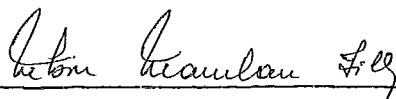


SOBRE O PROBLEMA DE CORTE A DUAS DIMENSÕES - UM
PROCEDIMENTO ALTERNATIVO DE CORTE

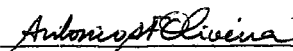
ARLINDO GOMES DE ALVARENGA

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE
PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JA
NEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO
GRAU DE DOUTOR EM CIÊNCIAS (D.Sc.)

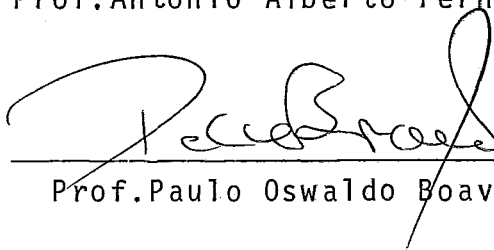
Aprovada por:



Prof. Nelson Maculan Filho
(Orientador)



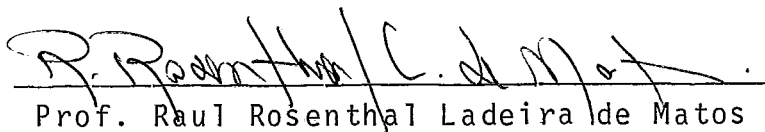
Prof. Antonio Alberto Fernandes Oliveira



Prof. Paulo Oswaldo Boaventura Netto



Prof. Ruderico Ferraz Pimentel



Prof. Raul Rosenthal Ladeira de Matos

Rio de Janeiro, RJ - Brasil

Novembro de 1982

ALVARENGA, ARLINDO GOMES DE

Sobre o Problema de Corte a Duas Dimensões - Um Procedimento Alternativo de Corte. [Rio de Janeiro] 1982.

ix, 95 p., 29,7 cm (COPPE-UFRJ, D.Sc., Engenharia de Sistemas, 1982).

Tese - Universidade Federal do Rio de Janeiro. Programa de Sistemas.

1. Otimização I. COPPE/UFRJ II. Título (Série)

Dedico este trabalho a Ana Luiza,
minha filha, Luiza Maria, minha es
posa, Manoel e Aparecida, meus pais,
e ao amigo casal, Adelino e Luiza.

AGRADECIMENTOS

Ao Professor Nelson Maculan Filho, pela orientação e estímulos recebidos durante a realização deste trabalho.

Ao Professor Antonio Alberto Fernandes Oliveira pelo seu trabalho de orientação durante a ausência do Professor Nelson Maculan, pelos ensinamentos recebidos e sua participação na banca.

Aos Professores: Paulo O. Boaventura Netto, Ruderico Ferraz Pimentel, Raul Rosenthal Ladeira de Matos, pela participação na banca.

Aos colegas Francisco Edmar, Manuel, Clécio e Francisco Negreiros, por discussões e sugestões recebidas e, em especial, por amigos que sempre foram.

Aos demais colegas, professores, secretárias e funcionários da COPPE pelo bom convívio e relacionamento.

Ao amigo ausente professor Paulo Brito Braz e Silva pelo estímulo e dedicação na resolução de meus problemas junto a UFPb.

À Universidade Federal da Paraíba pelo apoio financeiro.

À Norma e Gilmar, respectivamente, pela datilografia e confecção das figuras.

A todas as pessoas que, de uma forma ou de outra, contribuíram na realização deste trabalho.

R E S U M O

Neste trabalho consideramos o problema de corte a duas dimensões, ou seja, o problema de se atender a partir de placas retangulares (L, W) do estoque, a uma demanda de peças também retangulares, em número especificado.

Quando expressado como um modelo de programação linear inteira, a complexidade combinatória inerente do problema, torna a computação inviável.

Nosso objetivo se restringe a estudar o problema de corte a duas dimensões do tipo guilhotina, e um procedimento alternativo de corte para obter soluções aproximadas é sugerido. Uma versão Branch-Bound do procedimento foi implementada e os resultados computacionais mostram um bom desempenho do sistema à medida que crescemos o porte do problema, ou seja, um bom aproveitamento do material com tempo computacional relativamente modesto.

A B S T R A C T

In this work we consider the two-dimensional cutting-stock problem, i. e., the problem of supplying from rectangular plates (L, W) of the stock, an order for a specified number of pieces, also rectangular.

When It is expressed by an integer linear programming model, the combinatorial complexity inherent to the problem, is computationally overbundening.

Our objective is restricted to study the two-dimensional cutting-stock problem of the guilhotine type and an alternative cut procedure to get approximate solutions is suggested. A Branch-Bound version of this procedure was implemented, and computational results show a good performance of the system as problem size increase, i. e., a good material profit with a relatively inexpensive computational time.

R É S U M É

Dans ce travail on considère le problème du découpage à deux dimensions, c'est à dire celui de l'obtention d'une collection de pièces rectangulaires, en nombre spécifié pour chaque pièce, en partant d'un stock de plaques rectangulaires (L,W).

La complexité combinatoire du problème empêche la résolution du modèle de programmation entière qui lui peut être associé.

L'object du travail est limité à l'étude des problèmes bidimensionnels du type "guillotine", pour lesquels on suggère une procédure alternative pour le découpage, amenant à des solutions approchées.

Une version SEP de cette procédure a été implémentée, dont les résultats computationels maintiennent une efficacité assez bonne, au fur et à mesure que l'ordre de grandeur du problème s'accroît; c'est à dire, on obtient des solutions où on profite assez bien du matériel découpé et cela avec un temps de processement relativement modeste.

I N D I C E

	Página
INTRODUÇÃO.....	1
I. ASPECTOS TEÓRICOS GERAIS SOBRE O PROBLEMA DE CORTE..	4
I.1 - Introdução.....	5
I.2 - O Problema de Corte no Caso 1 - Dimensional....	5
I.3 - O Problema Knapsack.....	8
I.4 - O Problema de Corte a Duas Dimensões.....	12
I.5 - Corte Tipo Guilhotina.....	15
II. PROCEDIMENTO DE CORTE A DOIS ESTÁGIOS ALTERNADO.....	23
II.1 - Introdução.....	24
II.2 - Fluxograma do Procedimento de Corte a Dois Es tágios.....	24
III. SOBRE UM CASO ESPECÍFICO DO PROBLEMA DE CORTE.....	37
III.1 - Introdução.....	38
III.2 - Considerações Teóricas Gerais sobre o Proble- ma.....	38
III.3 - Procedimento de Corte a Dois Estágios (Alter- nativas de Melhoramento da Solução).....	51
IV. EXPERIÊNCIA COMPUTACIONAL E CONCLUSÕES.....	66
IV.1 - Introdução.....	67
IV.2 - Implementação do Procedimento de Corte a Dois Estágios.....	67
IV.3 - Conclusões.....	88
REFERÊNCIAS BIBLIOGRÁFICAS.....	89

I N T R O D U Ç Ã O

I N T R O D U Ç Ã O

Um problema de interesse relevante para algumas indústrias, é o de otimizar processos de layout a duas dimensões. Neste problema, é mantido em estoque um número de placas retangulares que devem ser cortadas, de forma a se atender uma demanda constituída de diversos tipos de peças retangulares, do mesmo material. O objetivo é cortar as peças da demanda a partir das placas de estoque, de forma a minimizar a perda do material envolvida no processo.

Dependendo sobre o processo técnico de corte envolvido, e mais especificamente, sobre a natureza do material a ser cortado, certas aplicações do problema como por exemplo, indústria de vidro, corte de placas de metal etc., requerem que o corte possua a característica específica de guilhotina. Neste trabalho, estudamos o problema de corte a duas dimensões, tendo como objetivo principal, analisar a sub-classe dos problemas de corte a duas dimensões do tipo guilhotina. Neste sentido, sugerimos, no Capítulo II, um procedimento heurístico de corte para obtenção de soluções aproximadas do problema, baseado em um esquema de corte a dois estágios, com a presença de uma guilhotina alternativa, no sentido de se incrementar a flexibilidade de reutilização de sub-placas geradas durante o processo. No Capítulo III é analisado um caso específico do problema de corte a duas dimensões do tipo guilhotina, quando a demanda é constituída de um único tipo de peça, e algumas idéias são sugeridas no sentido de se desenvolver alternativas de melhoramento da solução, obtida via o procedimento heurístico.

A implementação do procedimento de corte se fez sobre um Burroughs 6700 e, no Capítulo IV, discutimos duas versões do sistema, onde observamos, com base nos resultados computacionais obtidos, um bom desempenho da versão Branch-Bound à medida que crescemos o porte do problema, ou seja, um bom aproveitamento do material com tempo computacional envolvido relativamente modesto.

CAPÍTULO I - ASPECTOS TEÓRICOS GERAIS SOBRE O
PROBLEMA DE CORTE

I. ASPECTOS TEÓRICOS GERAIS SOBRE O PROBLEMA DE CORTE

I.1 - Introdução

Para a colocação do problema de corte a duas dimensões, procuramos neste capítulo, rever da literatura alguns resultados teóricos sobre o problema para o caso 1-dimensional. Tratamos também o caso específico de interesse de nosso trabalho, que é o modelo de corte tipo guilhotina a dois estágios, procurando abordar suas vantagens e desvantagens, no sentido de gerar motivação para o procedimento de corte que é sugerido no Capítulo II sob a denominação de procedimento de corte guilhotina a 2 estágios alternados.

I.2 - O Problema de Corte no Caso 1-dimensional

Suponhamos que sejam mantidas em estoque, placas contínuas de largura L , com L inteiro e positivo, de um certo material que pode ser, por exemplo, aço, ferro etc.; em um número ilimitado. Por outro lado, admitamos que se deseje a partir deste estoque, atender a uma demanda N_i de peças de largura l_i e de comprimento inteiro (que pode, sem perda de generalidade, ser pensado como sendo idêntico ao comprimento da placa de estoque), com N_i e l_i inteiros e positivos para $i = 1, 2, \dots, m$.

Em linhas gerais, o problema de corte a uma dimensão, pode ser colocado como o de atender à demanda a partir do estoque, de forma que seja minimizada a perda gerada no processo. Naturalmente, só é possível atender à demanda se $L \geq l_i$ pa-

ra todo $i = 1, 2, \dots, m$.

Por um modelo de corte, representamos uma maneira de se cortar a largura L nas larguras menores l_i , $i = 1, 2, \dots, m$.

Associemos ao j -ésimo modelo de corte, uma variável x_j inteira, que representará o número de vezes que tal modelo será utilizado no processo. Na figura 1 abaixo, ilustramos modelos de corte associados a placa de largura L .

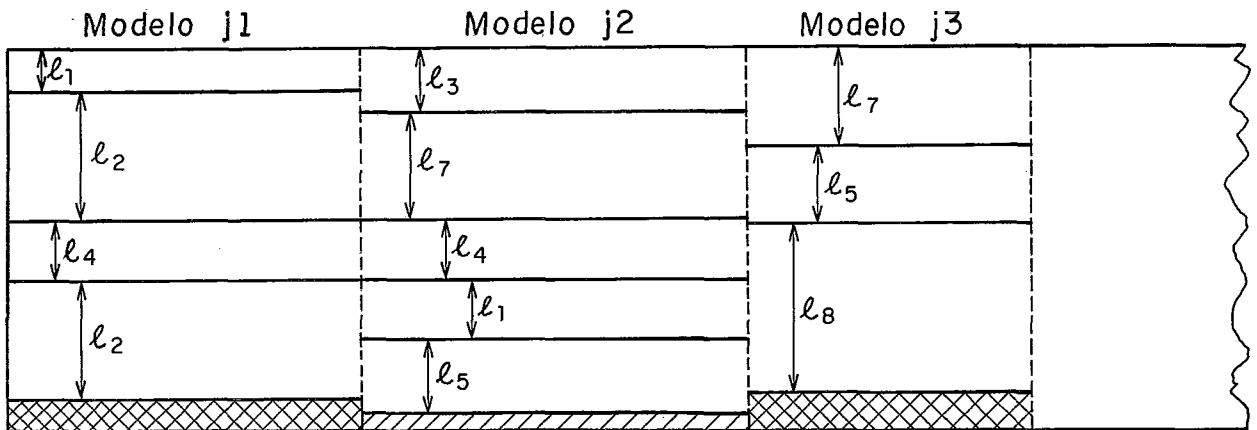


Figura 1

Representemos por a_{ij} : quantidade de faixas de largura l_i , geradas pelo j -ésimo modelo de corte. Desta forma, temos que para cada j :

$$a_{1j} l_1 + a_{2j} l_2 + \dots + a_{mj} l_m \leq L$$

Do fato de que a demanda deve realmente ser atendida, temos a seguinte restrição:

$$a_{i1} x_1 + a_{i2} x_2 + \dots + a_{in} x_n = N_i$$

para cada $i = 1, 2, \dots, m$.

Representemos por:

$$A = \left[a_{ij} \right]_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}} : \text{matriz } m \times n \text{ dos coeficientes inteiros } a_{ij}.$$

$$X = \left[x_1, x_2, \dots, x_n \right]^t : \text{vetor coluna de } \mathbb{R}^n \text{ de uma quantidade } n \text{ de variáveis inteiras } x_j.$$

$$\underline{1} = \left[1, 1, \dots, 1 \right] : \text{vetor linha de } \mathbb{R}^n \text{ com todas componentes unitárias.}$$

$$N = \left[N_1, N_2, \dots, N_m \right]^t : \text{vetor coluna de } \mathbb{R}^m \text{ das demandas inteiras } N_i.$$

Então, o problema de corte a uma dimensão pode ser estabelecido pelo seguinte modelo:

$$\left. \begin{array}{l} \text{minimizar } \underline{1}X \\ \text{sujeito a } AX = N \\ X \geq 0 \\ X: \text{inteiro} \end{array} \right\} \textcircled{1}$$

Pode ocorrer, em determinadas situações práticas, que o processo de corte permita produzir faixas acima do nível

da demanda existente. Neste caso, o modelo que se estabelece para o problema, é o seguinte:

$$\left. \begin{array}{l} \text{minimizar} \quad 1X \\ \text{sujeito a} \quad AX \geq N \\ \quad \quad \quad X \geq 0 \\ \quad \quad \quad X: \text{inteiro} \end{array} \right\} \textcircled{2}$$

I.3 - O Problema Knapsack

Se analisarmos o problema de corte, considerando que o número m de larguras l_i demandadas é relativamente grande, vemos que o número n de modelos de cortes viáveis a ele associado, torna-se expressivamente grande, mesmo considerando que o estoque é constituído de um único tipo de placa padrão de largura L . Este fato, adicionado à condição de integrabilidade, faz com que o modelo $\textcircled{1}$ apresente sérias dificuldades computacionais. Se relaxarmos a condição de integrabilidade, o modelo $\textcircled{1}$ pode ser resolvido no contexto de programação linear, e posteriormente, a solução arredondada para valores inteiros utilizando técnicas existentes na literatura.

Desta forma, damos origem ao seguinte modelo de programação linear:

$$\left. \begin{array}{l} \text{minimizar} \quad 1X \\ \text{sujeito a} \quad AX = N \\ \quad \quad \quad X \geq 0 \end{array} \right\} \textcircled{3}$$

O objetivo desta seção é colocar o procedimento desenvolvido por Gilmore & Gomory [21] para solução do modelo (3). Vamos admitir que para um estágio genérico de processamento do algoritmo simplex na solução do modelo (3), as variáveis básicas sejam dadas por x_1, x_2, \dots, x_m .

No modelo (3), o vetor $\underline{1}$ de elementos unitários pode ser pensado de forma mais geral como um vetor de custos C . Indicamos por P_i o vetor $(a_{1i}, a_{2i}, \dots, a_{mi})^t$, e por C_i o coeficiente de custo associado com a variável x_i ; seja $P = [a_1, a_2, \dots, a_m]^t$ um modelo de corte associado a uma largura L com custo C_0 . Representemos por A , a matriz de ordem m , cujas colunas são dadas por P_1, P_2, \dots, P_m associadas, respectivamente, às variáveis básicas x_1, x_2, \dots, x_m . Desta forma, temos que o conjunto de vetores $\{P_1, P_2, \dots, P_m\}$ forma uma base para o \mathbb{R}^m e assim, existe um vetor coluna U solução da seguinte equação:

$$AU = P$$

Agora, a coluna P pode ser introduzida na base se $CU > C_0$, onde $C = [c_1, c_2, \dots, c_m]$.

Indicando por b_1, b_2, \dots, b_m os coeficientes do vetor linha CA^{-1} temos que a coluna P será introduzida na base se existirem inteiros a_1, a_2, \dots, a_m satisfazendo as seguintes condições:

$$\ell_1 a_1 + \dots + \ell_m a_m \leq L \quad (4)$$

e

$$b_1 a_1 + \dots + b_m a_m > C_0 \quad (5)$$

A maneira de se determinar inteiros a_1, \dots, a_m satisfazendo às condições (4) e (5), é encontrar inteiros a_1, \dots, a_m satisfazendo à condição (4) para os quais, a função:

$$Z(P) = b_1 a_1 + \dots + b_m a_m \text{ seja máxima.}$$

É claro que se o grupo de inteiros a_i com $i = 1, 2, \dots, m$, para os quais $Z(P)$ é máxima, não satisfizer a condição (5) nenhum outro grupo satisfará. Desta forma, vemos que o problema de se determinar uma coluna não básica para entrar na base ou seja uma iteração de mudança de base na resolução do modelo (3) pelo algoritmo simplex, é solucionado resolvendo-se um sub-problema knapsack do tipo do modelo (6) abaixo:

$$\left. \begin{array}{l} \text{maximizar } Z(P) = b_1 a_1 + b_2 a_2 + \dots + b_m a_m \\ \text{sujeito a } l_1 a_1 + l_2 a_2 + \dots + l_m a_m \leq L \\ 0 \leq a_i \\ a_i \text{ inteiro, para } i = 1, \dots, m \end{array} \right\} \textcircled{6}$$

O modelo (6) pode ser resolvido por um método de programação dinâmica (Gilmore & Gomory^[21]), definindo $F_s(x)$ como o valor da melhor combinação de peças que pode ser cortada de uma placa de largura x , fazendo uso das s primeiras variáveis apenas. Temos, então, fazendo $\Pi_i = b_i$ para $i = 1, 2, \dots, m$:

$$F_s(x) = \text{máximo} \{ \Pi_s + F_s(x - l_s); F_{s-1}(x) \}: \textcircled{7}$$

para $s > 1$.

A recursão acima, é fácil de ser interpretada, dado o fato que o modelo fornecendo o valor de $F_s(x)$ para $s > 1$, usa ou não usa, a s -ésima variável para um nível positivo. Se a s -ésima variável é utilizada, então

$$F_s(x) = \Pi_s + F_s(x - \ell_s)$$

e se a s -ésima variável não é utilizada, então

$$F_s(x) = F_{s-1}(x)$$

Indiquemos por

$[x]$: maior inteiro menor ou igual a x . Então:

$F_1(x) = \Pi_1 \left[\frac{x}{\ell_1} \right]$ é facilmente obtido e $F_s(0) = 0$ para todo s . Desta forma, a recursão (7) pode ser utilizada para se calcular $F_s(x)$ em termos de $F_{s-1}(x)$ e $F_s(x')$ para $x' < x$, ou seja, podemos calcular $F_s(x)$ em termos das funções já calculadas. Sendo assim, $F_m(L)$ expressa a melhor solução para o modelo (6) e uma vez que $F_m(L)$ tenha sido computado, então $F_s(x)$ para todo s , com $1 \leq s \leq m$, e todo x com $0 \leq x \leq L$, também estará computado. Tendo sido computado $F_m(L)$, resta o problema de se recuperar o valor da seqüência de inteiros $\{a_1, a_2, \dots, a_m\}$ que representa a solução do modelo (6). Isto é feito, retornando a recursão (7), fazendo uso do seguinte procedimento: associamos com $F_1(x)$ o índice 1 se $x \geq \ell_1$ e o índice 0 em caso contrário, e para $s > 1$, quando $F_s(x)$ é computado, associamos a ela o índice s se $F_s(x) = \Pi_s + F_s(x - \ell_s)$ e em caso contrário associamos o índice

ce que foi associado a $F_{s-1}(x)$. Desta forma, o índice associado com $F_s(x)$ para quaisquer s e x é o maior índice r para o qual a_r é positivo no modelo fornecendo $F_s(x)$.

Logo, para se determinar o modelo fornecendo $F_m(L)$ isto é, a seqüência de inteiros solução de (6), olhamos para o índice associado; se este índice é r e $r > 0$, então o inteiro a_r , está presente no modelo com um valor no mínimo igual a 1; a seguir, olhamos para $F_m(L - \ell_r)$; se seu índice associado é r' ($r' = r$ é possível acontecer) e $r' > 0$, então uma peça de largura ℓ_r , foi utilizada em $F_m(L - \ell_r)$ e conseqüentemente em $F_m(L)$.

Uma vantagem computacional deste método que merece ser ressaltada é a seguinte: na computação da recursão (6) é necessário apenas manter a informação armazenada em $F_{s-1}(x)$ para a computação de $F_s(x)$ e também, na recuperação da seqüência $\{a_1, a_2, \dots, a_m\}$ solução do modelo (6) é requerido apenas a informação $F_m(x)$, assim, todos os valores $F_{s-2}(x)$ podem ser descartados antes de se iniciar a computação de $F_s(x)$.

