CORTE DE ESTOQUE BIDIMENSIONAL

eng. Reinaldo Morábito Neto

Orientador: dr. Marcos Nereu Arenales

marco 1989

Dissertação apresentada ao Instituo de Ciências Matemáticas de São Carlos, da Universidade de São Paulo, como parte dos requisitos para a obtenção do título de Mestre na Área "Ciências de Computação e Matemática Computacional"

RESUMO

Uma grande variedade de materiais são produzidos e estocados em grandes unidades que posteriormente são cortadas em unidades menores encomendadas por clientes. Evidentemente os processos de corte estão restritos pela natureza das máquinas utilizadas.

O Problema do Corte de Estoque Bidimensional consiste em cortar em função de um dado objetivo grandes placas de estoque em determinadas peças menores satisfazendo a demanda dos clientes. Associado a este *problema*, aparece um importante *sub-problema* que consiste em gerar um bom padrão de corte para cortar um certo número de peças encomendadas de uma única placa de estoque.

Este estudo revisa algumas técnicas que têm sido aplicadas para estes *problemas* incluindo algumas modificações. Também é sugerida uma nova representação do *sub-problema* em um *grafo-e-ou* que possibilitará outras abordagens para a solução.

ABSTRACT

A wide variety of materials are produced and supplied in large units, that will be cut into smaller order units. Evidently, cutting processes are constrained by the nature of the machinery being used.

Two-Dimensional Cutting-Stock Problem consists in cutting the stock plates into required smaller pieces to satisfy an order book, in such a way to optimize a given objective. Associated to this *problem*, there is an important *sub-problem* which consists to generate a good cutting pattern to cut a number of smaller pieces from a single stock plate.

This study reviews the techniques that have been applied to these *problems* including some modifications. Also, it sugests a new *and-or-graph* representation for the *sub-problem* which may provide other approaches to the solution.

CONTEÚDO

I - INTRODUÇÃO 4

Problema Geral, 4

Caso Bidimensional, 6

Objeto deste Estudo, 11

II - FORMULAÇÃO DO PROBLEMA 13

Formulação Inicial, 13

Geração de Padrões de Corte Bidimensional, 14

Formulação Relaxada, 16

III- PADRÃO DE CORTE 2(3)-ESTÁGIOS GUILHOTINADO 19

2-Estágios, 19

Problema da Mochila, 21

Terceiro Estágio, 25

IV - PADRÃO DE CORTE MULTI-ESTÁGIOS GUILHOTINADO 29

Representação em Grafo-E-Ou, 30

Procedimento Recursivo, 32

V - RESULTADOS COMPUTACIONAIS E CONCLUSÕES 44

Conclusões, 50

VI - REFERÊNCIAS 52

Ordem Alfabética, 52

Ordem Cronológica, 55

CORTE DE ESTOQUE BIDIMENSIONAL

INTRODUÇÃO

PROBLEMA GERAL

A economia de escala faz com que certos produtos sejam produzidos em grandes lotes, o que torna geralmente mais econômico a produção e provisão de alguns materiais em determinadas unidades (unidades de estoque).

Posteriormente, estas unidades de estoque são cortadas em diversas unidades menores (unidades de encomenda) em conformidade com a demanda dos clientes.

Considerando-se um processo de produção com estas duas etapas, aparecem dois problemas distintos associados respectivamente a cada uma delas.

O primeiro refere-se ao dimensionamento de unidades econômicas de estoque, ou seja, quais os melhores tamanhos em que as unidades de estoque deveriam ser produzidas? Este problema é conhecido na literatura como Problema da Classificação (*Assortment Problem*), e não será tratado neste estudo.

O segundo problema refere-se a seleção dos melhores padrões de corte destas unidades de estoque, ou seja, quais as melhores formas de se cortar as unidades de estoque para satisfazer a

demanda de unidades de encomenda requerida pelos clientes? Este problema é conhecido como Problema do Corte de Estoque (*Cutting-Stock Problem*).

CLASSIFICAÇÃO DOS PROBLEMAS

O Problema do Corte de Estoque tem sido categorizado pela dimensão. Portanto, o problema unidimensional é aquele em que apenas uma das dimensões das unidades de estoque e unidades de encomenda é relevante. Esta situação ocorre, por exemplo, no corte de barras de aço em que apenas o comprimento das barras será relevante.

Kantarovich [1939] inicialmente propos uma formulação matemática do problema unidimensional, sendo publicada em ingles somente vinte-e-um anos depois. Eisemann [1957] tratou-o como Problema da Apara (*Trim Problem*), e posteriormente Gilmore-Gomory [1961] [1963] como Problema do Corte de Estoque.

Inúmeros autores, desde então, tem estudado o caso unidimensional e sua aplicação estende-se ao corte de rolos de papel, de tecidos, de celofanes, de carpetes, e barras de aço, tubos metálicos, fibra-de-vidro. Arcaro [1988] recentemente abordou o caso unidimensional sugerindo alternativas aplicáveis à indústria do papel.

O caso bidimensional similarmente é aquele em que duas das dimensões das unidades de estoque e unidades de encomenda são significantes na determinação da solução.

Aplica-se ao corte de vidro, chapas de aço, grafite, filme fotográfico, plástico, papel, madeira. Também pode ser utilizado como parte do Problema do Corte de Estoque Tridimensional, por exemplo na produção de *crepe-rubber* conforme Schneider [1988].

Uma observação interessante é certa analogia existente entre

o Problema do Empacotamento (*Packing Problem*) e o Problema do Corte de Estoque: cortar uma unidade de estoque em diversas unidades de encomenda pode ser visto como análogo a empacotar diversas unidades de encomenda dentro de uma unidade de estoque. Isto aplica-se, por exemplo no caso tridimensional, ao transporte de carga ferroviária e ao empacotamento de *containers*, não tendo sido explorado pela literatura como observaram George-Robinson [1980].

CASO BIDIMENSIONAL

O Problema do Corte de Estoque Bidimensional (*Two-Dimensional Cutting-Stock Problem*) consiste em cortar em função de um objetivo grandes retângulos (placas de estoque (W,L)) em determinados retângulos menores (peças de encomenda (w,l)) satisfazendo a demanda dos clientes.

Associado a este problema, aparece um importante sub-problema que consiste em gerar um bom padrão de corte para cortar um certo número de peças de encomenda de uma única placa de estoque.

O caso bidimensional ocorre, por exemplo, no corte de placas de madeira para produzir móveis, no corte de chapas de aço para fabricação de vigas metálicas, no corte de lâminas de vidro em peças menores encomendadas.

Alguns autores, como Albano-Sapuppo [1980] e Farley [1988], estudaram o caso bidimensional aplicado a peças de encomenda irregulares (não-retangulares). Exemplos disto ocorrem na indústria têxtil, na construção naval, e no corte de couro; e não serão tratados neste estudo.

O material a ser cortado pode ser anisotrópico implicando

numa orientação das peças de encomenda dentro dos padrões de corte, ou isotrópico não implicando em nenhuma orientação. Por exemplo, uma porta ao ser cortada de uma placa de madeira deverá ter suas fibras na direção longitudinal. Por outro lado, a orientação de pequenas chapas de aço encomendadas poderá ser qualquer dentro da chapa de estoque.

As direções dos cortes sobre as placas de estoque serão sempre ortogonais; o que reduz a orientação das peças de encomenda dentro dos padrões de corte. Ou seja, caso o material seja anisotrópico, a peça de encomenda deverá ser cortada na mesma orientação da placa de estoque. Caso o material seja isotrópico (ou ortotrópico), a peça de encomenda poderá ser cortada na mesma orientação ou na orientação ortogonal a da placa de estoque.

De Cani [1978] observou que há situações particulares em que a restrição de cortes ortogonais acima descrita pode produzir uma pior solução para materiais isotrópicos. E propõe um modelo que permite rotações para as peças de encomenda dentro dos padrões de corte. Considerando-se a particularidade destas situações e as restrições dos equipamentos de corte, os cortes não-ortogonais não serão tratados neste estudo.

O padrão pode estar restrito ainda ao número máximo de vezes que cada peça de encomenda pode ser produzida de uma única placa de estoque. Por exemplo, poderíamos na fig.1 impor um limite de 2-vezes para a peça de encomenda 8 que aparece 3-vezes.

CLASSIFICAÇÃO DOS PADRÕES DE CORTE

A grande maioria dos processos de corte possuem uma característica comum; o corte inicia-se de um lado da placa de estoque e atravessa em linha reta até o lado oposto. Este tipo de

corte é devido em grande parte a simplicidade e a restrições técnicas dos equipamentos.

Um exemplo dele ocorre na indústria do papel onde as folhas são cortadas com uma guilhotina; por causa disto este tipo de corte foi denominado **guilhotinado.**

Basicamente os padrões de corte podem ser classificados como guilhotinados ou não-guilhotinados. Num padrão de corte não-guilhotinado (fig.1) as peças de encomenda assumem qualquer posição dentro da placa de estoque. As regiões hachuriadas representam as perdas de material.

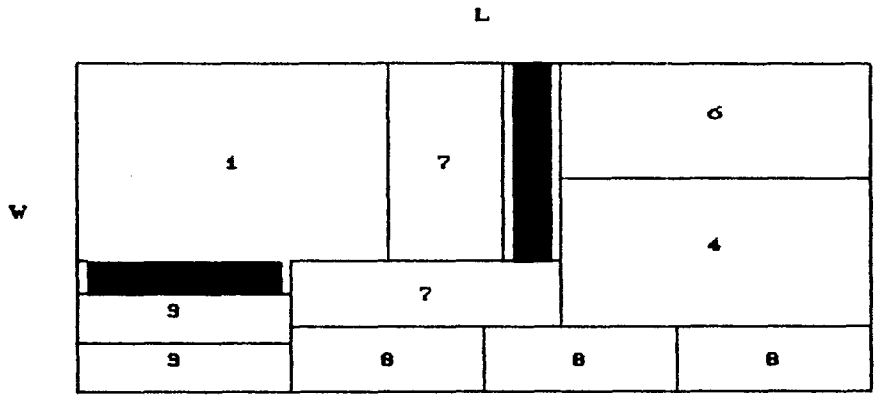

fig.1 - **PADRAO DE CORTE NAO-GUILHOTINADO**

Os padroes de corte guilhotinados ainda podem ser classificados como segue:

(i)Padrão de Corte Multi-Estágios (ou Não-Estagiado)

Os cortes são produzidos em diversos estágios, sendo que o número destes não é limitado.

Num primeiro estágio são feitos cortes sobre a placa de estoque em uma determinada direção de tal forma que são produzidos retângulos intermediários, por sua vez cortados em direções

perpendiculares a direção do estágio anterior, e assim sucessivamente até produzir as peças de encomenda. Em todos os estágios, os cortes sobre um retângulo qualquer deverão ir de uma de suas extremidades até a sua outra oposta (fig.2).

