

MÉTODOS COMBINATÓRIOS
PARA A SOLUÇÃO DO PROBLEMA
DE CORTE DE CHAPAS

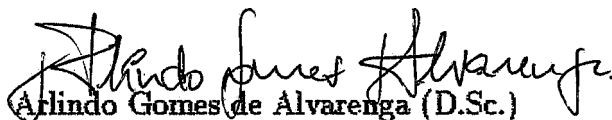
Almir Torquato Dantas Filho

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO EM ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO

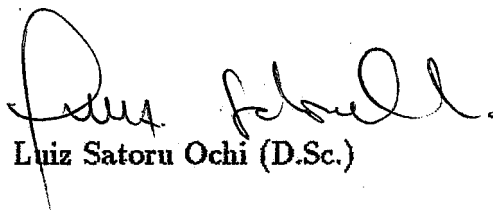
Aprovada por:



Nelson Maculan Filho (D.Sc.)
(Presidente)



Arlindo Gomes de Alvarenga (D.Sc.)



Luiz Satoru Ochi (D.Sc.)

JUNHO DE 1992

DANTAS FILHO, ALMIR TORQUATO

**Métodos Combinatórios para a
Solução do Problema de Corte de Chapas**

[Rio de Janeiro] 1992

IX, 107 p., 29.7 cm,

(COPPE/UFRJ,

M. Sc., ENGENHARIA DE SISTEMAS E COMPUTAÇÃO, 1992)

TESE - Universidade Federal do Rio de Janeiro, COPPE

1 - Corte de Chapas

2 - Programação Linear Inteira

3 - O Problema da Escolha de Estoques

4 - Algoritmos

I. COPPE/UFRJ II. Título(Série).

*"Todo erro humano é a impaciência,
uma renúncia prematura ao método,
uma ilusória fixação a uma ilusão"*

FRANZ KAFKA

*A minha mãe
Alcinéa*

*A meu pai
Almir Dantas e*

*A meus irmãos
Adriana, Alexandre, Anna e Alricéa*

AGRADECIMENTOS

Durante o período em que escrevi a tese morei em quatro cidades diferentes: Rio de Janeiro, Vitória, Porto Alegre e Novo Hamburgo. Sendo assim, fica difícil lembrar os nomes de todos aqueles que ajudaram no trabalho da tese de mestrado. Espero que me perdoem qualquer lapso de memória.

No início estive na pensão da dona Hilda, em Copacabana, onde além de conhecê-la, tive o coleguismo do Coca, do Waldomiro, do Antonio Carlos e do Marcelo.

Gostaria de agradecer a cooperação e companhia de meus colegas da "república das parreiras": Hércules, Wamberto, Fernando Rivas, Rui Chiou, André Jalles e Carlos Martignhon. Essas pessoas me deram grande colaboração técnica e muita amizade.

Ao pessoal da Coppe que veio também de Fortaleza: Marcos Negreiros, Claudia Linhares, Anatalia Saraiva, Rubens, Einstein, Emilia e Morganna. Aos demais colegas da Coppe do Rio e de outros estados. Esta é a parte mais difícil, mas vai lá: Ismael Regis, José Roberto Zappa, Murilo, Célia, Graça, todos da Otimização, Cristina e Claudia (as irmãs de Niterói) e o Paulo da Engenharia de Transportes. Àqueles que faltaram minhas sinceras desculpas.

Ao Romildo, de Fortaleza, que está em Vitória dando aula na UFES. Gostaria de agradecer a cooperação durante o período em que estive dando aula na UFES ao prof. Francisco Negreiros. Agradecimentos especiais: ao prof. Arlindo Gomes de Alvarenga. Ele prestou grande ajuda ao presente trabalho, incluindo referências bibliográficas, esclarecimento de dúvidas, além de ter proposto o tema da tese. Sem dúvida o prof. Arlindo foi peça fundamental neste trabalho. Gostaria também de agradecer em especial ao prof. Antonio Clecio Fontelles da UFC, sem o qual minha ida para a Coppe teria sido muito mais difícil.

Agradecimentos importantes ao meu orientador professor Nelson Maculan Filho que me prestigiou durante toda a demorada fase da tese.

Agradecimentos também aos meus colegas do setor de informática da Refinaria Alberto Pasqualini, pela sua paciência em me aturar solicitando esclarecimentos acerca do mundo da microinformática, em especial ao meu amigo Marco Costa.

Finalmente, o agradecimento mais importante não poderia deixar de ser dado a minha mãe. Suas palavras de carinho me deram ânimo para continuar a jornada com coragem, Ao meu pai, Almir Dantas, um abraço. Aos meus irmãos, espero que eles se renovem com a fé em Deus.

Resumo da Tese apresentada à COPPE como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M. Sc.)

Métodos Combinatórios para a Solução
do Problema de Corte de Chapas
Almir Torquato Dantas Filho
Maio de 1992

Orientador: Nelson Maculan Filho

Programa: Engenharia de Sistemas e Computação

A tese é um apanhado teórico acerca do que diversos pesquisadores escreveram sobre o assunto. O primeiro capítulo está baseado no artigo "The theory and Computation of Knapsack Functions" de P. C. Gilmore e R. E. Gomory. O segundo capítulo é baseado no artigo de J. C. Herz intitulado "Recursive Computational Procedure for Two-Dimensional Stock Cutting". O terceiro capítulo é baseado no trabalho de P. C. Gilmore e R. E. Gomory intitulados "A Linear Programming Approach to the Cutting Stock Problem" e "A Linear Programming Approach to the Cutting Stock Problem - Part II". Também foram bastante utilizados os trabalhos de Harald Dyckhoff, "A New Linear Programming Approach to the Cutting Stock Problem" e o livro de Vašek Chvátal sobre programação linear. O quarto capítulo foi baseado no artigo de P. C. Gilmore e R. E. Gomory intitulado "Multistage Cutting Stock Problems of Two and More Dimensions". O quinto capítulo foi baseado no artigo de Nicos Christofides e Charles Whitlock intitulado "An Algorithm for Two-Dimensional Cutting Problems". O sexto capítulo é baseado nos artigos de P. Y. Wang, "Two Algorithms for Constrained Two-Dimensional Cutting Stock Problems", e de José Fernando Oliveira e José Soeiro Ferreira, "An improved version of Wang's algorithm for two-dimensional cutting problems". O sétimo capítulo é baseado no artigo de David W. Pentico, "The Discrete Two-Dimensional Assortment Problem", de Willard I. Zangwill, "Minimum Concave Cost Flows in Certain Networks", e de M. L. Wolfson, "Selecting the Best Lengths to Stock". Finalmente, o último capítulo é baseado no artigo escrito por J. E. Beasley, "An Algorithm for the Two-Dimensional Assortment Problem".

Ao longo do texto, PPL representa um problema em programação linear. O texto se inicia com um estudo teórico sobre as funções mochila, sobre o qual todo o restante do trabalho se baseia. As funções mochila em duas dimensões com e sem restrições são largamente estudadas. O problema do corte de chapas e o seu relacionamento com o da escolha de estoques também é visto.

Abstract of Thesis presented to COPPE as partial fulfillment of the requirements for the degree of Master of Science (M. Sc.)

Título em inglês
Combinatoric Methods to Solve
the Cutting Stock Problems
Almir Torquato Dantas Filho
May, 1992

Thesis Supervisor: Nelson Maculan Filho
Department: Programa de Engenharia de Sistemas e Computação

The thesis contains theoretical discussion of the theme that many researches have written about. The first chapter is based upon the paper "The Theory and Computation of Knapsack Functions", by P. C. Gilmore and R. E. Gomory. The second chapter is based upon the paper by J. C. Herz titled "Recursive Computational Procedure for Two-Dimensional Stock Cutting". The third chapter is based upon the papers "A linear Programming Approach to the Cutting Stock Problem" and "A linear Programming Approach to the Cutting Stock Problem - Part II", by P. C. Gilmore and R. E. Gomory. The paper "A New Linear Programming Approach to the Cutting Stock Problem" by Haral Dyckhoff and the book about Linear Programming written by Va sek Chvátal were utilised, as well. The fourth chapter is based upon the paper "Multistage Cutting Stock Problems of Two and More Dimensions" by P. C. Gilmore and R. E. Gomory. The fifth chapter is based upon the paper by Nicos Christofides and Charles Whitlock titled "An Algorithm for Two-Dimensional Cutting Problems". The sixth chapter is based upon the papers "Two Algorithms for Constrained Two-Dimensional Cutting Stock Problems" by P. F. Wang, and "An Improved version of Wang's algorithm for two-dimensional cutting problems" by José Fernando Oliveira and José Soeiro Ferreira. The seventh chapter is based upon the papers "The Discrete Two-Dimensional Assortment Problem" by David Pentico, and "Minimum Concave Cost Flows in Certain Networks" by Willard I. Zangwill, and "Selecting the Best Lengths to Stock" by M. L. Wolfson. Finally, the last chapter is based upon the paper "An algorithm for the Two-Dimensional Assortment Problem" by J. E. Beasley.

