0,0312821

11	12	14	2
11	12	14	2
11	12	14	2
11	12	14	2
11	12	14	2
11	12	14	2
11	12	14	2
11	12	14	2

Fig. B6 - Solução com o algoritmo de Morabito e Arenales (2000) (valor 1,3474).

11	14	6	5
11	14	6	5
11	14	6	5
11	14	6	5
11	14	6	5
11			
11			

Fig. B7 – Solução com uso do algoritmo proposto (valor 1,37435).





**APÊNDICE C**  
**APLICAÇÃO DA FUNÇÃO CUSTO NO MÉTODO**  
**SIMPLEX COM GERAÇÃO DE COLUNAS**

Este Apêndice apresenta um pseudo-algoritmo do método Simplex com geração de colunas utilizado para a realização dos testes computacionais da nova função custo. Embora o método não seja descrito em grandes detalhes, o algoritmo apresentado reúne os pontos essenciais para que o leitor compreenda melhor como foi aplicada a função custo na resolução do problema de corte de estoque. O problema resolvido segue o modelo de programação linear apresentado por Gilmore e Gomory (1961):

$$\text{Minimizar } z = c_1x_1 + c_2x_2 + c_3x_3 + \dots + c_nx_n$$

$$\text{Sujeito a: } a_{i1}x_1 + a_{i2}x_2 + a_{i3}x_3 + \dots + a_{in}x_n \geq N_i, (i=1, \dots, m)$$

$$x_j \geq 0 \text{ e inteiro } (j=1, \dots, n),$$

onde  $c_j$  é o custo associado ao  $j$ -ésimo padrão,  $a_{ij}$  é o número de itens do tipo  $i$  no  $j$ -ésimo padrão,  $N_i$  é a demanda associada ao  $i$ -ésimo item,  $x_j$  é a quantidade de vezes que o  $j$ -ésimo padrão deve ser cortado.

**Definição das Variáveis:**

N	Total de diferentes tipos de chapas
C[j]	Custo associado ao $j$ -ésimo tipo de chapa ( $j=1, \dots, 10$ )
Z[i]	Vetor que contém os 10 últimos valores da função objetivo
otimiza	Indica se o problema deve continuar a ser otimizado (SIM) ou não (NAO)
otimo	Indica se a coluna gerada atende a condição de otimalidade (SIM) ou não (NÃO)
coluna	Coluna gerada (ou a ser gerada)
diff	Representa a diferença obtida entre a solução do padrão e o custo da sua produção (custo da chapa envolvida no processo adicionada ao tempo de corte do padrão gerado)