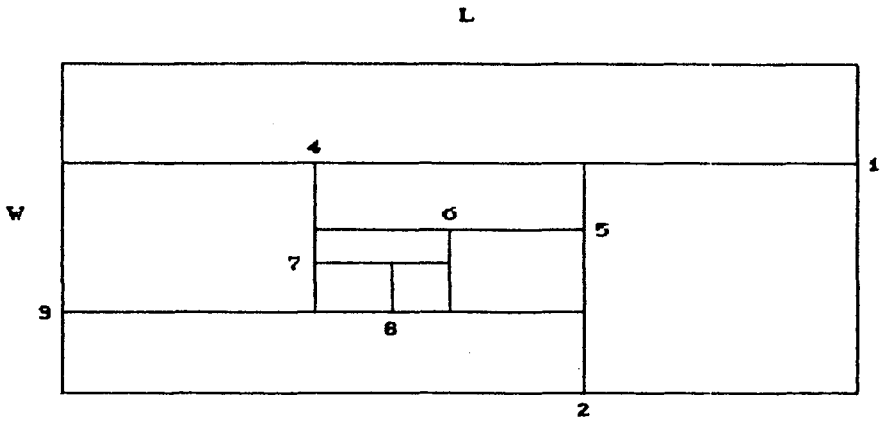

fig. 2- **PADRAO DE CORTE 8-ESTAGIOS GUILHOTINADO**

(ii)Padrão de Corte N-Estágios (ou Estagiado)

Os cortes são produzidos em N-estágios.

Temos o caso exato (fig.3.a) e o caso não-exato (fig.3.b) quando é permitido um estágio posterior somente para fazer a última apara na direção perpendicular à direção dos cortes do estágio anterior.

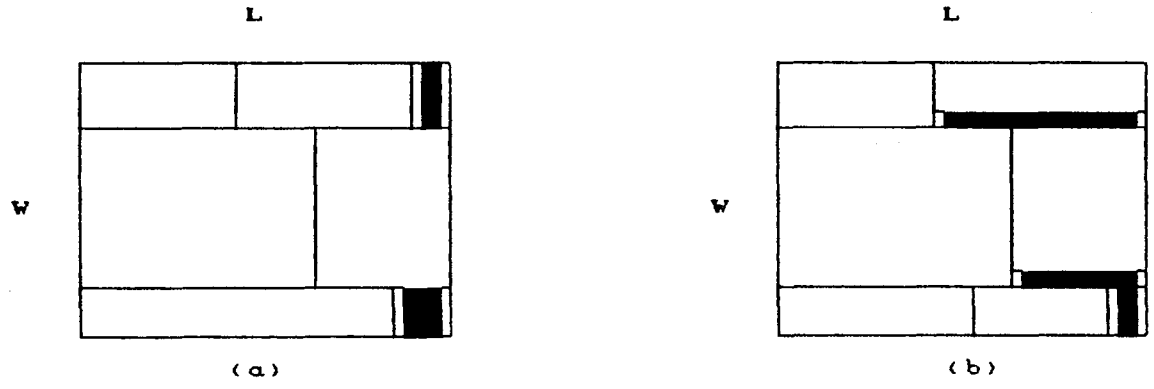

fig. 3 - **PADRAO DE CORTE 2-ESTAGIOS GUILHOTINADO**

HISTÓRICO

Alguns autores estudaram o Problema do Corte de Estoque Bidimensional conforme definido anteriormente; outros estudaram apenas o sub-problema associado da geração de padrões de corte bidimensional.

Gilmore-Gomory [1965] inicialmente estudaram o caso bidimensional e o multidimensional com programação linear, minimizando o número de placas de estoque a serem cortadas e atendendo a uma demanda mínima de peças de encomenda. Apresentaram um algoritmo iterativo através de programação dinâmica para gerar padrões de corte irrestrito em 2(3)-estágios guilhotinado.

Herz [1972] apresentou um algoritmo recursivo de busca em grafo para gerar padrões de corte irrestrito em multi-estágios guilhotinado. A técnica utilizou discretizações e limitantes para a busca no grafo. Comparações foram feitas entre o algoritmo recursivo e dois algoritmos iterativos.

Dyson-Gregory [1974] estudaram o caso bidimensional aplicado a produção de vidro com programação linear, minimizando a perda no corte do material e atendendo a uma demanda mínima. Apresentaram uma heurística para sequenciamento dos padrões de corte com o objetivo de minimizar a descontinuidade no atendimento das encomendas. Concluíram que os resultados da implementação com o sequenciamento de padrões não foram satisfatórios para aplicação.

Christofides-Whitelock [1977] apresentaram um algoritmo de busca em árvore para gerar padrões de corte restrito em multi-estágios guilhotinado. Utilizaram programação dinâmica e o Problema do Transporte (*Transportation Problem*) para produzir limitantes superiores durante a busca na árvore.

Outros autores como Adamowicz-Albano [1976] propuseram aproximações para o caso da placa de estoque ter o comprimento bem maior do que a largura, e De Cani [1978] estudou particularmente os cortes não-ortogonais. Hinxman [1980] fez um exame completo

apresentando uma taxonomia dos problemas e revisando as técnicas utilizadas para a solução.

Wang [1983] apresentou dois métodos combinatoriais para gerar padrões de corte restrito em multi-estágios guilhotinado e Farley[1983] apresentou dois procedimentos com uma restrição adicional no problema dirigido a indústria do vidro. Israni-Sanders [1984] estudaram algumas heurísticas e o efeito da intervenção humana, e Beasley [1985] apresentou um algoritmo de busca em árvore para gerar padrões de corte restrito não-guilhotinado.

Beasley [1985] utilizou programação dinâmica em fórmulas recursivas para gerar padrões de corte irrestrito em n-estágios e multi-estágios guilhotinado. Apresentou um algoritmo heurístico para problemas de tamanho grande.

OBJETO DESTE ESTUDO

Este presente estudo está dirigido para o Problema do Corte de Estoque Bidimensional.

No segundo capítulo serão revistos a formulação matemática do *problema* adaptada a um intervalo de demanda dos clientes que melhor representa os casos reais, e um método de solução através de programação linear.

Para resolver o *sub-problema* da geração de padrões de corte irrestrito em 2(3)-estágios guilhotinado, no terceiro capítulo serão analisados algoritmos que utilizam programação dinâmica e métodos enumerativos.

Será sugerida uma heurística para gerar o terceiro estágio a partir do padrão de corte em 2-estágios.

No quarto capítulo será revisto um algoritmo recursivo para o *sub-problema* de gerar padrões de corte irrestrito em multi-estágios guilhotinado, com discretizações geradas através de programação dinâmica e limitantes para a busca no grafo.

Será sugerida uma nova representação deste *sub-problema* em um grafo-e-ou *(and-or-graph)* que possibilitará outras abordagens para a solução.

No quinto capítulo serão apresentados alguns resultados computacionais das implementações, e as conclusões finais.

FORMULAÇÃO DO PROBLEMA

FORMULACÃO INICIAL

O problema consiste em selecionar o conjunto dos melhores padrões de corte das placas de estoque de forma a atender a demanda de peças de encomenda. Suponha que placas de estoque com dimensões (W,L) deverão ser cortadas em m peças menores (peças de encomenda) com dimensões (w_i,l_i) e uma demanda b_i.

Define-se um padrão de corte A_j como um vetor $[a_{1j},a_{2j},a_{3j},\ldots,a_{mj}]$, onde a_{ij} é o número de peças (w_i,l_i) que ocorrem em (W,L).

Por exemplo, a placa de estoque ilustrada na fig.1 foi cortada num padrão onde ocorrem: 1 peça (w_1,l_1), 2 peças (w_3,l_3), 1 peça (w_4,l_4), 1 peça (w_6,l_6), 2 peças (w_7,l_7), 3 peças (w_8,l_8); se m = 10, logo temos o padrão de corte $A_j=[1,0,2,1,0,1,2,3,0,0]$.

Inicialmente entenderemos como melhores padrões aqueles que nos levam a uma solução cuja perda f (ou todas as sobras de material de todos os cortes) seja mínima. Define-se a perda unitária do padrão de corte A_j como:

$$c_j = W.L - \sum_{i=1,m} a_{ij}.(w_i.l_i) \qquad (2.1)$$

Seja n o número de padrões de corte e x um vetor-de-variáveis $[x_1,x_2,x_3,\ldots,x_j,\ldots,x_n]$; temos:

$$\begin{aligned} &\textit{minimizar} && f = c.x \\ &\textit{sujeito a:} && A.x = b \\ &\textit{com:} && x \geq 0 \text{ , } \textit{inteiro} \end{aligned} \qquad (2.2)$$

onde, c é o vetor-de-perda-unitária $[c_1, c_2, c_3, \ldots, c_j, \ldots, c_n]$
b é o vetor-de-demanda $[b_1, b_2, b_3, \ldots, b_j, \ldots, b_m]$
A é a matriz-de-padrões $[A_1, A_2, A_3, \ldots, A_j, \ldots, A_n]$

A matriz-de-padrões A tem dimensao $m \times n$, ou seja, m linhas correspondendo a m diferentes peças encomendadas, e n colunas correspondendo a n possíveis padrões de corte distintos.

Considerando a grandeza de m e as dimensões das peças de encomenda e placas de estoque , a grandeza de n pode chegar a ordem de milhões de colunas na matriz A, sendo impraticável gerar todos estes padrões e resolver este problema de programação inteira.

Gilmore-Gomory [1965] sugeriram a relaxação da restrição de integralidade de (2.2), e analisaram o problema através de programação linear justificando que as soluções não-inteiras arredondadas poderiam ser satisfatórias. Propuseram também uma técnica de geração de colunas para a matriz A, que será vista a seguir.

GERAÇÃO DE PADRÕES DE CORTE BIDIMENSIONAL

Problemas de programação linear podem ser, em geral, resolvidos pelo Método Simplex.

Suponha que se tenha uma solução básica factível para o problema (2.2) relaxado da restrição de integralidade, cujo valor

da funçao objetivo e dado por:

$$f_o = c_B . B^{-1} . b \qquad (2.3)$$

onde, c_B é o vetor-de-perda das variáveis básicas
B^{-1} é a inversa da matriz-básica de A

Uma solucão geral do sistema $A.x = b$ pode ser escrita por:

$$x_B = B^{-1} . b - \sum_{j \in N} B^{-1} . A_j . x_j \qquad (2.4)$$

onde, N é o conjunto de índices de variáveis não-básicas

Substituindo (2.3) e (2.4) na expressão da função objetivo $f=c.x$, obtemos a seguinte expressão para todo x / $A.x = b$:

$$f = c_B . B^{-1} . b - \sum_{j \in N} c_B . B^{-1} . A_j . x_j + \sum_{j \in N} c_j . x_j = f_o - \sum_{j \in N} (\pi . A_j - c_j) . x_j \qquad (2.5)$$

onde, $\pi = c_B . B^{-1}$ é o vetor-multiplicador-de-Lagrange $[\pi_1, \pi_2, .., \pi_m]$

A condição de otimalidade do Método Simplex testa as variáveis não-básicas para avaliar se é vantajoso substituir alguma delas por uma variável básica.

