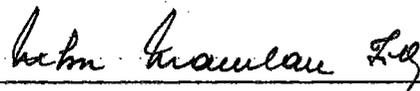


ALGUNS MODELOS PARA O PROBLEMA DE  
OTIMIZAÇÃO DE CORTES DE CHAPAS

OSWALDO DE SOUZA CAMPOS

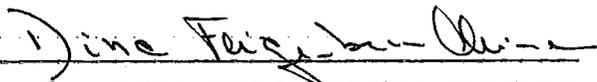
TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE  
PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE  
JANEIRO, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA OBTENÇÃO DO  
GRAU DE MESTRE EM CIÊNCIAS (M.Sc.)

Aprovada por:



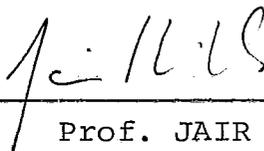
---

Prof. NELSON MACULAN FILHO  
(Presidente)



---

Profa. DINA FEIGEINBAUM CLEIMAN



---

Prof. JAIR KOILLER

RIO DE JANEIRO, RJ BRASIL

AGOSTO 1981

CAMPOS, OSWALDO DE SOUZA

ALGUNS MODELOS PARA O PROBLEMA DE OTIMIZAÇÃO DE  
CORTES DE CHAPAS (RIO DE JANEIRO) 1981.

, 61 p., 29,7 cm (COPPE-UFRJ, M.SC., Engenharia de Sistemas e  
Computação, 1981).

Tese - Universidade Federal do Rio de Janeiro. Faculdade de Engenharia.

1. Programação Matemática I. COOPE/UFRJ II. Título (série).

À minha esposa Dalva  
e filhos Herbert, Vi  
tor e Osvaldo.

AGRADECIMENTOS

Ao Prof. NELSON MACULAN FILHO que pela excelente orientação, apoio e solidariedade humana, possibilitou a conclusão deste trabalho.

A FUEL pelo auxílio financeiro e colaboração prestados para a realização deste trabalho.

A CAPES-PICD pela contribuição financeira fornecida durante a obtenção dos créditos e elaboração do trabalho.

Aos meus amigos, colegas, professores e funcionários da COPPE que direta ou indiretamente contribuíram para a execução deste trabalho.

A MARIA CRISTINA RUTZ DEBIAZIO, pelo excelente trabalho de datilografia.

Ao PAULO CESAR BERNARDI, pela execução dos desenhos.

## RESUMO

A finalidade deste trabalho foi apresentar uma colaboração didática sobre modelos de otimização de cortes de chapas.

Dentre tais modelos, fixamos mais a nossa atenção naqueles que utilizam Programação Linear. Além disso, incluimos alguns exemplos de aplicações.

ABSTRACT

The purpose of this work is to make a tutorial presentation an models of cutting stock problems otimization.

Among the model studied we concentrated our atention on the one that uses the Linear Programming. We have included, in addition, examples of applications.

ÍNDICE

CAPÍTULO I	- INTRODUÇÃO . . . . .	1
CAPÍTULO II	- O PROBLEMA DE CORTE EM UMA DIMENSÃO. . . . .	4
2.1	- Formulação do Problema . . . . .	4
2.2	- Outras Formulações do Problema de Corte. . . . .	9
CAPÍTULO III	- MÉTODOS DE SOLUÇÃO DO PROBLEMA DE CORTE E DO PROBLEMA KNAPSACK . . . . .	13
3.1	- Descrição do Algoritmo para Resolver o Problema de Corte . . . . .	13
3.2	- Um Método de Programação Dinâmica para Resolver o Problema Knapsack. . . . .	18
CAPÍTULO IV	- O PROBLEMA DE CORTE EM DUAS DIMENSÕES. . . . .	36
4.1	- Problema em Uma Dimensão . . . . .	36
4.2	- Problema Geral de Corte em Duas Dimensões. . . . .	38
4.3	- Corte de Guilhotina. . . . .	41
4.4	- Formulação do Problema de Corte de Guilhotina como (P.P.L.). . . . .	46
4.5	- Outros Problemas de Corte em Duas Dimensões. . . . .	49
CAPÍTULO V	- CONCLUSÃO. . . . .	55
5.1	- Outros Algoritmos para o Problema de Corte . . . . .	55
5.2	- Conclusão. . . . .	56

CAPÍTULO IINTRODUÇÃO

No problema de corte procura-se atender um pedido a custo mínimo de um número especificado de comprimento a ser cortado de materiais estocados cujos custos são conhecidos.

Na formulação como um problema de programação linear inteira, a grande número de variáveis geralmente envolvido torna o problema computacionalmente inviável.

Esta mesma dificuldade ainda persiste mesmo quando o problema é tratado como um problema de programação linear.

Neste trabalho formulamos o problema de corte como um problema de programação linear e descrevemos métodos para superar esta dificuldade.

A dificuldade surgida com o grande número de colunas nos problemas de corte, pode ser superada utilizando o método que será descrito neste trabalho.

O método consiste basicamente no seguinte: quando no método simplex revisado buscamos a coluna que entrando na base melhora a solução, o fazemos resolvendo um problema auxiliar, que é um problema Knapsack, ver referência (1).

Este problema Knapsack pode ser resolvido por vãrios métodos ver referência (3), nós descreveremos um método de programação dinâmica baseado na referência (4).

O problema de corte em duas dimensões considera um estoque de retângulos de largura  $W$  e comprimento  $L$ , indicado  $W \times L$ , e uma demanda de  $N_i$  retângulos de largura  $w_i$  e comprimento  $l_i$ ,  $i = 1, \dots, m$ , então o problema é cortar o mínimo de retângulos de estoque nos retângulos menores pedidos.

Trataremos o problema como um problema de programação linear. A dificuldade do grande número de colunas persiste aqui, não podendo em geral ser superada pois não há método eficiente para resolver o problema Knapsack generalizado que surge na solução do problema de corte de duas ou mais dimensões.

Estes problemas aparecem na produção industrial do vidro, chapas metálicas, grafite, filmes fotográficos, plásticos, etc.

Algumas considerações sobre o problema de corte quando os retângulos menores são ajustados de maneira arbitrária no retângulo  $W \times L$ , são feitas na referência (9).

Em muitas situações industriais o problema que aparece é o de cortar o material de uma extremidade a outra, cortando-se novamente as peças menores obtidas, sendo permitido ainda um terceiro corte conseguindo-se então os retângulos menores pedidos, resultando problemas ainda complexos mas tratáveis.

No capítulo II enunciamos o problema de corte em uma dimensão, formulamos como um problema de programação linear inteira, enumeramos as dificuldades resultantes desta formulação. Depois formulamos o problema como um problema de programação linear

(PPL) e apresentamos um método de solução. Em seguida apresentamos outras formulações do mesmo problema.

No capítulo III descrevemos um algoritmo para re solver o problema de corte em uma dimensão que pode ser usado tam bém na solução do problema em duas dimensões. Descrevemos também um algoritmo de programação dinâmica para o problema Knapsack. A presentamos em seguida dois exemplos ilustrativos.

Iniciamos o capítulo IV revisando o problema em uma dimensão, usando para gerar colunas o problema Knapsack visto em III.

Emitimos a formulação usual do problema de corte em duas dimensões. Formulamos o mesmo como um problema de programa ção linear e observamos que o método de geração de coluna, conduz a problemas Knapsack generalizados para os quais nenhum método eco nômico de solução é conhecido. Algumas classes especiais de proble mas em duas dimensões são solúveis.

Vemos que para gerar colunas no problema de corte em duas dimensões, podemos usar o método de decomposição de Dantzig e Wolfe (5).

Analizamos em seguida problema de corte de guilho tina em dois estágios.

No capítulo V concluimos analisando outros algo ritmos usados na solução dos problemas de corte, ver referência (9).

CAPÍTULO II

O PROBLEMA DE CORTE EM UMA DIMENSÃO

Suponhamos um estoque de chapas de comprimentos  $L_1, L_2, \dots, L_r$ , todas com a mesma largura, sem limitação no número de peças disponíveis de cada comprimento. O problema será cortálas de maneira que o pedido de pelo menos  $N_i$  peças de comprimento  $\ell_i$  ( $i = 1, 2, \dots, m$ ) seja atendido, minimizando o custo total de chapas cortadas. Portanto, se houver perda em cada chapa cortada, estaremos minimizando a perda total.

2.1 - Formulação do Problema

O problema de corte será da forma:

$$\text{minimizar } z = \sum_j c_j x_j$$

sujeito a:

$$\sum_j a_{ij} x_j \geq N_i, \quad (i = 1, 2, \dots, m)$$

$$x_j \geq 0, \text{ e inteiro para } \forall_j$$

onde,

$x_j$  = número de vezes que o  $j$ -ésimo modelo de corte é usado para cortar uma chapa, ( $j = 1, 2, \dots$ )

$c_j$  = custo da chapa cortada pelo  $j$ -ésimo modelo de corte, ( $j = 1, 2, \dots$ )

$a_{ij}$  = número de peças de comprimento  $\ell_i$  produzido cada vez que o

$j$ -ésimo modelo de corte é usado ( $i = 1, 2, \dots, m$ ) e ( $j = 1, 2, \dots$ )

$N_i$  = número pedido de peças de comprimento  $l_i$ .

Definimos o  $j$ -ésimo modelo de corte como sendo  $a_{ij}$  peças de comprimento  $l_i$ , produzidas de  $L_k$ .

Um pedido pode ser atendido enquanto  $l_i$  não exceder o  $\max_k \{L_k \mid k = 1, 2, \dots, r\}$ .

Consideremos o caso em que os cortes são feitos segundo linhas verticais, isto é, cada chapa deverá ser cortada em comprimentos  $l_i$  ( $i = 1, 2, \dots, m$ ).

Um modelo de corte é ilustrado na Figura II-1, onde são produzidas duas peças de comprimento  $l_1$ , três peças de comprimento  $l_2$ , uma peça de comprimento  $l_3$  e a parte hachuriada é a perda.

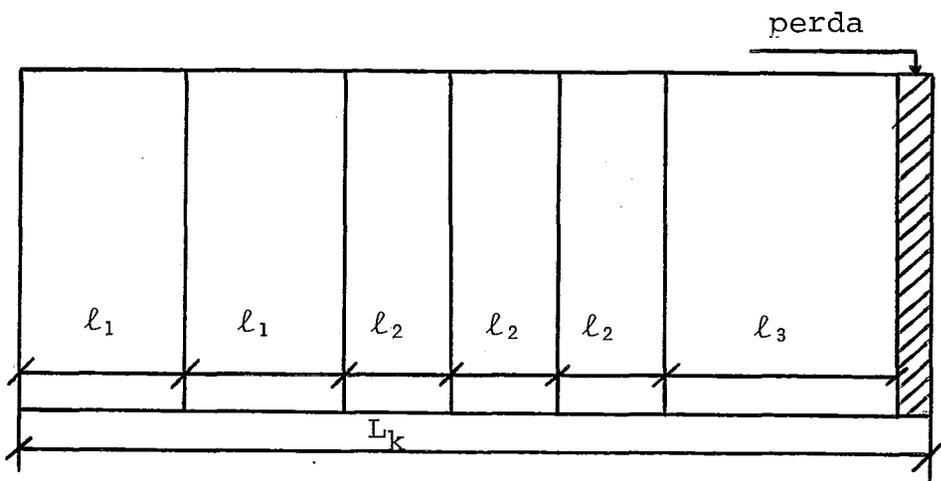


Fig. II-1

No problema formulado surgem duas dificuldades:

- i) os  $x_j$  são inteiros e o grande número de variáveis torna o problema computacionalmente intratável.
- ii) impossibilidade de enumerar antecipadamente todos os métodos de corte.