In the text, PPL means a linear programming problem. The thesis begins with a theoretical study about knapsack functions that gives a basis for the rest of the work. The constrained and unconstrained two-dimensional knapsack functions are studied in detail. The cutting stock problem and its relation with the assortment problem are discussed too.

Índice

I	A Teoria e Computação das Funções Mochila	2
I.1	Fundamentos Teóricos	5
I.2	Discussão Teórica de Funções Mochila de uma Dimensão	11
II	O Problema da Mochila em Duas Dimensões sem Restrições	25
II.1	Definição do Problema	25
II.2	Discretização	25
II.3	Simetrização	27
II.4	Exclusão	29
II.5	Recursão Principal	32
II.6	Uso de Cotas	32
III	O Problema do Corte em 1 Dimensão	36
III.1	Considerações sobre o Caso Geral	36
III.2	Um Procedimento para o Caso Geral de Cortes	39
III.3	Um Procedimento para a Solução do Problema da Mochila em uma Dimensão	43
III.4	O Problema dos Cortes com Reaproveitamento	55
III.5	Um Exemplo	48
III.6	Uma Técnica para se Determinar uma Solução Inicial para o Problema de Cortes	49
IV	Problemas de Corte em Duas e Três Dimensões	51
IV.1	O Problema Geral de Cortes em Duas Dimensões	51

IV.2	O Problema de Cortes Tipo Guilhotina em Dois Estágios	53
IV.3	Formulação do Problema como um PPL em Estágios	54
IV.4	O Problema de Cortes em 3 Dimensões a 3 Estágios	55
V	O Problema da Mochila em Duas Dimensões com Restrições	57
V.1	Definição do Problema	57
V.2	Procedimento Enumerativo	58
V.3	Efeitos de Simetria	58
V.4	Efeitos na Ordenação dos Cortes	58
V.5	Cortes Normais	59
V.6	Descrição do Algoritmo Enumerativo	61
V.7	Algoritmo de Busca em Árvore	61
V.8	Métodos de Ramificação	62
V.9	Uma Rotina para Solução do Problema	63
V.10	RESUMO FINAL	68
VI	Dois Algoritmos para o Problema da Mochila em Duas Dimensões com Restrições	70
VI.1	Definições	70
VI.2	Os Algoritmos	72
VI.3	Limites para o Erro e Condições Ótimas	73
VIA	O Problema do Corte de Chapas	79
VI.5	Uma Nova Proposta Melhorada	80
VII	Problema de Alocação Discreta em Duas Dimensões	83
VII.1	Introdução	83
VII.2	Formulação do Problema	84
VII.3	Representação em Redes	85
VII.4	Padrões de Estocagem Aninhados	87
VII.5	Características da Política Ótima	89

VII.6 Resultados Computacionais	91
VII.70 Problema do Custo Mínimo em Redes para uma Função Côncava	92
VII.80 Método de Wolfson para o Problema Unidimensional	96
VIII Algoritmo para o Problema de Alocação em duas Dimensões	97
VIII. Introdução	97
VIII. Formulação do Problema	97
VIII. Os Algoritmos Utilizados	100
VIII.3. Seleção dos Padrões de Corte	100
VIII.3. Seleção dos Retângulos do Estoque	104
IX CONCLUSÕES	105

Capítulo I

A Teoria e Computação das Funções Mochila

Este capítulo está baseado no paper escrito por Gilmore e Gomory referenciado em [19]. Os principais resultados e teoremas podem aí ser encontrados.

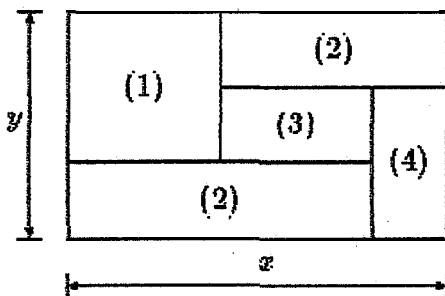
Definição de uma Função Mochila de uma dimensão:

Sejam dados $Z \ni l_i > 0$ e $0 \leq \pi_i \in R, \forall i, i = 1, \dots, m$.
 $F: R_+ \mapsto R$
 $F(x) = \max\{\pi_1 z_1 + \pi_2 z_2 + \dots + \pi_m z_m; l_1 z_1 + l_2 z_2 + \dots + l_m z_m \leq x \text{ e } z_i \geq 0, z_i \in Z\}$

Definição de uma Função Mochila Geral de duas dimensões:

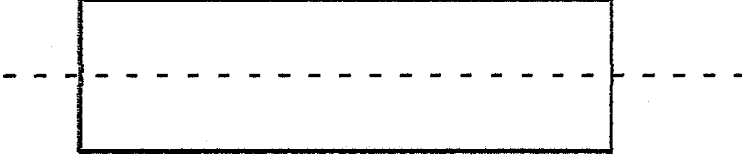
Sejam dados $(l_i, w_i) > (0, 0)$ e $\pi_i \geq 0, \forall i, i = 1, \dots, m$.
 $G: R_+ \times R_+ \mapsto R$
 $G(x, y) = \max\{\pi_1 z_1 + \pi_2 z_2 + \dots + \pi_m z_m; \exists \text{ uma forma de se cortar o retângulo com dimensões } (x, y) \text{ de modo a termos } z_1 \text{ retângulos com dimensões } (l_1, w_1), z_2 \text{ retângulos com dimensões } (l_2, w_2), \dots, z_m \text{ retângulos com dimensões } (l_m, w_m)\}$

Exemplo:



$$G(x, y) \geq \pi_1 + 2\pi_2 + \pi_3 + \pi_4$$

Consideraremos apenas um tipo especial de corte chamado de corte tipo guilhotina, definição adotada somente para retângulos. É um corte realizado de um lado ao outro diametralmente oposto e paralelo aos outros dois lados, conforme a figura:



Daqui para frente, trabalharemos com a função F abaixo:

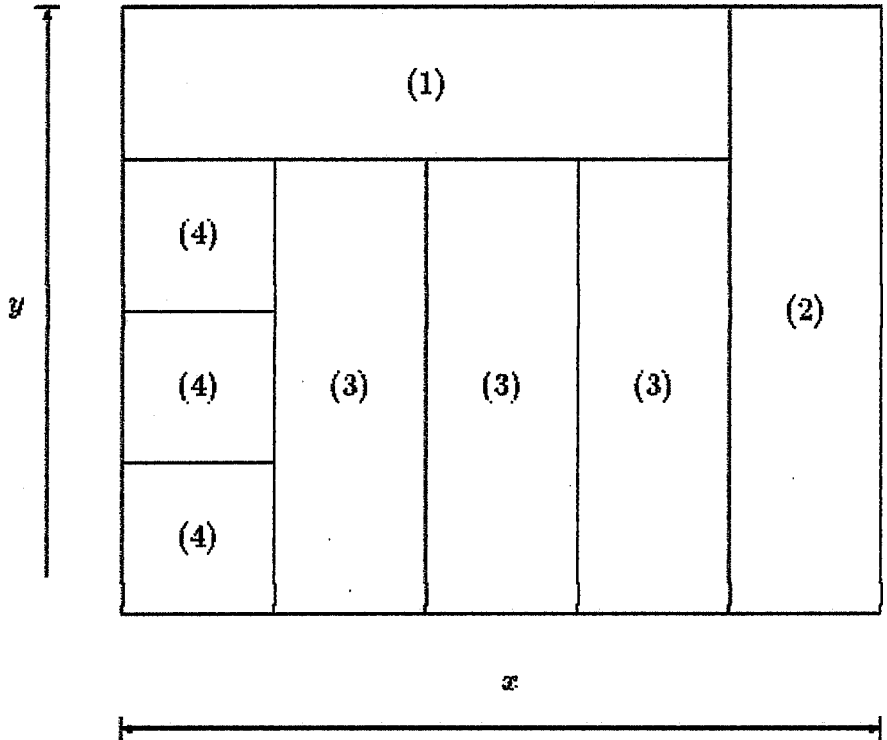
Sejam dados $(l_i, w_i) > (0, 0)$ e $\pi_i \geq 0$, $\forall i, i = 1, \dots, m$.