Desde que se quer minimizar f, deduz-se de (2.5) a seguinte condição para que uma variável não-básica x_j substitua uma variável básica:

$$-\partial f / \partial x_j = \pi . A_j - c_j > 0 \qquad (2.6)$$

Substituindo (2.1) em (2.6), temos:

$$\sum_{i=1,m} \pi_i . a_{ij} - (W.L - \sum_{i=1,m} a_{ij} . (w_i . l_i)) = \sum_{i=1,m} ((\pi_i + w_i . l_i) . a_{ij}) - W.L > 0$$

Seja $\Pi_i = \pi_i + w_i . l_i$; temos:

$$\Pi . A_j - W.L > 0 \qquad (2.7)$$

Como o termo $W.L$ é constante e estritamente positivo, a estratégia para gerar um padrão de corte A_j que satisfaça (2.7) é *maximizar* $\Pi . A_j$.

Foi proposto como gerador de padrões de corte o Problema da Mochila Generalizado (*Generalized Knapsack Problem*).

$$\textit{maximize} \quad g = \Pi_1 . a_1 + \Pi_2 . a_2 + \ldots + \Pi_m . a_m \qquad (2.8)$$

sujeito a condição de que $[a_1, a_2, \ldots, a_m]$ *corresponda a um padrão de corte, onde peças* (w_i, l_i) *preenchem uma placa de estoque* (W, L).

FORMULACÃO RELAXADA

A formulação relaxada do Problema de Corte de Estoque Bidimensional é basicamente a formulação (2.2) com a restrição de integralidade relaxada.

No entanto, algumas alterações serão feitas neste modelo inicial. Alguns clientes ao encomendar determinados produtos toleram uma certa variação na quantidade final b. Ou também pode

acontecer de se desejar para determinados produtos uma produção estratégica dentro de um limite inferior *b1* e superior b2 previstos de demanda. Portanto, a produção será canalizada no intervalo *[b1 ,b2]*, como segue:

$$b1 \leq A.x \leq b2 \quad \Rightarrow \quad A.x + s = b2 \quad \Rightarrow \quad s \geq 0 \qquad (2.9)$$
$$b1 \leq b2 - s \quad \Rightarrow \quad s \leq b2 - b1$$

onde, *b1* é o vetor-de-denanda-inferior $[b1_1, b1_2, b1_3, \ldots, b1_m]$
b2 é o vetor-de-demanda-superior $[b2_1, b2_2, b2_3, \ldots, b2_m]$
s é o vetor-de-variáveis-de-folga $[s_1, s_2, s_3, \ldots, s_m]$

Existem algumas vantagens adicionais na inclusão das variáveis de folga além de permitir o intervalo de demanda. Sem o uso delas como em (2.2), qualquer solução ótima para o problema será em geral em termos de * padrões de corte, enquanto que com o uso delas a solução ótima será provavelmente em termos de um número menor do que * padrões de corte.

Gilmore-Gomory [1963] discutiram para o caso unidimensional a mudança da função objetivo no caso da quantidade produzida poder variar dentro de um intervalo *[b1,b2]*, ao invés de ser fixada num valor *b*.

Ao produzir-se quantidades maiores, a perda total pode aumentar enquanto que a perda percentual, indicadora de eficiência técnica, pode diminuir. Foi sugerida uma função objetivo racional de minimizacao da perda percentual, definida em termos da perda unitária (2.1):

$$minimizar \quad \xi = \sum_{j=1,n} (c_j . x_j) \; / \; \sum_{j=1,n} x_j$$

Foi mostrado que a estratégia Simplex pode ser utilizada para encontrar a solução ótima neste problema com função objetivo racional.

Alguns autores propuseram ainda a *minimização de placas cortadas*, ou a *minimização do custo* associado as placas cortadas. Arcaro [1988] propos a substituição da função objetivo *minimização da perda percentual* pela *maximização do lucro*, mais adequada do ponto de vista econômico. Todas estas funções podem ser adaptadas para o caso bidimensional.

PADRÃO DE CORTE 2(3)-ESTÁGIOS GUILHOTINADO

2-ESTÁGIOS

A maior parte dos casos práticos pertencem a sub-classes mais particulares do padrão de corte bidimensional. Do ponto de vista industrial, uma das sub-classes mais importantes é a que envolve apenas cortes guilhotinados em 2-estágios.

Reescrevendo a expressão (2.8), temos:

$$\textit{maximize} \quad g = \Pi_1.a_1 + \Pi_2.a_2 + \ldots + \Pi_m.a_m \qquad (3.1)$$

sujeito a condição de que $[a_1,a_2,\ldots,a_m]$ *corresponda a um padrão de corte 2-estágios guilhotinado, onde peças* (w_i,l_i) *preenchem uma placa de estoque* (W,L).

Considerando-se inicialmente o caso não-exato, o material anisotrópico, e os cortes do primeiro estágio paralelos ao comprimento L da placa de estoque (fig.4.a), a expressão (3.1) sem perda de generalidade pode ser resolvida em duas etapas:

(i) *maximizar*
$$\Pi'_i = \sum_{k \in S_i} \Pi_k.y_k \quad , \; i=1,m \qquad (3.2)$$

sujeito a:
$$\sum_{k \in S_i} l_k.y_k \leq L$$

com: $y \geq 0$, *inteiro*

onde, $S_i = [\, k \;/\; w_k \leq w_i \;,\; k=1,m \,]$

(ii) *maximizar* $\quad g = \sum_{i=1,m} \Pi'_i . z_i$

sujeito a: $\quad \sum_{i=1,m} w_i \, z_i \leq W$

com: $\quad z \geq 0$, *inteiro*

Observe que cada um dos valores Π'_i de (i) são obtidos resolvendo um Problema da Mochila (*Knapsack Problem*) unidimensional onde são selecionadas apenas as peças (w_k, l_k) cuja largura w_k é menor ou igual a largura da peça (w_i, l_i).

Portanto, em (i) tratam-se de m Problemas da Mochila correspondendo a uma primeira etapa, quando m faixas (w_i, L) são preenchidas com as peças (w_k, l_k) tal que $w_k \leq w_i$, ao longo do comprimento L.

São calculados m valores Π'_i respectivamente a cada faixa, que a seguir serão utilizados no Problema da Mochila da segunda etapa. Ou seja, em (ii) a placa de estoque será preenchida com as faixas (w_i, L) ao longo da largura W. Note que o padrão de corte gerado A_j tem a seguinte expressão:

$$A_j = [a_1, a_2, \ldots, a_k, \ldots, a_m] = \qquad (3.3)$$

$$= [\, \sum_{i=1,m} y_1^{(i)} . z_i, \; \sum_{i=1,m} y_2^{(i)} . z_i, \ldots, \; \sum_{i=1,m} y_k^{(i)} . z_i, \ldots, \; \sum_{i=1,m} y_m^{(i)} . z_i \,]$$

onde, a_k representa o k-ésimo elemento do conjunto A_j

$y_k^{(i)}$ representa a k-ésima posição do vetor-solução y do i-ésimo Problema da Mochila de (i)

z_i a i-ésima posição do vetor-solução z

fig. 4 - **PRIMEIRO E SEGUNDO ESTAGIOS**

Na próxima seção serão analisados alguns algoritmos para a solução de (i) e (ii) e, em particular, uma técnica que permite resolver juntos os m Problemas da Mochila de (i).

PROBLEMA DA MOCHILA

Uma variedade de técnicas tem sido desenvolvidas para resolver o Problema da Mochila (unidimensional). Salkin-Kluyver [1975] classificaram os métodos nas seguintes categorias:

(a) programação dinâmica e variações
(b) programação inteira, *branch & bound*, algoritmos enumerativos
(c) busca heurística, métodos lagrangeanos
(d) aproximações em redes

Esta classificação é arbitrária, uma vez que um determinado algoritmo pode utilizar-se de mais de uma categoria. A técnica mais utilizada é programação dinâmica. Bellman [1957] apresentou a seguinte expressão:

$$F_{i+1}(x) = \max \{r.\Pi_{i+1} + F_i(x-r.l_{i+1})\} \qquad (3.4)$$

$$F_i(0) = 0$$

onde, $0 \leq r \leq [x/l_{i+1}]$

$1 \leq i \leq m$

$0 \leq x \leq L$

Gilmore-Gomory [1963] descreveram um método lexicográfico com algumas vantagens sobre (3.4), e posteriormente Gilmore-Gomory [1965] apresentaram um refinamento de (3.4), envolvendo menos aritmética, e comparativamente tão veloz quanto o método lexicográfico.

Supondo as peças (w,l) ordenadas em relação a largura w da forma: $w_1 \leq w_2 \leq w_3 \leq \ldots \leq w_m$, seja $F_i(L)$ definido como o resultado de cada um dos m Problemas da Mochila de (i) em (3.2):

$$F_i(x) = \max \{\Pi_i + F_i(x-l_i),\ F_{i-1}(x)\}, \quad 1 < i \leq m \qquad (3.5)$$

$$F_i(0) = 0, \qquad 1 \leq i \leq m$$

$$F_1(x) = \Pi_1.[x_1/l_1]$$

onde, $0 \leq x \leq L$

Observe que $F_i(x)$ representa o valor da melhor combinação dos comprimentos l preenchidos ao longo de um comprimento $x \leq L$, usando apenas as primeiras i peças (w,l), ou seja, com a ordenação das larguras w, então $S_i = [k \ / \ k=1,i]$ em (i).

Desta forma, o cálculo de $F_m(L)$ implica no cálculo de cada $F_i(L)$ para todo i, $0 \leq i \leq m$, que por sua vez implica no cálculo de cada $F_i(x)$ para todo x, $0 \leq x \leq L$. Portanto, o cálculo recursivo de $F_m(L)$ resulta em cada um dos valores Π'_i de (i).

Resta ainda óbter-se os valores de y de (i) que serão utilizados em (3.3). Para isto, faz-se m *backtrackings* na recursão (3.5). Associa-se inicialmente a $F_1(x)$ o índice 1 se $x \geq I_1$, e caso contrário o índice 0. Assim que $F_i(x)$, $i > 1$ é computado para cada x, associa-se o índice i se $F_i(x)=\Pi_i+F_{i-1}(x-I_i)$, e caso contrário o índice associado a $F_{i-1}(x)$.

Consequentemente, o índice associado a $F_i(x)$, qualquer que seja i e x, é o maior índice r para a qual $y_r^{(i)} > 0$ na combinação dos comprimentos ao longo de x resultante em $F_i(x)$. Finalmente, para encontrar a combinação resultante em $F_i(L)$ procura-se primeiro o seu índice associado; se ele for r e $r > 0$, então a peça (w_r, I_r) pertence a combinação e $y_r^{(i)} \geq 1$.

Repete-se em $F_i(L-I_r)$; se o seu índice associado for r' ($r'=r$ é possível) e $r' > 0$, então a peça $(w_{r'}, I_{r'})$ pertence a combinação e $y_{r'}^{(i)} \geq 1$. Continuando o processo, obtem-se toda combinação de comprimentos $y^{(i)}$ de (3.3).