P.C. Gilmore e R.E. Gomory (1) superaram estas dificuldades da seguinte maneira:

- i) relaxaram a condição de  $x_j$  ser inteiro, resultando um Problema de Programação Linear (PPL) e arredondaram a solução ótima de maneira apropriada.
- ii) ao invés de enumerar todos os modelos de corte antecipadamente, um modelo é considerado somente se melhorar a solução do (PPL).

O procedimento utilizado por P.C. Gilmore e R.E. Gomory (1) é um algoritmo do Simplex Revisado, no qual a coluna que entra na base é encontrada resolvendo ou mais problemas Knapsack.

### 2.1.1 - Solução do Problema de Corte

Consideremos o (PPL)

$$\text{minimizar } z = \sum_{j=1}^n C_j x_j \quad (1)$$

sujeito a:

$$\sum_{j=1}^n a_{ij} x_j \geq N_i, (i=1,2,\dots,m) \text{ e } (j=1,2,\dots,n) \quad (2)$$

$$x_j \geq 0, \text{ para } \forall_j \quad (3)$$

Introduzindo as variáveis de folga,  $x_{n+1}, \dots, x_{n+m}$ , não negativas, (2) poderá ser substituído por:

$$\sum_j a_{ij} x_j - x_{n+i} = N_i, \quad (i = 1, 2, \dots, m)$$

$$(j = 1, 2, \dots, n+m)$$

Numa iteração qualquer do Método Simplex, considere mos:

$B = (P_1, \dots, P_i, \dots, P_m)$  uma base, onde,  $P_i$  é um vetor coluna de  $m$  linhas, ( $i = 1, 2, \dots, m$ ).

$C_B = (C_1, C_2, \dots, C_m)$ , coeficientes da função objetivo associados aos vetores  $P_1, P_2, \dots, P_m$ .

Da teoria da Programação Linear, um modelo de Corte  $j$  (ou coluna), entrará na base se seu custo reduzido,

$$Z_j - C_j = C_B B^{-1} P_j - C_j$$

for positivo (problema de minimização), onde

$$P_j = (a_{1j}, a_{2j}, \dots, a_{mj})^T$$

é o vetor que representa o número de peças de comprimento  $l_i$  ( $i = 1, 2, \dots, m$ ), gerado pelo modelo  $j$ . Neste ponto, os elementos de  $P_j$  não são conhecidos, isto é, o novo modelo de corte não foi ainda determinado. A coluna que entrará na base será aquela que fornece o maior valor de  $Z_j - C_j$ , dentre todos os possíveis modelos não básicos. Isto é equivalente a resolver o

## Problema Knapsack

$$\text{maximizar } v = \sum_{i=1}^m W_i a_{ij}$$

sujeito a:

$$\sum_{i=1}^m \ell_i a_{ij} \leq L_k$$

$$a_{ij} \geq 0 \text{ e inteiro } (i = 1, 2, \dots, m)$$

onde,

$$W_i = i\text{-ésimo elemento de } C_B B^{-1}$$

O problema será resolvido para  $L_1, L_2, \dots, L_r$ , onde  $r$  é o número total de chapas disponíveis.

Seja  $p^k$  a solução ótima de cada problema e

$$W = C_B B^{-1} = (W_1, W_2, \dots, W_m). \text{ Então se,}$$

$$WP_j^k - C_k \leq 0 \text{ para cada } k, \quad (4)$$

a solução básica corrente,

$$x_B = B^{-1}N$$

é ótima, onde  $N = (N_1, N_2, \dots, N_m)$ . Caso contrário, a coluna que produz o maior valor positivo de  $WP_j^k - C_k$ , será introduzida na base.  $C_k$  é o custo da chapa de comprimento  $L_k$ , portanto é o custo da coluna  $P_j^k$ .

A solução dos  $r$  problemas Knapsack para gerar um modelo de corte pode ser muito trabalhosa do ponto de vista compu

tacional, então será utilizado o primeiro modelo que produz

$$WP_j^k - C_k > 0$$

A coluna que deverá deixar a base é escolhida após calcular  $B^{-1}P_j^k$ , segundo o pivoteamento usual. Uma nova coluna não básica (modelo de corte) será encontrada resolvendo problemas Knapsack e o (PPL) estará resolvido quando (4) for satisfeita.

A eficiência do algoritmo depende de quantos problemas Knapsack deverão ser resolvidos. Se existir um único comprimento, somente um problema será resolvido em cada iteração. Uma solução básica inicial pode ser obtida pelo método das duas fases.

## 2.2 - Outras Formulações do Problema de Corte

Lasdon (5) formula o problema considerando os custos unitários

$$\text{minimizar } Z = \sum_j x_j$$

sujeito a:

$$\sum_j a_{ij}x_j \geq N_i \quad (i = 1, 2, \dots, m)$$

$$x_j \geq 0 \quad \text{e inteiro}$$

O coeficiente de custo da variável  $x_j$  não-básica será:

$$\sum_{i=1}^m W_i a_{ij} - 1$$

onde,  $W_i$  são os multiplicadores de Lagrange da restrição (2).

TAHA (6), formula o problema de corte considerando  $C_j$ , como perda associada a cada modelo de corte minimizando a perda total. Em condições ótimas, um modelo  $j$  cortará uma peça  $L_k$  de modo a produzir menor perda possível e  $C_j$  deve ser estritamente menor do que  $\min_i \{\ell_i\}$  ( $i = 1, 2, \dots, m$ ).

Portanto o problema se tornará:

$$\text{minimizar } Z = \sum_j C_j x_j$$

sujeito a:

$$\sum_j a_{ij} x_j \geq N_i \quad (i = 1, 2, \dots, m)$$

$$x_j \geq 0 \quad \text{e inteiro para } \forall_j$$

onde,

$$C_j = L_k - \sum_{i=1}^m \ell_i a_{ij} \geq 0$$

É possível que  $a_{ij} = 0$ , para algum  $i = 1, 2, \dots, m$ .

### 2.2.1 - Exemplo Ilustrativo

Seja uma máquina que corta rolos de papel de 60 polegadas de largura. Desejamos cortá-los de maneira a atender um pedido de 30 rolos de 28 polegadas, 60 rolos de 20 polegadas e 48 rolos de 15 polegadas de largura.

Consideremos o quadro,

Larguras Pedidas em Polegadas.	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	Número de rolos pedidos
28	2	1	1	0	0	0	0	30
20	0	1	0	3	2	1	0	60
15	0	0	2	0	1	2	4	48
PERDAS	4	12	2	0	5	10	0	

A variável  $x_3$  representa o número de vezes que o rolo de 60 polegadas é cortado em um rolo de 28 polegadas e dois de 15 polegadas com uma perda de 2 polegadas. Logo o problema será:

$$\text{minimizar } 4x_1 + 12x_2 + 2x_3 + 5x_5 + 10x_6$$

sujeito a:

$$2x_1 + x_2 + x_3 \geq 30$$

$$x_2 + 3x_4 + 2x_5 + x_6 \geq 60$$

$$2x_3 + x_5 + 2x_6 + 4x_7 \geq 48$$

$$x_1, x_2, x_3, x_4, x_5, x_6, x_7 \geq 0$$

Uma solução ótima obtida por inspeção será:

$x_1 = 15$ ,  $x_4 = 20$  e  $x_7 = 12$ , com uma perda total de 60 polegadas.

As variáveis de folga serão iguais a zero, o que indica que não há superprodução. Esta solução significa que cortamos 15 rolos de 60

polegadas, cada um, em 2 rolos de 28 polegadas e 1 de 4 polegadas.  
Outra solução ótima será  $x_1 = 3$ ,  $x_3 = 24$  e  $x_4 = 20$  com perda to  
tal igual a 60.

CAPÍTULO IIIMÉTODOS DE SOLUÇÃO DO PROBLEMA DE CORTE E DO PROBLEMA KNAPSACK3.1 - Descrição do Algoritmo para Resolver o Problema de Corte

Consideremos o problema de corte,

$$\text{minimizar} \quad c_1x_1 + c_2x_2 + \dots + c_nx_n \quad (1)$$

sujeito a:

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n - x_{n+i} = N_i \quad (2)$$

$$(i = 1, 2, \dots, m)$$

$$x_j \geq 0 \quad \text{para } (j = 1, 2, \dots, n+m) \quad (3)$$

Sejam  $x_1, x_2, \dots, x_m$  uma solução básica viável,  $P_i = (a_{i1}, a_{i2}, \dots, a_{mi})$ ,  $c_i$  o vetor e o custo, associados a variável  $x_i$ . Se  $x_i$  for uma variável de folga, então  $c_i = 0$ , e o vetor terá a coordenada não nula  $-1$ . Consideremos  $P = (a_1, a_2, \dots, a_m)$  um novo modelo de corte de  $L$ , com custo  $c$  e seja  $A$ , uma matriz cujas colunas são  $P_1, P_2, \dots, P_m$ . Como estas colunas são linearmente independentes, existe um vetor coluna  $U$  satisfazendo a equação

$$AU = P \quad (4)$$

e o nosso modelo poderá ser utilizado em uma solução melhor que a anterior se, e somente se:

$$CU > c \quad (5)$$

onde  $C = (c_1, c_2, \dots, c_m)$ . Fazendo

$$CA^{-1} = (w_1, w_2, \dots, w_m)$$

podemos concluir de (4) e (5) que um modelo de corte L será mais vantajoso se, e somente se, existirem inteiros não-negativos  $a_1, a_2, \dots, a_m$  satisfazendo

$$l_1 a_1 + l_2 a_2 + \dots + l_m a_m \tag{6}$$

e

$$w_1 a_1 + w_2 a_2 + \dots + w_m a_m > c \tag{7}$$

Uma maneira de verificar se existem inteiros não-negativos, satisfazendo (6) e (7) será resolver o seguinte problema Knapsack

maximizar  $w_1 a_1 + w_2 a_2 + \dots + w_m a_m$

sujeito a:

$$l_1 a_1 + l_2 a_2 + \dots + l_m a_m \leq L$$

A seguir descreveremos o algoritmo conforme (1).

1) Determinar  $m$  modelos e seus custos como segue:

i) para cada  $i$ , escolher um  $L_k$  tal que,  $L_k > l_i$

ii) definir o  $i$ -ésimo modelo de corte como sendo o que fornece

$$a_{ii} = \left[ L_k / l_i \right]$$

peças de comprimento  $\ell_i$  de  $L_k$ . O custo do  $i$ -ésimo modelo será  $c_k$  de  $L_k$ . Sendo  $[x]$  maior inteiro  $\leq x$ .

2) Construir a matriz B, de ordem  $m + 1$

$$\begin{array}{ccccc} 1 & -c_1 & -c_2 & \dots & -c_m \\ 0 & a_{11} & 0 & & 0 \\ 0 & 0 & a_{22} & & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & a_{mm} \end{array}$$

onde,  $a_{ii}$  é o número de peças de comprimento  $\ell_i$  no modelo  $i$ , cujo custo é  $c_i$ .