$$F : R_+ \times R_+ \mapsto R$$

$F(x, y) = \max\{\pi_1 z_1 + \pi_2 z_2 + \dots + \pi_m z_m; \exists \text{ uma forma guilhotina de se cortar o retângulo com dimensões } (x, y) \text{ de modo a termos } z_1 \text{ retângulos com dimensões } (l_1, w_1), z_2 \text{ retângulos com dimensões } (l_2, w_2), \dots, z_m \text{ retângulos com dimensões } (l_m, w_m)\}$

Exemplo:

$$F(x, y) \geq 3\pi_4 + 3\pi_3 + \pi_2 + \pi_1$$



Afirmação:

$$F(x_1 + x_2, y) \geq F(x_1, y) + F(x_2, y) \text{ e } F(x, y_1 + y_2) \geq F(x, y_1) + F(x, y_2).$$

A demonstração é óbvia.

Dizemos, neste caso, que a função F satisfaz a uma propriedade distributiva.

Assumiremos as seguintes hipóteses para o restante do capítulo:

1. $x, y, l_i, w_i \in Z, \forall i, i = 1, \dots, m$;
2. $F : Z_+ \times Z_+ \mapsto R$ é uma função mochila;
3. Caso $l_i \notin Z \vee w_i \notin Z$, podemos escolher $\delta > 0$, e então substituímos l_i por $l_i/\delta \in Z$ e w_i por $w_i/\delta \in Z$;

I.1 Fundamentos Teóricos

As funções mochila satisfazem as seguintes desigualdades:

$$F(x, y) \geq 0 \tag{I.1}$$

$$\begin{cases} F(x_1 + x_2, y) \geq F(x_1, y) + F(x_2, y) \wedge \\ F(x, y_1 + y_2) \geq F(x, y_1) + F(x, y_2) \end{cases} \tag{I.2}$$

$$F(l_i, w_i) \geq \pi_i \quad (i = 1, \dots, m) \tag{I.3}$$

Teorema I.1 *F é a função mochila definida por retângulos (l_i, w_i) com valores π_i , $i = 1, \dots, m$, se e somente se, F satisfaz I.1, I.2, I.3 e:*

VG satisfazendo I.1, I.2, I.3, temos

$$F(x, y) \leq G(x, y) \quad \forall x, y \tag{I.4}$$

Demonstração

(\implies)

F satisfaz I.1, I.2 e I.3. Óbvio.

Queremos mostrar I.4. Dado (x, y) seja K o número de cortes tipo guilhotina que serão necessários para se chegar à solução ótima que define $F(x, y)$. A prova de I.4 será feita por indução sobre K . Representaremos a solução ótima por $a^* = (z_1^*, z_2^*, \dots, z_m^*)$.

Para $K = 0$, significa que não foi realizado nenhum corte. Assim sendo, devemos ter uma das seguintes situações:

(a) $\exists i; z_i^* = 1 \wedge (x, y) = (l_i, w_i) \wedge z_1^* = z_2^* = \dots = z_{i-1}^* = z_{i+1}^* = \dots = z_m^* = 0$. Neste caso, $F(x, y) = \pi_i$. Como G satisfaz I.3, concluímos que $G(x, y) = G(l_i, w_i) \geq \pi_i = F(x, y)$.

(b) O retângulo de dimensões $(x, y) \not\supset$ qualquer retângulo de dimensões $(l_i, w_i) \implies z_1^* = z_2^* = \dots = z_m^* = 0$. Logo, $F(x, y) = F(0, 0)$. Por I.1 e I.2, tiramos:

- $F(0, 0) \geq 0$
- $F(0 + 0, 0) \geq F(0, 0) + F(0, 0) \implies F(0, 0) \geq F(0, 0) + F(0, 0) \implies F(0, 0) \leq 0$

O que implica em $F(0, 0) = 0$. Como $G(x, y) \geq 0$ (G satisfaz I.1), temos:

$$G(x, y) \geq 0 = F(0, 0) = F(x, y) \implies G(x, y) \geq F(x, y)$$

Suponhamos o resultado válido para um certo K . Isto é

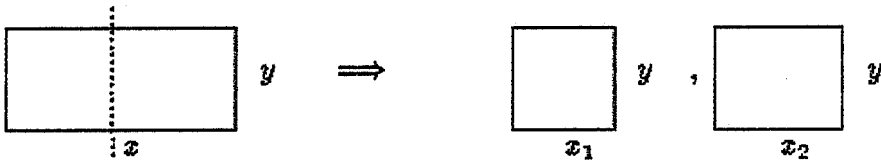


Figura I.1: O retângulo depois do corte

"Se um retângulo com dimensões (a, b) necessita de no máximo K cortes, $K \geq 0$, para fornecer a solução ótima que dá $F(a, b)$, então $G(a, b) \geq F(a, b)$ "

Vamos demonstrar o resultado para $K + 1$, isto é, estamos diante da seguinte situação: um retângulo com dimensões (x, y) necessita de $K + 1$ cortes, $K \geq 0$, para fornecer a solução ótima do problema de máximo. Queremos provar que $G(x, y) \geq F(x, y)$.

Como $K + 1 \geq 1$ pois $K \geq 0$, segue-se que pelo menos um corte foi realizado. Sem perda de generalidade, podemos supor que houve o primeiro corte no lado com dimensão x , deixando x_1 e x_2 tais que $x = x_1 + x_2$. Assim, com os cortes são do tipo guilhotina, nós tivemos (após o primeiro corte) a divisão do retângulo (x, y) como mostrada na figura I.1

Os retângulos (x_1, y) e (x_2, y) poderão ser posteriormente subdivididos em outros retângulos até atingirmos o ponto ótimo, após K cortes ($K \geq 0$). Então, consideremos $z^1 = (z_1^1, z_2^1, \dots, z_m^1)$ a solução ótima para o problema de cortes de (x_1, y) , ou seja, $F(x_1, y) = \pi_1 z_1^1 + \pi_2 z_2^1 + \dots + \pi_m z_m^1$. E também, $z^2 = (z_1^2, z_2^2, \dots, z_m^2)$ a solução ótima para o problema de cortes de (x_2, y) , ou seja, $F(x_2, y) = \pi_1 z_1^2 + \pi_2 z_2^2 + \dots + \pi_m z_m^2$.

Afirmção:

$z_i^* = z_i^1 + z_i^2 \forall i \in \{1, 2, \dots, m\}$. De fato? o primeiro corte na solução ótima para o retângulo (x, y) é o dado na figura I.1, perpendicular ao lado x , dividindo-o em x_1 e x_2 .

$$\begin{aligned} \text{Temos: } F(x_1, y) + F(x_2, y) &= \pi_1 z_1^1 + \pi_2 z_2^1 + \dots + \pi_m z_m^1 + \pi_1 z_1^2 + \pi_2 z_2^2 + \dots \\ &+ \pi_m z_m^2 = \pi_1 (z_1^1 + z_1^2) + \pi_2 (z_2^1 + z_2^2) + \dots + \pi_m (z_m^1 + z_m^2) = \sum_{i=1}^m \pi_i (z_i^1 + z_i^2) = \\ &= \sum_{i=1}^m \pi_i z_i^* = F(x, y) \end{aligned}$$

Como as soluções ótimas para (x_1, y) e (x_2, y) foram obtidas com no máximo K cortes, segue-se, por hipótese de indução: $G(x_1, y) \geq F(x_1, y)$ e $G(x_2, y) \geq F(x_2, y)$. Assim, por I.2 e pela igualdade acima: $G(x, y) = G(x_1 + x_2, y) \geq G(x_1, y) + G(x_2, y) \geq F(x_1, y) + F(x_2, y) = F(x, y)$.