A segunda etapa (ii) de (3.2) consiste no Problema da Mochila convencional. Dreyfus [1977] apresentou uma função de programação dinâmica eficiente para valores relativamente pequenos de W, e Akink [1983] descreveu um algoritmo enumerativo permitindo tratar valores relativamente grandes de W.

Será utilizado o primeiro método, a seguir:

$$F(x) = \max \{\Pi'_i + F(x-w_i)\}, \quad i = 1,m \qquad (3.6)$$

$$F(x) = 0, \quad x = 1,(w_1-1)$$

onde, $0 \leq x \leq W$

Observe que $F(x)$ agora representa o valor da melhor combinação de larguras w preenchidas ao longo de uma largura $x \leq W$, usando todas as peças (w,I). Desta forma, o cálculo de $F(x)$ implica nos cálculos de $F(x')$, $x' < x$. Portanto, o cálculo

recursivo de $F(W)$ resulta no valor g de (ii).

Para se obter os valores de z_i de (ii) que serão utilizados em (3.3), faz-se apenas um *backtracking* na recursão (3.6) começando com o índice associado a $F(W)$, de forma análoga ao que foi feito anteriormente.

CONSIDERAÇÕES GERAIS

Algumas considerações serão feitas com o objetivo de melhorar a eficiência da implementação dos dois algoritmos descritos.

Dentro de uma iteração as peças (w_a, l_a) cujo $\Pi_a \leq 0$ poderão ser eliminadas.

Da mesma forma, dadas duas peças (w_a, l_a) e (w_b, l_b) em (i), a segunda peça poderá ser eliminada se $l_a \leq l_b$ e $\Pi_a \geq \Pi_b$. Analogamente, dadas duas faixas (w_a, L) e (w_b, L) em (ii), a segunda faixa poderá ser eliminada se $w_a \leq w_b$ e $\Pi_a \geq \Pi_b$.

Como os algoritmos manipulam números inteiros, é conveniente o uso do máximo divisor comum (mdc) transformando o problema num outro equivalente com dimensões possivelmente menores.

Para evitar múltiplas soluções, ou seja, diferentes caminhos conduzindo a uma mesma solução ótima, é conveniente designar uma ordem arbitrária na qual as peças são escolhidas. Por exemplo, as peças serão escolhidas na ordem não-decrescente de seus respectivos Π.

Convém ressaltar que o resultado em (3.2) é função da ordem dos dois estágios definidos nas etapas (i) e (ii), ou seja, se ao contrário, os cortes no primeiro estágio forem produzidos

paralelos à largura W da placa de estoque (a primeira etapa (i) produzindo m faixas (W,l_i)), e em seguida, os cortes do segundo estágio forem produzidos paralelos ao comprimento L da placa de estoque (a segunda etapa (ii) combinando as faixas (W,l_i) ao longo de L), então o resultado possivelmente não será o mesmo do que foi inicialmente proposto em (3.2). Note que isto não é válido se a placa de estoque e as peças de encomenda forem quadradas.

O caso exato é um caso particular do caso não-exato. Para ser resolvido basta substituir na expressão $S_i=[k \ / \ w_k \leq w_i, \ k=1,m]$ de (i) em (3.2) a condição $w_k = w_i$.

Se o material for isotrópico, em cada padrão de corte as peças (w,l) não precisarão mais estar orientadas com a placa de estoque. A etapa (i) consistirá em $2m$ Problemas da Mochila com dimensão $2m$, correspondendo a m faixas (w,L) mais m faixas (l,L) que deverão ser preenchidas com m peças (w,l) mais m peças (l,w) ao longo do comprimento L.

A segunda etapa (ii), portanto, consistirá em mais um Problema da Mochila com dimensão $2m$, correspondendo ao preenchimento das $2m$ faixas ao longo da largura W. Os algoritmos e as considerações descritos para o material anisotrópico são válidos para o material isotrópico.

TERCEIRO ESTÁGIO

Outra sub-classe importante do padrão de corte bidimensional do ponto de vista prático é a que envolve cortes guilhotinados em 3-estágios. Problemas deste tipo aparecem na produção de papel e de vidro.

No primeiro estágio a placa de estoque é cortada na direção paralela ao comprimento L resultando em faixas (p_j, L).

No segundo estágio cada faixa é cortada na direção paralela a largura W resultando em retângulos intermediários (p_j, q_j).

Finalmente no terceiro estágio cada retângulo é cortado novamente na direção paralela ao comprimento L resultando em (w_i, q_j), que após uma apara (caso não-exato) resultam nas peças (w_i, l_i). Um exemplo desta sequência de cortes está ilustrada na fig.5.

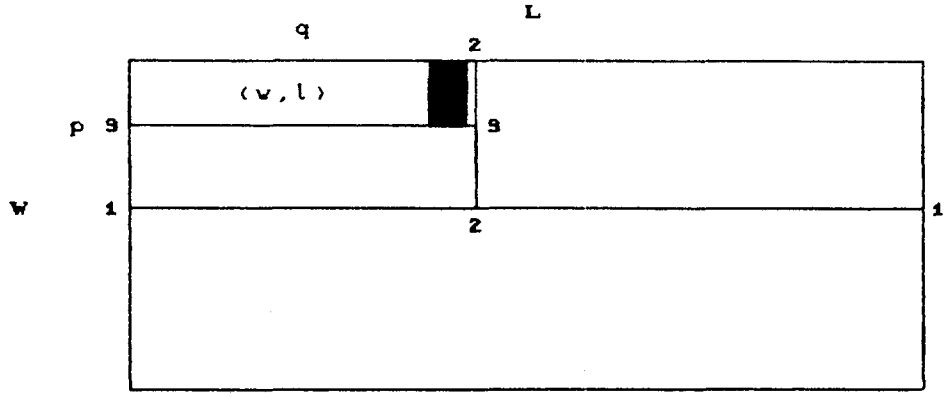

fig.5 - CORTES EM 3-ESTAGIOS

Seja α um vetor com m inteiros não-negativos. Um número finito destes vetores satisfazem a condição $\alpha.w \leq W$. Seja P definido como todas as possíveis discretizações $p = \alpha.w$ em W, ou seja, $P = \{ p \ / \ p \leq W \}$.

Analogamente a 2-estágios, para se resolver um problema de 3-estágios basta resolver numa primeira etapa um problema de 2-estágios para cada faixa (p, L) obtendo-se para cada uma um valor associado Π'.

Resta então, numa segunda etapa, resolver um Problema da Mochila igual a (ii) com dimensão igual a P, e cujas larguras w

correspondem as discretizaçoes p.

Observe que dependendo do número de discretizações possíveis, a técnica descrita acima pode ser muito dispendiosa. A seguir será discutida como alternativa uma heurística para o terceiro estágio.

TERCEIRO ESTÁGIO HEURÍSTICO

A heurística a ser descrita permite encontrar uma solução sub-ótima bem econômica do problema de 3-estágios fazendo algumas pequenas alterações nas duas etapas (i) e (ii) do problema de 2-estágios.

Supõe-se da mesma forma, que para cada faixa (w_i, L), um conjunto de peças (w_k, l_k) tal que $w_k \leq w_i$ são selecionadas para cada i de uma primeira etapa.

A heurística consiste em supor que para uma determinada peça $(w_{k'}, l_{k'})$ com $[w_i / w_{k'}] = \gamma$, então esta poderá preencher a faixa (w_i, L) sobrepondo-se γ vezes ao longo de w_i, conforme ilustrado na fig.6.

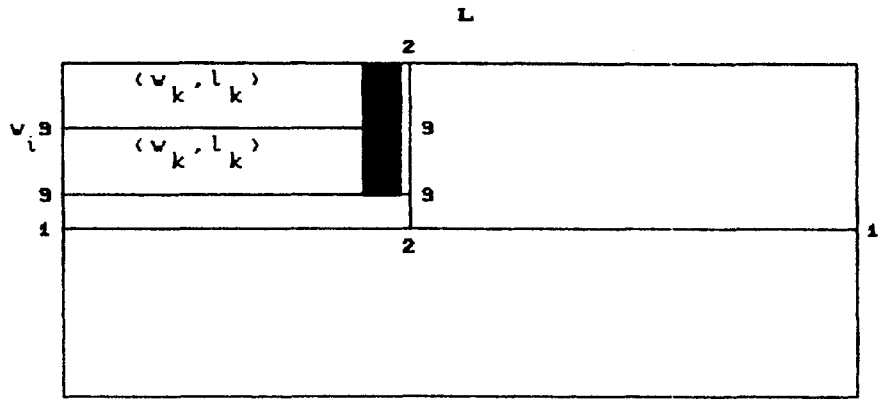

fig.6 - TERCEIRO ESTAGIO HEURISTICO

Portanto, para cada peça (w_k, l_k) temos:

$$\tilde{\Pi}_k = \Pi_k . [w_i / w_k] \qquad (3.7)$$

$$\tilde{y}_k = y_k . [w_i / w_k] \qquad (3.8)$$

Substituindo $\tilde{\Pi}_k$ em (i) de (3.2) e $\tilde{y}_k$ em (3.3), temos novamente o problema de 2-estágios para ser resolvido em duas etapas com os novos valores de (3.7) e (3.8).

Sem perda de generalidade, se $w_1 \leq w_2 \leq \ldots \leq w_m$, então tratam-se de m Problemas da Mochila (com dimensão variável $i=1,m$) na primeira etapa e mais um na segunda etapa (com dimensão = m) que poderão ser resolvidos através do algoritmo da expressão (3.6).

PADRÃO DE CORTE MULTI-ESTÁGIOS GUILHOTINADO

Os padrões de corte multi-estágios guilhotinado podem ser gerados resolvendo-se um Problema da Mochila Generalizado. Reescrevendo a expressão (2.8), temos:

$$\textit{maximize} \quad g = \Pi_1 . a_1 + \Pi_2 . a_2 + \ldots + \Pi_m . a_m \qquad (4.1)$$

sujeito a condição de que $[a_1, a_2, \ldots, a_m]$ *corresponda a um padrão de corte multi-estágios guilhotinado, onde peças* (w_i, l_i) *preenchem uma placa de estoque* (W, L).

Os métodos para a solução de (4.1) não são econômicos comparados com os do problema (3.1) no capítulo anterior. Gilmore-Gomory [1966] apresentaram dois algoritmos através de programação dinâmica para o problema (4.1); um algoritmo básico e um refinamento dele. Os algoritmos são iterativos, não utilizam limitantes na busca da solução ótima e ocupam grande espaço de memória.

Herz [1972] apresentou um procedimento recursivo e comparou-o com os dois algoritmos anteriores. Mostrou a desvantagem do algoritmo básico no número de *loopings* internos, e que o refinamento não leva a uma solução correta. Além disso, transformou o procedimento recursivo em um iterativo similar e comparou-os. O recursivo foi da ordem de 20% mais rápido que o iterativo nas implementações.