I.4 - O Problema de Corte a Duas Dimensões

Consideremos a seguinte situação: uma demanda $D = \{(\ell_i, w_i); i = 1, \dots, m\}$, de peças retangulares em quantidades N_i , onde ℓ_i , w_i e N_i são inteiros positivos para $i=1,2,\dots,m$, é para ser cortada de placas retangulares de dimensões (L, W) , que são mantidas em estoque em um número ilimitado. Admitimos, também, que o estoque seja constituído somente de placas padrão do tipo (L, W) , e que o problema seja viável ou seja, que para

todo $i = 1, \dots, m$ a peça do tipo $i \in D$ pode ser alocada na placa (L, W) . Então, o problema de corte a duas dimensões, é o de atender a demanda D , de forma que seja minimizada a perda envolvida no processo. Da característica específica das placas do estoque, o problema pode ser posto da seguinte forma: atender a demanda D , com o número mínimo de placas (L, W) . Utilizando as notações da seção 1.2 e um procedimento idêntico ao da seção 1.3 o modelo de programação linear que se adapta ao problema é do tipo ③ ou seja

$$\left. \begin{array}{l} \text{minimizar} \quad \mathbb{1} X \\ \text{sujeito a} \quad AX = N \\ \quad \quad \quad X \geq 0 \end{array} \right\} \text{⑧}$$

Agora, as colunas da matriz A são definidas da seguinte forma: para cada possível modelo de corte retangular de (L, W) corresponde uma coluna $[a_1, a_2, \dots, a_m]^t$ de A onde a_i representa o número de peças (l_i, w_i) que figura no modelo, para $i = 1, 2, \dots, m$. Por exemplo, na Figura 2, a seguir, é ilustrado um modelo de corte para o retângulo (L, W) com uma demanda de $m = 11$ peças distintas.

Desta forma, a coluna da matriz A representando o modelo de corte acima é dada por:

$$[2, 1, 0, 1, 1, 0, 3, 0, 0, 0, 0]^t$$

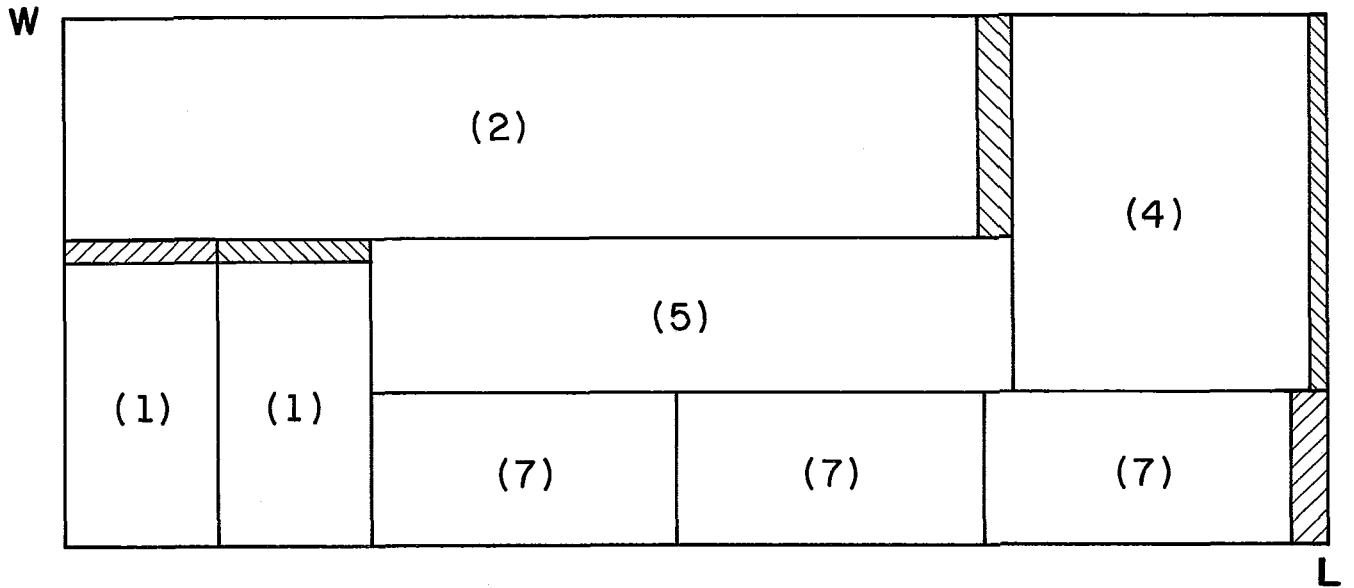


Figura 2

Utilizando a técnica de geração de colunas, análoga à da seção 1.3, o seguinte knapsack generalizado necessita ser resolvido a cada iteração de mudança de base quando da solução do modelo (8) pelo algoritmo simplex: representando por Π_i para $i = 1, 2, \dots, m$, o multiplicador de Lagrange, ou multiplicador simplex, associado à i -ésima equação em alguma solução básica viável temos:

$$\left. \begin{array}{l} \text{maximizar } \{ \Pi_1 a_1 + \Pi_2 a_2 + \dots + \Pi_m a_m \} \\ \text{sujeito a condição de que } [a_1, a_2, \dots, a_m]^t \\ \text{corresponde a um modelo de corte do retângulo} \\ \text{lo } (L, W), \text{ em peças } (l_i, w_i) \text{ para } i=1,2,\dots,m \end{array} \right\} \textcircled{9}$$

O knapsack generalizado do modelo (9), é desprovido de nenhum método econômico do ponto de vista computacional

para sua solução.

Um método clássico para sua solução, desenvolvido por Bellman [5], utiliza a técnica dos multiplicadores de Lagrange para transformá-lo em um knapsack do tipo (6), e posteriormente resolvê-lo pela técnica de programação dinâmica. Observamos, então, que a solução do modelo (8), mesmo utilizando a técnica de geração de colunas, neste contexto, pode demandar um tempo de computação relativamente significativo, se considerarmos o porte do problema de corte a duas dimensões, que comumente encontramos na prática.

1.5 - Corte Tipo Guilhotina

O objetivo de nosso trabalho se restringe a estudar uma sub-classe específica da classe dos problemas de corte a duas dimensões, a saber, aquela do corte tipo guilhotina. Esta sub-classe é caracterizada pela natureza do corte que é dado sobre a placa: qualquer corte dado sobre uma placa, ou qualquer sub-placa da mesma, possui a propriedade de iniciar em uma borda e de forma contínua, terminar na borda simétrica. Também, todo corte é paralelo a um dos lados da placa.

Na Figura 3, a seguir, ilustramos um modelo de corte tipo guilhotina.

A Figura 2, ilustra um modelo de corte a duas dimensões, que não pode ser obtido pelo esquema de guilhotina.

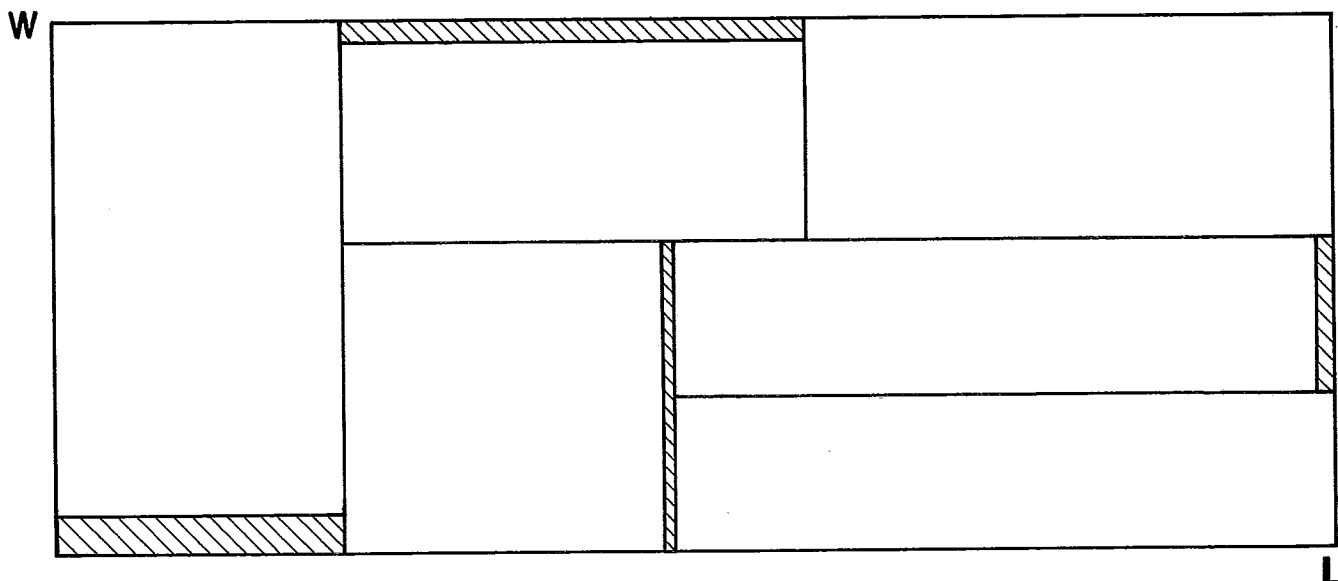


Figura 3

Dentre os modelos de corte tipo guilhotina, ressaltamos aqueles que podem ser obtidos a dois estgios, onde, em um primeiro estgio, o corte se processa em um sentido da placa, e no segundo estgio cada sub-placa produzida do 1 estgio,  analisada e cortada no outro sentido da placa original. A Figura 4 abaixo fornece uma ilustrao desta situao.

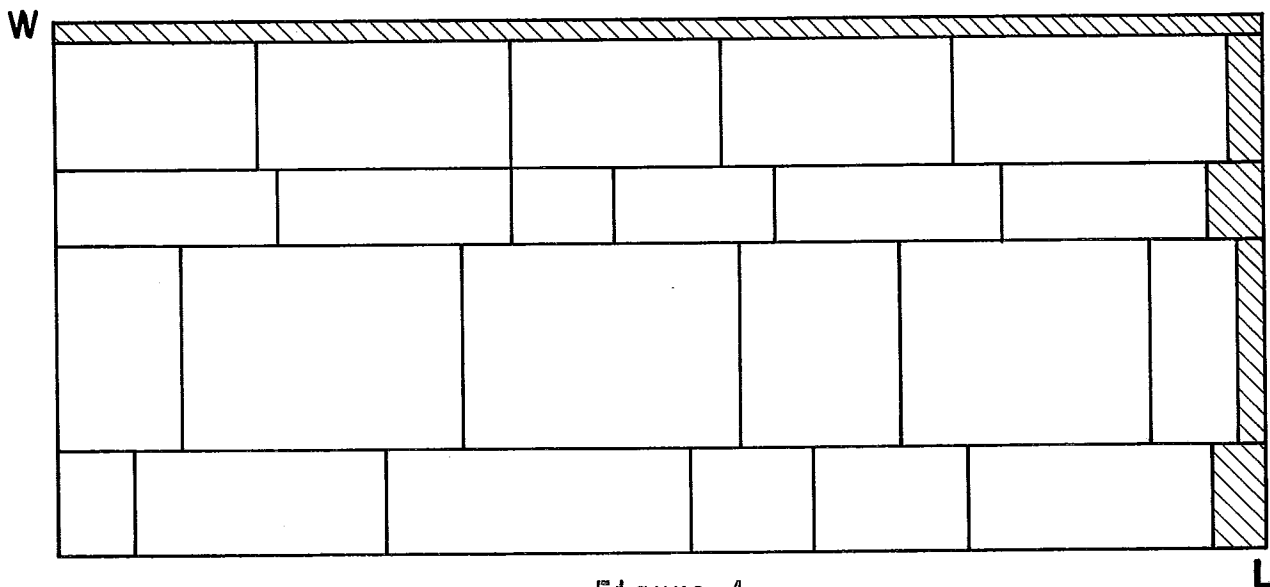


Figura 4

Algumas vezes, um terceiro estágio é permitido como ilustrado na Figura 5 abaixo, onde os retângulos obtidos no corte a dois estágios são cortados segundo seu comprimento para fornecer uma peça nas dimensões exatas de uma determinada peça da demanda.

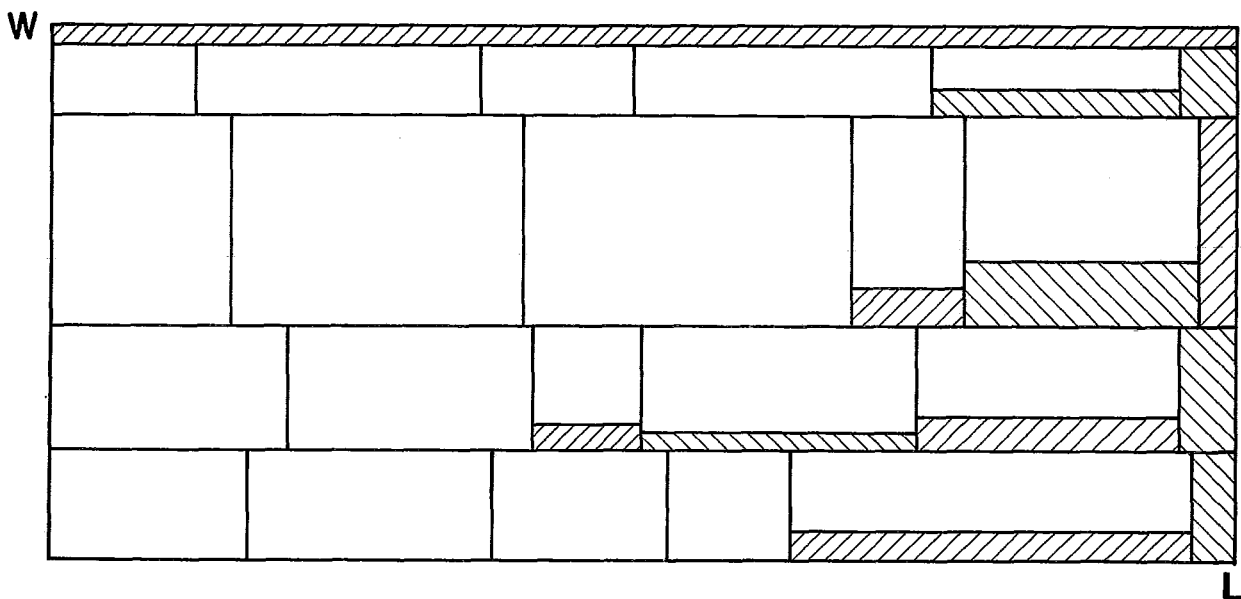


Figura 5

Associando um custo Π_i à i -ésima equação, e notando por a_i a quantidade de peças do tipo i , para $i=1, \dots, m$, presente no modelo, observamos que, a fim de se resolver o problema de cortes a dois estágios, é suficiente resolver o seguinte knapsack generalizado.

$$\left. \begin{array}{l} \text{maximizar } \{ \Pi_1 a_1 + \Pi_2 a_2 + \dots + \Pi_m a_m \} \\ \text{sujeito à condição que } [a_1, a_2, \dots, a_m]^t \\ \text{correspondente a um modelo de corte gui} \\ \text{thotina a dois estágios} \end{array} \right\} \textcircled{10}$$

Admitindo-se que a placa a ser cortada tenha di-

mensões (L, W) , o modelo (10) pode ser resolvido com um procedimento a dois estágios, Gilmore & Gomory [23], como descrito abaixo:

- i) para toda largura w_i , calculamos Π_i^* o valor ótimo obtido na alocação em faixa de largura w_i e comprimento L , de retângulos $\ell_j \times w_j$ on $w_j \leq w_i$. Observamos que, para cada $i=1,2,\dots,m$ este problema é um knapsack do tipo (6);
- ii) o valor ótimo da função objetivo $\Pi_1 a_1 + \Pi_2 a_2 + \dots + \Pi_m a_m$ do modelo (10), é então obtido resolvendo-se um outro knapsack do tipo (11) a baixo:

$$\left. \begin{array}{l} \text{maximizar } \{ \Pi_1^* b_1 + \dots + \Pi_m^* b_m \} \\ \text{sujeito a } w_1 b_1 + \dots + w_m b_m \leq W \end{array} \right\} \text{(11)}$$

Agora, se ordenarmos o conjunto de peças demandadas, de forma tal que $w_1 \leq w_2 \leq \dots \leq w_m$ e definimos $F_s(x)$ como sendo o valor da melhor combinação utilizando apenas as peças associadas às s primeiras larguras w_j , $j = 1, 2, \dots, s$ na geração de faixa de largura w_s e comprimento x , temos que, segundo a seção 1.3, os multiplicadores Π_i^* do estágio i são expressos por:

$$\Pi_i^* = F_i(L)$$

Desta forma, através de um processo recursivo, todos os multiplicadores Π_i^* do estágio i , estarão calculados, quan

do $F_m(L)$ estiver computado. Observamos, então, que o modelo (10) pode ser resolvido, resolvendo-se apenas dois sub-problemas knapsack do tipo (6) ou seja, que os modelos de corte guilhotina a dois estágios, permitem a adaptação de métodos de solução bem mais viáveis do ponto de vista de custo computacional envolvido, que os modelos de corte gerais.

Uma outra característica, inerente dos modelos de corte guilhotina a dois estágios, que merece ser observada, é que do ponto de vista operacional do processo, os mesmos apresentam uma vantagem sobre os modelos mais gerais, dado que, uma vez decidido o modelo de corte guilhotina a dois estágios sobre a placa (L, W) , como por exemplo o ilustrado na Figura 4, no primeiro estágio a dimensão W é toda cortada sem necessitar virar a placa, e no segundo estágio, cada sub-placa obtida do primeiro estágio e também cortada, agora no sentido do comprimento L sem a necessidade de virá-la.

Por outro lado, se analisarmos o procedimento a dois estágios, anteriormente descrito, para solução do modelo (10), observamos que o mesmo trabalha com as peças da demanda orientadas segundo a orientação da placa (L, W) , e para cada $i = 1, 2, \dots, m$ o cálculo dos multiplicadores Π_i^* do estágio i gera faixas como a ilustrada na Figura 6, abaixo.

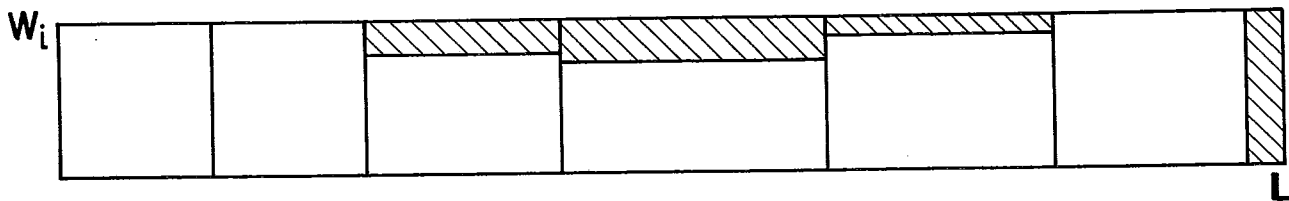


Figura 6

Observamos, então, que o procedimento deixa um acúmulo de pequenas áreas de resto (a parte hachureada na Figura 6), para cada faixa de largura w_i presente no modelo que, aparentemente, é uma característica não muito boa do procedimento.

Esta característica, aparentemente ruim, do modelo é fortemente influenciado pela presença do terceiro estágio de corte (veja Figura 5), a qual é amenizada quando trabalhamos com modelos permitindo apenas dois estágios (veja ilustração da Figura 4).

Analiseemos, agora, o procedimento a dois estágios para solução do modelo (10) através de um exemplo numérico.

Consideremos uma demanda D de peças dadas por:

	l_i	w_i	
Tipo 1	(20, 1)	-	com 1 peça
Tipo 2	(20, 2)	-	com 1 peça
Tipo 3	(20, 3)	-	com 1 peça
Tipo 4	(20, 5)	-	com 1 peça
Tipo 5	(11, 5)	-	com 1 peça

} D

orientadas segundo a orientação de uma placa $(L, W) = (25, 11)$. Desejamos utilizar o procedimento acima para cortar D de (L, W) .

A título apenas de notação, vamos dizer que o tipo i para $i = 1, \dots, 5$, gera a faixa do tipo i de largura w_i .

Para o Primeiro Estágio do Procedimento podemos assumir que para cada $i = 1, \dots, 5$, Π_i^* seja dada pela soma das áreas das peças utilizadas na geração da faixa respectiva, dividida pelo comprimento $L = 25$.

Desta forma, temos que:

$$\Pi_1^* = 0.8$$

$$\Pi_2^* = 1.6$$

$$\Pi_3^* = 2.4$$

$$\Pi_4^* = 4$$

$$\Pi_5^* = 2.2$$

e passamos ao Segundo Estágio do Procedimento com:

$$b_1 = b_2 = b_3 = b_4 = 1, b_5 = 0$$

e valor ótimo do modelo (11) dado por:

$$\Pi_1^* b_1 + \Pi_2^* b_2 + \Pi_3^* b_3 + \Pi_4^* b_4 = 8.8$$

e o modelo de corte é ilustrado na Figura 7, a seguir.

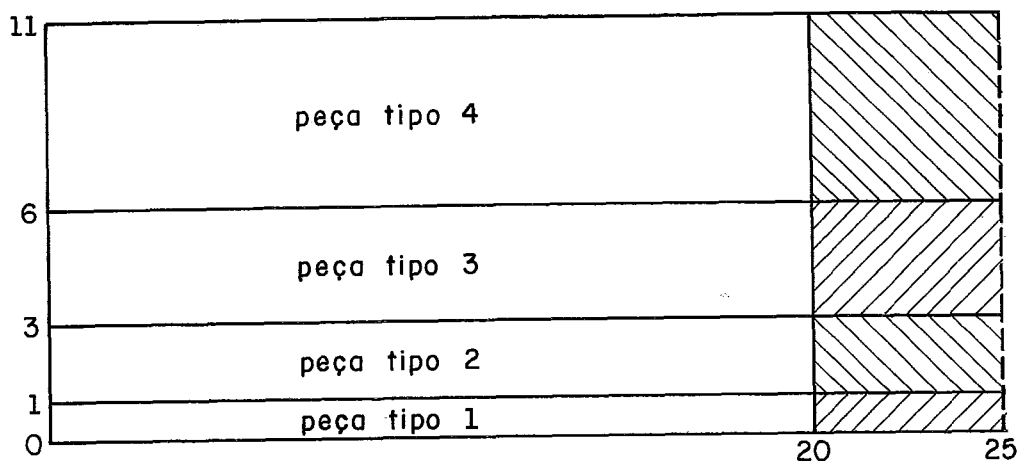


Figura 7

Observamos, então, que a peça do tipo 5 necessariamente necessita ser cortada de uma outra placa dado que as sub-placas de resto geradas na Figura 7 (sub-placas hachureadas na Figura 7), não comportam o tipo 5. Agora, o fato interessante é que se o corte for dado vertical, na altura 20 de L, a peça do tipo 5 pode ser retirada da parte hachureada da Figura 7 e o resto gerado no processo é nulo.