(\Leftarrow)

Sejam F_1 e F_2 duas funções tais que:

1. F_1 satisfaz I.1, I.2, I.3 e I.4
2. F_2 é a função mochila definida pelos retângulos (l_i, w_i) com valores π_i , $i = 1, 2, \dots, m$.

Como F_2 satisfaz I.1, I.2 e I.3, e F_1 satisfaz I.4, temos $F_1 \leq F_2$ (*).

Pelo que foi demonstrado em (\implies) devemos ter $F(2)$ satisfaz I.4. Assim, como F_1 também satisfaz I.1, I.2 e I.3, segue-se que $F_2 \leq F_1$ (**).

Por (*) e (**) vem: $F_1 = F_2$.

[C.Q.D]

Um ponto (x, y) é dito ser uma fonte para uma função G se as seguintes condições forem satisfeitas:

1. $\exists x_1, x_2 > 0; x = x_1 + x_2$ e $G(x_1 + x_2, y) = G(x_1, y) + G(x_2, y)$
2. $\exists y_1, y_2 > 0; y = y_1 + y_2$ e $G(x, y_1 + y_2) = G(x, y_1) + G(x, y_2)$

Teorema I.2 *Seja uma função F que satisfaz I.1 e I.2. Suponhamos que os retângulos $(l_1, w_1), (l_2, w_2), \dots, (l_m, w_m)$ sejam fontes de F no intervalo $[0, L] \times [0, W]$. Então:*

$$F : ([0, L] \cap Z) \times ([0, W] \cap Z) \mapsto R$$

$$(x, y) \rightsquigarrow F(x, y)$$

é a função mochila definida para $(l_1, w_1), (l_2, w_2), \dots, (l_m, w_m)$ com valores $\pi_1 = F(l_1, w_1), \pi_2 = F(l_2, w_2), \dots, \pi_m = F(l_m, w_m)$

Demonstração

Pelo Teorema anterior, só falta demonstrar I.4. Seja G satisfazendo I.1, I.2 e I.3. Suponhamos que, por absurdo, existam x, y^1 ; $G(x, y) < F(x, y)$. Necessariamente, temos $x \neq 0$ e $y \neq 0$, pois:

- $F(x, 0) = F(x, 0 + 0) \stackrel{I.2}{\geq} F(x, 0) + F(x, 0) \Rightarrow F(x, 0) \leq 0 \stackrel{I.1}{\implies} F(x, 0) = 0$
- $F(0, y) = F(0 + 0, y) \stackrel{I.2}{\geq} F(0, y) + F(0, y) \Rightarrow F(0, y) \leq 0 \stackrel{I.1}{\implies} F(0, y) = 0$

Assim, $G(x, 0) = F(x, 0) = 0$ e $G(0, y) = F(0, y) = 0 \forall x, y$. Portanto, existe um $X_0 = (x_0, y_0)$ satisfazendo:

- $x_0, y_0 \neq 0$;

¹ $x, y \in Z$, pois só trabalhamos com valores inteiros. Ver hipóteses à página 4.

- $G(x_0, y_0) < F(x_0, y_0)$;
- Se x e y são tais que $G(x, y) < F(x, y)$ então

$$\| (x, y) - (0, 0) \| \geq \| (x_0, y_0) - (0, 0) \|;$$

Esta última condição significa que (x_0, y_0) é o ponto mais próximo da origem, dentre aqueles tais que $G(x, y) < F(x, y)$.

Assim, (x_0, y_0) também satisfaz: $x < x_0 \Rightarrow G(x, y_0) \geq F(x, y_0) \wedge y < y_0 \Rightarrow G(x_0, y) \geq F(x_0, y)$.

Afirmção:

(x_0, y_0) não pode ser uma fonte para F . De fato, se fosse, existiria um i tal que $(x_0, y_0) = (l_i, w_i)$, pois, por hipótese,

CONJUNTO DAS FONTES DE $F = \{(l_i, w_i); i = 1, 2, \dots, m\}$

Nesse caso, teríamos $G(l_i, w_i) = G(x_0, y_0) < F(x_0, y_0) = F(l_i, w_i) = \pi_i$, contrariando o fato de que G satisfaz I.3.

Assim, pelo menos uma das duas:

- (a) $\exists \exists x_1, x_2 > 0$ tais que $x_0 = x_1 + x_2 \wedge F(x_0, y_0) = F(x_1 + x_2, y_0) = F(x_1, y_0) + F(x_2, y_0)$;
- (b) $\exists \exists y_1, y_2 > 0$ tais que $y_0 = y_1 + y_2 \wedge F(x_0, y_0) = F(x_0, y_1 + y_2) = F(x_0, y_1) + F(x_0, y_2)$;

Sem perda de generalidade, podemos supor válida a 2ª hipótese. Assim, de acordo com a definição de (x_0, y_0) , segue-se que $F(x_0, y_1) \leq G(x_0, y_1)$ e $F(x_0, y_2) \leq G(x_0, y_2)$ enquanto $G(x_0, y_0) < F(x_0, y_0)$. O que implica em:

$$G(x_0, y_0) < F(x_0, y_0) = F(x_0, y_1) + F(x_0, y_2) \leq G(x_0, y_1) + G(x_0, y_2),$$

contradizendo o fato de que G satisfaz I.2.

[C.Q.D.]

Vamos supor que o domínio de uma certa função mochila seja um quadrado finito contendo a origem. Pelo teorema I.1, podemos estabelecer um PPL para calcular $F(x, y)$ no domínio em que nossa função mochila F está definida. Denotemos por D o domínio de F , e (l_i, w_i) , $i = 1, 2, \dots, m$, seus pontos fontes.

$$\text{minimizar } \sum_{(x,y) \in D} G(x,y)$$

$$\begin{aligned} \text{sujeito a } & G(x,y) \geq 0, G(l_i, w_i) \geq \pi_i \quad \forall i, i = 1, \dots, m \\ & G(x_1 + x_2, y) \geq G(x_1, y) + G(x_2, y) \\ & G(x, y_1 + y_2) \geq G(x, y_1) + G(x, y_2) \\ & \forall x, y, x_1, x_2, y_1, y_2; (x, y), (x_1, y), (x_2, y), (x, y_1), (x, y_2), \\ & \quad (x_1 + x_2, y), (x, y_1 + y_2) \in D \end{aligned}$$

Este é um PPL, pois as variáveis cujos valores serão determinados de modo a resolvermos o problema, correspondem aos $G(x, y)$.

A pergunta é: existe solução para este problema e ela nos dá $F(x, y)$, que é a função mochila associada aos retângulos (l_i, w_i) ? A resposta é sim, pois o PPL acima sempre admite como solução viável a própria F mochila. E como, pelo teorema I.1, ela minimiza a função objetivo, segue-se o resultado.

A seguir, mostraremos uma maneira de se calcular $F(x, y)$ para uma série de retângulos (l_i, w_i) , $i = 1, \dots, m$ dados, usando programação dinâmica.

Teorema L3 *Sejam $\pi_j, (l_j, w_j)$ dados, $j \in Z$. Existe uma única função que é solução para a equação abaixo:*

$$\begin{aligned} F(x, y) = \max\{ & F_0(x, y), F(x_1, y) + F(x_2, y), F(x, y_1) + \\ & F(x, y_2); x \geq x_1 + x_2, 0 < x_1 \leq x_2, \\ & y \geq y_1 + y_2 \wedge 0 < y_1 \leq y_2 \} \end{aligned} \quad (L5)$$

onde

$$F_0(x, y) = \max\{0, \pi_j; l_j \leq x \wedge w_j \leq y, j = 1, \dots, m\} \quad (L6)$$

A função que é solução para a equação acima é a mochila correspondente a $\pi_j, (l_j, w_j)$, $j = 1, \dots, m$.

Demonstração

Para mostrarmos que a função F é a função mochila associada aos tamanhos (l_i, w_i) e aos pesos π_i , devemos antes provar que:

(2.1) F satisfaz I.1.

$F(x, y) \geq F_0(x, y) \geq 0$, esta última desigualdade vem da definição de $F_0(x, y)$.

(2.2) F satisfaz I.2.