- Marcar os modelos correspondentes as  $m$  últimas colunas de B.
- Formar os vetores coluna  $S_1, S_2, \dots, S_m$  de dimensão  $m + 1$ , correspondentes as variáveis de folga. Estes vetores possuem  $-1$  na  $(i + 1)$ -ésima linha e zero nos demais.
- Formar o vetor coluna  $N'$  de dimensão  $m + 1$  com zero na primeira linha e  $N_i$  na  $i$ -ésima linha.
- Calcular  $B^{-1}$

$$B^{-1} = \begin{vmatrix} 1 & c_1/a_{11} & c_2/a_{22} & \dots & c_m/a_{mm} \\ 0 & 1/a_{11} & 0 & \dots & 0 \\ 0 & 0 & 1/a_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 1/a_{mm} \end{vmatrix}$$

- Fazer  $N = B^{-1} N'$

Conhecidos  $B^{-1}$  e o vetor coluna  $P$  (novo modelo cuja primeira linha é menos o custo, e as demais  $m$  linhas  $a_{ij}$ ; no caso da  $i$ -ésima variável de folga  $P$  e  $S_i$ ), para verificar se a presente solução poderá ser melhorada fazendo uso de  $P$ , será necessário calcular somente o primeiro elemento de  $B^{-1}P$ . Se fôr positivo a solução melhorará, caso contrário, não.

- 3) Uma variável de folga que não aparece na presente solução, fornecerá uma solução melhor se, e somente se, o  $(i + 1)$ -ésimo elemento da primeira linha de  $B^{-1}$  for negativo.
- 4) Se nenhuma variável de folga melhorar a solução corrente será necessário verificar se a introdução de um novo modelo melhora a solução, determinando se existe um  $L$  com custo  $c$  para o qual as desigualdades (6) e (7) tem solução, onde  $w_1, w_2, \dots, w_m$  são os últimos  $m$  elementos da primeira linha de  $B^{-1}$ .

- Se as desigualdades não tiverem solução para  $L_1, L_2, \dots, L_k$  com custos respectivamente  $c_1, \dots, c_k$ , então a solução atual é mínima.
  - A solução atual e seu custo é obtido pelo  $N$  atual, onde a primeira linha deste vetor, é o custo e as  $m$  linhas restantes são em ordem os valores das variáveis correspondentes as  $m$  colunas de  $B^{-1}$ .
  - Se o novo modelo melhorar a solução corrente, formar o seu vetor coluna  $P = (-c, a_1, \dots, a_m)$ .
- 5) A introdução de uma variável de folga ou novo modelo melhora a solução atual, faça  $P$  o vetor coluna da variável.

Para determinar os novos  $B^{-1}$  e  $N$ , que fornecem uma solução melhor e os respectivos custos e que permita realizar 3), 4) e 5), procede-se da seguinte forma:

- Calcular  $B^{-1}P$ , sejam  $y_1, \dots, y_m, y_{m+1}$  os elementos deste vetor e  $x_1, x_2, \dots, x_m, x_{m+1}$  os elementos de  $N$ .
- Determinar  $i, i \geq 2$  para o qual  $y_i > 0, x_i \geq 0$  e  $x_i/y_i$  é mínimo e seja  $i = k$ .
- Se a razão for zero, teremos degeneração e trataremos da maneira usual. Se for diferente de zero, o  $k$ -ésimo elemento de  $P$ , isto é,  $y_k$  será o elemento pivô.
- O pivoteamento será feito na matriz  $G_{(m+1) \times (m+3)}$  formada de  $B^{-1}$ ,

acrescentando as novas colunas  $N$  e  $B^{-1}P$  respectivamente. As primeiras  $(m + 1)$  colunas de  $G'$  formarão a nova  $B^{-1}$  e a coluna  $(m + 2)$ -ésima de  $G'$ , formará a nova  $N$ .

Para evitar ciclos acrescentaremos a  $G$  uma nova coluna  $N^1$  de elementos  $x'_1, \dots, x'_m$ , independente de  $N$ , e o novo elemento pivô será o  $\min x'_i/y_i$  com  $x'_i > 0$  e procede-se o pivoteamento em  $G$ . A coluna adicional será mantida em  $G$  até encontrar  $i$  para o qual  $x'_i/y_i$  é positivo e finito, depois poderá ser abandonada. Se não atingirmos essa situação, com o acréscimo de  $N^1$ , adicionaremos uma coluna  $N^2$  de elementos positivos independentes de  $N^1$  e  $N$  que deverá ser usada no pivoteamento até que  $N$  ou  $N^1$  possa ser usada. Desde que tais colunas permanecem independentes após o pivoteamento, não necessitamos introduzir mais do que  $m$  de tais colunas. Cada coluna introduzida define um novo (PPL) prevenindo ciclos, caso não ocorra degenerência (21).

### 3.2 - Um Método de Programação Dinâmica para Resolver o Problema Knapsack

Uma das maiores dificuldades na solução de um problema de corte como um (PPL), é o número muito grande de colunas no problema. Esta dificuldade pode ser superada resolvendo um problema Knapsack em cada passo do pivoteamento simplex.

Sejam  $w_1, w_2, \dots, w_m$  os multiplicadores de Lagrange ou "preços sombra", associados a uma solução básica do (PPL) definido por (1), (2) e (3), então teremos o seguinte problema Knapsack.

maximizar  $w_1 a_1 + w_2 a_2 + \dots + w_m a_m$

sujeito a:

$$l_1 a_1 + l_2 a_2 + \dots + l_m a_m \leq L.$$

Nas referências (1) e (2), P.C. Gilmore e R.E. Gomory, descreveram dois métodos de recorrência para a solução do problema acima. Na referência (3), os mesmos autores desenvolveram um método de programação dinâmica mais eficiente do que os anteriores. O método consiste em definir,

$$F_s(x) = \max \sum_{i=1}^s w_i a_i$$

sujeito a:

$$\sum_{i=1}^s l_i a_i \leq x, \quad 1 \leq s \leq m, \quad 0 \leq x \leq L$$

e usar a recorrência

$$F_s(x) = \max \{w_s + F_s(x - l_s), F_{s-1}(x)\} \quad (8)$$

para  $s > 1$

Isto segue do fato de que no cálculo do valor de  $F_s(x)$  para  $s > 1$ , usamos ou não, a  $s$ -ésima variável. Se for usada então:

$$F_s(x) = w_s + F_s(x - l_s), \text{ se não, } F_s(x) = F_{s-1}(x).$$

Fazendo  $F_1(x) = w_1 \left[ \frac{x}{l_1} \right]$ ,  $F_s(0) = 0$  para todo  $s$ , calculamos  $F_s(x)$  em termos de  $F_{s-1}(x)$  e  $F_s(x')$  para  $x' < x$ , isto é, usando funções já conhecidas. Então,  $F_m(L)$  fornece o melhor va

lor para o problema Knapsack.

Agora teremos que achar um modelo que forneça o valor ótimo, procedendo da seguinte maneira: associamos a  $F_1(x)$  o índice 1 se  $x \geq l_1$  e zero, caso contrário. Como  $F_s(x)$  para  $s > 1$  é calculado para cada  $x$ , associamos a  $F_s(x)$  o índice  $s$ , se  $F_s(x) = w_s + F_{s-1}(x - l_s)$  e caso contrário, o índice associado com  $F_{s-1}(x)$ .

Como consequência, o índice associado com  $F_s(x)$  para qualquer  $s$  e  $x$ , será o maior índice  $r$  para o qual  $a_r$  é positivo no modelo que fornece  $F_s(x)$ . Portanto, para achar o modelo  $F_m(L)$ , examinamos inicialmente o índice associado. Se este for  $r > 0$ , então  $a_r$  está no modelo pelo menos uma vez. Consideramos em seguida  $F_m(L - l_r)$ , se o índice associado for  $r' > 0$ , então a peça de comprimento  $l_{r'}$ , foi usada em  $F_m(L - l_r)$  e portanto em  $F_m(L)$ . Continuando deste modo, obteremos todos os modelos.

No cálculo de (8), usaremos somente  $F_{s-1}(x)$  para calcular  $F_s(x)$ , e no retorno necessitaremos somente de  $F_m(x)$ . Logo todos os valores de  $F_{s-2}(x)$  poderão ser abandonados antes de iniciar o cálculo de  $F_s(x)$ .

Este método requer menos cálculo aritmético do que os métodos das referências (1) e (2) por um fator aproximado  $\alpha$ , onde

$$\alpha = \frac{1}{m} \left( \sum_i \left[ \frac{L}{l_i} \right] \right)$$

Este método poderá ser comparado com o método léxico gráfico descrito com "Knapsack Method" na referência (2), quanto a rapidez de execução. Contudo ao contrário dos métodos léxico gráfico não pareça ser adaptável aos problemas, onde o número de facas de corte é limitado.

### 3.2.1 - Exemplo de Aplicação do Método

Um pedido de 20 peças de comprimento 2, 10 peças de comprimento 3 e 20 peças de comprimento 4 deverá ser atendido cortando chapas de comprimento 5, 6 e 9 com custos respectivamente 6, 7 e 10.

#### Solução

Dados:	$N_1 = 20$	$l_1 = 2$
	$N_2 = 10$	$l_2 = 3$
	$N_3 = 20$	$l_3 = 4$

Tomando  $L = 5$ , teremos:

$$a_{11} = \left[ \frac{5}{2} \right] = 2, \quad a_{22} = \left[ \frac{5}{3} \right] = 1, \quad e \quad a_{33} = \left[ \frac{5}{4} \right] = 1$$

$$c_1 = 6 \quad c_2 = 6 \quad e \quad c_3 = 6$$

$$B = \begin{vmatrix} 1 & -6 & -6 & -6 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad N' = \begin{vmatrix} 0 \\ 20 \\ 10 \\ 20 \end{vmatrix}$$

$$B^{-1} = \begin{vmatrix} 1 & 3 & 6 & 6 \\ 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad N = B^{-1}N' = \begin{vmatrix} 240 \\ 10 \\ 10 \\ 20 \end{vmatrix}$$

Logo,  $w_1 = 3$ ,  $w_2 = 6$  e  $w_3 = 6$ .