REPRESENTAÇÃO EM GRAFO-E-OU

Certos processos de busca consistem em percorrer caminhos num grafo direcionado cujos nós representam um estado do problema e cada arco representa a relação entre os estados por ele conectados.

Considere a placa de estoque como um estado inicial e as peças de encomenda como estados finais. Considere ainda um processo de busca que deverá procurar um caminho através de um grafo, partindo do estado inicial e terminando em um ou mais estados finais.

Um conjunto de cortes horizontais e verticais do tipo guilhotinado pode ser produzido sobre a placa de estoque ((0) da fig.7) correspondendo ao grau de ramificação do estado inicial. Para cada um dos cortes resultam dois retângulos sucessores (A e B da fig.7) correspondendo aos estados intermediários. Observe que cada um destes cortes representam um movimento (ou arco) do estado inicial até estados intermediários ((1) da fig.7).

Associadas a cada movimento existem algumas regras, tais como direção do corte, restrição de simetria, ordenação dos cortes. Um sub-conjunto de cortes pode então ser produzido sobre cada um dos retângulos resultantes, e assim sucessivamente, até encontrar uma ou mais peças de encomenda como estados finais.

Note que neste processo frequentemente resulta o mesmo estado (ou retângulo) gerado através de diferentes sequências de cortes, o que evidencia o fato de não se tratar de uma busca em árvore (C da fig.7).

Uma representação do processo descrito é o grafo-e-ou (*and-or-graph*) ilustrado na fig.7. Um grafo-e-ou é um tipo de estrutura usada para representar soluções de problemas que podem ser resolvidos decompondo-os num conjunto de problemas menores,

todos os quais devendo ser resolvidos.

Esta decomposição (ou redução) gera arcos chamados arcos-e (*and-arcs*) que neste caso serão aos pares correspondendo aos dois retângulos produzidos por um corte sobre o estado antecessor.

Por outro lado, em cada estado podem suceder diversos arcos-ou (*or-arcs*) indicando uma variedade de caminhos (ou possíveis cortes), através de um dos quais o problema original poderá ser resolvido.

Um caminho completo neste grafo representa uma sequência de cortes sobre um estado inicial resultando em um ou mais estados finais.

O caminho a ser escolhido será aquele cujo valor *g* de (1) associado a todas as peças de encomenda geradas neste for maior. Este caminho representa o **melhor** padrão de corte do estado inicial.

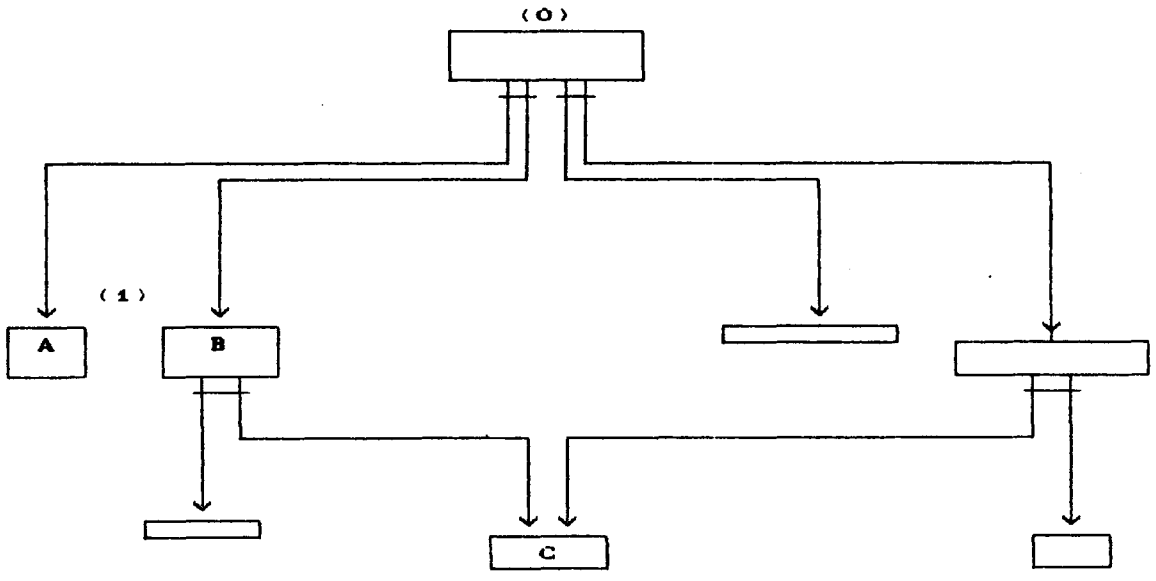

fig.7 - **GRAFO-E-OU**

Este grafo-e-ou a ser percorrido, em princípio, deve ser construído através de regras que definam os movimentos permitidos no espaço do problema.

Na prática, o grafo pode ser muito grande e uma parte dele pode não precisar ser explorada. Ao invés de construí-lo explicitamente e depois explorá-lo, um programa de busca pode representá-lo implicitamente nas regras e gerar explicitamente apenas as partes que ele decidir explorar.

PROCEDIMENTO RECURSIVO

Seja α um vetor com m inteiros não-negativos. Um número finito destes vetores satisfazem a condição $\alpha.w \leq p$ e o mesmo é verdadeiro para $\alpha.l \leq q$. Sejam P e Q definidos como todas as possíveis discretizações num retângulo (p,q), tais que:

$$P = [\ \alpha.w \ /\ \alpha.w \leq p\ ] \qquad (4.2)$$
$$Q = [\ \alpha.l \ /\ \alpha.l \leq q\ ]$$

Herz [1972] demonstrou que para qualquer padrão de corte sobre (p,q), existe um padrão de corte com valor maior ou igual ao primeiro e com todos os cortes em P e Q.

Isto reduz os exames envolvidos no algoritmo para o processo finito de percorrer P e Q. Um padrão de corte com todos os cortes em P e Q será chamado *padrão de corte canônico*.

Um padrão de corte canônico particular é aquele que envolve apenas uma peça de encomenda (w_i, l_i). Os cortes pertencem ao sub-conjunto $[w_i, 2w_i, 3w_i, \ldots]$ de P e ao sub-conjunto $[l_i, 2l_i, 3l_i, \ldots]$ de Q. Este padrão é chamado *padrão de corte homogêneo*.

SIMETRIA

Para um retângulo (p,q), são possíveis $p-1$ cortes ao longo da largura p e $q-1$ cortes ao longo do comprimento q. Portanto, o nó correspondente ao retângulo (p,q) produzirá $p+q-2$ ramificações, gerando duplicação de retângulos nos nós sucessores por causa do efeito de simetria. Estas duplicações podem ser facilmente eliminadas.

Suponha um corte horizontal p' sobre (p,q), produzindo dois retângulos (p',q) e $(p-p',q)$. Obviamente, estes dois retângulos também podem ser produzidos por outro corte horizontal $p-p'$, simetricamente oposto ao primeiro corte com relação a (p,q).

Isto pode ser evitado sem perder a generalidade, simplesmente limitando os cortes horizontais em $\lfloor p/2 \rfloor$ e os cortes verticais em $\lfloor q/2 \rfloor$.

EXCLUSÃO

Observe que as considerações de simetria aplicadas a um padrão de corte canônico reduzem o processo de percorrer P e Q. Foi assumido implicitamente para qualquer peça (w_i, l_i) que $w_i \leq W$ e $l_i \leq L$. No entanto, alguns dos retângulos intermediários (p,q) gerados durante as recursões podem ser menores que determinadas peças . Reescrevendo (4.2) temos:

$$P_{pq} = [\ \alpha.w \ /\ (l_i > q \Rightarrow \alpha_i = 0)\ e\ (\alpha.w \leq p/2)\ ] \qquad (4.3)$$

$$Q_{pq} = [\ \alpha.l \ /\ (w_i > p \Rightarrow \alpha_i = 0)\ e\ (\alpha.l \leq q/2)\ ]$$

Os conjuntos de (4.3) podem ser reescritos de uma forma mais prática para a geração deles. Tomemos inicialmente P_{pq}. Seja uma função $f'(\alpha) = max(l_i \ /\ \alpha_i > 0)$, $0 \leq i \leq m$. Define-se:

$$F(x) = \min(f'(\alpha) \ / \ \alpha.w = x), \quad \forall\ x \in P \qquad (4.4)$$
$$P_{pq} = [\ x \ / \ x \leq p/2,\ F(x) \leq q\ ]$$

Similarmente para Q_{pq}, seja a função $g'(\alpha) = \max(w_i \ / \ \alpha_i > 0]$, $0 \leq i \leq m$. Define-se:

$$G(y) = \min(g'(\alpha) \ / \ \alpha.l = y), \quad \forall\ y \in Q$$
$$Q_{pq} = [\ y \ / \ y \leq q/2,\ G(y) \leq p\ ]$$

Como F e G não dependem de p e q, elas podem ser computadas no início. Christofides-Whitlock [1977] apresentaram uma função recursiva $F_i(x)$ com $1 \leq i \leq m$ e $0 \leq x \leq p/2$ que resolve (4.4).

Uma pequena alteração foi feita sobre a função original com o objetivo de facilitar a implementação. Assuma que as m peças estão na ordem não-decrescente de suas larguras w_i:

$$F_i(x) = \min(F_{i-1}(x),\ \max(l_i,\ \min(F_i(x-k,w_i)))) \qquad (4.5)$$

onde, $\quad 1 \leq k \leq [x/w_i],\ k$ inteiro $\quad , x \geq w_i$

$$F_i(x) = F_{i-1}(x) \quad , 1 \leq x < w_i$$
$$F_i(0) = 0 \quad , i > 0$$
$$F_0(x) = infinito \quad , \forall\ x$$

Note em (4.5) que se $F_m(x) \leq q$, então x pertence a P_{pq} e representa a soma das larguras w_i de uma combinação de peças (w_i, l_i), cujos respectivos comprimentos l_i são todos menores ou iguais a q. A função $G(y)$ e o conjunto Q_{pq} podem ser gerados de maneira similar.

ESTRATÉGIA *BRANCH & BOUND*

A estratégia *branch & bound* consiste em gerar caminhos completos durante a busca, e retornar o melhor caminho encontrado. Cada caminho deve ser explorado verificando se o seu valor parcial não o torna pior do que o melhor caminho encontrado até então.

A recursão a ser descrita gera os caminhos de busca a partir do retângulo (p,q). Os possíveis padrões de corte em (p,q) são:

(a) o padrão de corte canônico homogêneo.

(b) qualquer padrão de corte canônico cujo primeiro corte é horizontal e os cortes restantes têm qualquer direção para cada um dos retângulos resultantes.

(c) qualquer padrão de corte canônico cujo primeiro corte é vertical e os cortes restantes têm qualquer direção para cada um dos retângulos resultantes.

O padrão de corte ótimo para o problema (4.1) é escolhido comparando-se os valores g de todos os padrões gerados nos itens anteriores. As recursões ocorrem obviamente nos itens (b) e (c).