Sejam x' e x'' tais que $(x', y), (x'', y)$ e $(x' + x'', y) \in D_F$, para algum y . Vamos analisar os seguintes casos:

(2.2.1) $x' = 0 \vee x'' = 0$. Sem perda de generalidade, podemos supor $x'' = 0$.

Assim:

$$\begin{aligned} F(x' + x'', y) &= F(x' + 0, y) = F(x', y) = F(x', y) + 0 = \\ F(x', y) + F(0, y) &= F(x', y) + F(x'', y) \geq F(x', y) + F(x'', y) \end{aligned}$$

(2.2.2) $a', x'' > 0$. Assim,

$$\begin{aligned} & ((x', x'') \vee (x'', x')) \in \{(x_1, x_2); \\ & 0 < x_1 \leq x_2 \wedge x_1 + x_2 \leq x' + x''\} \end{aligned}$$

$$\text{Logo, } F(x' + x'', y) \geq F(x', y) + F(x'', y)$$

Analogamente, sejam y' e y'' tais que (x, y') , (x, y'') e $(x, y' + y'') \in D_F$, para algum x . Vamos analisar os seguintes casos:

(2.2.3) $y' = 0 \vee y'' = 0$. Sem perda de generalidade, podemos supor $y'' = 0$.

$$\begin{aligned} F(x, y' + y'') &= F(x, y' + 0) = F(x, y') = \\ F(x, y') + 0 &= F(x, y') + F(x, 0) = F(x, y') + \\ &F(x, y'') \geq F(x, y') + F(x, y'') \end{aligned}$$

(2.2.4) $y', y'' > 0$. Assim,

$$\begin{aligned} & ((y', y'') \vee (y'', y')) \in \{(y_1, y_2); \\ & 0 < y_1 \leq y_2 \wedge y_1 + y_2 \leq y' + y''\}. \end{aligned}$$

$$\text{Logo, } F(x, y' + y'') \geq F(x, y') + F(x, y'').$$

Pelo teorema VII.2, a função F obtida por programação dinâmica é a função mochila associada aos ~~sem~~ pontos fonte. Sejam $\mathcal{A} =$ conjunto dos pontos fontes de F em um certo domínio $D \supset \{(l_j, w_j), j = 1, \dots, m\}$ e $F' =$ função mochila associada aos tamanhos (l_j, w_j) com valores π_j . Querem mostrar $F = F', \forall (x, y) \in D$.

É evidente que F satisfaz I.3, pois $F(l_j, w_j) \geq F_0(l_j, w_j) \geq \pi_j$, pela definição de F_0 . Logo, como F' satisfaz I.4, $F(x, y) \geq F'(x, y), \forall (x, y) \in D$ (*).

Seja $(x, y) \in \mathcal{A}$ (vide observação abaixo). Temos os seguintes casos a considerar:

- (i) existe algum j com retângulo $(l_j, w_j) \subset$ retângulo (x, y) . Neste caso, temos, então, $F'(x, y) \geq \pi_j$ para j retângulo máximo contido. Como $(x, y) \in \mathcal{A}$, segue-se que $F(x, y) = F_0(x, y) = \pi_j$, pois $(l_j \leq x \wedge w_j \leq y)$. Donde se conclui que $F'(x, y) \geq F(x, y)$.
- (ii) $\nexists j$; retângulo $(l_j, w_j) \subset$ retângulo (x, y) . Assim, $F'(x, y) = 0$. Como $(x, y) \in \mathcal{A}$, segue-se que $F(x, y) = F_0(x, y) = 0$, pois $(l_j > x \vee w_j > y), \forall j, j = 1, \dots, m$. Donde se conclui que $F'(x, y) = 0 = F(x, y) \Rightarrow F'(x, y) \geq F(x, y)$.

Assim, se considerarmos os pontos fontes de F e a própria F mochila veremos que, em relação a esses dois, F' satisfaz I.3. Como F é a mochila, ela deve satisfazer I.4 e, portanto, $F'(x, y) \geq F(x, y), \forall (x, y) \in D$. (**)

Por (*) e (**), $F = F'$, e o resultado está provado.

Observação: Nós usamos constantemente o fato de que, se $(x, y) \in A$ então $F(x, y) = F_0(x, y)$. Suponhamos, por absurdo, não ser este o caso. Assim, das duas uma:

- (i) $\exists \exists x_1, x_2, 0 < x_1 \leq x_2, x_1 + x_2 \leq x; F(x, y) = F(x_1, y) + F(x_2, y);$
 (ii) $\exists \exists y_1, y_2, 0 < y_1 \leq y_2, y_1 + y_2 \leq y; F(x, y) = F(x, y_1) + F(x, y_2).$

Sem perda de generalidade, vamos supor a primeira.

Se $x_1 + x_2 = x$, então x não poderia ser um ponto fonte de F . Logo, $x_1 + x_2 < x$. Sejam, então, $x'_1 \geq x_1$ e $x'_2 \geq x_2$ tal que $x'_1 + x'_2 = x$. Assim, $F(x'_1, y) \geq F(x_1, y)$ e $F(x'_2, y) \geq F(x_2, y)$.

Temos as seguintes possibilidades:

- (i) $F(x'_1, y) > F(x_1, y) \vee F(x'_2, y) > F(x_2, y)$. Nesse caso, não poderíamos ter:

$$\max\{F_0(x, y), F(x_1, y) + F(x_2, y), F(x, y_1) + F(x, y_2); x \geq x_1 + x_2, 0 < x_1 \leq x_2, y \geq y_1 + y_2 \wedge 0 < y_1 \leq y_2\} = F(x_1, y) + F(x_2, y)$$

- (ii) $F(x'_1, y) = F(x_1, y) \wedge F(x'_2, y) = F(x_2, y)$. Nesse caso, $x \notin A$ (absurdo!).

Portanto: $\exists x_1, x_2, 0 < x_1 \leq x_2, x_1 + x_2 \leq x; F(x, y) = F(x_1, y) + F(x_2, y)$. Temos: $F(x, y) = F_0(x, y)$

[C.Q.D.]

I.2 Discussão Teórica de Funções Mochila de uma Dimensão

Trataremos nesta secção de funções mochila de uma dimensão, isto é aquelas definidas a partir de comprimentos l_1, l_2, \dots, l_m , cujos valores são π_1, r, \dots, r^* , pela equação:

$$F(x) = \max\{z_1\pi_1 + \dots + z_m\pi_m; Z \ni z_i \geq 0 \wedge z_1l_1 + \dots + z_ml_m \leq x\} \quad (I.7)$$

Claramente: F satisfaz a primeira forma de 1.5:

$$\begin{cases} F(x) = \max\{F_0(x), F(x_1) + F(x_2); x \geq x_1 + x_2, \\ 0 < x_1 \leq x_2\}, \text{ onde} \\ F_0(x) = \max\{0, \pi_j; l_j \leq x, j = 1, \dots, m\} \end{cases} \quad (I.8)$$

Definição:

Dizemos que a seqüência (x_1, x_2, \dots, x_r) é uma partição de x se $0 < x_1 < \dots < x_{r-1} < x_r \leq x$.