Tomando a chapa de comprimento  $L_3 = 9$  e custo  $c = 10$ , podemos definir um maior número de modelos de corte. Determinaremos um novo modelo resolvendo o problema Knapsack,

$$\text{maximizar } w = 3a_1 + 6a_2 + 6a_3 \quad (9)$$

sujeito a:

$$2a_1 + 3a_2 + 4a_3 \leq 9 \quad (10)$$

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

usando o método 3.2, teremos:

x s	1	2	3	4	5	6	7	8	9
1	0	3	3	6	6	9	9	12	12
2	0	3	6	6	9	12	12	15	18
3	0	3	6	6	9	12	12	15	18

Tabela 1

$$F_s(0) = 0 \text{ para todo } s$$

$$F_1(x) = w_1 \left[ \frac{x}{\ell_1} \right]$$

$$F_1(1) = 3 \left[ \frac{1}{2} \right] = 0$$

$$F_1(2) = 3 \left[ \frac{2}{2} \right] = 3$$

$$F_1(3) = 3 \left[ \frac{3}{2} \right] = 3$$

$$F_1(4) = 3 \left[ \frac{4}{2} \right] = 6$$

$$F_1(5) = 3 \left[ \frac{5}{2} \right] = 6$$

$$F_1(6) = 3 \left[ \frac{6}{2} \right] = 9$$

$$F_1(7) = 3 \left[ \frac{7}{2} \right] = 9$$

$$F_1(8) = 3 \left[ \frac{8}{2} \right] = 12$$

$$F_1(9) = 3 \left[ \frac{9}{2} \right] = 12$$

$$F_s(x) = \max \{ w_s + F_s(x - \ell_r), F_{s-1}(x) \} \quad \text{para } s > 1.$$

$$F_2(x) = \text{m}\bar{\text{a}}\text{x} \{w_2 + F_2(x - \ell_2), F_1(x)\}$$

$$F_2(1) = \text{m}\bar{\text{a}}\text{x} \{6 + F_2(1 - 3), F_1(1)\} = F_1(1) = 0$$

$$F_2(2) = \text{m}\bar{\text{a}}\text{x} \{6 + F_2(2 - 3), F_1(2)\} = F_1(2) = 3$$

$$F_2(3) = \text{m}\bar{\text{a}}\text{x} \{6 + F_2(3 - 3), F_1(3)\} = 6$$

$$F_2(4) = \text{m}\bar{\text{a}}\text{x} \{6 + F_2(4 - 3), F_1(4)\} = 6$$

$$F_2(5) = \text{m}\bar{\text{a}}\text{x} \{6 + F_2(5 - 3), F_1(5)\} = 9$$

$$F_2(6) = \text{m}\bar{\text{a}}\text{x} \{6 + F_2(6 - 3), F_1(6)\} = 12$$

$$F_2(7) = \text{m}\bar{\text{a}}\text{x} \{6 + F_2(7 - 3), F_1(7)\} = 12$$

$$F_2(8) = \text{m}\bar{\text{a}}\text{x} \{6 + F_2(8 - 3), F_1(8)\} = 15$$

$$F_2(9) = \text{m}\bar{\text{a}}\text{x} \{6 + F_2(9 - 3), F_1(9)\} = 18$$

$$F_3(x) = \text{m}\bar{\text{a}}\text{x} \{w_3 + F_3(x - \ell_3), F_2(x)\}$$

$$F_3(x) = \text{m}\bar{\text{a}}\text{x} \{6 + F_3(x - 4), F_2(x)\}$$

$$F_3(1) = F_2(1) = 0$$

$$F_3(2) = F_2(2) = 3$$

$$F_3(3) = F_2(3) = 6$$

$$F_3(4) = \text{m}\bar{\text{a}}\text{x} \{6 + F_3(4 - 4), F_2(4)\} = \text{m}\bar{\text{a}}\text{x} \{6 + 0, 6\} = 6$$

$$F_3(5) = \text{m}\bar{\text{a}}\text{x} \{6 + F_3(5 - 4), F_2(5)\} = \text{m}\bar{\text{a}}\text{x} \{6 + 0, 9\} = 9$$

$$F_3(6) = \text{m}\bar{\text{a}}\text{x} \{6 + F_3(6 - 4), F_2(6)\} = \text{m}\bar{\text{a}}\text{x} \{6 + 3, 12\} = 12$$

$$F_3(7) = \text{m}\bar{\text{a}}\text{x} \{6 + F_3(7 - 4), F_2(7)\} = \text{m}\bar{\text{a}}\text{x} \{6 + 6, 12\} = 12$$

$$F_3(8) = \text{m}\bar{\text{a}}\text{x} \{6 + F_3(8 - 4), F_2(8)\} = \text{m}\bar{\text{a}}\text{x} \{6 + 6, 15\} = 15$$

$$F_3(9) = \text{m}\bar{\text{a}}\text{x} \{6 + F_3(9 - 4), F_2(9)\} = \text{m}\bar{\text{a}}\text{x} \{6 + 9, 18\} = 18$$

O valor ótimo será  $F_3(9) = 18$ .

Para determinar os valores de  $a_1$ ,  $a_2$  e  $a_3$ , construiremos a Tabela 2 de acordo com o método de 3.2.

$x \backslash s$	1	2	3	4	5	6	7	8	9
1	0	1	1	1	1	1	1	1	1
2	0	2	2	2	2	2	2	2	2
3	0	1	2	2	2	2	2	2	2

Tabela 2

Temos então o valor 2 associado com  $F_3(9)$ , o que significa que a segunda peça  $l_2 = 3$  será usada pelo menos uma vez. Associado com  $F_3(9 - 3) = F_3(6)$  temos 2 e novamente a segunda peça será usada pelo menos uma vez em  $F_3(6)$ . Associado com  $F_3(6 - 3) = F_3(3)$  temos 2, portanto, usaremos novamente a peça  $l_2$ . Associado a  $F_3(3 - 3) = F_3(0)$  temos zero, portanto:

$$a_1 = 0, \quad a_2 = 3, \quad a_3 = 0 \quad \text{que satisfaz}$$

$$3a_1 + 6a_2 + 6a_3 = 3 \cdot 0 + 6 \cdot 3 + 6 \cdot 0 = 18 > 10.$$

Assim,

$$G = \left| \begin{array}{cccccc} 1 & 3 & 6 & 6 & 240 & 8 \\ 0 & 1/2 & 0 & 0 & 10 & 0 \\ 0 & 0 & 1 & 0 & 10 & \textcircled{3} \\ 0 & 0 & 0 & 1 & 20 & 0 \end{array} \right|$$

e após o pivoteamento, teremos:

$$G' = \left| \begin{array}{cccccc} 1 & 3 & 10/3 & 6 & 640/3 & 0 \\ 0 & 1/2 & 0 & 0 & 10 & 0 \\ 0 & 0 & 1/3 & 0 & 10/3 & 1 \\ 0 & 0 & 0 & 1 & 20 & 0 \end{array} \right|$$

Como  $w_1 = 3$ ,  $w_2 = \frac{10}{3}$  e  $w_3 = 6$  são não-negativos, a introdução de uma variável de folga não melhorará a solução. Teremos então um novo problema Knapsack

$$\text{maximizar } 3a_1 + \frac{10}{3} a_2 + 6a_3$$

sujeito a:

$$2a_1 + 3a_2 + 4a_3 \leq 9.$$

Usando o método de 3.2, obteremos:

$$a_1 = 4, \quad a_2 = 0 \quad \text{e} \quad a_3 = 0, \quad \text{logo } P = (-10, 4, 0, 0)$$

$$B^{-1}P = \begin{pmatrix} 2 \\ 2 \\ 0 \\ 0 \end{pmatrix} \quad \text{e} \quad G = \begin{pmatrix} 1 & 3 & 10/3 & 6 & 640/3 & 2 \\ 0 & 1/2 & 0 & 0 & 10 & \textcircled{2} \\ 0 & 0 & 1/3 & 0 & 10/3 & 0 \\ 0 & 0 & 0 & 1 & 20 & 0 \end{pmatrix}$$

$$G' = \begin{pmatrix} 1 & 5/2 & 10/3 & 6 & 610/3 & 0 \\ 0 & 1/4 & 0 & 0 & 5 & 1 \\ 0 & 0 & 1/3 & 0 & 10/3 & 0 \\ 0 & 0 & 0 & 1 & 20 & 0 \end{pmatrix}$$

Novamente a introdução de uma variável de folga não melhora a solução. Resolvemos então o seguinte problema Knap sack.

$$\text{minimizar} \quad \frac{5}{2} a_1 + \frac{10}{3} a_2 + 6a_3$$

sujeito a:

$$2a_1 + 3a_2 + 4a_3 \leq 9$$

Obteremos a solução  $a_1 = 0$ ,  $a_2 = 0$  e  $a_3 = 2$ ,  $P = (-10, 0, 0, 2)$

$$B^{-1}P = \begin{pmatrix} 2 \\ 0 \\ 0 \\ 2 \end{pmatrix}, G = \begin{pmatrix} 1 & 5/2 & 10/3 & 6 & 610/3 & 2 \\ 0 & 1/4 & 0 & 0 & 5 & 0 \\ 0 & 0 & 1/3 & 0 & 10/3 & 0 \\ 0 & 0 & 0 & 1 & 20 & \textcircled{2} \end{pmatrix}$$

e

$$G' = \begin{pmatrix} 1 & 5/2 & 10/3 & 5 & 550/3 & 0 \\ 0 & 1/4 & 0 & 0 & 5 & 0 \\ 0 & 0 & 1/3 & 0 & 10/3 & 0 \\ 0 & 0 & 0 & 1/2 & 10 & 1 \end{pmatrix}$$

A introdução da variável de folga não melhora a solução. Poderemos experimentar  $L_2 = 6$  e teremos o seguinte problema Knapsack.

$$\text{maximizar } \frac{5}{2} a_1 + \frac{10}{3} a_2 + 5a_3$$

sujeito a:

$$2a_1 + 3a_2 + 4a_3 \leq 6$$

que terá como solução,  $a_1 = 3$ ,  $a_2 = 0$  e  $a_3 = 0$ , logo  $P = (-7, 3, 0, 0)$  e

$$B^{-1}P = \begin{array}{c} \left| \begin{array}{cccc} 1 & 5/2 & 10/3 & 5 \\ 0 & 1/4 & 0 & 0 \\ 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 1/2 \end{array} \right| \left| \begin{array}{c} -7 \\ 3 \\ 0 \\ 0 \end{array} \right| = \left| \begin{array}{c} 1/2 \\ 3/4 \\ 0 \\ 0 \end{array} \right| \end{array}$$

$$G = \begin{array}{c} \left| \begin{array}{cccccc} 1 & 5/2 & 10/3 & 5 & 550/3 & 1/2 \\ 0 & 1/4 & 0 & 0 & 5 & \textcircled{3/4} \\ 0 & 0 & 1/3 & 0 & 10/3 & 0 \\ 0 & 0 & 0 & 1/2 & 10 & 0 \end{array} \right| \end{array}$$

e

$$G' = \begin{array}{c} \left| \begin{array}{cccccc} 1 & 7/3 & 10/3 & 5 & 540/3 & 0 \\ 0 & 1/3 & 0 & 0 & 20/3 & 1 \\ 0 & 0 & 1/3 & 0 & 10/3 & 0 \\ 0 & 0 & 0 & 1/2 & 10 & 0 \end{array} \right| \end{array}$$

A introdução da variável de folga não melhorará o custo, logo teremos o seguinte problema Knapsack:

$$\text{maximizar } \frac{7}{3} a_1 + \frac{10}{3} a_2 + 5a_3$$

sujeito a:

$$2a_1 + 3a_2 + 4a_3 \leq 6$$

cuja solução será,  $a_1 = 1$ ,  $a_2 = 0$  e  $a_3 = 1$  e  $P = (-7, 1, 0, 1)$

$$B^{-1}P = \begin{pmatrix} 1/3 \\ 1/3 \\ 0 \\ 1/2 \end{pmatrix}, G = \begin{pmatrix} 1 & 7/3 & 10/3 & 5 & 540/3 & 1/3 \\ 0 & 1/3 & 0 & 0 & 20/3 & 1/3 \\ 0 & 0 & 1/3 & 0 & 10/3 & 0 \\ 0 & 0 & 0 & 1/2 & 10 & 1/2 \end{pmatrix}$$

e

$$G' = \begin{pmatrix} 1 & 2 & 10/3 & 5 & 520/3 & 0 \\ 0 & 1 & 0 & 0 & 20 & 1 \\ 0 & 0 & 1/3 & 0 & 10/3 & 0 \\ 0 & -1/2 & 0 & 1/2 & 0 & 0 \end{pmatrix}$$

A introdução da variável de folga ainda não melhora a solução. Resolveremos agora simultaneamente os problemas Knapsack,

$$\text{maximizar } 2a_1 + \frac{10}{3} a_2 + 5a_3$$

sujeito a:

$$2a_1 + 3a_2 + 4a_3 \leq 9$$

$$2a_1 + 3a_2 + 4a_3 \leq 6$$

$$2a_1 + 3a_2 + 4a_3 \leq 5$$

teremos:

$x \backslash s$	1	2	3	4	5	6	7	8	9
1	0	2	2	4	4	6	6	8	8
2	0	2	10/3	4	16/3	20/3	22/3	26/3	10
3	0	2	10/3	5	16/3	7	25/3	10	31/3

Tabela 3

$x \backslash s$	1	2	3	4	5	6	7	8	9
1	0	1	1	1	1	1	1	1	1
2	0	1	2	1	2	2	2	2	2
3	0	1	2	3	2	3	3	3	3

Tabela 4

Logo,  $F_3(5) = 5 \frac{1}{3}$  ,  $F_3(6) = 7$  e  $F_3(9) = 10 \frac{1}{3}$

Como  $F_3(5) = 5 \frac{1}{3} \leq 6$ ,  $F_3(6) = 7 \leq 7$ , deverão ser abandonados pois não verificam a condição,

$$2a_1 + \frac{10}{3} a_2 + 5a_3 > c$$

Como  $F_3(9) = 10 \frac{1}{3} > 10$ , satisfaz a condição acima e teremos a solução,  $a_1 = 1$ ,  $a_2 = 1$ ,  $a_3 = 1$  e  $P = (-10, 1, 1, 1)$ .

$$B^{-1}P = \begin{pmatrix} 1/3 \\ 1 \\ 1/3 \\ 0 \end{pmatrix}, G = \begin{pmatrix} 1 & 2 & 10/3 & 5 & 520/3 & 1/3 \\ 0 & 1 & 0 & 0 & 20 & 1 \\ 0 & 0 & 1/3 & 0 & 10/3 & 1/3 \\ 0 & -1/2 & 0 & 1/2 & 0 & 0 \end{pmatrix}$$

$$G' = \begin{pmatrix} 1 & 2 & 3 & 5 & 170 & 0 \\ 0 & 1 & -1 & 0 & 10 & 0 \\ 0 & 0 & 1 & 0 & 10 & 1 \\ 0 & -1/2 & 0 & 1/2 & 0 & 0 \end{pmatrix}$$

Novamente a introdução da variável de folga não melhora a solução. Resolvendo o problema Knapsack,

$$\text{maximizar } 2a_1 + 3a_2 + 5a_3$$

sujeito a:

$$2a_1 + 3a_2 + 4a_3 \leq 9$$

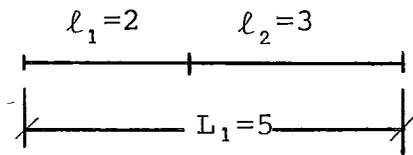
encontraremos,  $F_3(5) = 5$ ,  $F_3(6) = 7$  e  $F_3(9) = 10$ , portanto nenhum resultado satisfaz a condição

$$2a_1 + 3a_2 + 5a_3 > c$$

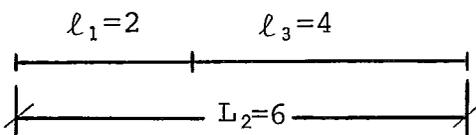
Logo, a solução ótima será cortar, 10 peças de comprimento 6 em 1 peça de comprimento 4 e uma peça de comprimento 2; 10 peças de comprimento 9 em uma peça de comprimento 2, uma peça de comprimento 3 e uma de comprimento 4. O custo será 170.

### 3.2.2 - Exemplo Ilustrativo

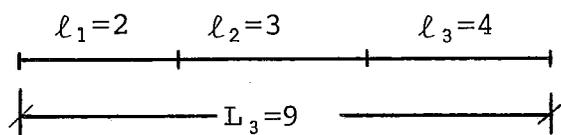
Consideremos um pedido de 20 peças de comprimento 2, de 10 peças de comprimento 3, de 20 peças de comprimento 4 a serem cortadas de peças de comprimentos 5, 6, e 9 ao custo respectivamente de 6, 7 e 10, segundo os modelos:



Modelo 1



Modelo 2



Modelo 3

Solução: Temos a matriz

$$A = \begin{vmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{vmatrix}$$

na qual as colunas são os modelos dados.

Fazendo:

$x_1$  = nº de peças cortadas pelo modelo 1.

$x_2$  = nº de peças cortadas pelo modelo 2.

$x_3$  = nº de peças cortadas pelo modelo 3.

Para atender o pedido a custo mínimo devemos:

minimizar  $z = 6x_1 + 7x_2 + 10x_3$

Sujeito a:

$$\begin{vmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{vmatrix} \begin{vmatrix} x \\ x \\ x \end{vmatrix} > \begin{vmatrix} 20 \\ 10 \\ 20 \end{vmatrix}$$

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0$$

Acrescentando as variáveis de folga fica:

minimizar  $z = 6x_1 + 7x_2 + 10x_3$

sujeito a:

$$x_1 + x_2 + x_3 - x_4 = 20$$

$$x_1 + x_3 - x_5 = 10$$

$$x_2 + x_3 - x_6 = 20$$

Resolvendo o PPL obtemos

$x_1 = 0$ ,  $x_2 = 10$ ,  $x_3 = 10$ , isto é, devemos cortar zero peças de comprimento 5, 10 peças de comprimento 6 e 10 peças de comprimento 9 para atender o pedido. O custo é 170.

Quando os modelos são explicitados o problema torna-se relativamente simples podendo ser resolvido por código de computador já existentes tanto para programação linear como para programa linear inteira.

CAPÍTULO IVO PROBLEMA DE CORTE EM DUAS DIMENSÕES

Neste capítulo analisaremos o problema geral de corte em duas dimensões e o problema de corte de guilhotina em dois estágios.

Para tornar mais claro o desenvolvimento iniciaremos com um problema de programação linear de corte em uma dimensão.

4.1 - Problema em Uma Dimensão

Uma chapa de largura  $L$  será cortada em peças de largura  $\ell_i$ , para atender a demanda  $N_i$ , ( $i = 1, \dots, m$ ) por qualquer número de peças de largura  $\ell_i$ , tal que a soma do comprimento total seja pelo menos  $N_i$ .

As demandas serão atendidas decidindo sobre os vários modelos de corte da chapa de largura  $L$ , conforme ilustrado na Figura IV-1. O  $j$ -ésimo modelo de corte é uma maneira de dividir a largura  $L$ , em larguras menores  $\ell_i$ , ( $i = 1, \dots, m$ ), aplicado a um comprimento  $x_j$  da chapa.

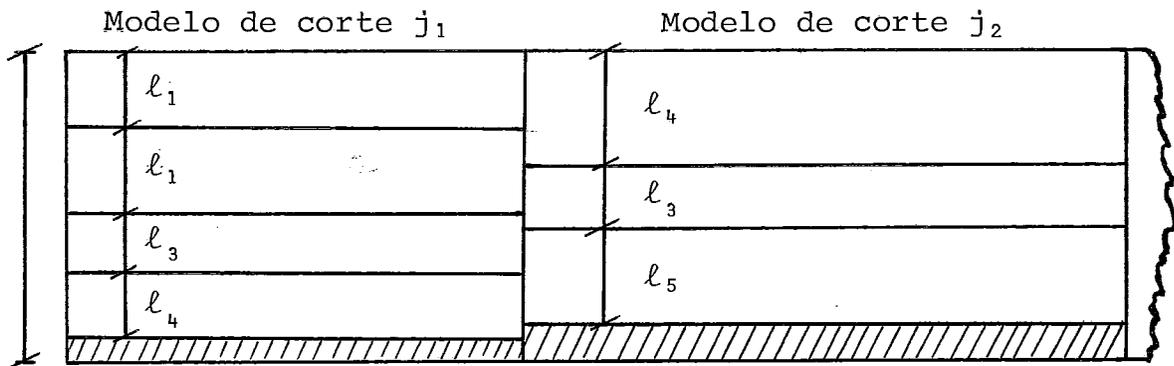


Fig. IV-1

Nesta formulação a matriz  $A$  do programa linear tem  $m$  linhas e um grande número de colunas, uma para cada um dos possíveis modelos de corte. Então cada conjunto  $\{a_1, a_2, \dots, a_1, \dots, a_m\}$  de inteiros satisfazendo.

$$l_1 a_1 + l_2 a_2 + \dots + l_i a_i + \dots + l_m a_m \leq L$$

é uma coluna da matriz. Teremos então o seguinte (PPL).

minimizar  $1 \cdot x$

sujeito a:

$$Ax \geq N$$

$$x \geq 0$$

onde,

$x$  = vetor coluna das variáveis, uma para cada coluna da matriz  $A$

$1 = (1, \dots, 1)$  é um vetor linha de elementos todos iguais a um.

$N = (N_1, N_2, \dots, N_m)$  é o vetor coluna das demandas  $N_i$ .

$A$  = matriz de  $m$  linhas cujas colunas ( $a_1, a_2, \dots, a_m$ ) são os possíveis modelos de corte.

$a_i$  = é o número de peças  $l_i$ , ( $i = 1, 2, \dots, m$ ) que ocorre no modelo.

Se acrescentarmos as restrições  $x_j$  e  $N_i$  inteiros, teremos um Problema de Programação Inteira (P.PI). Na prática resolvemos o (PPL) e arredondando os resultados poderemos obter uma solução satisfatória para o (PPI).

#### 4.2 - Problema Geral de Corte em Duas Dimensões

Neste problema o material de estoque usado consiste de uma barra de secção transversal retangular  $W \times L$ , que será cortada fornecendo uma quantidade  $N_i$  de uma barra de secção transversal retangular menor,  $w_i \times l_i$ , ( $i = 1, \dots, m$ ). A quantidade  $N_i$  pode ser atendida por qualquer número de barras  $w_i \times l_i$ .

O corte será executado do seguinte modo: escolhemos um certo número de modelos de corte retangular, cada modelo definido como uma maneira de ajustar retângulos menores  $w_i \times l_i$  dentro de um retângulo maior  $W \times L$ . O  $j$ -ésimo modelo descreve como a barra deverá ser cortada em um comprimento  $x_j$  para produzir barras menores conforme será mostrado na Figura IV-2.

O objetivo do problema será atender as demandas, usando o mínimo de material de estoque.

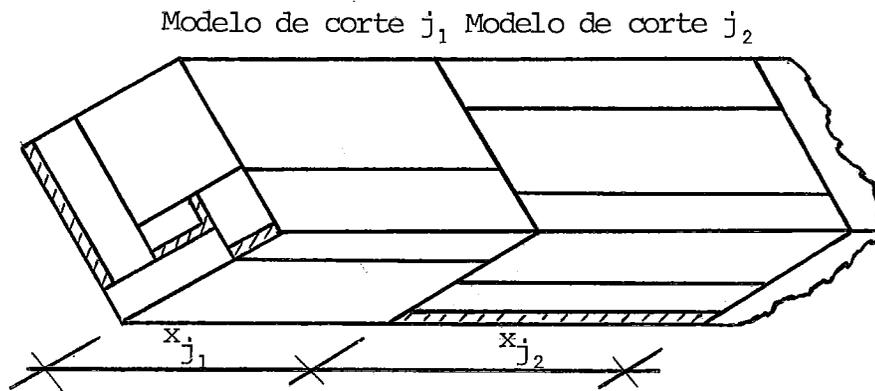


Fig. IV-2

O problema geral de corte em duas dimensões possui uma formulação semelhante ao de uma dimensão, logo teremos o (PPL).

minimizar  $lX$

sujeito a:

$$AX \geq N$$

$$X \geq 0$$

onde,

$A$  = matriz de  $m$  linhas cujas colunas ( $a_1, a_2, \dots, a_n$ ) são os possíveis modelos de corte retangular do retângulo  $W \times L$ .