Considerando-se que a busca percorre um grafo, um mesmo retângulo (p,q) pode aparecer em diferentes caminhos durante as recursões ((C) da fig.7). É aconselhável armazenar o seu valor ao invés de recomputá-lo em cada vez que aparecer. Define-se:

$$p_o = max(x \;/\; x \leq p,\; x \in P) \qquad (4.6)$$

$$q_o = max(y \;/\; y \leq q,\; y \in Q)$$

Como os padrões de corte de (p,q) e (p_o,q_o) têm o mesmo valor, então basta memorizar os valores somente para os elementos de P e Q.

Convem ressaltar que a memorização é fundamental para a performance geral do algoritmo recursivo.

LIMITANTES SUPERIORES

Um limitante superior para um retângulo (p,q) pode ser usado para reduzir o espaço de busca sem perder a solução ótima. Seja:

$$u(p,q) = (p.q) \; . \; max(\Pi_i/(w_i.l_i)) \; , \quad com \; i = 1,m \qquad (4.7)$$

Ao se procurar o padrão de corte ótimo para (p,q) através de todos os padrões possíveis para (p,q), a busca deve interromper assim que se tenha encontrado algum padrão de corte cujo valor seja igual a $u(p,q)$ em (4.7).

Suponha que em determinado momento, o melhor valor encontrado até então para (p,q) seja v, e que após um corte horizontal em r, o valor ótimo computado para o retângulo (r,q) é ω. Seja:

$$v_o = v - \omega$$

Note que se $v_o \geq u(p-r,q)$ é inútil investigar o outro retângulo $(p-r,q)$.

Suponha novamente que v é o melhor valor encontrado até então para (p,q), e que $v < v_o$. Então, será inútil investigar $(p-r,q)$ se $v_o - \omega \geq u(p-r,q)$. Como consequência, se $v_o \geq u(r,q) + u(p-r,q)$, então também será inútil investigar o retângulo (r,q).

Em alguns casos o grafo pode ser muito grande e o tempo associado a busca excessivo. O uso de limitantes arbitrários pode tornar possível obter soluções sub-ótimas mais rápidas. Nestes casos, deve-se definir limitantes através de alguma heurística que conduza a soluções boas.

ALGORITMO RECURSIVO

O algoritmo recursivo será descrito abaixo em pseudo linguagem estruturada.

O traçado da solução ótima é uma simples função recursiva para retornar da memória o vetor-solução a de (4.1). Para isto, a função percorre somente o caminho ótimo desde (W,L) armazenado na memória pela rotina de recursão.

Extensivamente, pode fornecer a sequência dos cortes e desenhar o padrão de corte ótimo.

```
{ início do algoritmo recursivo }
{ parâmetros m, W, L, w_i, l_i, Π_i, com i=1,m }
construa  F, G, P, Q  para (W,L)   (4.4)(4.5)
compute   g = recursão(W, L, 0)    (4.1)
compute   a = traçado da solução ótima  (4.1)
{ fim do algoritmo recursivo }
```

```
{ início da rotina de recursão }
{ parâmetros  p, q, v_o }
se  v_o ≥ u(p,q)   (4.7)
   então  retorne sem solução

   caso contrário
      início
      compute  p_o, q_o  (4.6)
      se o retângulo (p_o, q_o) existe na memória
         então  retorne com solução da memória
```

```
        caso contrário
            início
            { cortes homogêneos }
            enquanto a solução não for ótima faça para cada padrão
                início
                compute um padrão de corte homogêneo
                armazene a melhor solução v
                fim

            { cortes horizontais }
            enquanto a solução não for ótima faça para cada r∈ P_pq
                início
                compute  ω = recursão(r, q, max(v,v_o) - u(p-r,q))
                compute  φ = ω + recursão(p-r, q, max(v,v_o)-ω))
                armazene a melhor solução v
                fim

            { cortes verticais }
            enquanto a solução não for ótima faça para cada r∈ Q_pq
                início
                compute  ω = recursão(r, q, max(v,v_o) - u(p,q-r))
                compute  φ = ω + recursão(p, q-r, max(v,v_o)-ω))
                armazene a melhor solução v
                fim

            armazene na memória a melhor solução v de (p_o,q_o)
            retorne com melhor solução v
            fim
    fim
{ fim da rotina de recursão }
```

ORDENACÃO DOS CORTES

Suponha que num determinado nó um retângulo (p,q) sofre um corte horizontal r produzindo dois retângulos (r,q) e $(p-r,q)$. Suponha ainda que num nó sucessor o retângulo resultante $(p-r,q)$ sofre um corte horizontal s, $r \leq s \leq [(p-r)/2]$ produzindo mais dois retângulos (s,q) e $(p-r-s,q)$.

Observe que estes tres retângulos (r,q), (s,q) e $(p-r-s,q)$ poderiam ter sido produzidos fazendo-se primeiro o corte horizontal s sobre (p,q), e em seguida o corte horizontal r sobre $(p-s,q)$, conforme ilustrado na fig.8.

Este tipo de duplicação pode ser evitado sem perder a generalidade simplesmente introduzindo uma ordem arbitrária nos cortes.

Por exemplo, se um retângulo sofre um corte horizontal t, então todos os cortes horizontais subsequentes sobre os dois retângulos resultantes deverão ser maiores ou iguais a t. Esta restrição impõe limitantes inferiores para os cortes subsequentes.

Para implementá-la no algoritmo recursivo acima descrito, basta substituir para os cortes horizontais a condição "faça para cada $r \in P_{pq}$" por "faça para cada r tal que $r \in P_{pq}$ e $r \geq t$ ", onde t representa o corte horizontal produzido no nó antecessor. O valor de t pode ser passado do nó antecessor para os nós sucessores através de um simples parâmetro de valor na rotina de recursão.

Observe que a ordenação dos cortes descrita reduz o grafo-e-ou da fig.8 numa árvore-e-ou (*and-or-tree*), mas não altera o grafo-e-ou da fig.7.

Portanto, a representação do problema continua sendo um grafo-e-ou com tamanho reduzido. Procede-se de maneira similar para os cortes verticais.

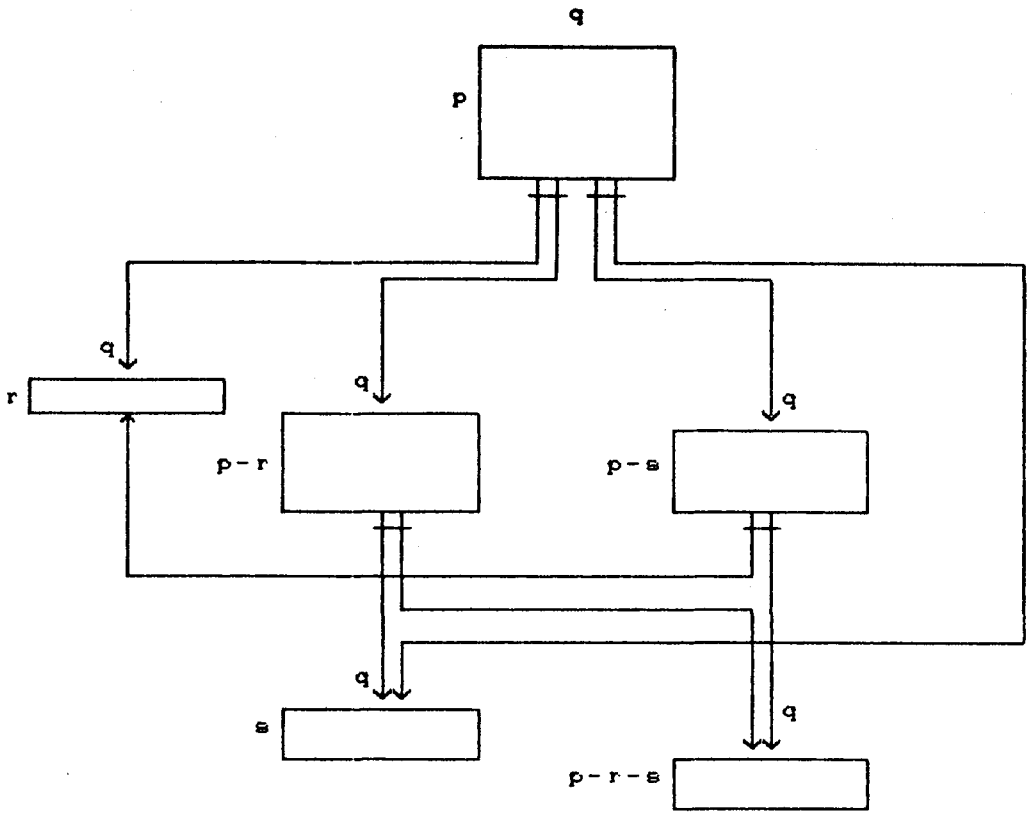

fig.8 - SEM ORDENACAO NOS CORTES HORIZONTAIS

HEURÍSTICA PARA PROBLEMAS GRANDES

Beasley [1985] sugeriu uma heurística para reduzir o custo computacional de problemas com os conjuntos P e Q grandes.

Redefine-se P e Q de (4.2) de forma que eles se tornem menores. Será ilustrado a redefinição de P (note que a redefinição de Q é similar).

Seja M um limitante da dimensão de P , tal que $dim\ (P) \leq M$. M pode ser determinado considerando-se o número de operações (ou capacidade de armazenagem) requerido pela rotina de recursão, e

escolhendo-se o máximo número de operações (ou máxima capacidade de armazenagem) que se considere computacionalmente viável. Para se redefinir P usa-se o seguinte procedimento:

(1) Seja $N = [1,2,\ldots,m]$, onde N representa as peças de encomenda.

(2) $P = [\, x / x = \sum_{i \in N} \alpha_i w_i,\ 1 \leq x \leq M - min(w_j / j \in N),\ \alpha_i \geq 0\ int.\ e\ i \in N]$

(3) Se $dim(P) \leq M$, então pare com P; caso contrário faça (4)

(4) Defina: $w_j = min(\, w_i \ / \ i \in N\,)$

$N = N - [j]$ e faça (2)

Basicamente, este procedimento descrito vai eliminando a peça de encomenda i com menor largura w_i do conjunto N até que o conjunto P tenha a dimensão desejada M. Obviamente, com esta redefinição de P e Q não se pode garantir que a solução será ótima.

CONSIDERACÕES GERAIS

O procedimento recursivo é válido para gerar padrões de corte irrestrito, ou seja, quando não há limitantes para o número de ocorrências de cada peça de encomenda na solução. Esta suposição torna possível considerar material isotrópico e anisotrópico (com direções ortogonais) ao mesmo tempo.

Suponha que algumas peças devem ter uma orientação horizontal, enquanto que outras podem ter orientação horizontal ou

vertical. Portanto, basta duplicar as últimas com as dimensões trocadas, e tratar o problema como se todas as peças fossem diferentes e devessem ter uma orientação horizontal.

O procedimento recursivo utiliza basicamente busca em profundidade primeiro (*depth-first-search*), que pode ser visto claramente através de uma representação do problema em grafo-e-ou.