Nestas condições, observamos que o procedimento a dois Estágios anteriormente discutido oferece pouca flexibilidade de reutilização para as sub-placas de resto geradas no processo de corte. O que se pretende no próximo capítulo, é apresentar um procedimento alternativo de corte denominado procedimento de corte a dois estágios alternado, onde a filosofia do procedimento a dois estágios de Gilmore & Gomory^[23] se faz presente (dada a flexibilidade operacional que o mesmo oferece ao processo), em conjunção com uma guilhotina vertical, no sentido de se tentar oferecer ao processo uma melhor reutilização das áreas de resto.

CAPÍTULO II - PROCEDIMENTO DE CORTE A DOIS ESTÁGIOS ALTERNADO

II. PROCEDIMENTO DE CORTE A DOIS ESTÁGIOS ALTERNADO

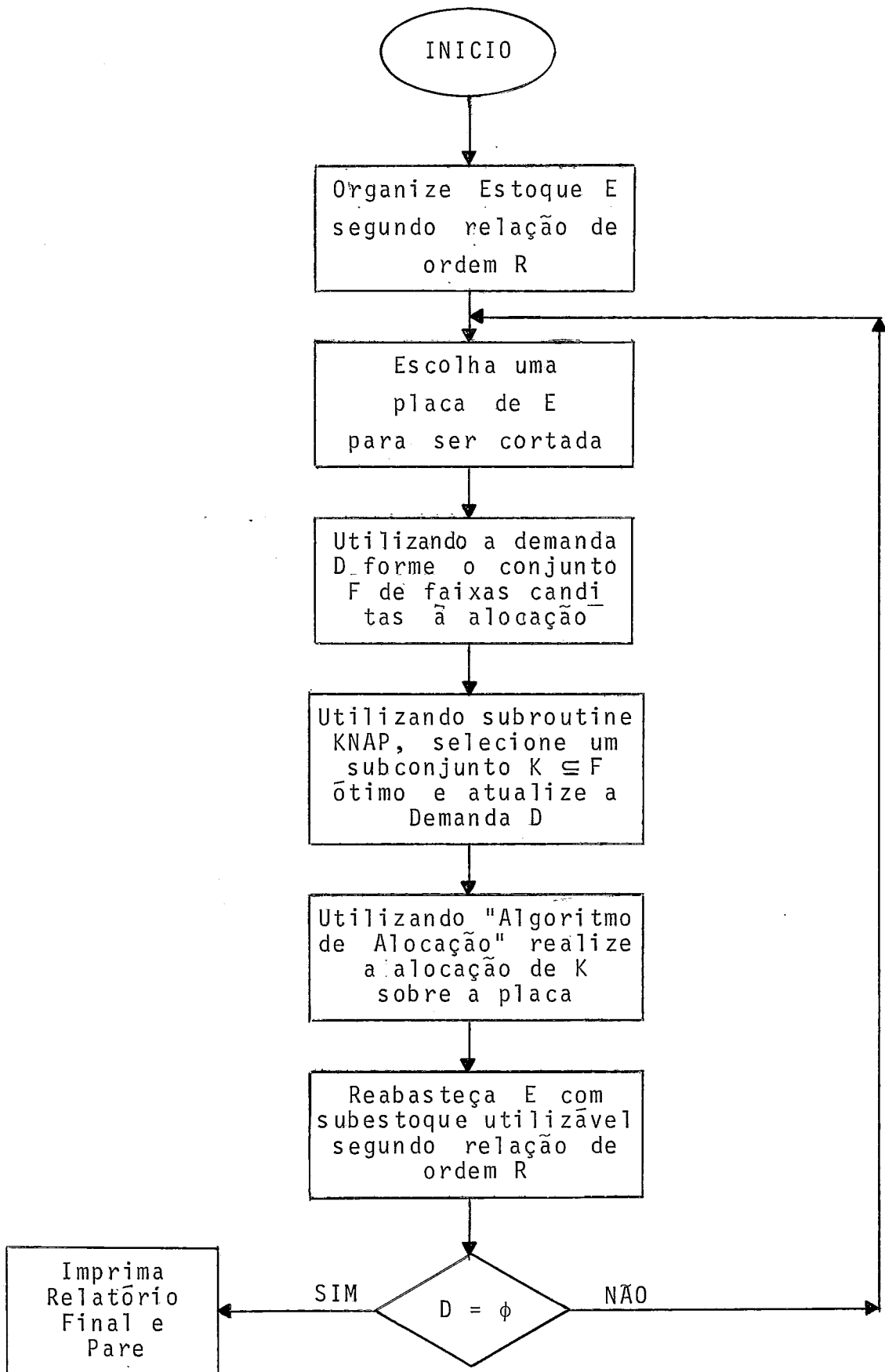
II.1 - Introdução

Neste capítulo, nos propomos a apresentar um procedimento alternativo para a solução do problema de corte a duas dimensões do tipo guilhotina, objetivando explorar a estrutura do modelo (10) discutido no Capítulo I. O procedimento como um todo, é uma mistura de técnicas aproximadas (heurísticas), e exatas, no sentido de gerar e alocar faixas sobre a placa (L, W) a partir das peças da demanda.

Apresentamos um fluxograma do procedimento acompanhado de uma discussão detalhada de seus módulos. No sentido de se fazer uma boa reutilização das áreas de resto, o algoritmo de alocação procura oferecer novas alternativas ao procedimento a dois estágios do Capítulo I.

II.2 - Fluxograma do Procedimento de Corte a Dois Estágios

II.2.1 - Na seção 1.4 do Capítulo I, o problema de corte a duas dimensões foi colocado, admitindo-se que o estoque E era constituído de um número ilimitado de placas padrão (L, W). Podemos pensar que o número de placas presentes no estoque E seja limitado e, desta forma, o problema só admitirá solução se esta limitação do estoque for suficiente para o atendimento da demanda D. Orientamos o processo de corte no seguinte sen



tido: uma vez que uma placa (L, W) foi escolhida para ser cortada, a escolha de uma nova placa padrão só se fará, quando esgotadas todas as possibilidades de corte sobre a placa anterior. Organizamos o estoque E em uma lista que é iterativamente reabastecida com novas subplacas geradas no processo de corte. Desta forma, passamos a encarar E como sendo heterogêneo, dada a presença de novas placas, caracterizadas como subestoque distintas das placas padrão. Definimos, então, a relação de ordem R sobre a lista E , como sendo a relação que organiza E na ordem decrescente de grandeza de área de seus elementos, no sentido de percorrer a lista do início para seu fim. Também, o procedimento de inserção em E de subestoque, se faz obedecendo a relação de ordem R existente sobre E .

II.2.2 - O procedimento de escolha de placa para ser cortada se faz acessando a lista E através de um apontador, onde o mesmo aponta sempre para a última placa inserida na lista. Com isto, estamos objetivando evitar, ao máximo, o retorno para o estoque dos retalhos de placas gerados durante o processo de corte de placas padrão.

II.2.3 - Seguindo a filosofia do procedimento de corte guilhotina a dois estágios, o conjunto F de faixas candidatas à alocação, possui a característica de que toda faixa é formada com base em um único tipo de peça da demanda.

De acordo com a seção 1.4 do Capítulo I, vamos admitir que a demanda D seja dada por $D = \{(l_i, w_i); i = 1, 2, \dots, m\}$ em quantidades inteiras N_i , $i = 1, 2, \dots, m$, e que a placa a ser cor

tada tenha dimensões (L, W).

Agora, para cada i com $i = 1, 2, \dots, m$ tal que $N_i \neq 0$ determinamos inteiros m_1^i e m_2^i , tais que: m_1^i resolve o problema:

$$\left. \begin{array}{l} \text{maximizar } m^i l_i \\ \text{sujeito a } m^i l_i \leq L \\ 0 \leq m^i \leq N_i \\ m^i: \text{ inteiro} \end{array} \right\} \textcircled{12}$$

e m_2^i resolve o problema:

$$\left. \begin{array}{l} \text{maximizar } m^i w_i \\ \text{sujeito } m^i w_i \leq L \\ 0 \leq m^i \leq N_i \\ m^i: \text{ inteiro} \end{array} \right\} \textcircled{13}$$

A partir das soluções dos modelos $\textcircled{12}$ e $\textcircled{13}$ acima, definimos:

$$C^i = \text{m\bar{a}ximo} \{m_1^i l_i, m_2^i w_i\}$$

e

$$m_{\text{m\bar{a}x}}^i = \begin{cases} m_1^i & \text{se } C^i = m_1^i l_i \\ m_2^i & \text{se } C^i = m_2^i w_i \end{cases}$$

Desta forma, para cada $i = 1, \dots, m$ tal que $N_i \neq 0$

definimos a faixa $F^i \in F$ de comprimento C^i e de Largura H^i dada por:

$$H^i = \begin{cases} w_i & \text{se } C^i = m_1^i l_i \\ l_i & \text{se } C^i = m_2^i w_i \end{cases}$$

Neste ponto, o procedimento se preocupa em fazer um bom aproveitamento do material, inicialmente, tentando utilizar o máximo possível da dimensão L da placa.

Observamos que ambos os modelos (12) e (13) apresentam cada um, os mesmos coeficientes na função objetivo e na restrição, são do tipo knapsack com variáveis limitadas e de fácil solução. Esta característica de fácil solução, nem sempre se observa para problemas gerais desta natureza, como podemos analisar com o seguinte exemplo: considerando o chamado problema do particionamento; onde n inteiros positivos $\{a_1, a_2, \dots, a_n\}$ são dados, e a seguinte questão é colocada: existe um conjunto de índices I ,

$$I \subseteq \{1, 2, \dots, n\}$$

de forma que:

$$\sum_{i \in I} a_i = b = \frac{1}{2} \sum_{i=1}^n a_i?$$

Uma maneira alternativa de solucionar o problema do particiona-

mento é considerar o seguinte problema knapsack:

$$\left. \begin{array}{l} \text{maximizar} \quad a_1 x_1 + a_2 x_2 + \dots + a_n x_n \\ \text{sujeito a} \quad a_1 x_1 + a_2 x_2 + \dots + a_n x_n \leq b = \frac{1}{2} \sum_{i=1}^n a_i \\ x_i \in \{0, 1\} \text{ para } i = 1, 2, \dots, n \end{array} \right\} (*)$$

e observamos que o problema do particionamento apresenta uma solução se e somente se (*) tem uma solução com:

$$a_1 x_1 + a_2 x_2 + \dots + a_n x_n \geq b$$

e dada a integrabilidade dos a_j vemos que \underline{b} se faz igual a:

$$b = \left\lfloor \frac{1}{2} \sum_{i=1}^n a_i \right\rfloor \text{ onde } \left\lfloor \frac{1}{2} \sum_{i=1}^n a_i \right\rfloor \text{ representa o maior inteiro menor ou igual a } \frac{1}{2} \sum_{i=1}^n a_i. \text{ Por outro lado, então, tomando o problema:}$$

$$\left. \begin{array}{l} \text{maximizar} \quad \sum_{j=1}^n a_j x_j \\ \text{sujeito a} \quad \sum_{j=1}^n a_j x_j \leq \left\lfloor \sum_{j=1}^n a_j / 2 \right\rfloor \\ x_j \in \{0, 1\} \quad (j = 1, 2, \dots, n) \end{array} \right\} (**)$$

Vemos que em condições especiais sobre os coeficientes a_j , o modelo (**) é um hard-knapsack para ser solucionado por algoritmos recursivos. Não é objetivo específico nosso aqui, nos alongarmos neste problema e para maiores detalhes referenciamos

mos o trabalho de Chvátal [7]..

II.2.4 - Suponhamos que o conjunto F seja dado por:

$$F = \{F^1, F^2, \dots, F^r\}$$

com $1 \leq r \leq m$, e associemos a cada F^i para $1 \leq i \leq r$, um novo índice j , $1 \leq j \leq m$ correspondente ao tipo da peça da demanda, utilizada na formação da faixa. Como em II.2.3, (L, W) representa a placa utilizada na formação do conjunto F .

Associemos à faixa F_j^i , o parâmetro $\Pi_j^i = \left(C_j^i / L \right) * H_j^i$ (para cada $i = 1, \dots, r$) de custo da mesma, ou seja, um custo proporcional à área de utilização de F_j^i , onde C_j^i e H_j^i são como definidos em II.2.3.

Indiquemos por X_j^i para $i = 1, 2, \dots, r$ a quantidade de vezes que a faixa F_j^i será utilizada no modelo de corte. Então, de posse do conjunto F nosso objetivo agora é selecionar $K \subseteq F$ de forma a se fazer uma boa utilização da largura W da placa. Neste sentido o seguinte knapsack se coloca:

$$\left. \begin{array}{l} \text{maximizar} \quad \sum_{i=1}^r \Pi_j^i X_j^i \\ \text{sujeito a} \quad \sum_{i=1}^r H_j^i X_j^i \leq W \\ 0 \leq X_j^i \leq \lfloor N_j^i / m_{\max}^j \rfloor \\ X_j^i: \text{ inteiro} \end{array} \right\} \textcircled{14}$$

onde $\lfloor N_{j/m_{\max}^j}^i \rfloor$ representa maior inteiro menor ou igual $N_{j/m_{\max}^j}^i$ e N_j^i representa a demanda da peça do tipo j utilizada na formação da faixa F^i .

Agora, para $k = 1, 2, \dots, r$ e $y = 0, 1, \dots, w$ definimos:

$$F(k, y) = \text{máximo} \sum_{i=1}^k \Pi_j^i X_j^i$$

sujeito a

$$\begin{cases} \sum_{i=1}^k H_j^i X_j^i \leq y \\ 0 \leq X_j^i \leq \lfloor N_{j/m_{\max}^j}^i \rfloor \\ X_j^i: \text{inteiro} \end{cases}$$

Observamos que $F(k, 0) = 0$ para $k = 1, \dots, r$. Suponhamos que para um dado k , $k = 2, \dots, r$ seja conhecido $F(k-1, y)$ para todos os possíveis valores de $y = 0, 1, \dots, W$. Então, podemos escrever:

$$F(k, y) = \text{máximo} \left\{ \Pi_j^k X_j^k + \sum_{i=1}^{k-1} \Pi_j^i X_j^i \right\}$$

sujeito a

$$\begin{cases} \sum_{i=1}^{k-1} H_j^i X_j^i \leq y - H_j^k X_j^k \\ 0 \leq X_j^i, \text{ e inteiro para } i = 1, 2, \dots, k-1 \\ X_j^k = 0, 1, 2, \dots, u_k \end{cases}$$

onde $u_k = \text{mínimo} \left\{ \lfloor y / H_j^k \rfloor; \lfloor N_{j/m_{\max}^j}^k \rfloor \right\}$

Desta forma, para um dado valor de X_j^k , com

$0 \leq X_j^k \leq u_k$ temos que:

$$F(k,y) = \Pi_j^k X_j^k + \left\{ \text{m\bar{a}ximo} \sum_{i=1}^{k-1} \Pi_j^i X_j^i \right\}$$

$$\text{sujeito a } \begin{cases} \sum_{i=1}^{k-1} H_j^i X_j^i \leq y - H_j^k X_j^k \\ 0 \leq X_j^i \text{ e inteiro para } i = 1, \dots, k-1 \end{cases}$$

Observamos que a parcela entre chaves \bar{e} exatamente $F(k-1, y - H_j^k X_j^k)$ a qual \bar{e} conhecida, uma vez que $y = 0, 1, \dots, W$ e $0 \leq X_j^k \leq u_k$ implicando que $y - H_j^k X_j^k$ \bar{e} um inteiro n\~ao negativo, que n\~ao excede W .

Desta forma, temos a fun\~cao recursiva:

$$F(k,y) = \text{m\bar{a}ximo} \left\{ \Pi_j^k X_j^k + F\left(k-1, y - H_j^k X_j^k\right) \right\}$$

$$X_j^k = 0, 1, \dots, u_k$$

utilizada pela subroutine KNAP, no sentido de se resolver o modelo (14). Para maiores detalhes sobre a recurs\~ao acima vide Sal-kin [42]. Com base na solu\~cao do modelo (14), realizamos a atualiza\~ao da demanda, e o processamento prossegue normalmente.

II.2.5 - Passemos agora a descrever o "algoritmo de Aloca\~ao".

Suponhamos que o subconjunto $K \subseteq F$ selecionado pe

la subroutine KNAP seja dada por:

$K = \{F^1, F^2, \dots, F^s\}$ onde, $1 \leq s \leq r$ já ordenado em relação à ordem decrescente de comprimento de faixa, isto é, para todo i com $1 \leq i \leq s - 1$ tem-se que:

$C^i \geq C^{i+1}$, onde C^i , para $i = 1, \dots, s$ é como definido em II.2.3.

Realizamos a alocação de K sobre a placa (L, W) de acordo com a ilustração da Figura 8, abaixo:

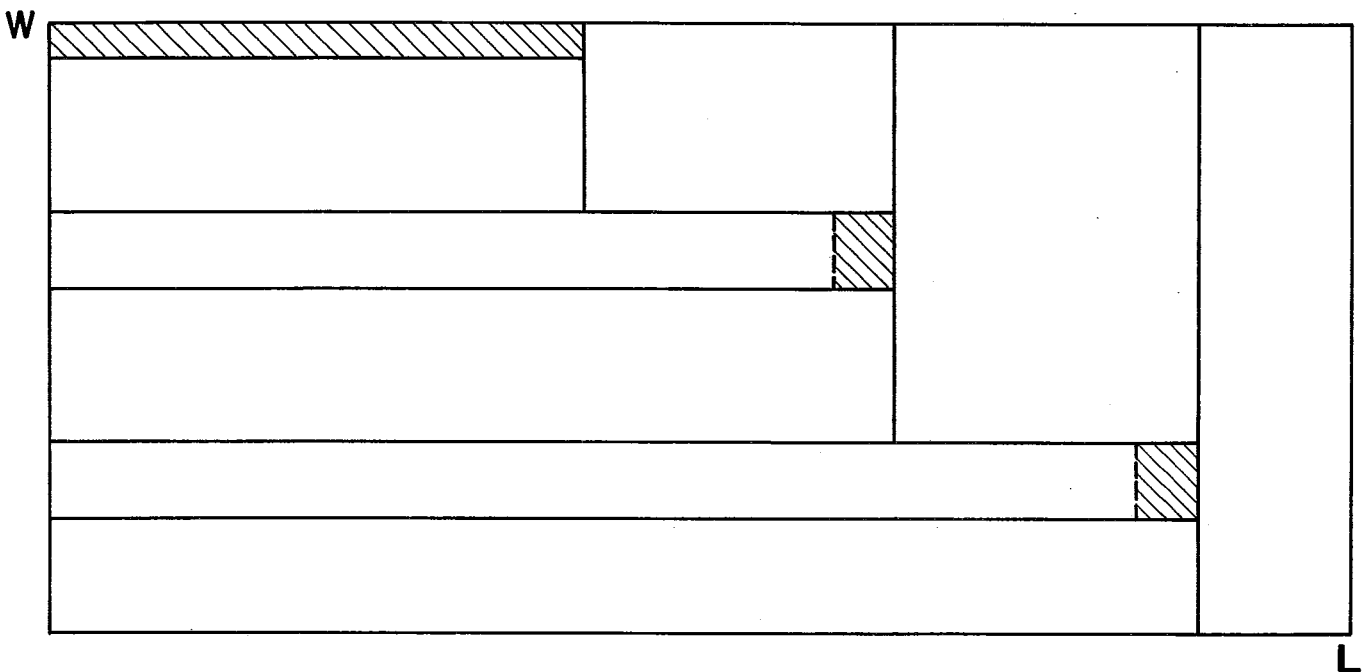


Figura 8

Neste estágio, nossa preocupação é no sentido de direcionarmos a guilhotina (se horizontal ou vertical), de forma a se tentar fazer um bom aproveitamento das áreas de resto. Em função da demanda D atualizada fornecida da KNAP, orientamos o processo de corte adotando o seguinte procedimento: para cada área de resto a ser alocada, no sentido de reaproveitá-la, reali-

mos uma pesquisa exaustiva sobre D e decidimos que a guilhotina sã serã horizontal, se nã existir em D nenhuma peã que se ajuste sobre a ãrea analisada. Uma vez constatada a utilizaãõ de uma determinada ãrea de resto, entãõ o procedimento procura tentar incrementar a flexibilidade de uso da mesma, orientando a guilhotina na vertical. Uma ilustraãõ desta situaãõ ã dada na Figura 8.

Vamos, agora, fazer mais algumas consideraãões: indiquemos por $VET1(J)$ para $J = 1, \dots, s$ uma variãvel binãria definida da seguinte maneira:

$$VET1(J) = \begin{cases} 1 & \text{se a guilhotina ã horizontal} \\ 0 & \text{se a guilhotina ã vertical} \end{cases}$$

Se para algum J , com $1 \leq J \leq s$ tivermos que $C^J = L$ entãõ $C^i = L$ para todo i com $1 \leq i \leq J$ e assumimos, neste caso, $VET1(i) = 1$ para todo $1 \leq i \leq J$ dado que, neste caso, nã existe ãrea de resto e simplesmente nos preocupamos de cortar de (L, W) as J faixas F^i para $1 \leq i \leq J$. Por outro lado, se para algum J com $1 \leq J \leq s - 1$ ocorrer $C^J = C^{J+1}$ e $VET1(J) = 0$ entãõ $VET1(J+1) = 0$ e os cortes J e $J+1$ sãõ coincidentes.

Agora, para $k = 1, 2, \dots, s$, indiquemos por $VET2(k)$ e $XOT(k)$, respectivamente a largura da faixa F^k e o vetor de alocaãõ fornecido pela KNAP, isto ã, $XOT(k)$ representa quantas vezes a faixa F^k serã utilizada no modelo de corte. De posse destas informaãões, passemos aos passos do Algoritmo de Alocaãõ.

Passo 1: (Inicialização)

$$\text{Faça} \begin{cases} \rho = L \\ \text{ISOBC} = \rho - C^1 \\ C^{s+1} = 0 \\ \text{KTEMP2} = 0 \end{cases}$$

e vá para o Passo 2.

Passo 2: Se $k = s + 1$, vá para o Passo 6. Caso contrário, vá para o Passo 3.

Passo 3: Se a placa (ISOBC, $W - \text{KTEMP2}$) for aproveitável, faça $\text{VET1}(k) = 0$ e vá para o Passo 4. Caso contrário, faça $\text{VET1}(k) = 1$ e vá para o Passo 5.