$a_i$  = número de retângulos  $w_i \times l_i$ , ( $i = 1, 2, \dots, m$ ) que ocorre no modelo.

$x = (x_1, x_2, x_3, \dots)$ , é um vetor coluna cujas componentes indicam os comprimentos onde a barra deverá ser cortada para produzir barras menores.

$N = (N_1, N_2, \dots, N_m)$ , vetor coluna das demandas.

$1 = (1, 1, \dots)$  vetor linha cujas componentes são todas iguais a 1.

Na figura IV-3, ilustramos um modelo de corte retangular, no qual ocorrem

1 retângulo	$W_1 \times l_1$
2 retângulos	$W_3 \times l_3$
1 retângulo	$W_4 \times l_4$
3 retângulos	$W_8 \times l_8$
1 retângulo	$W_6 \times l_6$

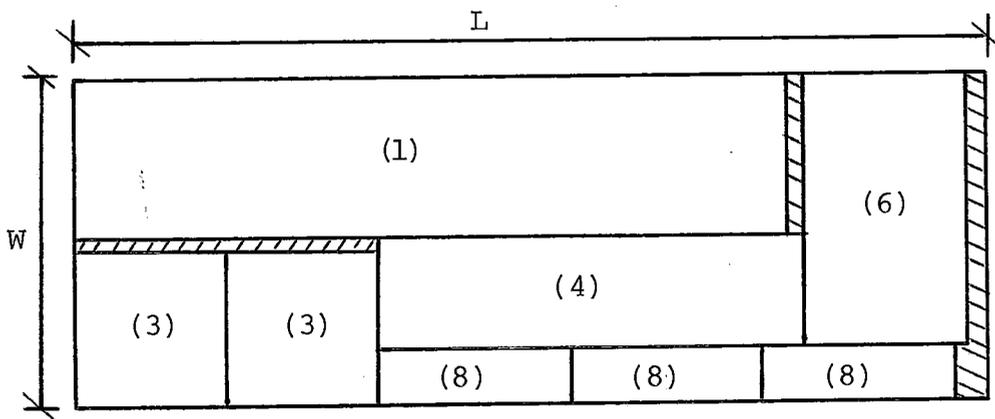


Fig. IV-3

Portanto se  $m = 10$  para um problema com este possível modelo de corte, uma das colunas de  $A$  será  $(1, 0, 2, 1, 0, 1, 0, 3, 0, 0)$ .

Como no caso de uma dimensão, a dificuldade para resolver o problema em duas dimensões é o grande número de colunas que pode ocorrer na matriz  $A$ . Se aplicarmos métodos de geração de coluna para este tipo de problema, teremos o seguin

te problema Knapsack:

$$\text{maximizar } \pi_1 a_1 + \pi_2 a_2 + \dots + \pi_m a_m$$

sujeito a:

$(a_1, a_2, \dots, a_m)$  correspondendo a condição de um modelo de corte retangular do retângulo  $W \times L$ .

$\pi_i$ ,  $(i = 1, \dots, m)$  é um multiplicador de Lagrange ou "preço sombra", correspondente a  $i$ -ésima equação de uma solução básica viável.

Na prática aparecem problemas particulares de corte em duas dimensões resultantes da utilização de certas máquinas que cortam o material em linha reta de uma extremidade a outra, como por exemplo o Cortador Guilhotina, usado para cortar papel. Em seguida analisaremos problemas de corte de guilhotina.

#### 4.3 - Corte de Guilhotina

É o corte executado num material em linha reta de um extremo a outro.

Um exemplo de corte de guilhotina será mostrado na Figura IV-4, onde os cortes serão numerados na ordem em que foram feitos.

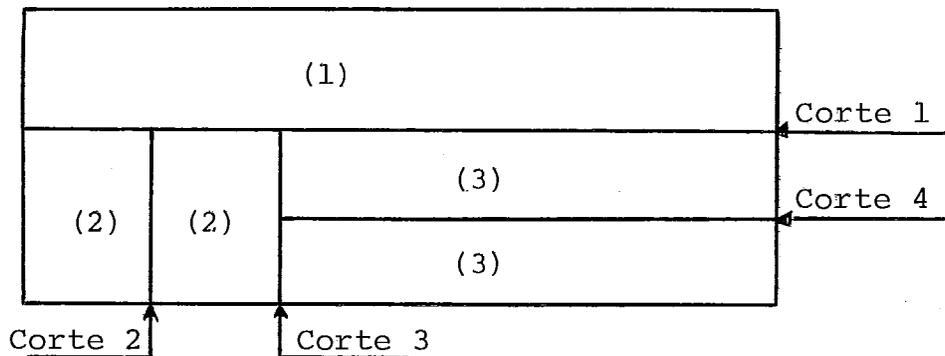


Fig. IV-4

O modelo da Figura IV-3, não poderá ser produzido por corte de guilhotina. O corte de guilhotina limita muito os modelos de corte em duas dimensões, apesar disso modelos bastante gerais podem ser obtidos, como o ilustrado na Figura IV-5.

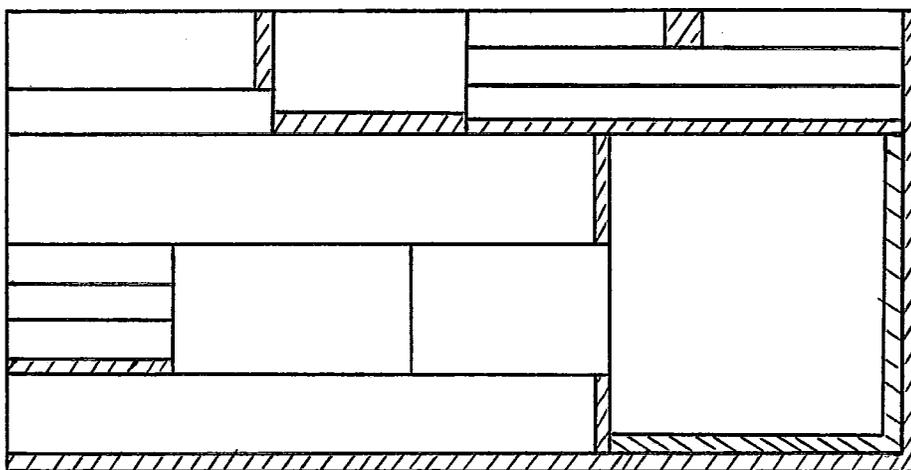


Fig. IV-5

Um problema envolvendo cortes de guilhotina poderá ser resolvido usando a métodos de Programação Dinâmica.

Uma fórmula de recorrência usada para este fim é fornecida pela referência (4).

Porém, nos limitaremos a subclasses especiais, menos difíceis de modelar e que correspondem aos métodos de corte usados em certas indústrias.

#### 4.3.1 - Cortes de Guilhotina em Dois Estágios

Uma subclasse importante de modelos envolvendo somente cortes de guilhotina é aquela em que os cortes são considerados em dois estágios. Será ilustrado na Figura IV-6, um modelo produzido desta forma, onde o retângulo  $W \times L$  é cortado segundo o comprimento em tiras e cada uma destas tiras cortadas segundo a largura. As vezes um terceiro estágio é considerado como ilustrado na Figura IV-7.

Os retângulos produzidos nos dois primeiros estágios serão cortados segundo o comprimento para atender os tamanhos pedidos.

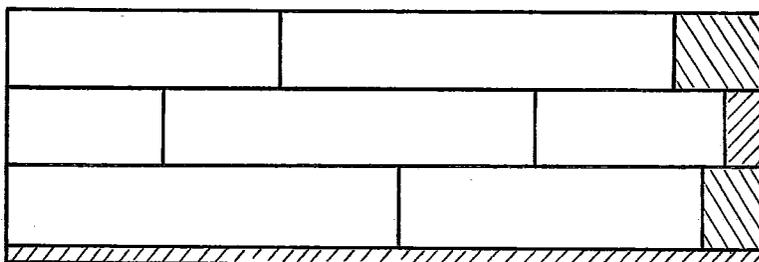


Fig. IV-6

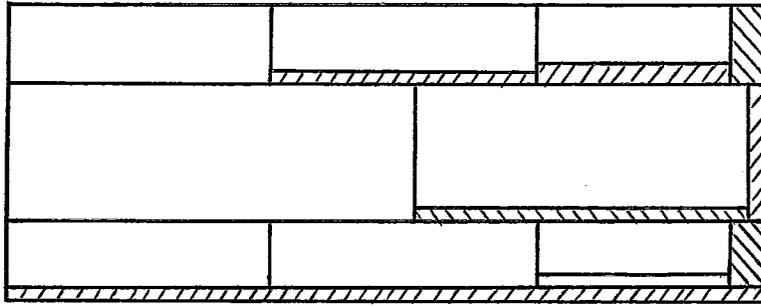


Fig. IV-7

O problema Knapsack que aparece neste caso é mais complicado que aquele, no caso de dois estágios.

#### 4.3.2 - Formulação Usual do Problema Envolvendo Cortes de Guilhotina em Dois Estágios

Para resolver o problema de corte de guilhotina em dois estágios, basta resolvermos o problema,

$$\text{maximizar } \pi_1 a_1 + \pi_2 a_2 + \dots + \pi_m a_m$$

sujeito a:

$(a_1, a_2, \dots, a_m)$  correspondendo a um modelo de corte de guilhotina em dois estágios do retângulo  $W \times L$ .

Este problema poderá ser resolvido em dois estágios:

- i) Para toda largura  $w_i$ , calcular  $\pi_i^*$ , o valor ótimo obtido ajustando retângulos  $w_j \times l_j$ , onde  $w_j < w_i$ , de extremo a extremo em uma tira de largura  $w_i$  e comprimento  $L$ . Para cada  $i$  temos um problema Knapsack do tipo visto em 3.1.

ii) O valor ótimo da função objetivo

$$\pi_1 a_1 + \pi_2 a_2 + \dots + \pi_m a_m$$

é obtido resolvendo o problema Knapsack

$$\text{maximizar } \pi_1^* b_1 + \dots + \pi_m^* b_m$$

sujeito a:

$$W_1 b_1 + \dots + W_m b_m < W.$$

Os problemas Knapsack de (i) poderão ser resolvidos, usando o método desenvolvido em 3.2. Ordenando os retângulos de forma que  $w_1 < w_2 < \dots < w_m$ , sendo  $\pi_i^* = F_i(L)$ , quando calculamos  $F_m(L)$  obteremos  $\pi_i^*$  para todo  $i$ . Portanto o problema Knapsack do corte de guilhotina de dois estágios será solucionado resolvendo dois problemas Knapsack do tipo tratado em 3.1.

#### 4.4 - Formulação do Problema de Corte de Guilhotina como (P.P.L.)

É possível formular o problema de corte de guilhotina em dois estágios como um (PPL) e resolvê-lo pelo método visto em III. ou pelo método de decomposição de Dantzig e Wolfe.

O problema é tratado como um (PPL) em dois estágios. O primeiro estágio corresponderá ao corte do retângulo  $W \times L$  em tiras com larguras iguais às larguras dos retângulos  $W_i \times l_i$ , ( $i = 1, \dots, m$ ) pedidos. O segundo estágio corresponderá ao corte das tiras em comprimentos iguais aos comprimentos dos retângulos pedidos.