Definição:

(z_1, z_2, \dots, z_m) é uma partição realizando $F(x)$ se

$$\begin{cases} z_1 l_1 + z_2 l_2 + \dots + z_m l_m \leq x \wedge \\ F(x) = \pi_1 z_1 + \pi_2 z_2 + \dots + \pi_m z_m \end{cases}$$

Definição:

$(z_1^*, z_2^*, \dots, z_m^*)$ é a partição associada a $F(x)$ se $(z_1^*, z_2^*, \dots, z_m^*)$ é uma partição realizando $F(x)$ e, se (z_1, z_2, \dots, z_m) é uma outra partição realizando $F(x)$ então ela satisfaz $z_m \leq z_m^*$ e também:

$$\begin{aligned} z_m^* &= z_m, z_{m-1}^* = z_{m-1}, \dots, z_{r+1}^* = z_{r+1} \\ &\implies z_r^* \geq z_r, \forall r \in \{1, 2, \dots, m-1\} \end{aligned}$$

Definição

Seja F uma função mochila.

$$l_1(x) = \begin{cases} 1 & \text{se } x \leq 0 \\ \max\{1, i; z_i \neq 0\} & \text{se } x > 0, \text{ onde } (z_1, z_2, \dots, z_m) \text{ é a partição} \\ & \text{associada a } F(x) \end{cases} \quad (I.9)$$

$$l_2(x) = \max\{1, i; F(x) = F(x - l_i) + \pi_i \wedge x \geq l_i\}$$

Lema I.1 $l_1(x) = l_2(x), \forall x$

Demonstração

Se $x < \min\{l_i; i = 1, \dots, m\}$ então $l_1(x) = 1$ pois a partição associada a $F(x)$ será um vetor nulo e $l_2(x) = 1$ pois $\{i; F(x) = F(x - l_i) + \pi_i \wedge x \geq l_i\} = \emptyset$.
Se $x \geq \min\{l_i; i = 1, \dots, m\}$, seja então (z_1, z_2, \dots, z_m) a partição associada a $F(x)$.
Sejam

$$\begin{aligned} i_1 &= \max\{i; z_i \neq 0\} \\ i_2 &= \max\{i; F(x) = F(x - l_i) + \pi_i \wedge x \geq l_i\} \end{aligned}$$

Consideremos $(z'_1, z'_2, \dots, z'_m)$ a partição associada a $F(x - l_{i_2})$, isto é, $F(x - l_{i_2}) = \pi_1 z'_1 + \pi_2 z'_2 + \dots + \pi_m z'_m$ e $z'_1 l_1 + z'_2 l_2 + \dots + z'_m l_m \leq x - l_{i_2}$.

Assim, $z'_1 l_1 + z'_2 l_2 + \dots + (z'_{i_2} + 1) l_{i_2} + \dots + z'_m l_m \leq x$. Como $F(x - l_{i_2}) + \pi_{i_2} = \pi_1 z'_1 + \pi_2 z'_2 + \dots + \pi_{i_2} (z'_{i_2} + 1) + \dots + \pi_m z'_m$ e $F(x) = F(x - l_{i_2}) + \pi_{i_2}$, vemos que $(z'_1, z'_2, \dots, z'_{i_2} + 1, \dots, z'_m)$ é uma partição que realiza $F(x)$.

$i_1 < i_2, (z_1, z_2, \dots, z_m)$ é a partição associada a $F(x) \Rightarrow z'_{i_2} + 1 = 0 \Rightarrow z'_{i_2} = -1 < 0$ (absurdo!)

Assuma $i_1 > i_2$. Se $r < l_{i_1}$, então $z_{i_1} = 0$, o que é absurdo. Logo $x \geq l_{i_1}$ e temos que analisar as duas possibilidades:

1. $F(x - l_{i_1}) + \pi_{i_1} > F(x)$, absurdo pela definição de $F(x)$
2. $F(x) > F(x - l_{i_1}) + \pi_{i_1} \Rightarrow z_{i_1} = 0$, pois $z_{i_1} \geq 1 \Rightarrow F(x) = F(x - l_{i_1}) + \pi_{i_1}$.
Absurdo, pois z_{i_1} não pode ser nulo.

Assim, $F(x) = F(x - l_{i_1}) + \pi_{i_1} \Rightarrow i_1 \in \{i; F(x) = F(x - l_i) + \pi_i \wedge x \geq l_i\}$.
Portanto, $i_1 \leq i_2$. Absurdo!

Dessa forma, $i_1 = i_2 \Rightarrow l_1(x) = l_2(x)$.

[C.Q.D.]

Daqui para frente, chamaremos simplesmente $l(x) = l_1(x) = l_2(x)$.

Teorema I.4 A função mochila $F(x)$ em uma dimensão é definida pela equação

$$F(x) = \max\{0, F(x - l_i) + \pi_i; i \geq l(x - l_i), x \geq l_i; \forall i = 1, 2, \dots, m\} \quad (\text{I.10})$$

Demonstração

Se $x < \min\{l_i; i = 1, 2, \dots, m\}$ então $\{i; i \geq l(x - l_i), x \geq l_i\} = \emptyset$ e o segundo membro da equação em I.10 é igual a zero, que é o mesmo valor de $F(x)$.

Suponhamos, então, $x \geq \min\{l_i; i = 1, 2, \dots, m\}$. Seja (z_1, z_2, \dots, z_m) a partição associada a $F(x)$ e $r = l(x)$. Assim:

$$\begin{cases} F(x) = \pi_1 z_1 + \pi_2 z_2 + \dots + \pi_r z_r, \\ z_r > 0 \wedge z_{r+1} = \dots = z_m = 0 \end{cases}$$

Logo, concluímos:

$$\begin{aligned} F(x) &= F[(x - l_r) + l_r] \geq F(x - l_r) + F(l_r) \geq F(x - l_r) + \pi_r \Rightarrow \\ F(x - l_r) &\leq F(x) - \pi_r = \pi_1 z_1 + \dots + \pi_{r-1} z_{r-1} + \pi_r (z_r - 1) \end{aligned} \quad (\text{I.11})$$

Por outro lado, como $z_1 l_1 + z_2 l_2 + \dots + (z_r - 1) l_r \leq x - l_r$, segue-se que:

$$F(x - l_r) \geq \pi_1 z_1 + \pi_2 z_2 + \dots + \pi_r (z_r - 1) \quad (\text{I.12})$$

Por I.11 e I.12, tiramos $F(x - l_r) = \pi_1 z_1 + \pi_2 z_2 + \dots + \pi_r (z_r - 1)$

Afirmação:

$(z_1, z_2, \dots, z_r - 1, 0, \dots, 0)$ é a partição associada a $F(x - l_r)$. De fato, só nos resta provar que, se $(z'_1, z'_2, \dots, z'_m)$ é uma outra partição que realiza $F(x - l_r)$, ela satisfaz as condições abaixo:

(i) $(z'_m = z'_{m-1} = \dots = z'_{r+1} = 0)$;

(ii) $(z'_r \leq z_r - 1)$;

(iii) Se $z'_r = z_r - 1$; $z'_{r-1} = z_{r-1}, \dots, z'_{s+1} = z_{s+1}$ então $z'_s \leq z_s \forall s \in \{1, 2, \dots, r-1\}$.

Consideremos, então, o vetor dado por $(z'_1, z'_2, \dots, z'_r + 1, \dots, z'_m)$. Ele satisfaz as relações abaixo:

$$\begin{aligned} z'_1 l_1 + z'_2 l_2 + \dots + (z'_r + 1) l_r + \dots + z'_m l_m &= \\ (z'_1 l_1 + z'_2 l_2 + \dots + z'_r l_r + \dots + z'_m l_m) + l_r &\leq (x - l_r) + l_r = x, \\ z'_1 \pi_1 + z'_2 \pi_2 + \dots + (z'_r + 1) \pi_r + \dots + z'_m \pi_m &= F(x - l_r) + \pi_r. \end{aligned}$$

Como $r = l(x)$, pelo lema I.1, segue-se que $F(x) = F(x - l_r) + \pi_r$. Assim, a partição $(z'_1, z'_2, \dots, z'_r + 1, \dots, z'_m)$ realiza $F(x)$. Pela definição de partição associada a $F(x)$ e, levando-se em conta que $(z_1, z_2, \dots, z_r, 0, \dots, 0)$ é a partição associada a $F(x)$, podemos concluir (i), (ii) e (iii).

Portanto, $l(x - l_i) \leq r$ e $x \geq l_i$. Logo,

$$\begin{aligned} \max\{F(x - l_i) + \pi_i; i \geq l(x - l_i), x \geq l_i\} &\geq \\ F(x - l_r) + \pi_r &= F(x) \end{aligned}$$

Portanto, $\max\{0, F(x - l_i) + \pi_i; i \geq l(x - l_i), x \geq l_i\} = F(x)$

[C.Q.D.]

Daqui para frente, consideraremos a seguinte ordem para os tamanhos dados l_1, l_2, \dots, l_m :

$$\rho_1 \leq \rho_2 \leq \dots \leq \rho_m, \text{ onde } \rho_i = \pi_i / l_i, i = 1, 2, \dots, m \quad (\text{I.13})$$

Para cada par r e s , $1 \leq r \leq s \leq m$, $F_{(r,s)}$ é a função mochila associada aos tamanhos l_r, l_{r+1}, \dots, l_s cujos valores são $\pi_r, \pi_{r+1}, \dots, \pi_s$.