Passo 4:

$$\text{Faça} \begin{cases} \rho = C^k \\ \text{ISOBC} = \rho - C^{k+1} \\ \text{KTEMP2} = \text{VET2}(k) * \text{XOT}(k) + \text{KTEMP2} \\ k = k + 1, \text{ e vá para o Passo 2} \end{cases}$$

Passo 5:

$$\text{Faça} \begin{cases} \text{ISOBC} = \rho - C^{k+1} \\ \text{KTEMP2} = \text{VET2}(k) * \text{XOT}(k) + \text{KTEMP2} \\ k = k + 1 \end{cases}$$

e vá para o Passo 2.

Passo 6: Se a placa (ISOBC, W - KTEMP2) for aproveitável, armazene-a e pare.

Caso contrário, armazene-a como resto não aproveitável e pare.

II.2.6 - Nesta seção, basicamente discutimos o critério de parada do procedimento. Uma exigência inerente do problema de corte é que a demanda D seja totalmente atendida, envolvendo perda mínima.

Utilizamos, então, a satisfação da demanda como critério de parada do procedimento. Uma vez a demanda satisfeita, imprimimos relatório final de corte e paramos o procedimento.

CAPÍTULO III - SOBRE UM CASO ESPECÍFICO DO PROBLEMA DE CORTE

III. SOBRE UM CASO ESPECÍFICO DO PROBLEMA DE CORTE

III.1 Introdução

Neste capítulo, o problema de corte tipo guilhotina a duas dimensões é tratado no caso em que a demanda é constituída de um único tipo de peça. Com esta particularidade, o problema de minimização de perda é estudado considerando-se o seguinte problema: Dados, uma placa (L, H) e uma peça do tipo (ℓ, h) com L, H, ℓ e h números inteiros positivos, determinar o número máximo de peças (ℓ, h) que pode ser cortado da placa (L, H) de forma que toda peça (ℓ, h) , originado do modelo de corte, possua seus lados paralelos aos lados de (L, H) e que o corte seja do tipo guilhotina.

Na seção III.2, o problema é abordado de forma geral, procurando rever da literatura alguns resultados teóricos no sentido de se fazer uma análise do procedimento de corte do Capítulo II para o problema de nosso interesse.

Na seção III.3 é sugerido um procedimento a dois estágios para obtenção de uma solução aproximada do problema.

III.2 Considerações Teóricas Gerais sobre o Problema

Consideremos os inteiros positivos ℓ, h, L e H , com ℓ e h relativamente primos e o problema de empacotar no retângulo

lo (L, H) o número máximo de retângulos $(\ell; h)$. Discutiremos com detalhes no Lema 1, o caso em que $\ell = 1$ expressando uma fórmula analítica para a área de resto, que depende apenas dos resíduos das classes de congruência de L e H , $(\text{mod } h)$.

Lema 1: Consideremos $L, H \geq h$ com

$$L \equiv a \pmod{h}$$

$$H \equiv b \pmod{h}$$

onde:

$$0 \leq a < h$$

$$0 \leq b < h$$

e indiquemos por R_1 a área de resto no melhor empacotamento de (L, H) com retângulos $(1, h)$. Temos, então, que:

$$R_1 = \begin{cases} ab & \text{se } a + b \leq h \\ (h - a)(h - b) & \text{se } a + b \geq h \end{cases}$$

Prova: Dados que $L \equiv a \pmod{h}$ e $H \equiv b \pmod{h}$ temos que h divide $L - a$ e $H - b$ respectivamente. Este resultado adicionado ao fato que H e a são números inteiros, nos permite dizer, seguindo a ilustração da Figura 9, a seguir, que se cortarmos de (L, H) o retângulo (a, b) então a área que sobra pode ser totalmente coberta com retângulos do tipo $(1, h)$.

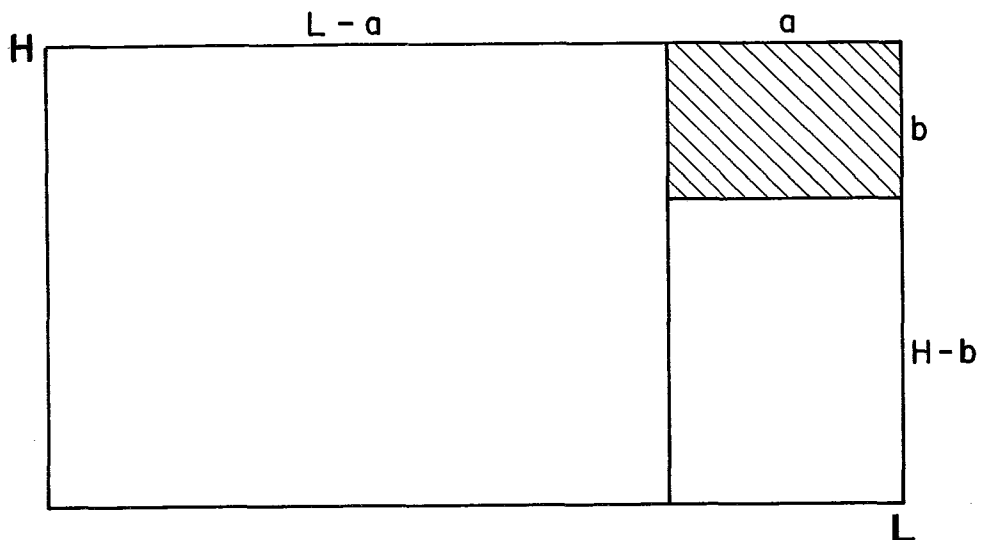


Figura 9

Tomando agora a ilustração da Figura 10, abaixo, considerando os fatos de que h divide $(L - a)$ e $(H - b)$, respectivamente, com a , b e h números inteiros, concluimos que os quatro

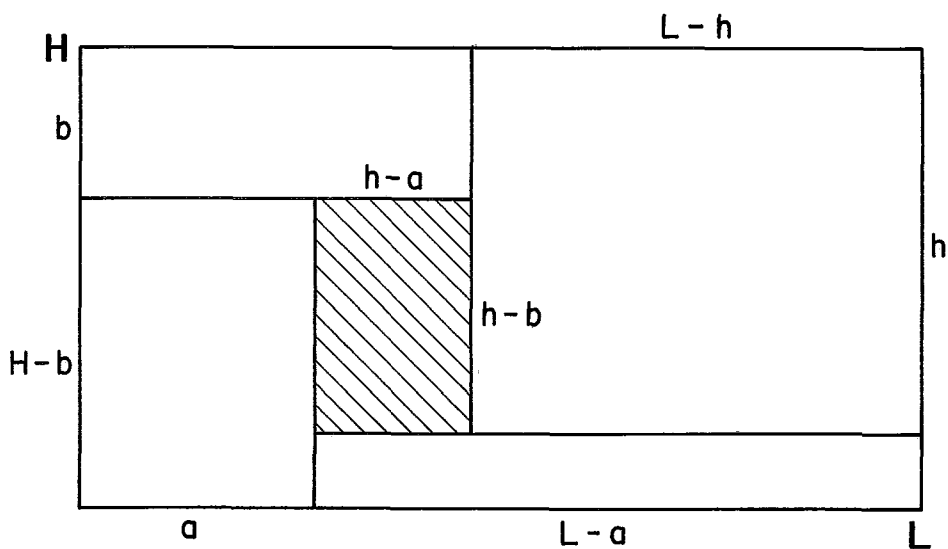


Figura 10

retângulos externos desta Figura, podem ser totalmente empacota-

dos com retângulos $(1, h)$ deixando uma área de resto representada pelo retângulo central hachurado, expressa por $(h - a)(h - b)$.

Desta maneira, observamos que o problema de empacotar o retângulo (L, H) com retângulo do tipo $(1, h)$ fornece uma área de resto que não ultrapassa a quantidade R_1 do Lema 1. Particionaremos, agora, o quadrante positivo em quadrados unitários segundo a ilustração da Figura 11.

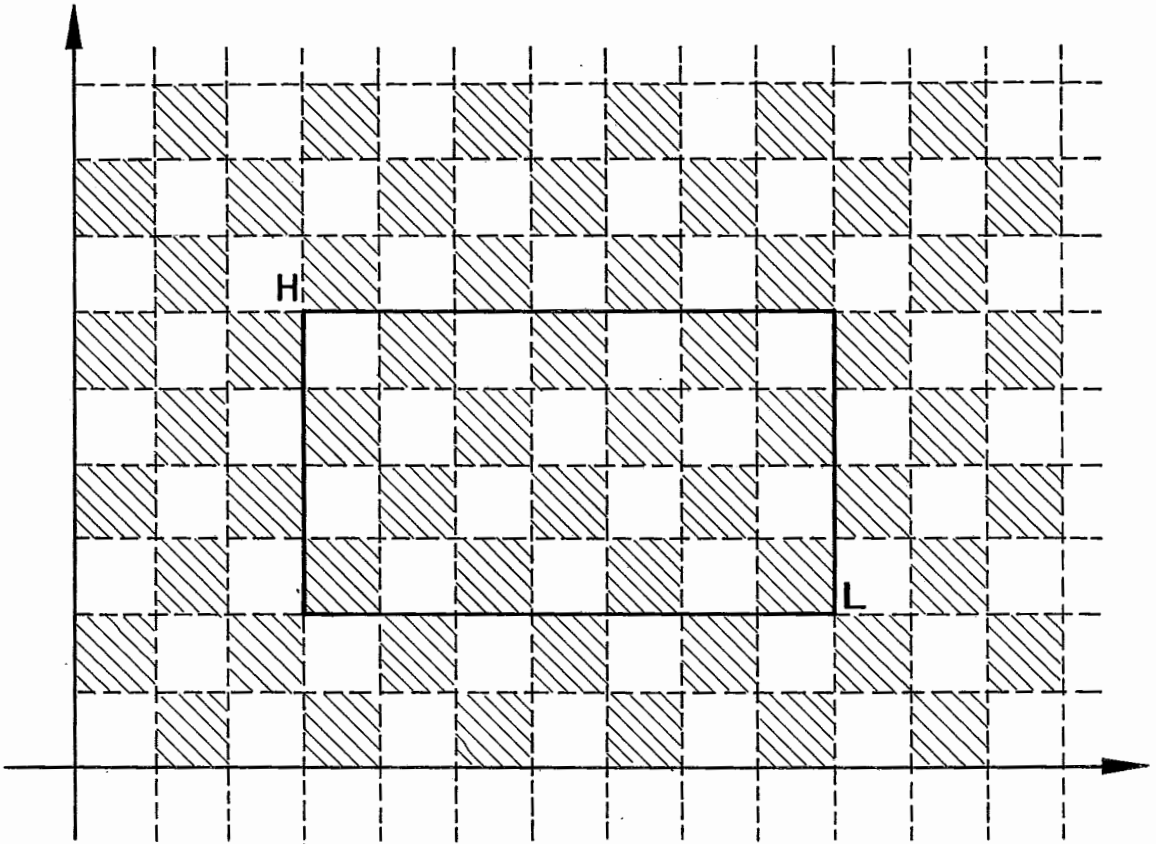


Figura 11

Com o objetivo de auxiliar a interpretação do que se segue, vamos ilustrar a Figura 11 para o caso de $h = 2$, onde destacamos as diagonais múltiplas de 2 em hachureado. Se o retângulo (L, H) for posicionado neste plano, de forma que o mesmo contenha k quadrados hachureados então é claramente impossível empacotar mais do que k retângulos do tipo $(1, h)$ em seu inte-

rior. Observamos na Figura 11, acima, que (L, H) contém, em seu interior, 14 quadrados hachureados e, como pode ser visto, o número máximo de retângulos $(1, 2)$ que pode ser empacotado em $(L, H) = (7, 4)$ é 14.

Desta forma, o empacotamento de (L, H) com retângulos do tipo $(1, h)$ fornece uma área de resto pelo menos da ordem de $LH - kh$. Agora, é fácil ver que um retângulo com perímetro $\leq 2h$ pode ser posicionado entre duas diagonais hachureadas, de forma a não conter nenhum quadrado hachureado.

Sendo assim, quando $a + b \leq h$ de forma que $2(a+b) \leq 2h$, o retângulo (L, H) da Figura 9 pode ser posicionado tal que seu sub-retângulo (a, b) não contenha nenhum quadrado hachureado em seu interior. Logo, o empacotamento de (L, H) com retângulos do tipo $(1, h)$ fornece uma área coberta que contém, exatamente, $k = \frac{LH - ab}{h}$ quadrados hachureados e, assim, a área de resto é, pelo menos, $LH - kh = ab$.

Por outro lado, quando $a + b \geq h$, temos que:

$$2(a + b) \geq 2h = 4h - 2h \rightarrow 4h - 2(a + b) \leq 2h$$

Mas, $4h - 2(a + b)$: representa o perímetro do retângulo hachureado da Figura 10 e, desta forma, utilizando o que já foi mostrado anteriormente, temos que quando $a + b \geq h$, o retângulo (L, H) da Figura 10 pode ser posicionado no plano tal que seu sub-retângulo $(h - a, h - b)$ hachureado não contenha em seu interior nenhum quadrado hachureado.

Com um argumento semelhante ao anterior mostramos também, neste caso, que a área de resto é dada por $(h - a)(h - b)$ o que mostra o resultado do Lema 1.

Indiquemos por R_2 , a área de resto no melhor empacotamento de (L, H) com retângulos (l, ℓ) , segundo o conteúdo do Lema 1, e definimos:

$R = R(L, H)$: como sendo o menor número inteiro não negativo tal que:

$R \geq \text{máximo} \{R_1, R_2\}$ com $R \equiv R_1 \pmod{h}$ e $R \equiv R_2 \pmod{\ell}$. De posse disto, enunciamos o seguinte teorema:

Teorema: Se L e H são suficientemente grandes, então a área de resto no melhor possível empacotamento do retângulo (L, H) com retângulos do tipo (ℓ, h) é precisamente $R(L, H)$.

Omitimos a prova do teorema, e para maiores detalhes referencie Barnes [4].

Neste estágio, desejamos fazer uma análise do procedimento de corte do Capítulo II para o problema que estamos considerando. Vamos admitir que o procedimento forme faixas segundo a dimensão ℓ e que a situação após a alocação seja representada na Figura 12, a seguir.

Temos, então, que existem inteiros m_1, r_1, m_2, r_2 tal que $L = m_1\ell + r_1$ com $0 \leq r_1 < \ell$ e $H = m_2h + r_2$ com $0 \leq r_2 < h$,

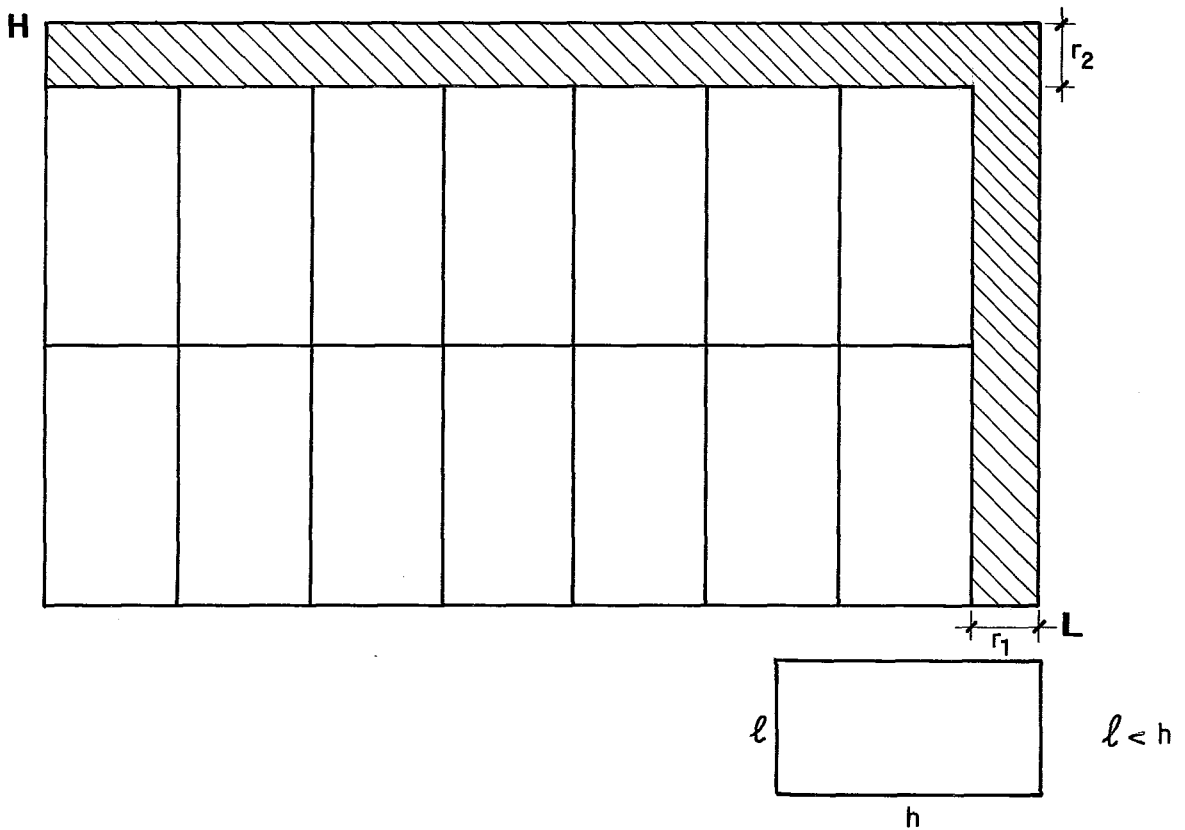


Figura 12

ou seja, $L \equiv r_1 \pmod{l}$ e $H \equiv r_2 \pmod{h}$. Nestas condições, a área de resto R representada pela parte hachureada na Figura 12 é dada por:

$$R = Hr_1 + (L - r_1)r_2$$

Considerando, agora, o empacotamento de (L, H) segundo o procedimento de corte, com retângulos do tipo $(l, 1)$ temos que o processo de alocação se faz segundo o ilustrado na Figura 13, a seguir. Desta forma, existem inteiros m_3, r_3 tal que $H = m_3l + r_3$ com $0 \leq r_3 < l$, ou seja, que $H \equiv r_3 \pmod{l}$ e o resto R_1 representado pela área hachureada na Figura 13 é expresso por $R_1 = r_1 r_3$.

De forma análoga, considerando o empacotamento de (L, H) com retângulos (l, h) temos que o modelo de corte é ilustrado na Figura 14, abaixo. Sendo assim, existem inteiros m_4 e r_4 tal que $L = m_4 h + r_4$, com $0 \leq r_4 < h$, ou seja, que $L \equiv r_4 \pmod{h}$.

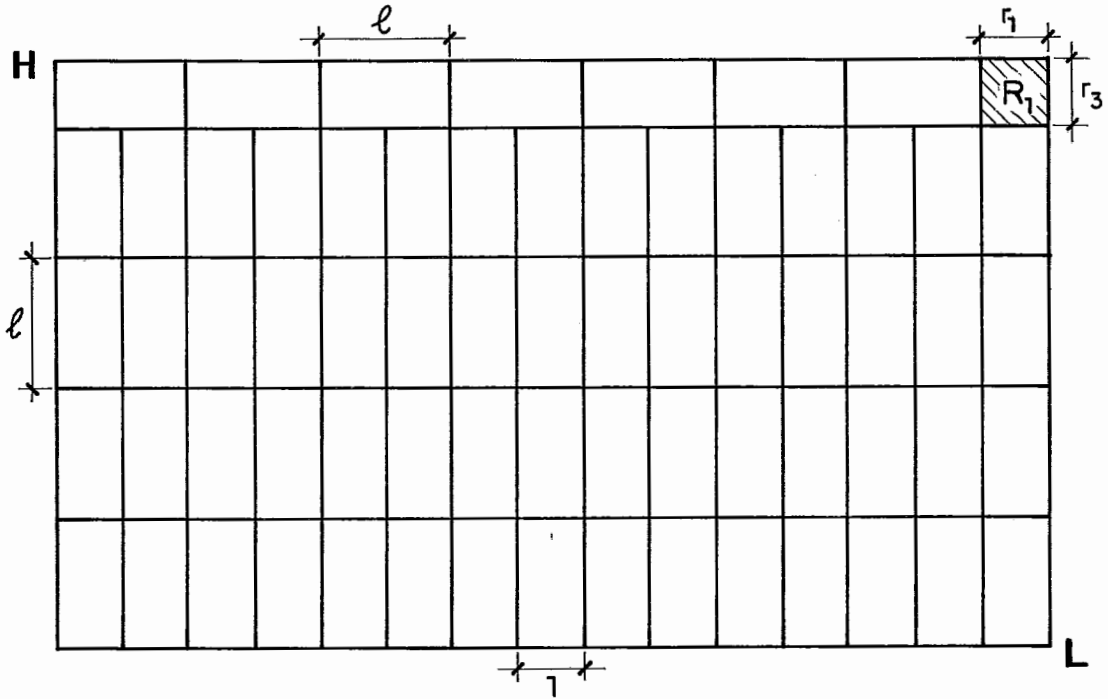


Figura 13

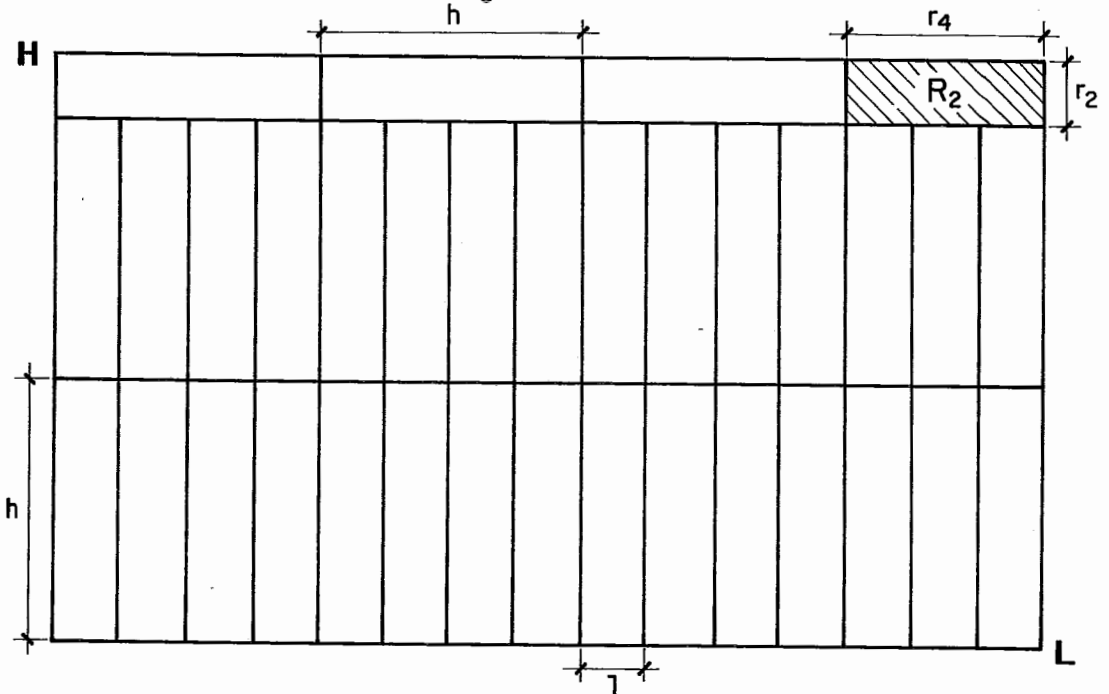


Figura 14

O resto R_2 representado pela área hachureada da

Figura 14 é expresso por $R_2 = r_2 r_4$.