Consideremos a matriz  $A$  com  $(2m)$  linhas particionada verticalmente em  $(m + 2)$  matrizes  $A_s$ . Seja  $\bar{x}$  um vetor coluna particionado horizontalmente de acordo com a matriz  $A$  em,  $(\bar{x}_0, \bar{x}_1, \dots, \bar{x}_m, \bar{u})$ , onde  $\bar{u}$ , é o vetor coluna das variáveis de folga. Seja  $N$  um vetor coluna de  $2m$  linhas, particionado em um vetor nulo de dimensão  $m$  e um vetor  $\bar{N}$  do número de retângulos pedidos, ordenados em tamanhos de larguras crescentes.

As colunas de  $A$  correspondem aos modelos que cortam os retângulos  $W \times L$  em tiras. Especificamente, na  $j$  - ésima coluna que corresponde ao  $j$  - ésimo modelo de corte de tiras, os primeiros  $m$  elementos são:  $(b_1, b_2, \dots, b_m)$  inteiros não negativos satisfazendo a desigualdade,

$$\sum_{i=1}^m b_i w_i \leq W \quad (1)$$

e os últimos  $m$  elementos são todos nulos.

Para cada conjunto de inteiros  $\{b_1, \dots, b_m\}$  satisfazendo (1), corresponderá um modelo de corte do retângulo  $W \times L$  em  $b_i$  tiras de largura  $w_i$ , ( $i = 1, \dots, m$ ).  $A_s$ , ( $s = 1, \dots, m$ ) será o conjunto de modelos que toma as tiras de largura  $w_s$  e corta-as em retângulos. As colunas de  $A_s$  contêm zero nas  $m$  primeiras linhas, exceto na  $s$ -ésima linha, onde aparece o elemento  $-1$ . Existem  $a_i$  inteiros não-negativos nas linhas  $(m + i)$ , ( $i = 1, \dots, s$ ) e zero, abaixo dessas linhas, com os inteiros satisfazendo a igualdade,

$$\sum a_i \ell_i \leq L \quad (i = 1, \dots, m) \quad (2)$$

Para cada conjunto de inteiros satisfazendo (2), existe uma coluna de  $A_s$  correspondendo ao modelo que corte uma tira de largura  $w_s$  em  $a_i$  retângulos de comprimento  $\ell_i$ , ( $i = 1, 2, \dots, s$ ).

Finalmente  $A_{m+1}$ , será uma matriz  $2m \times m$  com as primeiras  $m$  linhas nulas e as últimas  $m$  linhas iguais a  $-I$  (matriz identidade), formado pelos coeficientes das variáveis de folga das  $m$  últimas equações. Teremos então o seguinte (PPL)

$$\text{minimizar} \quad \sum_j x_j \quad (3)$$

sujeito a:

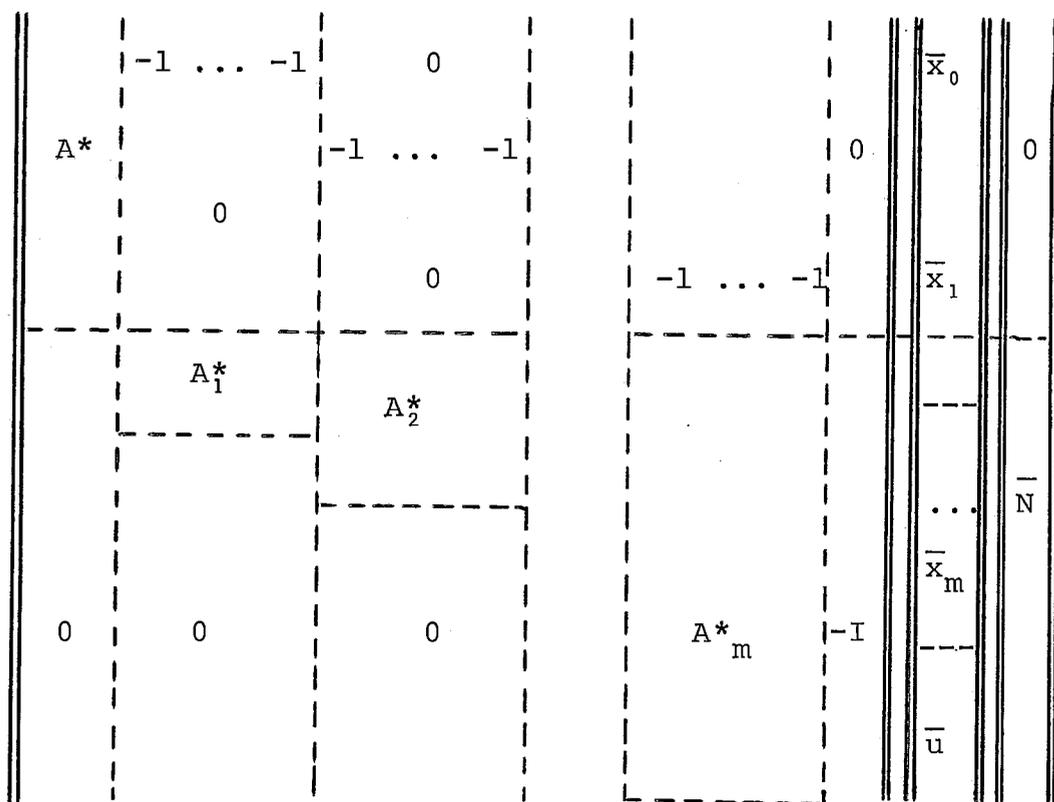
$$\overline{AX} \geq \tilde{N} \quad (4)$$

onde,

$x_1^0, x_2^0, \dots$ , são o número de vezes que cada um dos vários modelos para cortar tiras de  $W \times L$  são usados.

$x_1^s, x_2^s, \dots$ , são o número de vezes que cada um dos vários modelos de corte de tiras de largura  $w_s$  são usados.

Definindo  $A_0^*$ , como as  $m$  primeiras linhas de  $A_0$ ,  $A_s^*$  como sendo as linhas  $m + 1, \dots, m + s$  de  $A_s, (s = 1, \dots, m)$  podemos escrever (4) da seguinte maneira:



As primeiras  $m$  equações estabelecem que as tiras produzidas no primeiro estágio são usadas exatamente no segundo estágio. O segundo conjunto de equações estabelece que o número de retângulos pedidos  $w_i \times l_i$  deve ser atendido cortando as  $t_i$  ras.

Podemos resolver o problema definido por (3) e (4) pelo método visto em 3.1 ou pelo método de decomposição de Dantzig e Wolfe.

No método 3.1, para achar a coluna melhorada de  $A_0$ , usamos os "preços sombras" de uma solução inicial, resolvendo um problema Knapsack. Para achar a coluna melhorada entre todas  $A_s$ ,  $1 < s < m + 1$ , poderemos resolver os vários problemas Knapsack simultaneamente, usando 3.2. O tratamento dado as variáveis de folga será o mesmo de 3.1. No cálculo de uma coluna melhorada de  $A_0$ , teremos uma matriz inversa  $m \times m$ , no caso de  $A_s$  teremos uma matriz inversa  $2m \times 2m$  e, provavelmente o dobro de pivôs. Uma matriz inversa menor poderá ser obtida pela decomposição de Dantzig e Wolfe, que particiona a matriz  $A$  em uma parte superior  $A_1$ , consistindo das  $m$  primeiras linhas e uma inferior  $A_2$ , formada pelas  $m$  últimas linhas. Assim trabalharemos com uma matriz inversa  $m \times m$ . Contudo para cada pivoteamento no problema com a matriz  $A_1$ , terá que ser resolvido um problema de corte de programação linear para a matriz  $A_2$ .

#### 4.5 - Outros Problemas de Corte em Duas Dimensões

Trataremos agora de subclasses de corte de guilhotina em dois estágios que aparecem especialmente indústrias do vidro e do aço. Estas subclasses são aquelas nas quais o segundo estágio deve ser realizado simultaneamente em algumas das chapas resultantes dos cortes do primeiro estágio. Por exemplo, podemos reunir as chapas em dois grupos e cortar simultaneamente todas as chapas de um dos grupos. Na figura IV-1 temos um modelo onde ocorrem exatamente dois grupos,

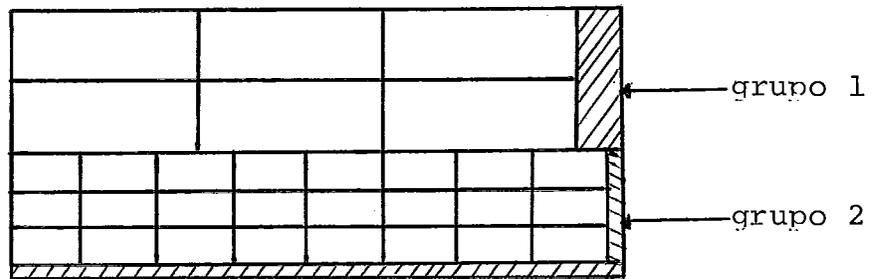


Fig. IV-1

e na Figura IV-2 um modelo de exatamente um grupo.

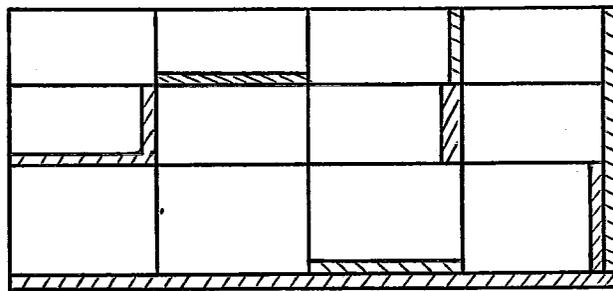


Fig. IV-2

Em geral teremos exatamente  $p$ -grupo de chapas para um  $p$  determinado. O problema de dois estágios discutido anteriormente, pode ser considerado como o caso onde  $p$  é ilimitado, e será referido como um problema de dois estágios livre.

#### 4.5.1 - O problema de 1 - Grupo

A classe dos problemas 1-grupo, isto é, para  $p = 1$  é muito mais restrita que a classe de dois estágios livres.

Chamaremos de problema não exato aquele em que um terceiro estágio de corte é permitido e de problema exato aquele em que não é permitido.

Se o corte é ou não exato afeta muito a dificuldade de solução dos problemas deste tipo. Num problema não exato, um retângulo  $w_i \times l_i$  pode ser cortado de qualquer retângulo  $w \times l$  cujas dimensões satisfazem as condições  $w \geq w_i$  e  $l \geq l_i$ . Por exemplo, no caso não exato, a chapa da figura IV-3 forneceria 3 retângulos de dimensões  $w_1 \times l_1$ ,  $w_2 \times l_2$  e  $w_3 \times l_3$ . No caso exato ela forneceria, somente, um retângulo  $w_3 \times l_3$ .

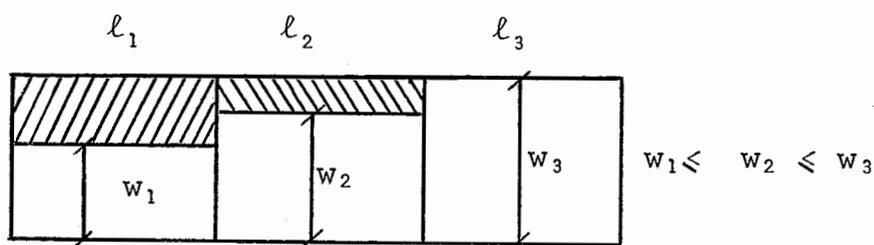


Fig. IV-3

Os problemas Knapsack no caso não exato são mais difíceis que aqueles do caso exato. Considerando  $l_i \neq l_j$  se  $i \neq j$ , então usamos somente a largura em qualquer modelo. Se  $w_i \neq w_j$ , implica  $l_i \neq l_j$  o problema permanece fácil. Neste caso podem atender comprimentos diferentes usando a mesma largura como ilustrado na figura IV-5.