Existem outras alternativas a serem exploradas para uma busca mais eficiente.

ALGORITMO AO*

Rich [1983] descreveu o algoritmo AO* de busca heurística em grafo-e-ou.

Em síntese, percorre-se o grafo partindo do nó inicial e seguindo o melhor caminho, e acumula-se o conjunto de nós deste caminho que não foram ainda expandidos.

Escolhe-se um deles e o expande. Adiciona-se ao grafo os seus sucessores e se computa um valor para cada um deles através de uma função de avaliação.

Altera-se o valor do nó recentemente expandido, para refletir a nova informação proveniente dos seus sucessores. Propaga-se estas alterações regressivamente (*backward*) através do grafo. Em cada nó visitado durante esta propagação, decide-se qual dos seus arcos sucessores é o mais promissor e se marca-o como parte do melhor caminho atual. Isto deve causar alteração no melhor caminho atual. Os nós expandidos devem ser re-examinados para que o melhor caminho seja selecionado. É importante que os seus valores estejam bem estimados, o que implica na necessidade de uma boa escolha da função de avaliação.

A adaptação deste algoritmo ao Problema do Corte de Estoque Bidimensional depende de uma boa escolha da função de avaliação. Melhores resultados têm sido encontrados ao experimentar-se uma busca híbrida, ou seja, combinando-se outras buscas dentro do algoritmo AO*.

Uma técnica que tem se mostrado eficaz é a busca estagiada (*staged search*) descrita em Pearl [1984] que basicamente combina duas estratégias diferentes: "subindo-morro" (*hill climbing*) com "busca-o-melhor-primeiro" (*best-first-search*).

Os trabalhos ainda estão em andamento, e não serão analisados neste estudo.

RESULTADOS COMPUTACIONAIS E CONCLUSÕES

Os algoritmos foram codificados na linguagem de programação *PASCAL* e os tempos de execução referem-se à utilização de um micro-computador do tipo IBM-PC.

O material será suposto anisotrópico, ou seja, não se permite rotações das peças (w_i, l_i) dentro da placa (W, L). Os cortes serão do tipo guilhotinado e a direção dos cortes do primeiro estágio poderá ser paralela a largura W ou ao comprimento L.

Nas tabelas 2 e 5, os resultados exibidos correspondem a soluções arredondadas. A última coluna refere-se ao tempo de execução de todas as iterações do algoritmo Simplex.

Foi implementado uma heurística de corte da maneira descrita por Gilmore-Gomory [1963] que interrompe o algoritmo Simplex caso a convergência torne-se lenta. O critério utilizado foi parar quando em 10 iterações a perda não se reduzir em 0.1%.

Adotou-se também uma tolerância de 0.1% para a condição de otimalidade discutida no segundo capítulo.

Os tres problemas A,B,e C da tabela.1 possuem respectivamente $m = 5, m = 10$, e $m = 20$; e um intervalo de demanda $[b1, b2]$. A placa de estoque (W, L) está disponível para cada um dos problemas em quantidade supostamente ilimitada.

A tabela.2 mostra os resultados computacionais da solução dos problemas A,B,e C utilizando os algoritmos para corte em 2-estágios, 3-estágios heurístico, e multi-estágios. $Dmax$ representa, apenas para o corte em multi-estágios, o número máximo de discretizações, ou seja, o maior número de elementos dos conjuntos P e Q.

A		B		C		demanda	
w	l	w	l	w	l	$b1$	$b2$
30	60	30	60	30	60	100	100
60	60	60	60	60	60	100	100
90	45	90	45	90	45	200	200
45	135	45	135	45	135	100	150
60	45	60	45	60	45	100	200
		50	100	50	100	100	100
		75	50	75	50	100	100
		75	75	75	75	180	220
		50	120	50	120	200	200
		100	115	100	115	108	132
				50	60	100	100
				75	60	100	100
				100	45	200	200
				45	115	100	150
				60	120	100	200
				50	90	100	100
				90	75	100	100
				30	75	180	220
				50	135	200	200
				45	30	108	132
placa disponivel $(W,L) = (170,230)$							

TABELA 1 - PROBLEMAS A,B,C

	numero estagios	dmax	perda (%)	placas cortadas	numero padroes	numero iteracoes	tempo (min)
A	$B = 5$						
	2	--	13.5	66	5	13	0.1
	3	--	10.4	65	4	12	0.1
	multi	12	9.8	65	4	14	0.4
B	$B = 10$						
	2	--	7.5	185	10	30	0.6
	3	--	7.1	182	9	27	0.6
	multi	30	6.6	181	10	39	4.3
C	$B = 20$						
	2	--	4.0	352	18	91	5.7
	3	--	3.8	348	20	119	7.5
	multi	35	3.3	351	18	154	42.9

TABELA 2 - RESULTADOS COMPUTACIONAIS (prob. A,B,C)

Observe que na soluçao do problema A a perda em multi-estágios é inferior à perda em 3-estágios, apesar do número de placas cortadas ser o mesmo.

Isto indica que a solução em multi-estágios produziu mais peças do que a outra, obtendo assim um melhor aproveitamento da mesma área representada pelas placas cortadas. Evidentemente, este aumento de produção respeita o intervalo de demanda de cada peça.

Observe ainda que na solução do problema C o número de placas cortadas em 3-estágios é inferior ao multi-estágios; o que não implica numa perda global menor.

Uma vez que estamos minimizando a perda de material e não o número de placas cortadas, basta verificar que:

$0.038 * 348 > 0.033 * 351$

Uma outra observação é o aumento sensível do número de iterações em função do aumento do número de peças **n**.

Nestes tres problemas, a perda percentual decresce conforme se aumenta o número máximo de estágios permitidos.

Bealey [1985] gerou sub-problemas aleatórios restringindo as dimensões das pecas i da seguinte forma:

$$0.25 * W <= w_i <= 0.75 * W \qquad (5.1)$$

$$0.25 * L <= l_i <= 0.75 * L$$

Conforme os seus resultados, além do número de discretizações associado a cada sub-problema gerado ter sido relativamente pequeno, as soluções N-estagiadas comparadas com as multi-estagiadas resultaram pouco diferentes.

Será proposto um outro intervalo para substituir (5.1) que

dificulta a soluçao do sub-problema, mas representa melhor casos reais:

$$0.15 * W <= w_i <= 0.65 * W \qquad (5.2)$$

$$0.15 * L <= l_i <= 0.65 * L$$

Usando (5.2) foram gerados 7 problemas aleatórios com m = 10 e demanda 100 <= b_i <= 1000. Os problemas 1,2,e 3 aparecem na tabela.3 e os problemas 4,5,e 6 na tabela.4.

	1			2			3	
W	L		W	L		W	L	
100	156		253	294		318	473	
w	l	b	w	l	b	w	l	b
27	32	704	130	130	566	141	128	336
16	50	435	49	134	363	109	151	489
47	66	164	45	52	178	76	142	187
37	46	797	64	134	977	204	124	192
31	51	246	38	100	674	110	165	750
47	51	322	121	146	946	142	168	920
48	44	234	48	123	851	82	238	383
49	45	899	83	79	539	169	75	920
59	78	109	42	187	769	165	185	338
21	87	713	112	92	409	162	184	981

TABELA 9 - PROBLEMAS 1,2,3

A tabela.5 mostra os resultados computacionais dos problemas 1,2,3,4,5,6,7 com cortes em 2-estágios, 3-estágios heurístico e multi-estágios, conforme anteriormente.

Com o objetivo de testar a heurística sugerida por Beasley

[1985] e descrita no capítulo anterior, *dmax* foi inicialmente limitado em *M* = 100 para os cortes em multi-estágios.

Observe que nos problemas 2,3,e 7 estas soluções sub-ótimas resultaram em perdas maiores do que as encontradas sem limitar *dmax*. E inclusive maiores do que as soluções em 2(3)-estágios.

4			5			6		
W	*L*		*W*	*L*		*W*	*L*	
501	556		750	806		507	999	
w	*l*	*b*	*w*	*l*	*b*	*w*	*l*	*b*
136	113	513	205	163	704	110	488	225
81	177	420	121	257	435	108	176	715
237	237	193	356	343	164	182	332	656
184	163	911	276	236	797	275	583	199
153	182	351	230	264	246	229	449	487
233	183	441	350	266	322	90	619	151
241	155	183	361	225	234	238	579	750
246	160	925	369	232	899	179	360	824
296	278	161	445	404	109	103	501	436
103	310	599	155	449	713	192	396	135

TABELA 4 - PROBLEMAS 4,5,6

Os tempos de execução (vide tabela.5) dos cortes em multi-estágios são bem maiores que os 2(3)-estágios; o que também verifica-se nos tempos médios por iteração.

É de se esperar que nos cortes em multi-estágios o custo por iteração esteja relacionado com *dmax*. No entanto, nos problemas 2 e 3, apesar dos *dmax* associados serem aproximadamente iguais, o tempo médio por iteração nos dois problemas foi bem diferente. O que caracteriza uma certa imprevisão da busca no grafo.

	numero estagios	dmax	perda (%)	placas cortadas	numero padroes	numero iteracoes	tempo (min)
1	(W,L) = (100,156)						
	2	--	11.7	601	10	8	0.2
	3	--	11.1	599	10	10	0.3
	multi	63	8.5	580	10	16	5.2
2	(W,L) = (253,294)						
	2	---	6.7	849	10	22	0.9
	3	---	6.7	849	10	22	0.9
	multi	100	7.0	849	10	16	5.1
	multi	131	6.5	846	10	21	31.6
3	(W,L) = (318,473)						
	2	---	14.8	900	10	21	0.6
	3	---	14.4	897	10	27	0.8
	multi	100	16.4	916	10	27	6.5
	multi	139	12.9	880	10	23	7.6
4	(W,L) = (501,556)						
	2	---	11.3	635	10	8	0.2
	3	---	10.9	632	10	11	0.4
	multi	100	10.1	625	10	11	4.4
	multi	111	9.0	616	10	16	11.0
5	(W,L) = (750,806)						
	2	---	13.1	608	10	13	0.4
	3	---	11.9	599	10	16	0.5
	multi	100	11.8	599	10	11	5.4
	multi	114	9.9	586	10	30	21.7
6	(W,L) = (506,999)						
	2	---	14.2	794	10	22	0.6
	3	---	12.8	781	10	16	0.5
	multi	100	12.8	782	10	16	4.9
	multi	156	12.8	781	10	13	5.3
7	(W,L) = (785,882)						
	2	---	13.2	837	9	19	0.8
	3	---	13.2	837	9	19	0.8
	multi	100	17.8	886	10	20	13.2
	multi	164	11.9	825	10	27	42.6

TABELA 5 - RESULTADOS COMPUTACIONAIS (prob.1,2,3,4,5,6,7)

Nos problemas 2 e 7, as soluções encontradas em 2-estágios e 3-estágios foram idênticas, ou seja, todos os padrões de corte da solução ótima em 3-estágios são, na verdade, padrões de corte em 2-estágios.