Vejamos, agora, o seguinte:

$$\begin{aligned} R - R_1 &= Hr_1 + (L - r_1)r_2 - r_1r_3 = \\ &= (H - r_3)r_1 + (L - r_1)r_2 \end{aligned}$$

Temos que $H \equiv r_3 \pmod{\ell} \rightarrow (H - r_3)$ é divisível por $\ell \rightarrow (H - r_3)r_1$ é divisível por ℓ .

Também, $L \equiv r_1 \pmod{\ell} \rightarrow (L - r_1)$ é divisível por $\ell \rightarrow (L - r_1)r_2$ é divisível por ℓ . Sendo assim, $(H - r_3)r_1 + (L - r_1)r_2 = R - R_1$ é divisível por ℓ , ou seja, que:

$$R \equiv R_1 \pmod{\ell}$$

Tomando $R - R_2 = Hr_1 + (L - r_1)r_2 - r_2r_4 = (H - r_2)r_1 + (L - r_4)r_2$. Temos que:

$H \equiv r_2 \pmod{h} \rightarrow H - r_2$ é divisível por $h \rightarrow (H - r_2)r_1$ é divisível por h .

Também, $L \equiv r_4 \pmod{h} \rightarrow (L - r_4)$ é divisível por $h \rightarrow (L - r_4)r_2$ é divisível por h .

Desta forma, $(H - r_2)r_1 + (L - r_4)r_2 = R - R_2$ é divisível por h , ou seja, que:

$$R \equiv R_2 \pmod{h}$$

Agora, como estamos admitindo que $\ell < h$, sendo $L = m_1\ell + r_1$ com $0 \leq r_1 < \ell$, então claramente $r_1 < h$. Por outro lado, do fato de $H = m_2h + r_2$ com $0 \leq r_2 < h$ pode ocorrer que $r_2 > \ell$, ou seja, que na faixa horizontal hachureada de dimensões (L, r_2) da Figura 12, possam ser alocadas peças no sentido de h paralelo a L , e ℓ paralelo a r_2 , obviamente o paralelismo é o considerado entre as respectivas retas suportes.

Desta forma, temos que existem inteiros m_3 e r_3 tal que $L = m_3h + r_3$ com $0 \leq r_3 < h$ e $r_3 > r_1$.

Também existem inteiros m_4 e r_4 tal que $r_2 = H - m_2h = m_4\ell + r_4$ com $0 \leq r_4 < \ell$, ou seja, que $r_2 \equiv r_4 \pmod{\ell}$. Se gue, então, que a área de resto R é agora expressa por:

$$R = r_1(H - r_2) + (L - r_3)r_4 + r_3r_2$$

Segundo a Figura 14, temos que $R_2 = r_2r_4$, onde $L = m_4h + r_4$ com $0 \leq r_4 < h$. Então, tomando $R - R_2 = r_1(H - r_2) + (L - r_3)r_4 + r_3r_2 - r_2r_4 = r_1(H - r_2) + (L - r_3)r_4 + r_2(r_3 - r_4)$

$$\text{Agora, } \begin{cases} L = m_3h + r_3 \\ L = m_4h + r_4 \end{cases}$$

e a característica destas relações, nos permite concluir que $r_3 = r_4$ e $m_3 = m_4 \rightarrow r_2(r_3 - r_4) = 0$.

Observando que:

$$\left. \begin{array}{l} H \equiv r_2 \pmod{h} \rightarrow h \text{ divide } r_1(H - r_2) \\ L \equiv r_3 \pmod{h} \rightarrow h \text{ divide } r_4(L - r_3) \end{array} \right\} \text{concluimos:}$$

h divide $r_1(H - r_2) + r_4(L - r_3) + r_2(r_3 - r_4) = R - R_2$, ou seja,
 $R \equiv R_2 \pmod{h}$.

Por outro lado,

$$\begin{aligned} R - R_1 &= r_1(H - r_2) + (L - r_3)r_4 + r_3r_2 - r_1r_3 = \\ &= r_1(H - r_3) + r_3(r_2 - r_4) + Lr_4 - r_1r_2 = \\ &= r_1(H - r_3) + r_3(r_2 - r_4) + r_4m_1\ell + r_4r_1 - r_1r_2 = \\ &= r_1(H - r_3) + (r_3 - r_1)(r_2 - r_4) + r_4m_1\ell \end{aligned}$$

$$\text{Temos que } \begin{cases} r_2 \equiv r_4 \pmod{\ell} \\ H \equiv r_3 \pmod{\ell} \end{cases}$$

e $r_3 > r_1 \rightarrow r_3 - r_1 > 0$ e assim $R - R_1 \geq 0$ e $R \equiv R_1 \pmod{\ell}$.

Agora, o limite inferior $R(L, H)$ oferecido pelo Teorema, não é atingido seguindo os passos do procedimento de corte. Uma justificativa natural para este fato, é que o problema que estamos considerando é bem mais restrito que o problema tratado no contexto do Teorema, senão vejamos: "A partir de uma placa (5,5), gostaríamos de cortar retângulos (3,2), em um núme-

ro máximo". Na Figura 15, abaixo, ilustramos os dois modelos de corte, guilhotina e não guilhotina. Observamos que o modelo não guilhotina, é bem melhor que o guilhotina no sentido que $R_1 < R_2$ e também qualquer que seja o procedimento de corte guilhotina direcionado para a placa (5,5), é impossível melhorar a área de res

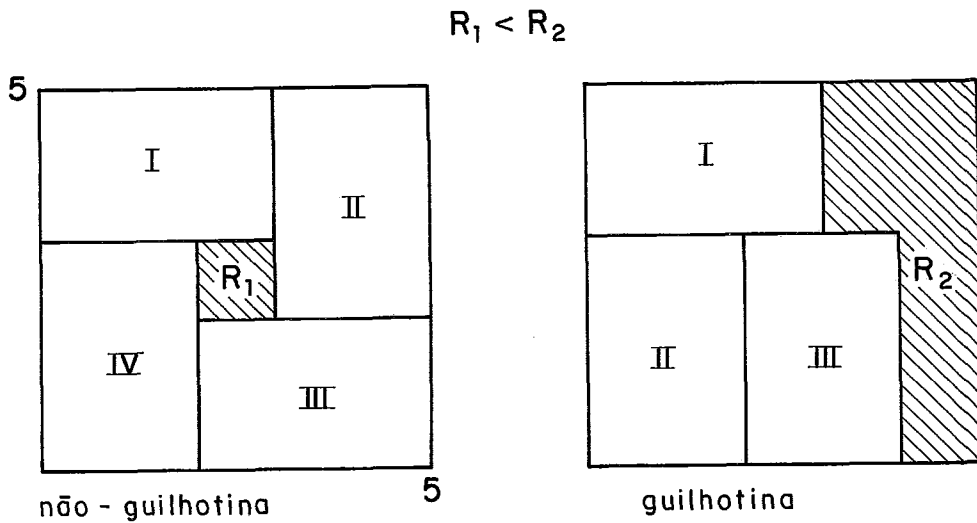


Figura 15

A influência do tipo de corte no conteúdo do Teorema pode ser observada com o seguinte exemplo: Tomando uma placa (5,11) e segundo o Lema 1, tomemos $a = b = 2, h = 3$. Observamos que:

$$\left. \begin{array}{l} 5 \equiv 2 \pmod{3} \\ 11 \equiv 2 \pmod{3} \end{array} \right\} \text{ e na Figura 16, a seguir, ilustra}$$

mos o Lema 1 quando desejamos empacotar retângulos $(1, h)=(1, 3)$ com a restrição do corte ser do tipo guilhotina.

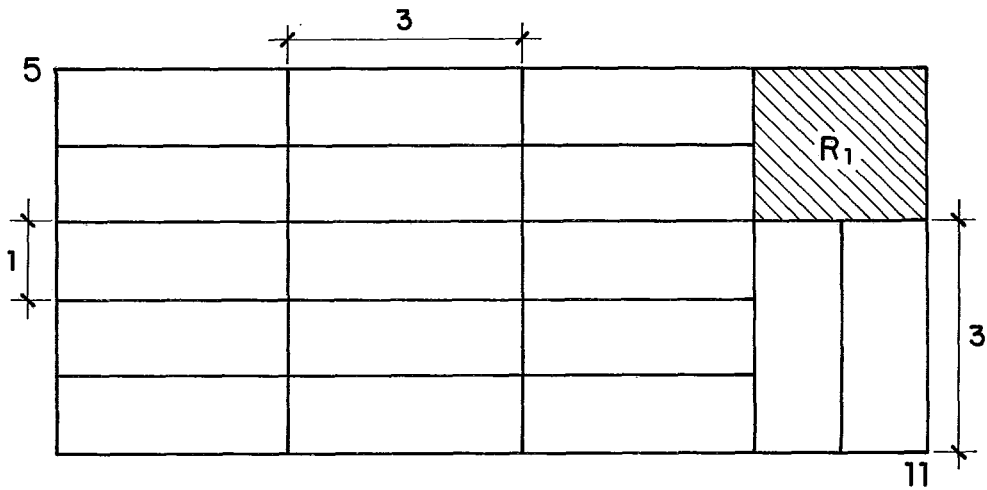


Figura 16

Temos que $a + b = 2 + 2 = 4 > 3 = h$. De acordo com o Lema 1, a área de resto R_1 deveria ser igual a $R_1 = (h - a)(h - b) = (3 - 2)(3 - 2) = 1$.

No entanto, observamos, na Figura 16, que $R_1 = 2 \times 2 = 4$ unidades de área.

Por outro lado, na Figura 17, a seguir, obtemos realmente $R_1 = 1$ mas agora a característica do corte ser guilhotina é perdida, e o Lema 1 falha para o caso em que o corte é tipo guilhotina.

Desejamos, agora, a partir da placa (40, 20) cortar segundo o procedimento, o número máximo de retângulos (4,3). O modelo de corte obtido segundo o procedimento, é o ilustrado na Figura 18, a seguir, com um resto $R = 80$, expresso em unidades de área.

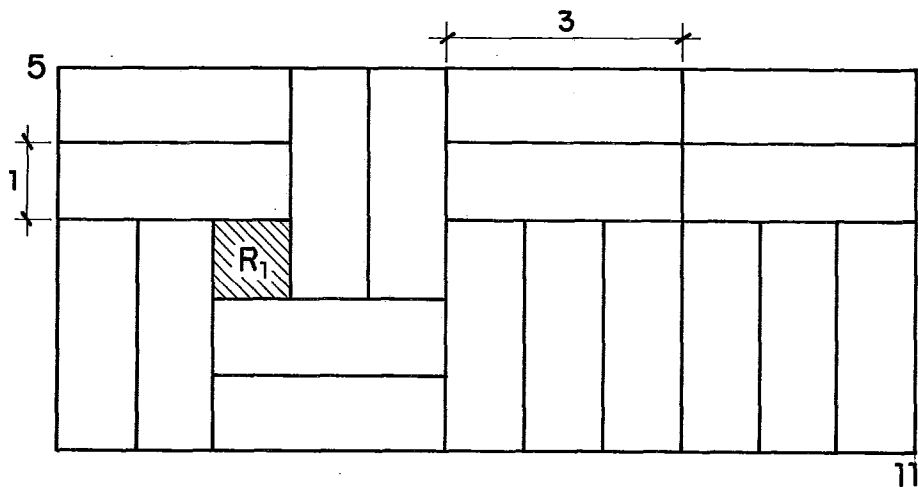


Figura 17

Por outro lado, com a restrição do corte ser guilhotina, a solução apresentada na Figura 19 é melhor que a da Figura 18, apresentando um resto R expresso por $R = 8$ unidades de área. Observamos do exemplo anterior que a utilização direta do procedimento de corte a dois estágios alternado do Capítulo II para a sua solução não é recomendável, dada a grandeza das áreas de resto envolvidas e a característica particular do problema que estamos tratando.

III.3 - Procedimento de Corte a Dois Estágios (Alternativas de Melhoramento da Solução)

Consideremos a placa de dimensões (L, H) e a peça (ℓ, h) , sendo L, H, ℓ, h números inteiros positivos com $h \leq \ell$ e admitimos que o problema seja viável, isto é, que seja possível cortar peças da placa.

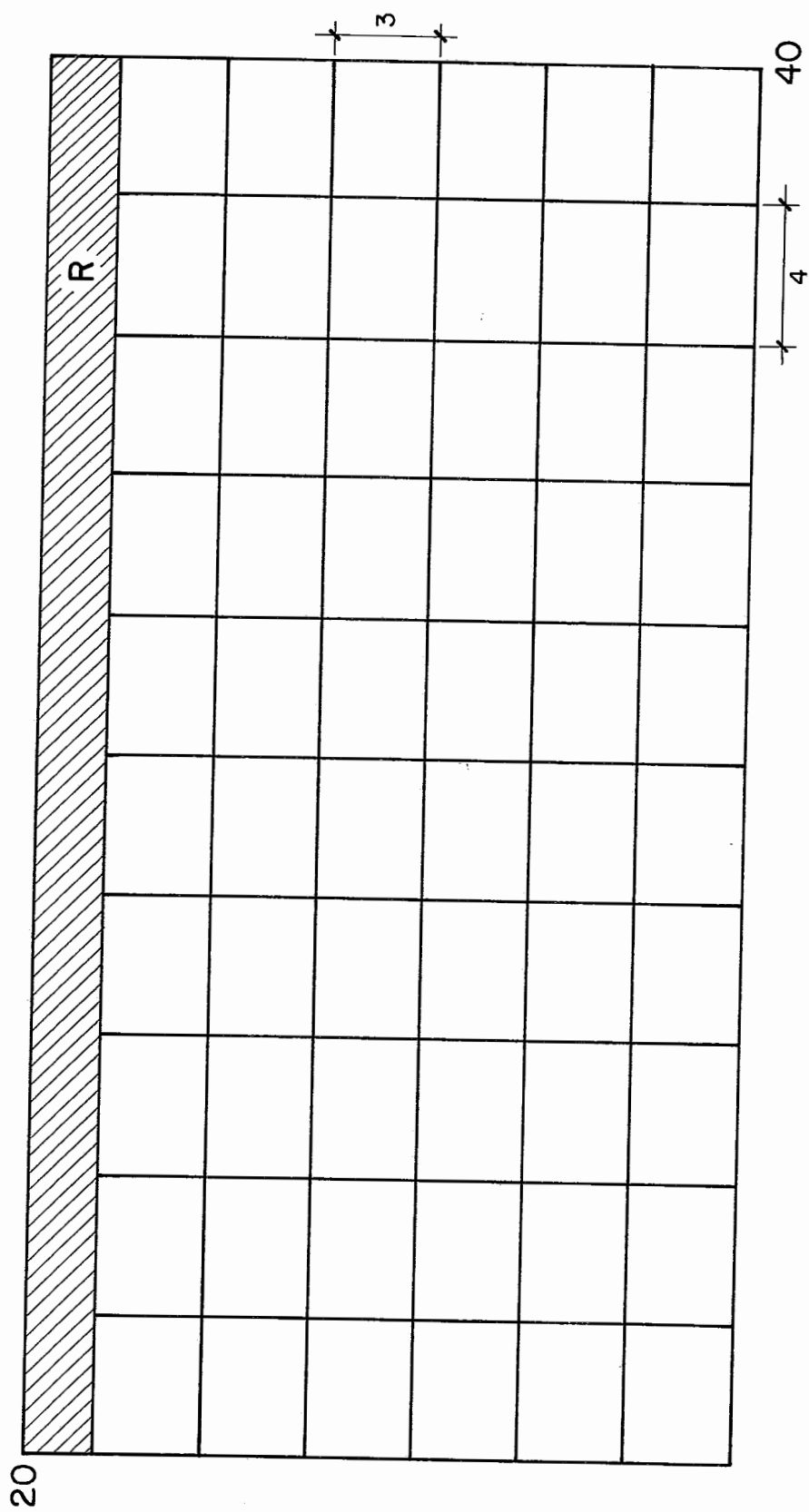


Figura 18

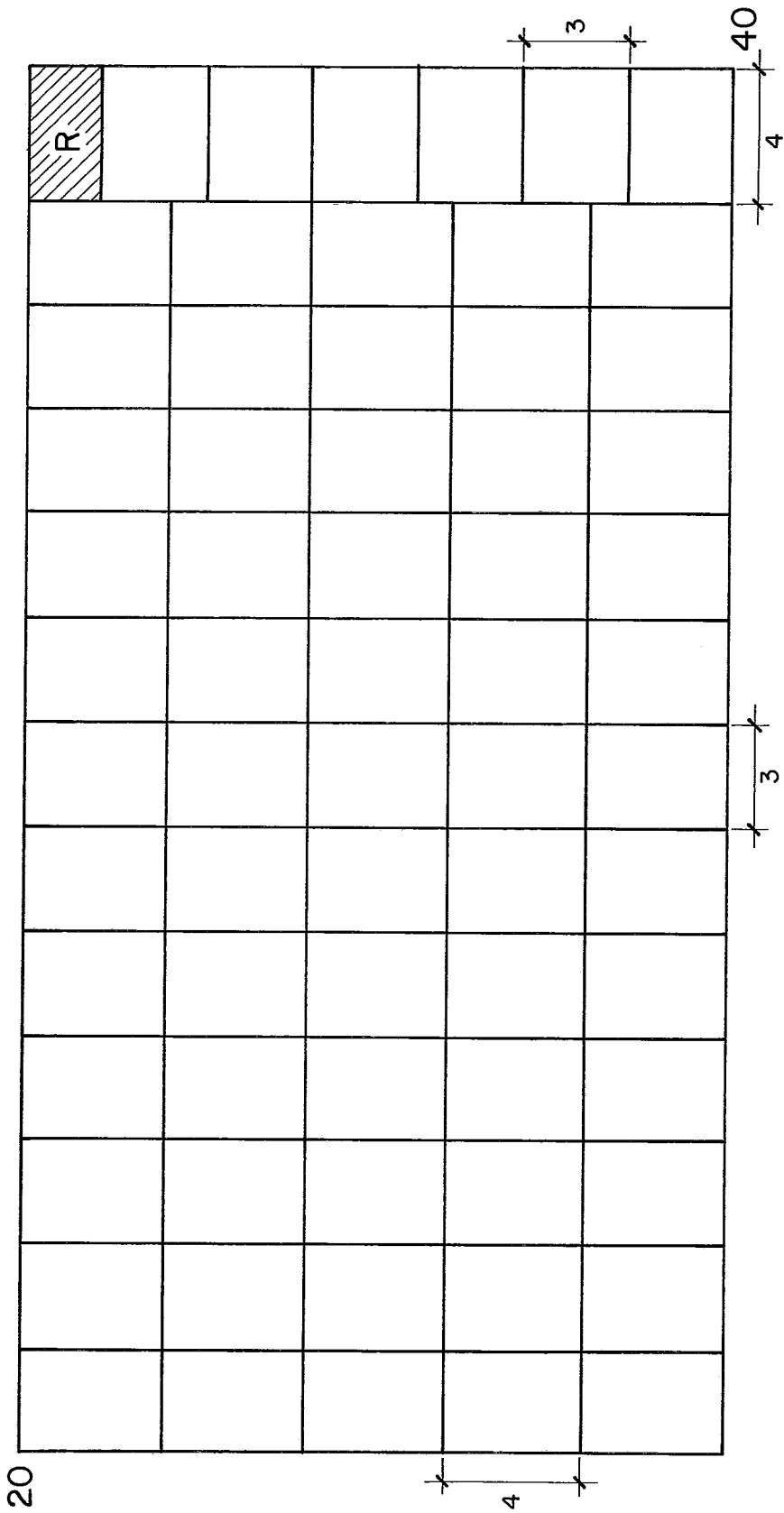


Figura 19

Primeiro Estágio:

Utilizando a idéia de geração de faixas geramos dois modelos de corte, um para cada posição fixada na alocação da peça, e tomamos o melhor, ou seja, aquele que deixa na Figura 20 a menor expressão para R.

Segundo Estágio:

O procedimento, agora, procura melhorar a solução obtida no Primeiro Estágio. Vamos admitir que o modelo de corte representado na Figura 20, seja tal que:

$$\left. \begin{array}{l} L = m_1 h + r_1 \\ H = m_2 \ell + r_2 \end{array} \right\} \text{ com } \begin{array}{l} 0 \leq r_1 < h \\ 0 \leq r_2 < \ell \end{array}$$

e, também, $r_2 < h$.