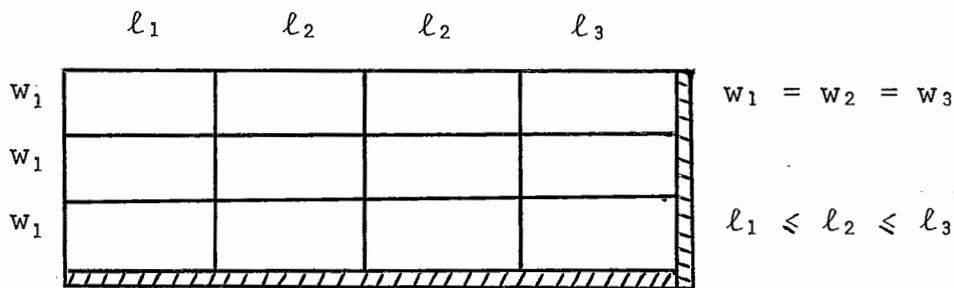


Fig. IV-5

Para resolver o problema Knapsack neste caso avalia-se o valor  $v_i$  da chapa de largura  $w_i$  por qualquer método disponível usando somente os  $l$ 's associados. O valor total do retângulo será então.

$$\max_i v_i \left[ \frac{W}{w_i} \right]$$

Em geral tem que resolver mais do que  $m$  problemas Knapsack do tipo visto em 3.2 um para cada  $w_i$ , porém o número de variáveis no problema não excede  $m$ .

#### 4.5.2 - O problema de 2-Grupos

Este problema é encontrado na indústria do vidro. Sendo uma combinação de dois problemas do 1-grupo parece ser mais difícil do que 4.5.1 no caso não exato. Se para todo  $i$  e  $j$   $w_i \neq w_j$  implica  $l_i \neq l_j$  o problema é tratável. Associa-se  $v_i$  com cada chapa de largura  $w_i$  para todo  $i$ , resolve-se o problema Knapsack somente para dois  $w_i$  diferentes, resultando o modelo da fig. IV-1.

#### 4.5.3 - Corte de Guilhotina em 3 estágios

Num terceiro estágio do corte de guilhotina re torna-se a largura cortada, para cortá-la novamente. Isto foi feito antes no processo de corte nos problemas não exatos, mas não afetou fundamentalmente os métodos de solução. Com cortes em três estágios em geral permitidos os métodos anteriores de solução não são muito aplicáveis. Problemas deste tipo são encontrados nos cortes de folhas de papel e chapas de vidro. Na fig. IV-6 damos

um exemplo de um modelo de corte de guilhotina em 3 estágios.

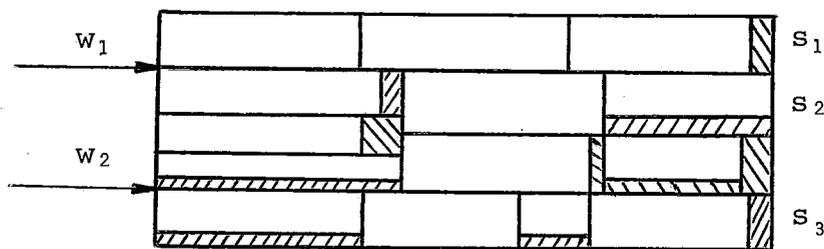


Fig. IV-6

O modelo foi obtido cortando o retângulo  $W \times L$  nos pontos indicados pelas setas, cortando em seguida as chapas resultantes  $S_1$ ,  $S_2$  e  $S_3$  e tomando os retângulos de  $S$  e cortando-os novamente, obtendo os retângulos pedidos. Note que num processo de corte deste tipo, as larguras das chapas resultantes dos cortes do retângulo  $W \times L$  podem ser dados por

$$W = \sum_{i=1}^m a_i w_i$$

sendo  $a_i$  inteiros não negativos.

Para resolver modelos de corte deste tipo é necessário resolver um problema de corte de guilhotina de dois estágios para cada retângulo  $W \times L$ , onde  $0 \leq w \leq W$ , para obter o valor  $\pi(w)$  associado com cada  $w$ . Resta ainda resolver um problema Knapsack no qual o número de larguras  $w$  é bastante grande para determinar que chapa seria cortada do retângulo  $W \times L$ .

Quando o corte do terceiro estágio é limitado ou pelo número de cortes, ou por modelos envolvendo somente uma lar gura a solução é mais fácil.

CAPÍTULO VCONCLUSÃO5.1 - Outros Algoritmos para o Problema de Corte

J.C. HERZ (10) desenvolveu um algoritmo recursivo que aumenta a velocidade da solução dos problemas de cortes em duas dimensões.

Quando o corte é de guilhotina mostra que a solução ótima é encontrada mais facilmente por programação recursiva do que pelos métodos convencionais, possibilitando economia de memória no aspecto computacional.

Compara o algoritmo recursivo com dois algoritmos iterativos desenvolvidos anteriormente por GILMORE-GOMORY (7). Diante as limitações dos algoritmos e exibe resultados numéricos.

NICOS CHRISTOFIDES e CHARLES WHITLOCK (11) apresentam um algoritmo de busca em árvore para problemas de corte em duas dimensões, no qual há uma restrição no número máximo de cada tipo de peça produzido. O algoritmo limita a busca em árvore estabelecendo e impondo condições para otimizar o modelo de corte.

Um procedimento de programação dinâmica para a solução dos problemas não restritos e um método de avaliação de um nó baseado num problema de transporte são usados para estabelecer os limites superiores durante a busca.

O desempenho computacional do algoritmo é testado em um grande número de problemas gerados aleatoriamente. Os resultados indicam que o algoritmo é eficiente para resolver problemas de corte de tamanho médio.

Relacionamos outros problemas estudados por GILMORE e GOMORY (4).

O problema em três estágios é considerado como uma extensão do de dois estágios. Consideram também vários itens de estoque, ou uma das dimensões do estoque variando continuamente.

Também uma chapa pode não ser uniforme e apresentar defeitos ou variação de espessura ou densidade. Concluem com uma aplicação dos métodos a produção de caixas corrugadas, na qual há problemas de planejamento de máquinas.

O método de programação linear (4) com a técnica de HERZ (10) aplicada ao problema Knapsack auxiliar é o mais potente algoritmo para o problema de corte em duas dimensões (9).

## 5.2 - Conclusão

Segundo HINXMAN (9) os melhores métodos para problemas de corte são os de programação linear de GILMORE e GOMORY, referências (1) e (4), adaptados a métodos melhorados para resolver o problema Knapsack. Contudo são computacionalmente dispendiosos. Além do mais estes métodos são destinados somente a minimização das perdas.

Nos casos práticos outros custos devem ser consi  
derados. R.S. STAINTON(9) cita 15 fatores principais que influem  
vantajosamente muito dos quais não são facilmente quantificáveis.

Outra limitação é a necessidade de resolver o pro  
blema Knapsack.

Os métodos disponíveis para resolver o problema  
Knapsack exigem restrições de ordem geométrica.

Para superar estas limitações poderão ser usados  
métodos heurísticos.

Alguns aspectos comuns aos métodos heurísticos são  
os seguintes:

(i) o estabelecimento de um nível desejado para saber se o mode  
lo encontrado pode ser usado.

(ii) redução da repetição exaustiva.

(iii) o uso de valores heurísticos, para levar em conta fatores  
tais como máquina e custos de quebra de produção.

No atual estado da arte de modelagem de um proble  
ma de corte é razoável desenvolver um método próprio, a menos que  
os métodos de GILMORE e GOMORY sejam aplicáveis. Tais métodos se  
limitam quase sempre a métodos heurísticos.

Contudo métodos heurísticos tem a tendência de ba  
searem-se em problemas particulares, e problemas semelhantes reque

rem métodos heurísticos diferentes.

A criação de novos algoritmos depende do desenvolvimento da formulação matemática que modelaria a complexidade dos problemas e baratearia os custos computacionais.

BIBLIOGRAFIA

- (1) GILMORE, P.C. e GOMORY, R.E. - A Linear Programming Approach to the Cutting Stock Problem, Opus. Res. 9, 849-859 (1961)
- (2) GILMORE, P.C. e GOMORU, R.E. - A Linear Programming Approach to the Cutting Stock Problem Part II, Opus Res. 11, 863-888 (1963).
- (3) SALKIN, H.M. e De KLUYVER, C.A. - The Knapsack Problem: a Survey, Naval Research Logistics Quartely, March, 1975, vol. 22, n. 1, 127-141.
- (4) GILMORE, P.C. e GOMORY, R.E. - Multistage cutting stock problems of two and more dimensions, Operations Research 13 (1965) 94-120.
- (5) LASDON, L.S. - Optimization Theory for Lurge Systems, The Macmillan Company, Inc., 1970.
- (6) TAHA, H.A. - Integer Programming Theory Applications, and Computations, Academic Press, Inc, 1975.
- (7) GILMORE, P.C. e GOMORY, R.E. - The Theory and Computation of Knapsack Funtions, Operations Res. 14(1966)1045-1074.
- (8) CREENBERG, H. e HEGERICH, R.L. - A Branch Search Algorithm for the Knapsack Problem, Managemente Science, vol. 16,n. 5, January, 1970.

- (9) HINXMAN, A.I. - The Trim - Lora and Assortment Problems: A Survey, North-Holland Publishing Company, European Journal of Operational Research 5(1980) 8-18.
- (10) HERZ, J. - Recursive Computational Procedure of two dimensional Stock Cutting, IBM J.Res. Develop. 16(1972) 462-469.
- (11) CHRISTOFIDES, N. e WHITLOCK, C. - An Algorithm for Two Dimensional Cutting Problems, Operations Research, vol. 25, n. 1, January - February (1977).
- (12) SALKIN, H.M. - Integer Programming Addison - Wesley Publishing Company, Inc. 1975.
- (13) DANTZIG, G. B. - Discrete Variable Extremum Problems, Opus. Res. 5, 266-277 (1957)
- (14) GASS, S. - Linear Programming, McGraw - Hill Kogakusha, Ltd. 1975.
- (15) HADLEY, G. Linear Programming, Addison - Wesley Publishing Company, 1975.
- (16) MACULAN, N.F. e PEREIRA, M.V.F. - Programação Linear, Editora Atlas S.A. (1980).
- (17) HU, T.C. - Integer Programming and Network Flows, Addison Wesley, 1969.

- (18) ELLENRIEDER, A. Von e BARROSO, S.E.G. Programação Linear. São Paulo, Almeida Neves, 1971.
- (19) MACHADO, Hilton. Programação Linear. 10º Colóquio Brasileiro de Matemática. Poços de Caldas. Julho de 1975. Rio de Janeiro: IMPA, 1975.
- (20) PUCCINI, A.L. Introdução à Programação Linear. Rio de Janeiro. Livros Técnicos e Científicos. 1975.
- (21) BREGALDA, P.F.; OLIVEIRA, A.A.F. & BORNSTEIN, C.T. - Introdução à Programação Linear. Editora Campus Ltda., 1981.