Teorema I.5 Para cada r , $1 \leq r \leq m-1$, existe $x_0 = x_0(r)$ tal que $\forall x \exists x_1$ e x_2 ; $F(x) = F_{(1,r)}(x_2) + F_{(r+1,m)}(x_1)$

$$x_2 < x_0 \text{ e } x_1 + x_2 \leq x \quad (\text{I.14})$$

Demonstração

Sejam x e r dados, onde $1 \leq r \leq m-1$. Para cada i , $1 \leq i \leq r$ sejam $a_i, b_i \in Z$ tal que $a_i l_i = b_i l_{r+1} =$ múltiplo comum de (l_i, l_{r+1}) .

Faça $x_0 = a_1 l_1 + \dots + a_r l_r$. Tomemos (z_1, z_2, \dots, z_m) satisfazendo

$$F(x) = \pi_1 z_1 + \pi_2 z_2 + \dots + \pi_m z_m \text{ onde}$$

$$l_1 z_1 + \dots + l_r z_r = \min\{l_1 z'_1 + \dots + l_r z'_r \text{ tal que}$$

$$(z'_1, z'_2, \dots, z'_m) \text{ é uma partição realizando } F(x)\}. \quad (*)$$

Faça $x_2 = l_1 z_1 + \dots + l_r z_r$ e $x_1 = l_{r+1} z_{r+1} + \dots + l_m z_m$

Temos $x_1 + x_2 \leq x$.

$$F(x) = F_{(1,r)}(x_2) + F_{(r+1,m)}(x_1)$$

Se $x_2 < x_0$, o teorema está provado. Suponha $x_2 \geq x_0$. Logo, $\exists i, 1 \leq i \leq r; z_i \geq a_i$.

$$a_i l_i = b_i l_{r+1} \Rightarrow l_i = b_i l_{r+1} / a_i$$

$$\rho_i \leq \rho_{r+1} \Rightarrow \frac{\pi_i}{l_i} \leq \frac{\pi_{r+1}}{l_{r+1}} \Rightarrow \frac{\pi_i}{\left(\frac{b_i l_{r+1}}{a_i}\right)} \leq \frac{\pi_{r+1}}{l_{r+1}}$$

$$\Rightarrow a_i \pi_i / b_i \leq \pi_{r+1} \Rightarrow a_i \pi_i \leq b_i \pi_{r+1}$$

Como $z_i \geq a_i$, podemos por $x'_2 = x_2 - a_i l_i = l_1 z_1 + \dots + l_r z_r - a_i l_i = l_1 z_1 + \dots + (z_i - a_i) l_i + \dots + l_r z_r$. Assim, $F(x'_2) \geq \pi_1 z_1 + \dots + \pi_i (z_i - a_i) + \dots + \pi_r z_r$. Por outro lado, fazendo $x'_1 = x_1 + b_i l_{r+1} = l_{r+1} (b_i + z_{r+1}) + l_{r+2} z_{r+2} + \dots + l_m z_m$, retiramos $F(x'_1) \geq \pi_{r+1} (b_i + z_{r+1}) + \pi_{r+2} z_{r+2} + \dots + \pi_m z_m$.

Portanto,

$$F(x'_1) + F(x'_2) \geq \pi_1 z_1 + \dots + \pi_i z_i - \pi_i a_i + \dots + \pi_r z_r +$$

$$\pi_{r+1} b_i + \pi_{r+1} z_{r+1} + \pi_{r+2} z_{r+2} + \dots + \pi_m z_m \geq F(x).$$

Logo, $F(x) = F(x'_1) + F(x'_2)$. Absurdo, pois encontramos

$$(z_1, z_2, \dots, z_i - a_i, \dots, z_r) \in \{(z'_1, z'_2, \dots, z'_r);$$

$$(z'_1, z'_2, \dots, z'_r, z'_{r+1}, \dots, z'_m) \text{ é uma partição realizando } F(x)\}$$

$$\text{com } l_1 z_1 + l_2 z_2 + \dots + l_i (z_i - a_i) + \dots + l_r z_r < l_1 z_1 + l_2 z_2$$

$$+ \dots + l_r z_r \text{ contradizendo } (*).$$

[C.Q.D.]

Consideremos o caso $r = m-1$. Segundo o teorema 1.5, existe um x_0 com o seguinte valor:

$$x_0 = a_1 l_1 + a_2 l_2 + \dots + a_{m-1} l_{m-1}, \text{ onde}$$

$$a_i l_i = b_i l_m, \forall i \in \{1, 2, \dots, m-1\}$$

Assim, este x_0 pode ser reescrito como:

$$x_0 = b_1 l_m + b_2 l_m + \cdots + b_{m-1} l_m = (b_1 + b_2 + \cdots + b_{m-1}) l_m$$

Assim: $F(x_0) = (b_1 + b_2 + \cdots + b_{m-1}) \pi_m$, pois ρ_m é de todas as densidades, a maior. Isto implica em:

$$F(x_0) = (b_1 + b_2 + \cdots + b_{m-1}) \pi_m = [(b_1 + b_2 + \cdots + b_{m-1}) - 1] \pi_m + \pi_m = F(x_0 - l_m) + \pi_m$$

Também, pelo teorema I.5, temos que para cada x existem x_1 e x_2 satisfazendo $x_1 + x_2 \leq x$, $F(x) = F_{(m,m)}(x_1) + F_{(1,m-1)}(x_2)$ e $x_2 < x_0$.

$$F(x) = F_{(m,m)}(x_1) + F_{(1,m-1)}(x_2) = [(x - x_2)/l_m] \cdot \pi_m + F_{(1,m-1)}(x_2),$$

onde $[y]$ denota o maior inteiro não excedendo y .

Seja $x > x_0$. Vamos supor que $[(x - x_2)/l_m] = 0$. Logo, $F(x) = F_{(1,m-1)}(x_2) \leq F(x_2)$. Como $x \geq x_2$ segue-se $F(x) = F(x_2)$. Já que $x_2 < x_0 < x$, concluímos: $F(x_2) = F(x_0) = F(x)$.

Porém, $F(x_0) = F(x_0 - l_m) + \pi_m \Rightarrow F(x) = F(x_0 - l_m) + \pi_m$. Assim:

$$F(x) = F[(x - l_m) + l_m] \geq F(x - l_m) + F(l_m) \geq F(x - l_m) + \pi_m \Rightarrow F(x_0 - l_m) + \pi_m \geq F(x - l_m) + \pi_m \Rightarrow F(x_0 - l_m) \geq F(x - l_m). \text{ Assim } F(x - l_m) = F(x_0 - l_m).$$

Assim: $F(x) = F(x_0) = F(x_0 - l_m) + \pi_m = F(x - l_m) + \pi_m$.

Agora suponha $[(x - x_2)/l_m] > 0$. Nesse caso, se chamarmos $z_m = [(x - x_2)/l_m]$, podemos escrever: $F(x) = z_m \pi_m + F_{(1,m-1)}(x_2) = \pi_m + (z_m - 1) \pi_m + F_{(1,m-1)}(x_2)$ (**).

Donde se conclui que:

$$\begin{aligned} x - l_m &\geq z_m l_m + x_2 - l_m = (z_m - 1) l_m + x_2 \Rightarrow F(x - l_m) \geq \\ &F[(z_m - 1) l_m + x_2] \geq F[(z_m - 1) l_m] + F(x_2) = (z_m - 1) \pi_m + F(x_2) \geq \\ &(z_m - 1) \pi_m + F_{(1,m-1)}(x_2) = F(x) - \pi_m \text{ por (**)} \end{aligned}$$

Assim, $F(x - l_m) + \pi_m \geq F(x)$. Por outro lado,

$$F(x) = F[(x - l_m) + l_m] \geq F(x - l_m) + F(l_m) \geq F(x - l_m) + \pi_m$$

De onde concluímos uma vez mais $F(x) = F(x - l_m) + \pi_m$.