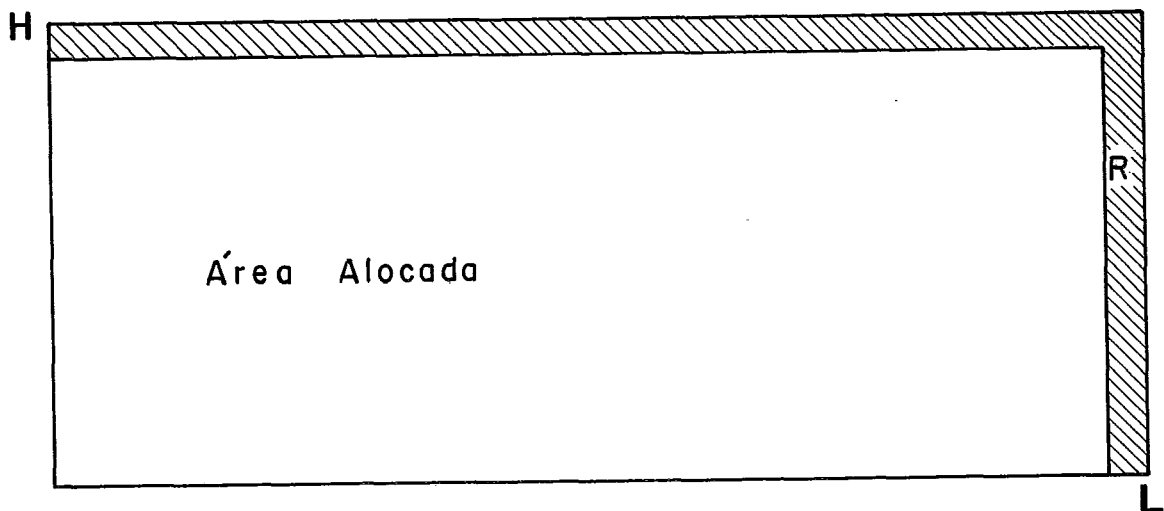


Figura 20

III.3.1-Alternativa I de Melhoramento

Restringindo-se à dimensão H da placa, procuramos determinar um número inteiro m pertencente ao intervalo $[1, m_2]$ de forma que $m\ell + r_2$ seja um múltiplo de h , ou seja, que exista um inteiro positivo m_3 tal que:

$$m\ell + r_2 = m_3h$$

Se um tal m inteiro existir temos:

$$\begin{aligned} H &= m_2\ell + r_2 = m_2\ell + r_2 - m_3h + m_3h = m_2\ell - m\ell + \\ &+ m_3h = (m_2 - m)\ell + m_3h \end{aligned}$$

Desta forma, determinamos um corte tipo guilhotina paralelo à reta suporte da dimensão L da placa e de altura $(m_2 - m)\ell$, dividindo a placa (L, H) em duas novas sub-placas $(L, (m_2 - m)\ell)$ e (L, m_3h) . Tomamos agora a sub-placa $(L, (m_2 - m)\ell)$ e restringindo-se à dimensão L da mesma procuramos determinar um número inteiro n no intervalo $[1, m_1]$ de tal forma que $nh + r_1$ seja um múltiplo de ℓ , ou seja, que exista um inteiro positivo m_4 tal que $nh + r_1 = m_4\ell$. Se um tal n existir, temos que:

$$\begin{aligned} L &= m_1h + r_1 = m_1h + r_1 - m_4\ell + m_4\ell = \\ &= m_1h - nh + m_4\ell = (m_1 - n)h + m_4\ell \end{aligned}$$

Desta forma, determinamos um corte guilhotina na

sub-placa $(L, (m_2 - m)l)$ paralelo a reta suporte da dimensão $(m_2 - m)l$ e de altura $(m_1 - n)h$, dividindo a mesma em duas novas sub-placas $((m_1 - n)h, (m_2 - m)l)$ e $(m_4l, (m_2 - m)l)$. Considerando agora o paralelismo entre as respectivas retas suportes temos que a sub-placa (L, m_3h) é cortada em peças (l, h) , alocadas de forma que l seja paralelo a L e h paralelo a H ; e o mesmo ocorrendo com a sub-placa $(m_4l, (m_2 - m)l)$. Naturalmente a sub-placa $((m_1 - n)h, (m_2 - m)l)$ é totalmente cortada sem deixar área de resto, com peças (l, h) , alocadas de forma que h seja paralelo a L e l paralelo a H .

Uma representação geométrica da Alternativa I de Melhoramento, é dada na Figura 21, abaixo, onde observamos que o procedimento procura transformar a área de resto R da Figura 20 na região hachureada da Figura 21, objetivando uma diminuição de R .

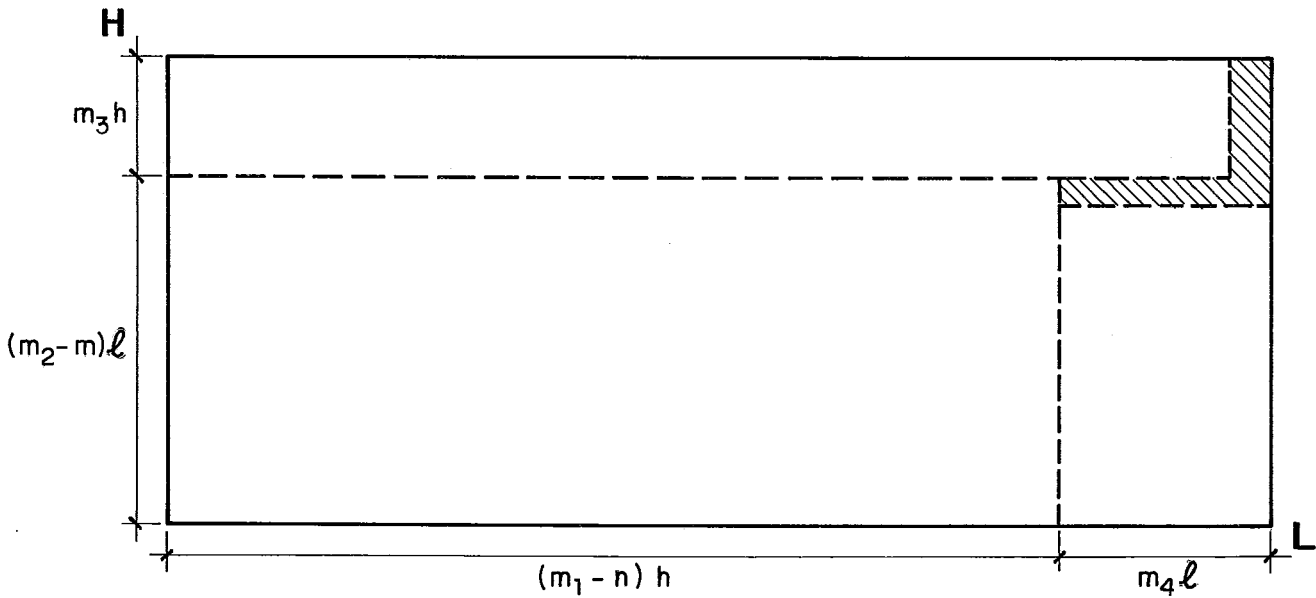


Figura 21

III.3.2-Alternativa II de Melhoramento

Neste caso basicamente o raciocínio é idêntico ao utilizado na Alternativa I, somente que o procedimento se restringe inicialmente à dimensão L da placa. A representação geométrica desta alternativa é dada na Figura 22.

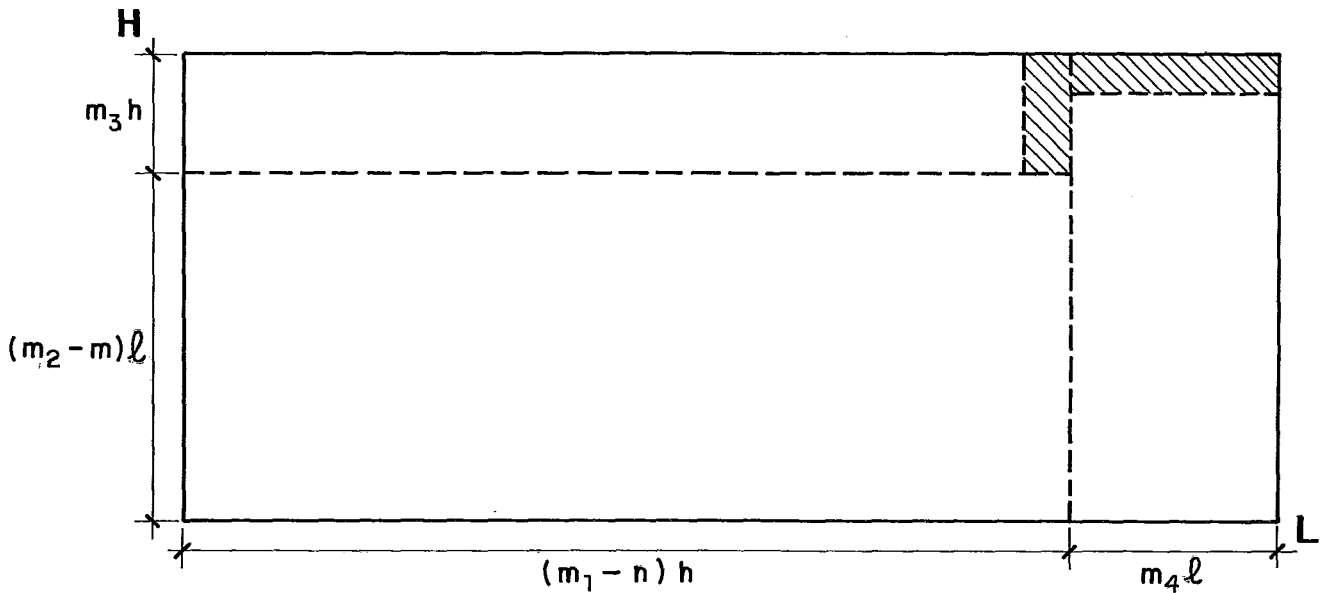


Figura 22

Dentre os modelos de corte da Figura 21 e 22, ficamos com o melhor, aquele que expressa menor grandeza para a área de resto.

Observamos que a posição da primeira guilhotina a ser dada na placa, se horizontal ou vertical, tem forte influência no sentido de se obter uma boa solução para o problema.

III.3.3-Alternativa III de Melhoramento

Passamos agora a analisar o caso em que não seja

possível determinar os inteiros \underline{m} e \underline{n} como descrito na Alternativa I. Aqui, o procedimento procura encontrar um inteiro $s \in [1, m_2]$ de forma que $s\ell + r_2 = m_5h + r_3$ com $0 < r_3 < h$ e $r_3 < r_2$ para algum inteiro m_5 pertencente a um domínio inteiro dependente dos dados da solução do Estágio I.

Se um tal inteiro \underline{s} existir, então temos:

$$\begin{aligned} H &= m_2\ell + r_2 = m_2\ell + r_2 + m_5h + r_3 - m_5h - r_3 = \\ &= m_2\ell + (r_2 - m_5h - r_3) + m_5h + r_3 = \\ &= m_2\ell - s\ell + m_5h + r_3 = (m_2 - s)\ell + m_5h + r_3 \end{aligned}$$

Desta forma, restringindo-se à dimensão H da placa determinamos uma guilhotina paralela à reta suporte da dimensão L e de altura $(m_2 - s)\ell$. Por outro lado, restringindo-se à dimensão L da placa, o procedimento procura encontrar um inteiro $r \in [1, m_1]$ de forma que:

$$rh + r_1 = m_6\ell + r_4 \text{ com } 0 < r_4 < h \text{ e } r_4 < r_1$$

e de forma análoga, m_6 pertencente a um domínio inteiro dependente dos dados da solução do Estágio I. Sendo assim, se o inteiro \underline{r} existir, temos que:

$$\begin{aligned} L &= m_1h + r_1 = m_1h + r_1 + m_6\ell + r_4 - m_6\ell - r_4 = \\ &= m_1h + (r_1 - m_6\ell - r_4) + m_6\ell + r_4 = (m_1 - r)h + m_6\ell + r_4 \end{aligned}$$

e uma guilhotina \tilde{e} determinada, paralela \tilde{a} reta suporte da dimens \tilde{a} o H e de altura $(m_1 - r)h$. Agora, o modelo de corte a se esco \tilde{l} her, \tilde{e} o melhor dentre as representações geométricas das Figuras 23 e 24, no sentido de se oferecer a menor grandeza para a respectiva \tilde{a} rea em hachureado.

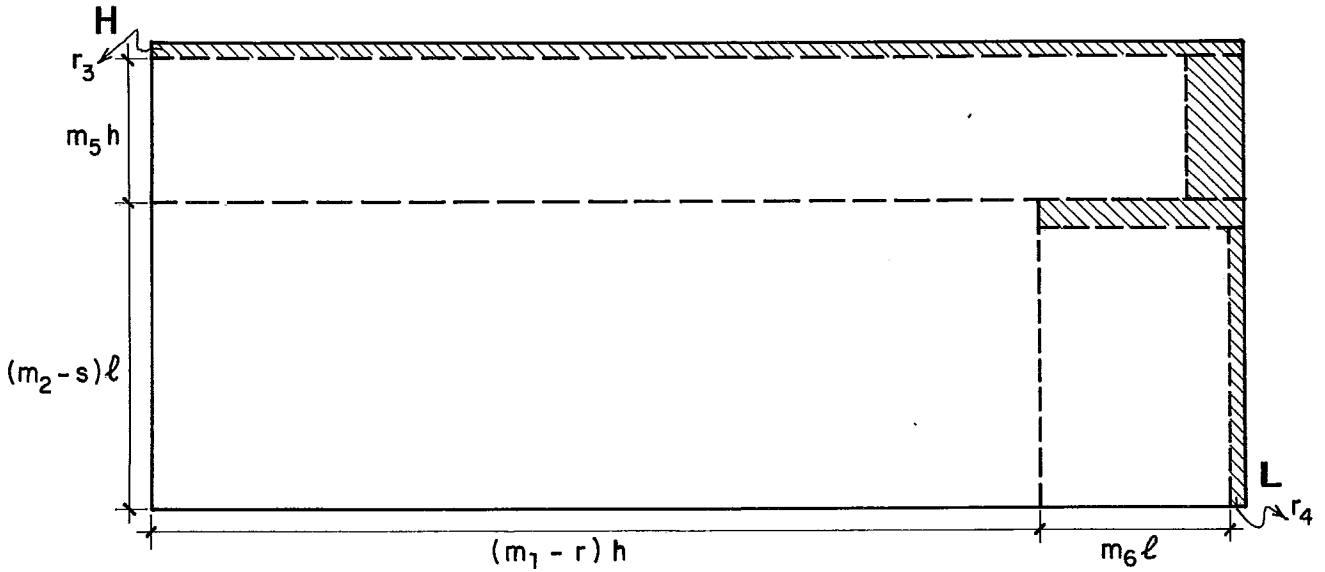


Figura 23

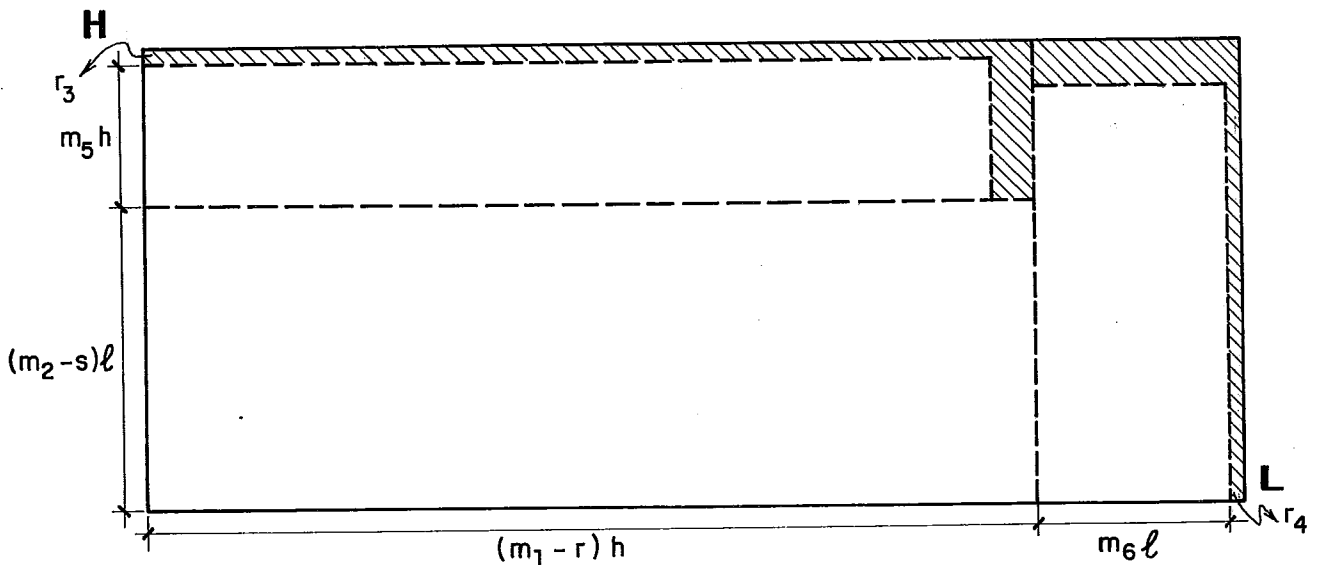


Figura 24

Para os casos em que a Alternativa I acontece em

relação a uma dimensão apenas da placa, o procedimento é semelhante, procurando determinar a posição da primeira guilhotina, analisando os dois casos.

Agora, se as três alternativas de melhoramento falharem para o problema que se está estudando, prevalece a solução do Estágio I que se está tentando melhorar.

Agora, observamos que se considerarmos que ℓ e h sejam relativamente primos, L e H dimensões da placa com $L, H \geq \ell * h$ então podemos sempre expressar:

$$\left. \begin{aligned} L &= x_1 * \ell + y_1 * h \\ H &= x_2 * \ell + y_2 * h \end{aligned} \right\} \begin{array}{l} \text{para } x_1, x_2, y_1, y_2: \text{ inteiros} \\ \text{não negativos} \end{array}$$

Desta forma, nestas condições as alternativas I e II de melhoramento podem ser aplicadas e ainda mais, para quaisquer inteiros L_1 e H_1 , com $L_1 \geq L$, $H_1 \geq H$ e $L_1 \equiv L \pmod{\ell * h}$

$$H_1 \equiv H \pmod{\ell * h}$$

temos que a placa (L_1, H_1) produz a mesma área de resto da placa (L, H) quando da aplicação das alternativas I e II de melhoramento, no corte de peças (ℓ, h)

De fato, admitindo-se que o melhor modelo de corte para a placa (L, H) oferece uma área de resto R . Agora, sendo $L_1 \equiv L \pmod{\ell * h}$ e $H_1 \equiv H \pmod{\ell * h}$ temos:

$$\begin{cases} L_1 = L + x\ell h \\ H_1 = H + y\ell h \end{cases} \quad \text{para inteiros } x \text{ e } y$$

Então, a placa (L_1, H_1) pode ser cortada segundo a representação geométrica da Figura 25.

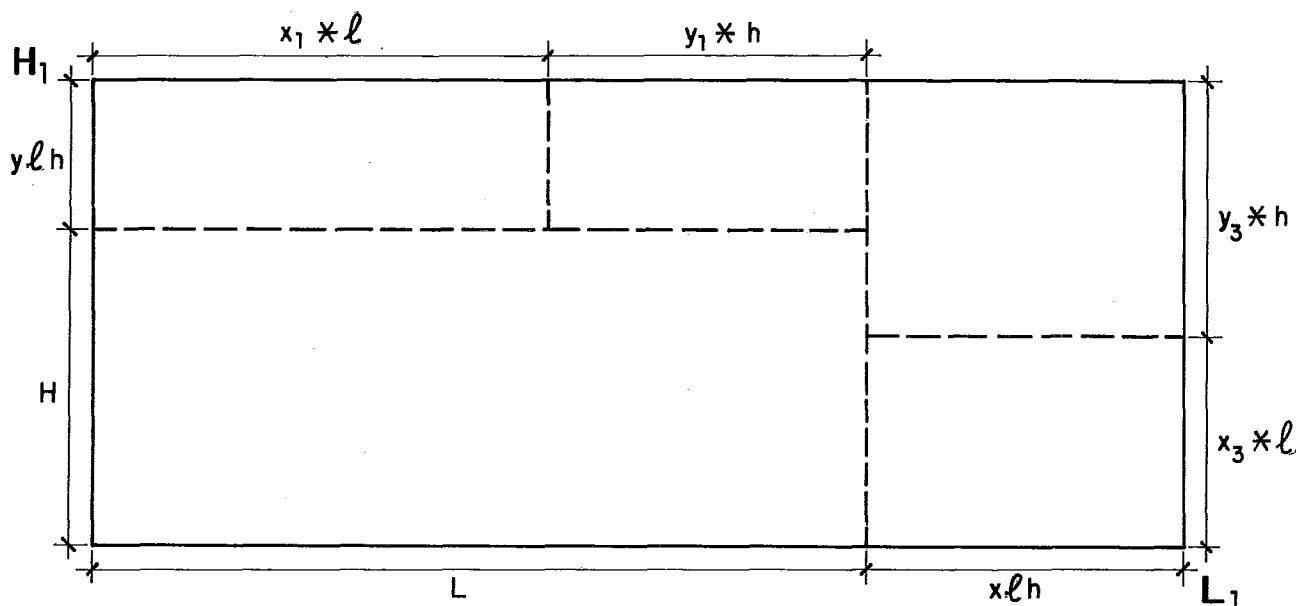


Figura 25

Dado que $L = x_1 \cdot \ell + y_1 \cdot h$, temos que a sub-placa $(L, y\ell h)$ pode ser cortada em duas novas sub-placas: $(x_1 \cdot \ell, y \cdot \ell \cdot h)$ e $(y_1 \cdot h, y \cdot \ell \cdot h)$ cada uma sendo totalmente cortada a guilhotina, sem nenhum resto. Agora, sendo $H_1 \geq H \geq \ell \cdot h$ com ℓ e h relativamente primos expressamos $H_1 = x_3 \cdot \ell + y_3 \cdot h$ para inteiros não negativos x_3, y_3 . Desta forma, a sub-placa $(x \cdot \ell \cdot h, H_1)$ pode ser cortada em duas novas sub-placas $(x \cdot \ell \cdot h, x_3 \cdot \ell)$ e $(x \cdot \ell \cdot h, y_3 \cdot h)$, naturalmente cada uma é cortada em peças (ℓ, h) a guilhotina, sem nenhum resto. Desta maneira, observamos que o corte em (L_1, H_1) produz o mesmo resto R de (L, H) .