Observe que os tempos de execução destas soluções são iguais, o que mostra que a inclusão do terceiro estágio heurístico não implica num aumento de custo considerável com relação aos cortes em 2-estágios.

CONCLUSÕES

O modelo proposto por Gilmore-Gomory [1965] para o Problema do Corte de Estoque Bidimensional mostrou-se eficiente através do *Método Simplex*, e da *programação dinâmica* utilizada para resolver o sub-problema da geração de padrões de corte guilhotinado e irrestrito em 2-estágios.

O terceiro estágio heurístico aqui proposto também mostrou-se eficiente. A sua implementação a partir do modelo anterior é simples, envolvendo um pequeno aumento de custo computacional por iteração.

Os resultados mostraram o decréscimo significativo da perda de material ao utilizar-se o terceiro estágio.

O *procedimento recursivo* proposto por Herz [1972] para o sub-problema da geração de padrões de corte guilhotinado e irrestrito em multi-estágios, com algumas modificações e adaptado ao *Método Simplex*, mostrou-se eficiente para resolver problemas de tamanho moderado.

A medida que o número de discretizações aumenta, o grafo pode ultrapassar dezenas de milhares de nós tornando sua busca

exaustiva em cada iteração do *Método Simplex*.

Entretanto, os resultados mostraram que em certos problemas o corte em multi-estágios pode encontrar soluções com perdas consideravelmente menores do que os cortes 2(3)-estágios.

Beasley [1985] generalizou os sub-problemas N-estagiados e multi-estagiados (não-estagiados) através de fórmulas recursivas e sugeriu uma heurística para sub-problemas grandes.

Os resultados aqui encontrados mostraram que esta heurística deve ser usada com cautela, uma vez que ela pode levar a soluções sensivelmente inferiores à solução ótima. Há casos em que o uso da heurística levou a perdas inferiores aos cortes em 2(3)-estágios.

Uma nova representação do sub-problema da geração de padrões de corte em multi-estágios foi aqui sugerida através de um *grafo-e-ou*. Esta nova representação permite visualizar o procedimento recursivo proposto por Herz [1972] como uma *busca em profundidade primeiro*.

Além disto, abre espaço para a pesquisa de outras buscas em *grafo-e-ou* possivelmente mais eficientes para a solução. Por exemplo, o *algoritmo AO** descrito em Rich [1983] ou a *busca estagiada* descrita em Pearl [1984].

Desta forma, o Problema do Corte de Estoque Bidimensional que tem sido mais explorado pelas técnicas da Pesquisa Operacional, poderia também beneficiar-se de técnicas mais exploradas dentro da Inteligência Artificial.

REFERENCIAS:

(em ordem alfabética)

Adamowicz, M. and Albano, A. [1976] - "A Solution of the Rectangular Cutting Stock Problem", IEEE Transactions on Systems, Man, and Cybernetics vol.SMG-6, 302-310.

Akink, U. [1983] - "An Algorithm for the Knapsack Problem", IIE Transactions 15, 31-36.

Albano, A. and Sapuppo, G. [1980] - "Optimal Allocation of Two Dimensional Irregular Shapes Using Heuristic Search Methods", IEEE Transactions on Systems, Man, and Cybernetics vol.SMC-10, 242-248.

Arcaro, V. [1988] - "Recorte de Estoque Unidimensional", Dissertação de Mestrado, ICMSC-USP.

Beasley, J.E. [1985]# - "An Exact Two Dimensional Non Guillotine Cutting Tree Search Procedure", Operations Research 33, 49-64.

Beasley, J.E. [1985] - "Algorithms for Unconstrained Two Dimensional Guillotine Cutting", Journal of Operatinal Research Society 4, 297-306.

Bellman, R. [1957] - "Dynamic Programming", Princeton University Press.

Christofides, N. and Whitlock, C. [1977] - "An Algorithm for Two Dimensional Cutting Problems", Operations Research 25, 30-44.

De Cani, P. [1978] - "A Note on the Two Dimensional Rectangular Cutting Stock Problem", Journal of the Operational Research Society 29, 703-706.

Dreyfus, S.E. [1977] - "The Art and Theory of Dynamic Programming", Academic Press, New York.

Dyson, R.G. and Gregory, A.S. [1974] - "The Cutting Stock Problem in the Flat Glass Industry", Operational Research Quartely

25, 41-53.

Eisemann, K. [1957] - "The Trim Problem", Management Science 3, 279-284.

Farley, A. [1983] - "Trim Loss Pattern Rearrangement and its Relevance to the Flat Glass Industry", European Journal of Operational Research 14, 386-392.

Farley, A. [1988] - "Mathematical Programming Models for Cutting Stock Problems in the Clothing Industry", Journal of Operational Research Society 39, 41-53.

George, J.A. and Robinson, D.F. [1980] - "A Heuristic for Packing Boxes into a Container", Comp. & Ops.Res. 7, 147-156.

Gilmore, P.C. and Gomory, R.E. [1961] - "A Linear Programming Approach to the Cutting Stock Problem", Operations Research 9, 849-859.

Gilmore, P.C. and Gomory, R.E. [1963] - "A Linear Programming Approach to the Cutting Stock Problem, Part II", Operations Research 11, 863-888.

Gilmore, P.C. and Gomory, R.E. [1965] - "Multistage Cutting Stock Problems of Two and More Dimensions", Operations Research 13, 94-120.

Gilmore, P.C. and Gomory, R.E. [1966] - "The Theory and Computations of Knapsack Functions", Operations Research 14, 1045-1074.

Herz, J.C. [1972] - "Recursive Computational Procedure for Two Dimensional Stock Cutting", IBM J.Res.Develop. 16, 462-469.

Hinxman, A.I. [1980] - "The Trim Loss and Assortment Problems: A Survey", European Journal of Operational Research 5, 8-18.

Israni, S.S. and Sanders, J.L. [1984] - "Performance Testing of Rectangular Parts Nesting Heuristics", International Journal of Production Research, 437-456.

Kantarovich, L.V. [1939] - "Mathematical Methods of Organizing and Planning Production", reedição na Management Science 6, 366-422.

Pearl, J. [1984] - "Heuristics: Intelligent Search Strategies for Computer Problem Solving", Addison-Wesley Publishing Company, 65-69.

Rich, E. [1983] - "Artificial Intelligence", International Student Edition - McGraw-Hill, 87-94.

Salkin, H.M. and De Kluyver, C.A. [1975] - "The Knapsack Problem: A Survey", Nav.Res.Log.Quart 22, 127-144.

Schneider, W. [1988] - "Trim Loss Minimization in a Crepe Rubber Mill; Optimal Solution versus Heuristic in the 2(3) Dimensional Case", European Journal of Operational Research 34, 273-281.

Wang, P.Y. [1983] - "Two Algorithms for Constrained Two Dimensional Cutting Stock Problems", Operations Research 31, 573-586.

REFERENCIAS:

(em ordem cronológica)

Kantarovich, L.V. [1939] - "Mathematical Methods of Organizing and Planning Production", reedição na Management Science 6, 366-422.

Eisemann, K. [1957] - "The Trim Problem", Management Science 3, 279-284.

Bellman, R. [1957] - "Dynamic Programming", Princeton University Press.

Gilmore, P.C. and Gomory, R.E. [1961] - "A Linear Programming Approach to the Cutting Stock Problem", Operations Research 9, 849-859.

Gilmore, P.C. and Gomory, R.E. [1963] - "A Linear Programming Approach to the Cutting Stock Problem, Part II", Operations Research 11, 863-888.

Gilmore, P.C. and Gomory, R.E. [1965] - "Multistage Cutting Stock Problems of Two and More Dimensions", Operations Research 13, 94-120.

Gilmore, P.C. and Gomory, R.E. [1966] - "The Theory and Computations of Knapsack Functions", Operations Research 14, 1045-1074.

Herz, J.C. [1972] - "Recursive Computational Procedure for Two Dimensional Stock Cutting", IBM J.Res.Develop. 16, 462-469.

Dyson, R.G. and Gregory, A.S. [1974] - "The Cutting Stock Problem in the Flat Glass Industry", Operational Research Quartely 25, 41-53.

Salkin, H.M. and De Kluyver, C.A. [1975] - "The Knapsack Problem: A Survey", Nav.Res.Log.Quart 22, 127-144.

Adamowicz, M. and Albano, A. [1976] - "A Solution of the Rectangular Cutting Stock Problem", IEEE Transactions on Systems,

Man, and Cybernetics vol.SMG-6, 302-310.

Dreyfus, S.E. [1977] - "The Art and Theory of Dynamic Programming", Academic Press, New York.

Christofides, N. and Whitlock, C. [1977] - "An Algorithm for Two Dimensional Cutting Problems", Operations Research 25, 30-44.

De Cani, P. [1978] - "A Note on the Two Dimensional Rectangular Cutting Stock Problem", Journal of the Operational Research Society 29, 703-706.

George, J.A. and Robinson, D.F. [1980] - "A Heuristic for Packing Boxes into a Container", Comp. & Ops.Res. 7, 147-156.

Albano, A. and Sapuppo, G. [1980] - "Optimal Allocation of Two Dimensional Irregular Shapes Using Heuristic Search Methods", IEEE Transactions on Systems, Man, and Cybernetics vol.SMC-10, 242-248.

Hinxman, A.I. [1980] - "The Trim Loss and Assortment Problems: A Survey", European Journal of Operational Research 5, 8-18.

Akink, U. [1983] - "An Algorithm for the Knapsack Problem", IIE Transactions 15, 31-36.

Farley, A. [1983] - "Trim Loss Pattern Rearrangement and its Relevance to the Flat Glass Industry", European Journal of Operational Research 14, 386-392.

Wang, P.Y. [1983] - "Two Algorithms for Constrained Two Dimensional Cutting Stock Problems", Operations Research 31, 573-586.

Rich, E. [1983] - "Artificial Intelligence", International Student Edition - McGraw-Hill, 87-94.

Israni, S.S. and Sanders, J.L. [1984] - "Performance Testing of Rectangular Parts Nesting Heuristics", International Journal of Production Research, 437-456.

Pearl, J. [1984] - "Heuristics: Intelligent Search Strategies for Computer Problem Solving", Addison-Wesley Publishing Company, 65-69.

Beasley, J.E. [1985]# - "An Exact Two Dimensional Non

Guillotine Cutting Tree Search Procedure", Operations Research 33, 49-64.

Beasley, J.E. [1985] - "Algorithms for Unconstrained Two Dimensional Guillotine Cutting", Journal of Operatinal Research Society 4, 297-306.

Schneider, W. [1988] - "Trim Loss Minimization in a Crepe Rubber Mill; Optimal Solution versus Heuristic in the 2(3) Dimensional Case", European Journal of Operational Research 34, 273-281.

Farley, A. [1988] - "Mathematical Programming Models for Cutting Stock Problems in the Clothing Industry", Journal of Operational Research Society 39, 41-53.

Arcaro, V. [1988] - "Recorte de Estoque Unidimensional", Dissertação de Mestrado, ICMSC-USP.