gdiff      Representa o maior valor de diff dentre as soluções dos padrões obtidos para cada um dos diferentes tipos de chapas testados.

```
PROCEDIMENTO SIMPLEXGC()  
  
// Procedimento principal para a aplicação da função custo na resolução  
// do método Simplex com Geração de Colunas  
Inicio  
    Considera a chapa de menor custo relativo (custo da chapa/dimensões  
    da chapa). Gere uma base inicial composta de padrões homogêneos e  
    otimize o problema;  
  
    z[1]=valor corrente da função objetivo;  
    i=1;  
    otimiza=SIM;  
  
    // Enquanto houver possibilidade do problema ser otimizado  
    Enquanto ( otimiza==SIM ) faça  
        Inicio  
            // Resolução do problema é feita inicialmente considerando-se  
            // a geração de padrões do tipo tabuleiro  
            coluna=Gerar(tabuleiro);  
  
            // Se a condição de otimalidade for atingida para padrões  
            // tabuleiros, então o problema passa a ser resolvido  
            // considerando-se padrões do tipo não tabuleiro.  
            Se ( coluna==nenhuma ) faça coluna=Gerar(naotabuleiro);  
  
            // Se a condição de otimalidade for atingida para padrões  
            // tabuleiros e não tabuleiros, então a solução ótima do problema  
            // foi encontrada e, portanto o procedimento deve ser encerrado.  
            // Caso contrário, a coluna gerada será inserida na base  
            Se ( coluna==nenhuma ) faça otimiza=NÃO;  
            Senão faça  
                Inicio  
                    Insira a coluna gerada e otimize problema;  
                    Se ( i<10 ) faça z[++i]=valor corrente da função objetivo;  
                    Senão faça  
                        Inicio  
                            Para i=1 ate 9 faça z[i]=z[i+1];  
                            z[10]=valor corrente da função objetivo;  
  
                            // O procedimento deve ser encerrado caso o valor da  
                            // solução seja melhorado em pelo menos 0,1% nas  
                            // últimas 10 iterações  
                            Se ( z[1]-z[10]<0.01*z[10] ) faça otimiza=NAO;  
                        Fim-Senão;  
                Fim-Senão;  
            Fim-Enquanto;  
        Fim-SimplexGC().
```

```

FUNÇÃO GERAR(tipopadrao)

// Esta função retorna a melhor coluna gerada considerando-se os
// diferentes tipos de chapas e o tipo de padrão de corte especificado
// pelo argumento tipopadrao. A função retorna nenhuma, caso não seja
// constatado colunas que resultem na otimização do problema
Início
  gdiff=0;
  otimo=NAO;
  Para j=1 ate N faça
    Início
      Se ( tipopadrao==tabuleiro ) gere o melhor padrão tabuleiro;
      Senão gere o melhor padrão não tabuleiro;
      Determine a coluna correspondente ao padrão gerado;

      // Testa a condição de otimalidade do problema para cada
      // tipo de chapa considerando-se a adição do valor da função
      // custo ao custo da chapa. A função custo varia conforme o
      // padrão gerado. Considera-se que as chapas estejam
      // especificadas em ordem crescente por custo relativo
      // (custo da Chapa/dimensões da chapa)
      Se ( valor do padrão>C[j]+f(padrao) ) faça
        Início
          diff=valor do padrão-( C[j]+f(padrao) );
          Se ( diff>0.000001 ) e ( diff>gdiff ) faça
            Início
              otimo=SIM;
              gdiff=diff;
            Fim-Se;
          Fim-Se;
        Fim-Para;

      // Se a condição de otimalidade não for satisfeita por nenhuma das
      // colunas geradas para cada tipo de chapa, então o problema não
      // poderá ser melhorado com o uso do tipo de padrão considerado.
      // Caso contrário, a melhor coluna gerada deve ser retornada para
      // ser inserida na base
      Se (otimo==NAO) nenhuma;
      Senão retorne a coluna que foi determinada;
    Fim-Gerar();
  Fim-Para;

```



**APÊNDICE D**  
**RESULTADOS DOS TESTES COMPUTACIONAIS COMPARANDO O**  
**“BALANÇO” ENTRE PADRÕES TABULEIROS EXATOS E PADRÕES 2-**  
**ESTÁGIOS EXATOS**

TABELA D1 – “BALANÇO” ENTRE PADRÕES TABULEIROS EXATOS E  
PADRÕES 2-ESTÁGIOS EXATOS

<b>Custo fixo</b>	<b>Custo mínimo</b>	<b>Desperdício (m<sup>2</sup>)</b>	<b>Desperdício (%)</b>	<b>Placas cortadas</b>	<b>Proporção de padrões não tabuleiros (%)</b>
0,00	349,60	69,38	2,92	349,60	84,34
0,01	352,55	69,38	2,92	349,60	84,34
0,02	355,43	70,51	2,97	349,77	80,88
0,03	358,25	70,86	2,98	349,82	80,28
0,05	363,86	70,98	2,99	349,84	80,13
0,10	377,55	81,99	3,43	351,46	74,23
0,15	388,28	184,73	7,42	366,59	39,43
0,18	391,14	296,86	11,41	383,11	11,65
0,20	391,76	333,90	12,66	388,56	4,12
0,22	392,08	333,90	12,66	388,56	4,12
0,24	392,40	333,90	12,66	388,56	4,12
0,26	392,72	333,90	12,66	388,56	4,12
0,28	393,04	333,90	12,66	388,56	4,12
0,30	393,36	333,90	12,66	388,56	4,12
0,32	393,68	333,90	12,66	388,56	4,12
0,34	394,00	333,90	12,66	388,56	4,12
0,36	394,32	333,90	12,66	388,56	4,12
0,38	394,64	333,90	12,66	388,56	4,12
0,40	394,96	333,90	12,66	388,56	4,12
0,42	395,28	333,90	12,66	388,56	4,12
0,44	395,60	333,90	12,66	388,56	4,12
0,46	395,92	333,90	12,66	388,56	4,12
0,48	396,24	333,90	12,66	388,56	4,12
0,50	396,56	333,90	12,66	388,56	4,12
0,52	396,65	388,82	14,44	396,65	0