A seguir, damos um Exemplo ilustrativo do procedi

mento. Consideremos peças do tipo $(3, 5)$ a serem cortadas de uma placa $(43, 17)$.

Estágio 1:

Solução 1: Alocação de faixas geradas segundo o sentido $(3, 5)$. Veja Figura 26.

Solução 2: Alocação de faixas geradas segundo o sentido $(5, 3)$. Veja Figura 27.

Desta forma, o Estágio 1 oferece a Solução 2 como a melhor, a ser melhorada. Observamos que na solução 2, a dimensão 43 é totalmente utilizada, logo a alternativa que resta é tentar melhorar na vertical.

Devemos procurar $m \in [1, 5]$, inteiro de forma que $3m + 2$ seja múltiplo de 5. De fato $m = 1$ fornece um tal múltiplo. Então, uma guilhotina é dada na altura $(5 - m)3 = 12$ e paralelo à dimensão horizontal.

O modelo de corte obtido é representado na Figura 28, a seguir, com $R = 11$ unidades de área e um total de 48 peças cortadas.

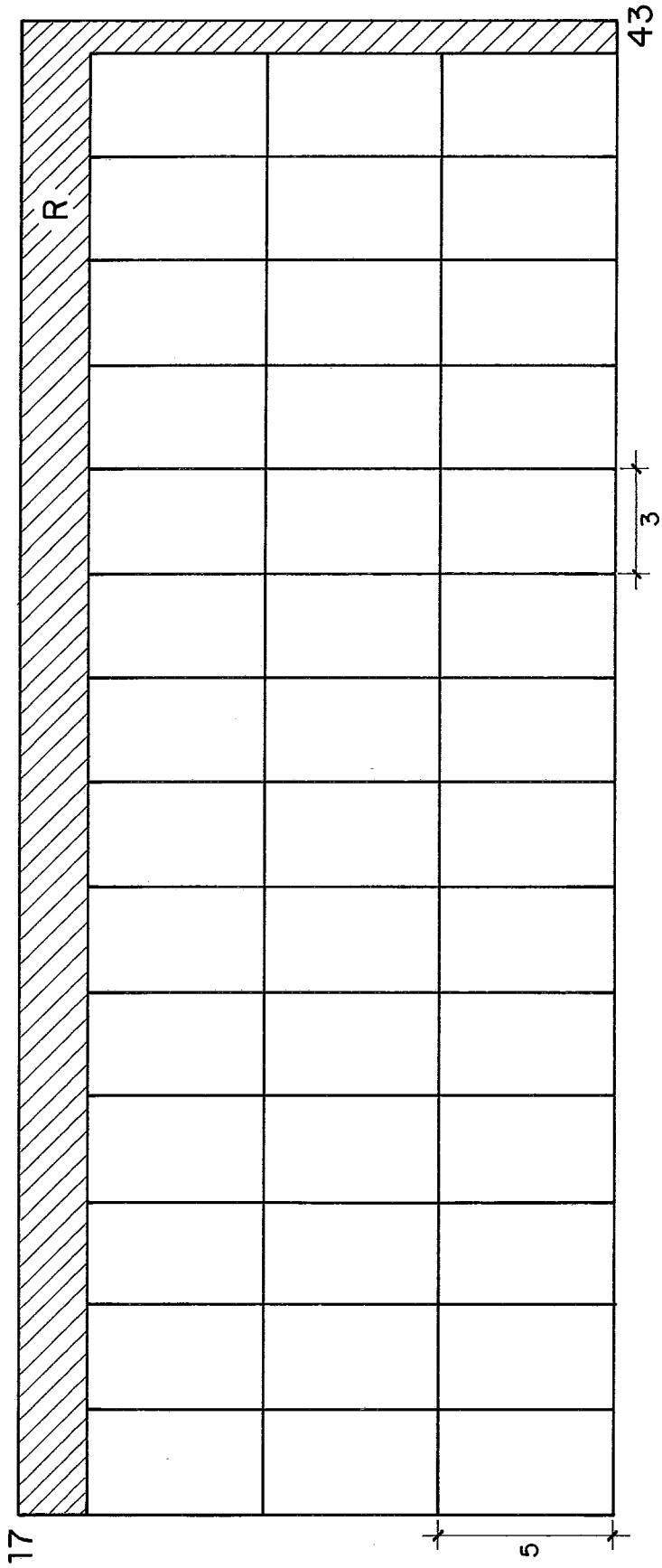


Figura 26

Temos que $R = 86 + 15 = 101$ unidades de área. Total de peças cortadas = 42 peças

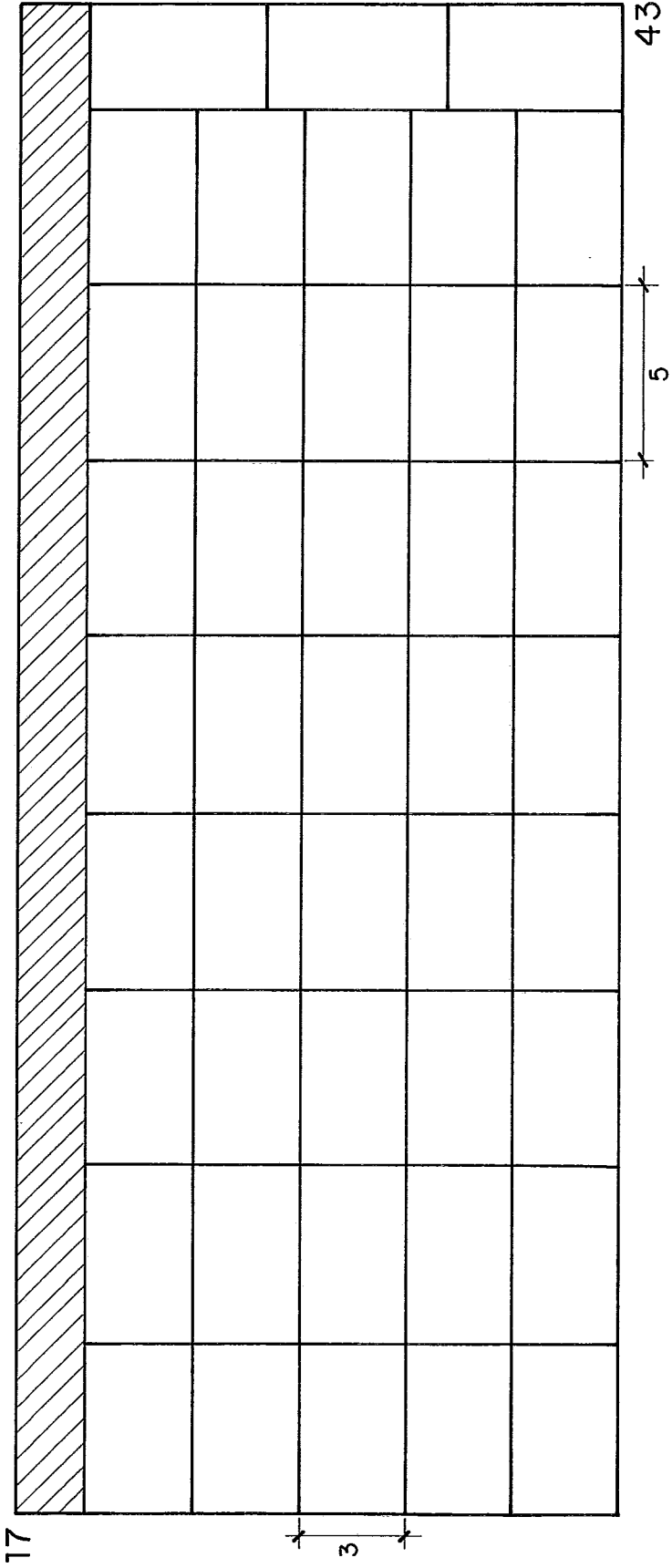
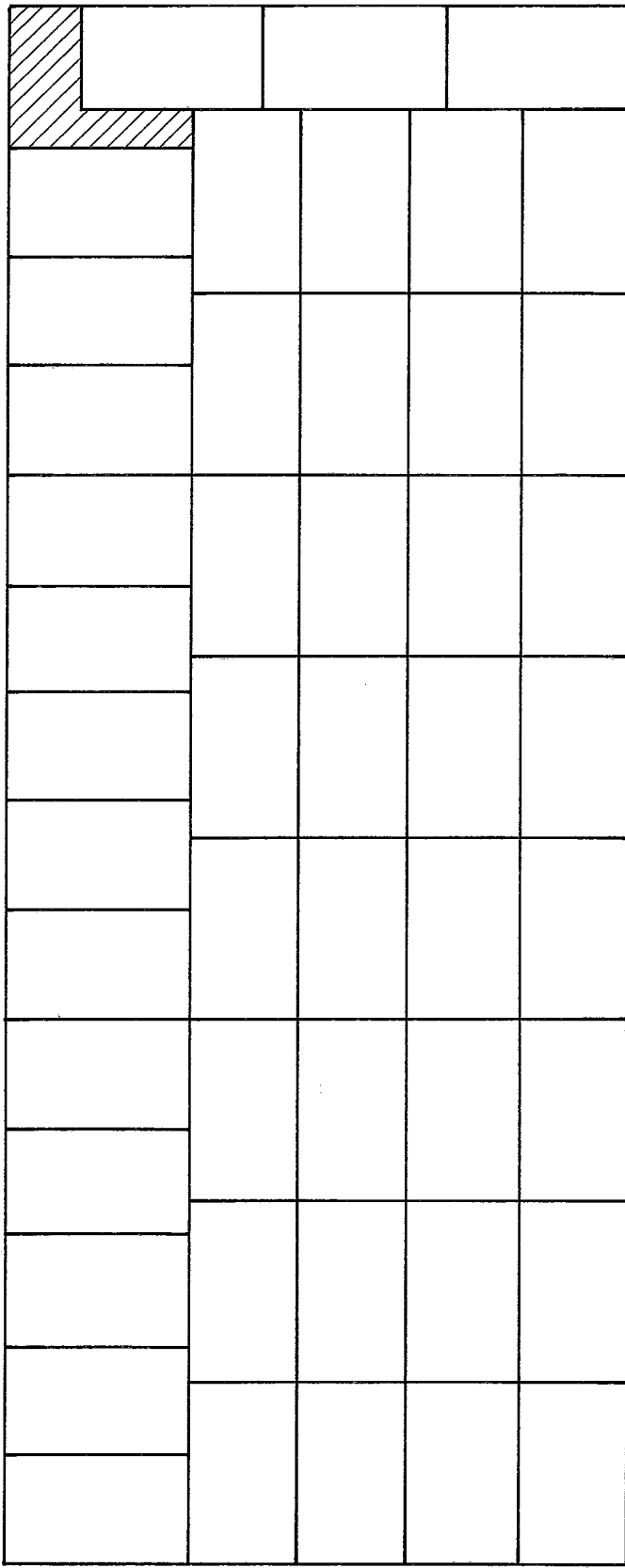


Figura 27

Temos que $R = 86$ unidades de área. Total de peças cortadas = 43 peças

17



43

Figura 28

CAPÍTULO IV - EXPERIÊNCIA COMPUTACIONAL E CONCLUSÕES

IV. EXPERIÊNCIA COMPUTACIONAL E CONCLUSÕES

IV.1 - Introdução

Neste capítulo abordaremos a implementação do Procedimento de Corte a Dois Estágios Alternado, sugerido no Capítulo II, procurando discutir todos os passos do desenvolvimento do trabalho. Em uma primeira fase, a implementação se fez utilizando uma versão dinâmica para o módulo da SUBROUTINE KNAP como descrito no Capítulo II. Baseados em resultados parciais obtidos desta primeira fase de implementação, partimos no sentido de se obter uma melhor performance para o sistema, no que diz respeito a tempo de processamento e tentativa de uma melhor utilização do material.

Desta forma, passamos a uma segunda fase de implementação do procedimento, onde uma versão Branch-Bound para o módulo da SUBROUTINE KNAP foi utilizado, bem como a adaptação de uma rotina mais eficiente para toda ordenação realizada no programa. Na seção IV.2 discutimos todos os detalhes destas duas fases de implementação do procedimento.

IV.2 - Implementação do Procedimento de Corte a Dois Estágios Alternado

Em uma Primeira Fase, a implementação se fez utilizando uma versão dinâmica para o módulo da SUBROUTINE KNAP, e os detalhes teóricos e técnicos foram discutidos no Capítulo II. Neste ponto desejamos, de forma rápida, analisar determinados elemen-

tos do ponto de vista computacional sobre a versão dinâmica acima citada. Para isto, consideremos o seguinte problema:

$$\left. \begin{array}{l} \text{maximizar} \quad \sum_{j=1}^n c_j x_j \\ \text{sujeito a} \quad \sum_{j=1}^n a_j x_j \leq b \\ x_j \in \{0, 1\} \quad j = 1, 2, \dots, n \end{array} \right\} \textcircled{15}$$

onde c_j , a_j para $j = 1, 2, \dots, n$ e b são números inteiros e positivos. A partir deste problema passemos a analisar o seguinte problema: para um dado valor inteiro $M > 0$, perguntamos se o sistema de desigualdades $\textcircled{16}$ abaixo,

$$\left. \begin{array}{l} \sum_{j=1}^n c_j x_j \geq M \\ \sum_{j=1}^n a_j x_j \leq b \end{array} \right\} \textcircled{16}$$

admite solução binária, ou seja, se existem inteiros $x_j \in \{0, 1\}$ para $j = 1, 2, \dots, n$, solução de $\textcircled{16}$. Uma maneira de solucionarmos o problema $\textcircled{16}$ é utilizar um algoritmo de programação dinâmica, veja Salkin^[42], do tipo do descrito no Capítulo II, que passamos a descrever: para $k = 1, 2, \dots, n$ e $y = 1, 2, \dots, b$ definimos $F(k, y)$ como sendo o máximo valor de $\sum_{j=1}^k c_j x_j$ sujeito a $\sum_{j=1}^k a_j x_j \leq y$, onde $x_j \in \{0, 1\}$.

Desta forma, temos que $F(k, y)$ satisfaz a condição inicial:

$$F(1, y) = \begin{cases} 0, & \text{se } y < c_1 \\ a_1, & \text{se } y \geq c_1 \end{cases}$$

e a forma de recorrência

$$F(k,y) = \text{m\`aximo} \{F(k-1, y), F(k-1, y-ak) + ak\}$$

para $k = 2, 3, \dots, n$

Assim, o problema (16) pode ser resolvido montando uma tabela da função $F(k,y)$ e, então, testando, se a condição $F(n,b) \geq M$ é satisfeita. Considerando, agora, $f(n)$ e $g(n)$ funções definidas no conjunto dos números inteiros positivos, tomando valores no conjunto dos números reais positivos, dizemos que $f(n) = O(g(n))$ se existe uma constante $c > 0$, tal que para n suficientemente grande ocorrer:

$$f(n) \leq c \cdot g(n)$$

Com isto, observamos que o procedimento dinâmico descrito acima para a resolução de (16) requer $O(nb)$ operações de adição e comparações. Admitamos que a representação binária de cada dos coeficientes a_j e c_j envolva z dígitos, isto é:

$$1 \leq a_j, c_j \leq 2^z - 1$$

e que a capacidade b do knapsack (15) seja aproximadamente da ordem de $n(2^z - 1)$ podendo, desta forma, apresentar uma representação binária com uma quantidade de dígitos da ordem de $z + \log_2 n$.

Vemos, assim, que o armazenamento total dos dados do problema (16) se faz através de um ARRAY de tamanho da ordem de $O(nz)$ implicando, desta forma, que o esforço computacional en-

volvido é da ordem de:

$$O(n^2 z 2^z)$$

que para z suficientemente grande dentro da capacidade da máquina, não é limitado por um polinômio em nz , ou seja, que o procedimento dinâmico nestas condições não é polinomialmente limitado. É claro que uma limitação polinomial para o procedimento pode ser obtida, se uma limitação superior uniforme para os inteiros a_j e c_j for dada, rejeitando qualquer dado que viole esta limitação. Neste contexto estamos analisando um caso específico originado do problema (15), ou seja, o problema knapsack de decisão dado por (16). Uma análise do procedimento dinâmico do tipo da acima realizada pode ser direcionada para o caso específico do problema knapsack (15). O procedimento dinâmico para a resolução de 15 só se comporta de maneira eficiente do ponto de vista computacional quando a capacidade b do knapsack for pequena; quando este valor se torna relativamente grande, as dificuldades computacionais começam a surgir, dada a quantidade de armazenamento requisitada em função disto, veja Toth & Martello^[46] e Dantzig^[10].

Observamos, agora, que um problema knapsack geral com variáveis limitadas pode ser transformado em um problema knapsack $\{0, 1\}$, segundo o procedimento abaixo. Representemos por:

$\lceil x \rceil$: menor número inteiro, maior ou igual a x , e

$\lfloor x \rfloor$: maior número inteiro, menor ou igual a x ,

para um dado número real x , e consideremos o problema:

$$\left. \begin{array}{l}
 \text{maximizar} \quad \sum_{j=1}^n c_j x_j \\
 \text{sujeito a} \quad \sum_{j=1}^n a_j x_j \leq b \\
 0 \leq x_j \leq b_j \quad \text{para } j = 1, 2, \dots, n \\
 x_j: \text{ inteiro}
 \end{array} \right\} \textcircled{17}$$

onde c_j , a_j , b_j para $j = 1, \dots, n$ e b , são números inteiros positivos.

Agora, para cada $j = 1, 2, \dots, n$ fazemos:

$$x_j = \sum_{i=1}^{\ell} w_i 2^{i-1} \quad \text{com } w_i \in \{0, 1\}$$

e ℓ é escolhido de forma que a desigualdade $b_j < 2^{\ell} - 1$ seja preservada.

Desta forma, temos que para cada $j = 1, 2, \dots, n$ x_j pode ser expressa em um total de:

$$m = \lceil \log_2 b_j \rceil \text{ variáveis binárias.}$$

Os quatro passos do procedimento abaixo transforma $\textcircled{17}$ em um problema knapsack $\{0, 1\}$, veja Toth & Martello [45].

Passo 1: (Inicialização)

Faça $k = 0$, $j = 1$ e vá para o Passo 2.

Passo 2:

Faça $c = b_j$ e $v\bar{a}$ para o Passo 3.

Passo 3:

$$\text{Faça } \begin{cases} e = \lceil c/2 \rceil \\ k = k + 1 \\ \bar{p}_k = e \cdot c_j \\ \bar{w}_k = e \cdot a_j \\ c = \lfloor c/2 \rfloor \end{cases}$$

Se $c > 0$ repita o Passo 3. Caso contrário, $v\bar{a}$ para o Passo 4.

Passo 4:

Se $j < n$, faça $j = j + 1$ e $v\bar{a}$ para o Passo 2. Caso contrário, pare. O problema knapsack $\{0, 1\}$ transformado é dado por (18) abaixo.

$$\left. \begin{array}{l} \text{maximizar } \sum_{i=1}^k \bar{p}_i y_i \\ \text{sujeito a } \sum_{i=1}^k \bar{w}_i y_i \leq b \\ y_i \in \{0, 1\} \text{ para } i = 1, \dots, k \end{array} \right\} \text{(18)}$$

Observamos, então, que o problema (17) de n variáveis é transformado no problema (18) com um total de k variáveis,

onde:

$$k = \sum_{j=1}^n \lceil \log_2 b_j \rceil$$

mostrando, assim, que a transformação acima sō serā viāvel do ponto de vista computacional para valores pequenos de b_j para $j = 1, 2, \dots, n$. De posse disto, observamos que podemos direcionar uma anālise anāloga ā realizada para o problema (15), para o procedimento dināmico utilizado na resoluçāo dos problemas tipo knapsack gerais com variāveis limitadas que sāo gerados no procedimento de corte a dois estāgios alternado do Capītulo II. Sem o desejo de prolongarmos esta discussāo, em linhas gerais isto justifica a atitude por nōs adotada, de procurarmos adaptar um algoritmo mais eficiente para o mōdulo da SUBROUTINE KNAP.

Neste sentido, passamos, agora, ā segunda Fase de Implementaçāo do Procedimento, discutindo a versāo Branch-Bround, utilizada para o mōdulo da SUBROUTINE KNAP.

Tomemos, entāo, o problema (17) e assumimos que:

i) c_j, a_j, b_j : sāo inteiros positivos ($j=1, \dots, n$) como jā considerado no modelo (17);

ii) $\sum_{j=1}^n b_j a_j > b$ e $b_j a_j \leq b$ ($j = 1, 2, \dots, n$).

e que os itens tenham sido ordenados de forma tal que:

$$\frac{c_j}{a_j} \geq \frac{c_{j+1}}{a_{j+1}} : j = 1, 2, \dots, n-1.$$

De posse disto, Toth & Martello [45] desenvolveram um algoritmo do tipo Branch-Bound para a resolução do problema 17 baseado no seguinte resultado:

Teorema: Seja: $\ell = \text{m\`a}x \left\{ j / \sum_{i=1}^j b_i a_i \leq b \right\}$

$$c = \sum_{i=1}^{\ell} b_i c_i + \left[(b - \sum_{i=1}^{\ell} b_i a_i) / a_{\ell+1} \right] c_{\ell+1}$$

$$b' = b - \sum_{i=1}^{\ell} b_i a_i - \left[(b - \sum_{i=1}^{\ell} b_i a_i) / a_{\ell+1} \right] a_{\ell+1}$$

$$U_1 = c + \left[b' c_{\ell+2} / a_{\ell+2} \right]$$

$$U_2 = c + \left[c_{\ell+1} - (a_{\ell+1} - b') c_{\ell} / a_{\ell} \right]$$

Ent\`ao, $\text{m\`a}x \{U_1, U_2\}$ \u00e9 um limite superior para o problema (17).

Prova: Da forma como foi definido, c representa o m\`aximo que se obt\u00e9m inserindo na mochila a quantidade m\`axima dos itens iniciais, como permitido pelos seus limites, at\u00e9 encontrarmos um item (o item de ordem $\ell + 1$ no caso), onde o seu limite superior n\`ao \u00e9 atingido, e ent\`ao, o n\u00famero m\`aximo poss\u00edvel deste item \u00e9 selecionado. Naturalmente, b' expressa o que restou de b , ap\u00f3s a inser\u00e7\~ao acima realizada.

Agora, a solu\u00e7\~ao \u00f3tima para o problema (17) pode ser obtida de duas maneiras:

- i) sem inserir na mochila novos elementos do item de ordem $\ell + 1$;
- ii) inserindo na mochila algum elemento do item de ordem $\ell + 1$.

Para o caso i), claramente a solução de (17) não pode exceder o valor de U_1 . Por outro lado, o caso ii) só é possível se, pelo menos, um elemento de algum item da inserção, realizada, for removido. Desta forma, o limite U_2 é obtido supondo que o elemento para ser removido tem exatamente o mínimo valor necessário dos pesos a_j , ou seja, $a_{\ell+1} - b'$ e o menor valor de $\frac{c_j}{a_j}$, ou seja, $\frac{c_\ell}{a_\ell}$.

E, assim, conclue-se que máximo $\{U_1, U_2\}$ é um limite superior para o problema (17).

Para a descrição dos Passos do algoritmo Branch-Bound utilizado, consideremos o seguinte:

y_j : para $j = 1, \dots, n$ representa uma solução parcial para o problema (17), na estrutura Branch-Bound.

$c = \sum_{j=1}^n c_j y_j$: representa o respectivo custo parcial.

x_j : para $j = 1, \dots, n$ representa a solução ótima para o problema (17).

$$V = \sum_{j=1}^n c_j x_j \text{ representa o custo \u00f3timo de } \textcircled{17}.$$

Passo 1: (Inicializa\u00e7\u00e3o)

Ordene todos os itens do problema $\textcircled{17}$ de forma tal que:

$$\frac{c_j}{a_j} \geq \frac{c_{j+1}}{a_{j+1}} \text{ para } j = 1, 2, \dots, n-1$$

$$\text{Fa\u00e7a } \begin{cases} b' = b - b_1 a_1 \\ c = b_1 c_1 \\ y_1 = b_1 \\ i = 2 \end{cases}$$

e v\u00e1 para 1.1

$$1.1 \text{ - Fa\u00e7a } s = \left\lfloor \frac{b'}{a_i} \right\rfloor \text{ Se } s < b_i, \text{ v\u00e1 para } \underline{1.2}$$

$$\text{Caso contr\u00e1rio, fa\u00e7a } \begin{cases} b' = b' - b_i a_i \\ c = c + b_i c_i \\ y_i = b_i \\ i = i + 1 \end{cases}$$

e repita 1.1.

$$1.2 \quad - \quad \text{Faça} \quad \begin{cases} b' = b' - s a_i \\ c = c + s c_i \\ y_i = s \end{cases}$$

$$\text{Se } b' = 0 \quad \text{faça} \quad \begin{cases} V = c \\ x_j = y_j \quad \text{para } j = 1, 2, \dots, i \\ x_j = 0 \quad \text{para } j = i+1, \dots, n \end{cases}$$

e Pare.

$$\text{Caso contrário, faça} \quad \begin{cases} c_{n+1} = 0 \\ a_{n+1} = 1 \\ b_{n+1} = b + 1 \\ V = 0 \\ y_j = 0 \quad \text{para } j = i+1, \dots, n \end{cases}$$

Calcule:

$$\text{Lim} = \text{máximo} \left\{ \left(c + \left[b' \frac{c_i + 1}{a_i + 1} \right] \right), \left(c + \left[c_i - (a_i - b') \frac{c_i - 1}{a_i - 1} \right] \right) \right\} \text{ e}$$

$$(\min_j = \text{mínimo} \{ a_k / j < k \leq n \}, j = 1, \dots, n-1)$$

$$\min_n = b + 1$$

e vá para o Passo 4

Passo 2: (Encontre o próximo item que pode ser inserido na solução parcial atual)

$$\text{Faça } s = \left\lfloor \frac{b'}{a_j} \right\rfloor$$

$$\text{Se } s > 0, \text{ faça } \begin{cases} b'' = b' \\ c'' = c \\ j = i \end{cases}$$

e vá para o Passo 3.

$$\text{Caso contrário, se } V \geq c + \left\lfloor b' \frac{c_{i+1}}{a_{i+1}} \right\rfloor$$

vá para o Passo 5, se não, faça $i = i + 1$ e repita o Passo 2.

Passo 3: (Construção de uma nova solução parcial)

$$\text{Se } s \geq b_j, \text{ então faça } \begin{cases} b'' = b'' - b_j a_j \\ c'' = c'' + b_j c_j \\ j = j + 1 \\ s = \left\lfloor \frac{b''}{a_j} \right\rfloor \end{cases}$$

e repita o Passo 3.

Caso contrário, se $V \geq c'' + \left\lfloor b'' \frac{c_j}{a_j} \right\rfloor$ vá para o Passo 6, se não, existem duas possibilidades:

$$3.1 \quad - \quad s \cdot a_j < b'' \text{ e } j < n, \text{ então faça } \begin{cases} b' = b'' - s \cdot a_j \\ c = c'' + s \cdot c_j \\ y_h = b_h \text{ para } h = i, \\ \quad \quad \quad \dots, j-1 \\ y_j = s \\ i = j \end{cases}$$

e vá para o Passo 4.

$$3.2 \quad - \quad s \cdot a_j = b'' \text{ ou } j \geq n, \text{ então}$$

$$\text{Faça } \begin{cases} v = c'' + s \cdot c_j \\ x_h = y_h \text{ para } h = 1, \dots, i-1 \\ x_h = b_h \text{ para } h = i, \dots, j-1 \\ x_j = s \\ x_h = 0 \text{ para } h = j+1, \dots, n \end{cases}$$

Se $v = \text{Lim}$, então pare. Caso contrário, vá para o Passo 6.

Passo 4: (Teste se a solução parcial atual pode ser melhorada)

Faça $i = i + 1$. Se $b' < \text{Min}_{i-1}$ vá para o Passo 5.

Caso contrário, vá para o Passo 2.

Passo 5: (Teste se é possível realizar reposição de itens)

$$\text{Se } V < c, \text{ faça } \begin{cases} V = c \\ x_j = y_j, j = 1, \dots, n \end{cases}$$

e se $V = \text{Lim}$, então pare.

Caso contrário (isto é, $v \geq c$ ou $v \neq \text{Lim}$)

$$\text{Se } y_n > 0, \text{ faça } \begin{cases} b' = b' + y_n a_n \\ c = c - y_n c_n \\ y_n = 0 \end{cases}$$

e em qualquer caso, vá para o Passo 6.

Passo 6: (Backtrack)

Encontre o maior $k < i$, para o qual $y_k > 0$

Se nenhum k nestas condições existe, então, pare.

Caso contrário,

$$\text{Faça } \begin{cases} R = b' \\ b' = b' + a_k \\ c = c - c_k \\ y_k = y_k - 1 \end{cases}$$

Se $V \geq c + \left[b' \frac{c_{k+1}}{a_{k+1}} \right]$, então

$$\text{Faça } \begin{cases} b' = b' + y_k \cdot a_k \\ c = c - y_k c_k \\ y_k = 0 \\ i = k \end{cases}$$

e repita o Passo 6.

Caso contrário, se $R \geq \text{Min}_k$, faça $i = k + 1$ e vá para o Passo 2.

$$\text{Se não, faça } \begin{cases} m = k + 1 \\ i = m \end{cases}$$

e vá para o Passo 7.

Passo 7: (Tente substituir elementos do item de ordem m por um elemento do item de ordem k).

Faça $D = a_m - a_k$. Existem três possibilidades:

7.1 - $D = 0$. Então, vá para o Passo 8.

7.2 - $D > 0$. Se $D > R$ ou $v \geq c + c_m$, então vá para o Passo 8.

Caso contrário, faça:

$$\left\{ \begin{array}{l} y = c + c_m \\ x_j = y_j \quad \text{para } j = 1, \dots, k \\ x_j = 0 \quad \text{para } j = k+1, \dots, n \text{ e } j \neq m \\ x_m = 1 \end{array} \right.$$

Se $v = \text{Lim}$, então pare.

$$\text{Se não, faça } \left\{ \begin{array}{l} R = R - D \\ k = m \end{array} \right.$$

e \bar{v} para o Passo 8.

7.3 - $D < 0$: Faça $T = R - D$

Se $T < \text{Min}_{m-1}$, então \bar{v} para o Passo 8.

$$\text{Caso contrário, faça } \left\{ \begin{array}{l} s = \left\lfloor \frac{b'}{a_m} \right\rfloor \\ j = m \\ i = j \\ b'' = b' \\ c'' = c \end{array} \right.$$

e \bar{v} para o Passo 3.

Passo 8: (Verifique se itens sucessores ao item de ordem m , podem ser utilizados).

$$\text{Se } m = n \text{ ou } V \geq c + \left[b' \frac{c_{m+1}}{a_{m+1}} \right]$$

então vá para o Passo 6.

Caso contrário, faça $m = m + 1$ e vá para o Passo 7.

Para os testes computacionais com o procedimento de corte a dois estágios alternado, foram utilizados os dados de Albano & Orsini^[2] mostrados na Tabela 1, onde para uma placa de dimensões 2550 x 12030 (em mm), gerou-se aleatoriamente sobre a mesma um modelo de corte guilhotina de forma que todo material fosse utilizado.

Tabela 1

Tipo de Peça	Nº Demandado	Comprimento (mm)	Largura (mm)
1	1	71	1708
2	1	71	403
3	1	71	439
4	1	69	174
5	3	66	174
6	10	69	167
7	30	66	167
8	1	69	706
9	3	66	706
10	10	438	120

Tabela 1 (continuação)

Tipo da Peça	Nº Demandado	Comprimento (mm)	Largura (mm)
11	50	438	119
12	1	4380	85
13	25	4380	70
14	1	69	353
15	14	63	353
16	1	846	73
17	4	846	70
18	1	1590	76
19	13	1590	64
20	3	69	908
21	1	192	71
22	9	192	62
23	30	64	66
24	1	87	112
25	11	86	112
26	11	87	107
27	121	86	107
28	1	68	1289
29	8	63	1289
30	10	1705	66
31	1	1705	85
32	1	1705	84
33	1	1705	111
34	2	79	1610
35	5	129	322

Tabela 1 (continuação)

Tipo da Peça	Nº Demandado	Comprimento (mm)	Largura (mm)
36	50	124	322
37	1	178	65
38	8	178	63
39	2	89	153
40	12	89	148
41	1	910	135
42	3	908	135
43	1	226	134
44	16	213	134
45	1	367	558
46	9	363	558
47	1	730	124
48	4	726	124
49	13	730	123
50	52	726	123
51	1	95	2550
52	1	81	2550

Ambas as versões foram rodadas em um Burroughs 6700 para os dados da Tabela 1, e os resultados são ilustrados na Tabela 2. A análise dos resultados se concentra, basicamente, sobre dois parâmetros: porcentagem de aproveitamento do material e tempo de processamento envolvido. Neste ponto salientamos mais alguns detalhes sobre ambas as versões implementadas. Observamos que a versão Branch-Bound requer uma ordenação da lista de itens

Tabela 2

Problema	Dimensões da Placa de Estoque (em mm)	Nº de Placas Necessárias	Nº de Peças de Demanda	Versão Dinâmica		Versão Branch-Bound	
				% de Aproveitamento do Material	Tempo de Processamento (em seg.)	% de Aproveitamento do Material	Tempo de Processamento (em seg.)
1	12030x2550	2	563	94.65	120	96.39	8
2	12030x2550	4	1689	99.05	255	98.12	9
3	12030x2550	7	3378	98.34	275	98.85	12
4	12030x2550	10	5067	97.98	485	98.99	12
5	12030x2550	13	6756	-	-	99.20	22

do problema (17) (veja Passo 1 do algoritmo) segundo a relação de ordem estabelecida. Como observado no Capítulo II, o reabastecimento da lista de placas de estoque com sub-placas geradas durante o procedimento de corte requer também uma ordenação destas sub-placas sempre que a inserção for realizada. Outras ordenações se fazem necessárias para o estágio de alocação das peças já cortadas. Por outro lado, enfatizamos que a versão dinâmica trabalha sem a necessidade de ordenação da lista de itens como exigido na versão Branch-Bound. Algoritmos distintos de ordenação de uma lista de n números inteiros positivos dados foram utilizados.

Para a versão dinâmica todas as ordenações se fizeram utilizando um algoritmo de ordenação de complexidade $O(n^2)$. Já para a versão Branch-Bound todas as ordenações necessárias se fizeram utilizando um algoritmo mais eficiente, de complexidade $O(n \log n)$ no pior caso, referencie Hedetniemi & Goodman^[30] para maiores detalhes.

A característica dos problemas 2, 3, 4 e 5 é a seguinte: a partir da Tabela 1, a demanda foi multiplicada por 3, 6, 9 e 12 originando, respectivamente, as demandas para os problemas 2, 3, 4 e 5, onde para cada problema, os modelos de corte tipo guilhotina aleatoriamente gerados, deixam resto zero. Uma análise da Tabela 2 nos mostra a superioridade da versão Branch-Bound sobre a Dinâmica. Uma característica expressiva da versão Branch-Bound é que a medida que aumentamos o porte do problema, o procedimento tende a fazer uma melhor utilização do material com um esforço computacional relativamente modesto.

IV.3 - Conclusões

A escolha de um procedimento de corte geral aproximado se justifica pela alta complexidade combinatória do problema aqui considerado, referencie Dyson & Chambers [14], Gilmore & Gomory [21,22,23].

Neste contexto, como sugerido no Capítulo II, o Procedimento de Corte a Dois Estágios Alternado, procura gerar modelos de corte de forma que seja feita uma boa utilização do material, com um custo computacional relativamente barato. No Capítulo III concentramos algumas idéias sobre um caso bastante particular do problema aqui estudado, e em nenhum ponto desejamos esgotar o assunto a respeito do mesmo. Uma alternativa de estudo seria talvez procurar alocar resultados, que garantam, de forma mais abrangente a aplicabilidade das alternativas de melhora sugeridas. Dentre as duas versões implementadas podemos dizer que a Branch-Bound apresentou uma melhor performance para o sistema.

Dos resultados computacionais expressos na Tabela II, observamos que o sistema melhor se adapta à medida que crescemos o porte do problema, detalhe este importante dada as características de dimensionamento dos problemas desta natureza que, geralmente, são apresentados em situações práticas.

REFERÊNCIAS BIBLIOGRÁFICAS

1. Adamowicz, M. & Albano, A. - "A Solution of the Rectangular Cutting-Stock Problem". IEEE Transactions on Systems, Man, And Cybernetics, Vol. SMC -6, pp: 302 -310, April, 1976.
2. Albano, A. & Orsini, R. "A Heuristic Solution of the Rectangular Cutting-Stock Problem". The Computer Journal, Vol. 23, Number 4, 1980.
3. Barnett, S. & Kynch, G.J. - "Exact Solution of a Simple Cutting Problem". Operations Research, Vol. 15, pp: 1051-1056, 1967.
4. Barnes, F.W. - "Packing the Maximum Number of $m \times n$ Tiles in a Large $p \times q$ Rectangle". Discrete Mathematics, Vol. 26, pp: 93-100, 1979.
5. Bellman, R. - "Comment on Dantzig's Paper on Discrete Variable Extremum Problems". Operations Research, Vol. 5, No 4-6, August - December, 1957.
6. Cabot, A.V. - "An Enumeration Algorithm for Knapsack Problems". Operations Research, Vol. 18, 1970.
7. Chvátal, V. - "Hard Knapsack Problems". Operations Research, Vol. 28, No 6, 1980.
8. Christofides, N. & Whitlock, C. - "An Algorithm for Two-Dimensional Cutting Problems". Operations Research, Vol. 25, No 1, January - February, 1977.

9. Coverdale, I. & Wharton, F. - "An Improved Heuristic Procedure for a Nonlinear Cutting-Stock Problem". Management Science Vol. 23, pp: 78-86, 1976.
10. Dantzig, G.B. - "Discrete Variable Extremum Problems". Operations Research, Vol. 5, 1957.
11. De Cani, P. - "A Note on the Two-Dimensional Rectangular Cutting-Stock Problems". Journal of Operational Research Soc., Vol. 29, pp: 703-706, 1978.
12. Dembo, R.S. & Hammer, P.L. - "A Reduction Algorithm for Knapsack Problems". Methods of Operations Research, Vol. 36, 1979.
13. Dyson, R.G. & Gregory, A.S. - "The Cutting-Stock Problem in the Flat Glass Industry". Operational Research Quarterly, Vol. 25, pp: 41-53, 1974.
14. Dyson, R.G. & Chambers, M.L. - "The Cutting-Stock Problem in the Flat Glass Industry - Selection of Stock Sizes". Operational Research Quarterly, Vol. 27, pp: 949 - 957, 1976.
15. Dickhoff, H. - "A New Linear Programming Approach to the Cutting-Stock Problem". Operations Research, Vol. 29, No 6, November - December, 1981.
16. Eilon, S. & Christofides, N. - "The Loading Problem". Management Science, Vol. 17, No 5, January, 1971.

17. Faaland, B.H. - "The Multiperiod Knapsack Problem". Operations Research, Vol. 29, No 3, May - June, 1981.
18. Frank, C.R. - "A Note on the Assortment Problem". Management Science, Vol. 11, pp: 724-726, 1965.
19. Freeman, H. & Haims, M.J. - "A Multistage Solution of the Template Layout Problem". IEEE Transactions on Systems Science and Cybernetics, Vol. SSC-6, No 2, April, 1970.
20. George, J.A. & Robinson, D.F. - "A Heuristic for Packing Boxes into A Container". Computer & Operational Research, Vol. 7, pp: 147-156, 1980.
21. Gilmore, P.C. & Gomory, R.E. - "A Linear Programming Approach to the Cutting-Stock Problem". Operations Research, Vol. 9, pp: 849-859, 1961.
22. Gilmore, P.C. & Gomory, R.E. - "A Linear Programming Approach to the Cutting-Stock Problem, Part II". Operation Research, Vol. 11, pp: 863-888, 1963.
23. Gilmore, P.C. & Gomory, R.E. - "Multistage Cutting - Stock Problems of Two and More Dimensions". Operations Research, Vol. 13, pp: 94-120, 1965.
24. Gilmore, P.C. & Gomory, R.E. - "The Theory and Computation of Knapsack Functions". Operations Research, Vol. 14, pp: 1045-1074, 1966.

25. Greenberg, H. & Hegerich, R.L. - "A Branch Search Algorithm for the Knapsack Problem". Management Science, Vol. 16, No 5, 1970.
26. Gribov, A.B. - "Algorithm for Solving the Problem of a Plane Cutting Layout". Cybernetics, Vol. 9, Number 6, pp: 1036-1043, November - December, 1973.
27. Haessler, R.W. - "An Heuristic Programming Solution to a Non-Linear Cutting-Stock Problem". Management Science, Vol. 17, B-793/B-802, 1971.
28. Haessler, R.W. - "Solving the Two-Stage Cutting-Stock Problem" OMEGA - The Int. JI. of Mgmt. Sci., Vol. 7, No 2, pp: 145-151, 1979.
29. Hahn, S.G. - "On the Optimal Cutting of Defectives Sheets". Operations Research, Vol. 16, pp: 1100-1114, 1968.
30. Hedetniemi, S.T. & Goodman, S.E. - "Introduction to the Design and Analysis of Algorithms". Mc Graw Hill Computer Science Series.
31. Herz, J.C. - "Recursive Computational Procedure for the Two-Dimensional Cutting-Stock". IBM Journal Research Development Vol. 16, pp: 462-469, 1972.
32. Hinxman, A.I. - "Problem Reduction and the Two - Dimensional Trim-loss Problem". Artificial Intelligence and Simulation (A.I.S.B.) Summer Conference, University of Edinburgh, pp: 158-165, 1976.

33. Horowitz, E. & Sahni, S. - "Computing Partitions with Applications to the Knapsack Problem". Journal of A.C.M., Vol. 21, 1974.
34. Hung, S.M. & Brown, J.R. - "An Algorithm for A Class of Loading Problems". Naval Research Logistics Quarterly, Vol. 25, No 2, pp: 281-297, June, 1978.
35. Kantorovich, L.V. - "Mathematical Methods of Organizing and Planning Production". Management Science, Vol. 6, pp: 366-422, 1960.
36. Kolesar, P.J. - "A Branch and Bound Algorithm for the Knapsack Problem". Management Science, Vol. 13, No 9, 1967.
37. Madsen, O.B.G. - "Glass Cutting in a Small Firm". Mathematical Programming, Vol. 17, pp: 85-90, 1979.
38. Marconi, R. - "Generalization of a Mathematical Procedure for the Optimal Solution of Two-Dimensional Cutting Problems". Centro Studi IBM of Pisa, Technical Report CSP 004/513-4001, April, 1970.
39. Nemhauser, G.L. & Ullmann, Z. - "Discrete Dynamic Programming and Capital Allocation". Management Science, Vol. 15, No 9, 1969.
40. Oldenburg, P.B. - "NP-Complete Operations Research Problems and Approximation Algorithms". Zeitschrift für Operations Research, Vol. 23, pp: 73-94, 1979.

41. Rocha, T.G. - "Aprovisionamento de Bobinas de Cabos Telefônicos: Uma Aplicação de Programação Combinatória". Tese (M.Sc.), COPPE/UFRJ, Novembro de 1978.
42. Salkin, H.M. - "Integer Programming". Addison-Wesley Publishing Company.
43. Salkin, H.M. & Kluyver, C.A. - "The Knapsack problem: A SURVEY". Naval Research Logistics Quarterly, Vol. 22, pp: 127-144, 1975.
44. Stainton, R.S. - "The Cutting-Stock Problem for the Stock-Holder of Steel Reinforcement Bars". Operational Research Quarterly, Vol. 28, pp: 139-149, 1977.
45. Toth, P. & Martello, S. - "Branch and Bound Algorithms for the Solution of General Unidimensional Knapsack Problem". Advances in Operations Research - Proceedings of EURO II - Stockholm, November 29 - December 1, 1976.
46. Toth, P. & Martello, S. - "An Upper Bound for the 0-1 Knapsack Problem and a Branch and Bound Algorithm". European Journal of Operational Research, Vol. 1, pp: 169-175, 1977.
47. Wilson, R.C. - "A Packaging Problem". Management Science, Vol. 12, Nº 4, December, 1965.