Em resumo, $F(x) = F(x - l_m) + \pi_m \forall x > x_0$. Então, se nós definirmos,

$$f(x) = F(x) - [x/l_m] \pi_m, \quad (I.15)$$

teremos o seguinte, para todo x , $x > x_0$:

$$\begin{aligned} f(x) &= F(x - l_m) + \pi_m - [x/l_m] \pi_m = F(x - l_m) + \pi_m - (([x/l_m] - 1) + 1) \pi_m = \\ &F(x - l_m) + \pi_m - [(x - l_m)/l_m] \pi_m - \pi_m = F(x - l_m) - [(x - l_m)/l_m] \pi_m = f(x - l_m) \end{aligned}$$

Assim: $f : (x_0, \infty) \mapsto R$, dada por $f(x) = F(x) - [x/l_m]\pi_m$ é periódica de período $= l_m$, onde $x_0 = x_0(m-1)$ (x_0 é função de $m-1$).

Seja $p(x) = \max\{0, F(x - l_i) + \pi_i - [x/l_m]\pi_m; i \geq l(x - l_i), x \geq l_i\}$.

Afirmção:

$$p(x) = f(x) \quad \forall x > x_0.$$

Seja x qualquer fixo. Temos duas possibilidades:

1. $\exists i; i \geq l(x - l_i), x \geq l_i$. Assim, pela equação L10, $F(x) = 0$ e, pelo definido acima, $p(x) = 0$. Conseqüentemente, temos:

$$\begin{aligned} f(x) &= F(x) - \overbrace{[x/l_m]}^{=0, \text{ pois } x < l_m} \cdot \pi_m = \\ &= 0 - 0 \cdot \pi_m = 0 = p(x) \end{aligned}$$

2. $\{i; i \geq l(x - l_i) \text{ e } x \geq l_i\} \neq \emptyset$. Nesse caso, por L.10, $\exists i_0$ satisfazendo $F(x) = F(x - l_{i_0}) + \pi_{i_0}$. Assim, por definição de $F(x)$ e pela equação L.10, temos:

$$\begin{aligned} F(x) &= F(x - l_{i_0}) + \pi_{i_0} \geq [x/l_m]\pi_m \text{ e} \\ F(x - l_{i_0}) + \pi_{i_0} - [x/l_m]\pi_m &\geq F(x - l_i) + \pi_i - [x/l_m]\pi_m \\ \forall i; i &\geq l(x - l_i), x \geq l_i. \end{aligned}$$

$$\text{Logo, } p(x) = \overbrace{F(x - l_{i_0}) + \pi_{i_0}}^{=F(x)} - [x/l_m]\pi_m = f(x).$$

Isto conclui a prova. Então:

$$\begin{cases} f(x) = \max\{0, F(x - l_i) + \pi_i - [x/l_m] \cdot \pi_m; \\ \quad i \geq l(x - l_i), x \geq l_i\} \\ f(x - l_i) = F(x - l_i) - [(x - l_i)/l_m] \cdot \pi_m \text{ desde que } x - l_i > x_0 \end{cases} \Rightarrow$$

$$\begin{aligned} f(x) &= \max\{0, f(x - l_i) + [(x - l_i)/l_m] \cdot \pi_m + \pi_i - [x/l_m] \cdot \pi_m; i \geq l(x - l_i), x \geq l_i\} = \\ &= \max\{0, f(x - l_i) + \pi_i - ([x/l_m] - [(x - l_i)/l_m]) \cdot \pi_m; i \geq l(x - l_i), x \geq l_i\}, \\ &\quad \text{desde que } x - l_i > x_0 \quad \forall i; i \geq l(x - l_i) \text{ e } x \geq l_i. \end{aligned}$$

Agora, consideremos a seguinte notação:

$$\begin{aligned} \pi'_i &= \pi_i - [l_i/l_m] \cdot \pi_m \\ r(x/y) &= \text{o resto da divisão de } x \text{ por } y \\ l'_i &= r(l_i/l_m) \end{aligned}$$

Assim, podemos escrever:

$$\begin{aligned}
 & f(x - l_i) + \pi_i - \left(\left[\frac{x}{l_m} \right] - \left[\frac{x - l_i}{l_m} \right] \right) \cdot \pi_m = \\
 & f(x - l_i) + \pi'_i + \left[\frac{l_i}{l_m} \right] \cdot \pi_m - \left(\left[\frac{x}{l_m} \right] - \left[\frac{x - l_i}{l_m} \right] \right) \cdot \pi_m = \\
 & f(x - l_i) + \pi'_i + \left(\left[\frac{l_i}{l_m} \right] - \left[\frac{x}{l_m} \right] + \left[\frac{x - l_i}{l_m} \right] \right) \cdot \pi_m = f(x - l_i) + \\
 & \pi'_i - \left(\left[\frac{x}{l_m} \right] - \left[\frac{l_i}{l_m} \right] - \left[\frac{x - l_i}{l_m} \right] \right) \cdot \pi_m \stackrel{Obs}{=} \\
 & f(x - l_i) + \pi'_i - \left[\frac{x}{l_m} - \left[\frac{l_i}{l_m} \right] - \left[\frac{x - l_i}{l_m} \right] \right] \cdot \pi_m = f(x - l_i) + \\
 & \pi'_i - \left[\left(\frac{x - l_i}{l_m} + \frac{l_i}{l_m} \right) - \left[\frac{l_i}{l_m} \right] - \left[\frac{x - l_i}{l_m} \right] \right] \cdot \pi_m = \\
 & f(x - l_i) + \pi'_i - \left[\left(\frac{x - l_i}{l_m} - \left[\frac{x - l_i}{l_m} \right] \right) + \left(\frac{l_i}{l_m} - \left[\frac{l_i}{l_m} \right] \right) \right] \cdot \pi_m = f(x - l_i) + \\
 & \pi'_i - \left[\frac{r(\frac{x-l_i}{l_m})}{l_m} + \frac{r(\frac{l_i}{l_m})}{l_m} \right] \cdot \pi_m = f(x - l_i) + \pi'_i - \left[\frac{r(\frac{x-l_i}{l_m}) + l'_i}{l_m} \right] \cdot \pi_m
 \end{aligned}$$

Obs.: $[x + z] = [x] + z$ desde que $z \in \mathbb{Z}$.

Donde se conclui que:

$$\begin{aligned}
 f(x) &= \max\{0, f(x - l_i) + \pi'_i - \left[\frac{r(\frac{x-l_i}{l_m}) + l'_i}{l_m} \right] \cdot \pi_m; x \geq l_i, i \geq l(x - l_i)\}, \\
 &\text{desde que } x - l_i > x_0 \forall i \text{ tal que } i \geq l(x - l_i) \text{ e } x \geq l_i.
 \end{aligned}$$

A função f é definida para toda x , porém, é mais interessante nos concentrarmos apenas em um intervalo correspondente a um período. Ou seja, estaremos interessados na função abaixo:

$$\begin{aligned}
 \varphi &: [0, l_m) \mapsto \mathbb{R} \\
 \varphi(l) &= f(Nl_m + l), \text{ onde } N \text{ é um inteiro satisfazendo } Nl_m > x_0
 \end{aligned} \tag{I.16}$$

Lema I.2 *Suponhamos que $\{i; i \geq l(x - l_i) \wedge x \geq l_i\} \neq \emptyset$. Então, podemos concluir que:*

$$\begin{aligned}
 f(x) &= \max\{f(x - l_i) + \pi'_i - \left[\frac{r(\frac{x-l_i}{l_m}) + l'_i}{l_m} \right] \cdot \pi_m; \\
 &\quad i \geq l(x - l_i) \text{ e } x \geq l_i\} \\
 &\text{desde que } x - l_i > x_0 \forall i; i \geq l(x - l_i) \text{ e } x \geq l_i.
 \end{aligned}$$

Demonstração

É só acompanhar o raciocínio desenvolvido antes do lema.

Lema I.3 Se $x \geq Nl_m$, então $f(x) = \varphi(r(\frac{x}{l_m}))$.

Demonstração

Dado x , podemos escrever:

$$x = \underbrace{[\frac{x}{l_m}] \cdot l_m}_{\downarrow} + r(\frac{x}{l_m}).$$

Se $x > Nl_m$ então $N = [\frac{x}{l_m}]$ e, dessa forma, colocamos:

$$\begin{aligned} x &= Nl_m + ([\frac{x}{l_m}] - N) \cdot l_m + r(\frac{x}{l_m}) \Rightarrow \\ f(x) &= f(Nl_m + r(\frac{x}{l_m})) + ([\frac{x}{l_m}] - N) \cdot l_m = \\ &= f(Nl_m + r(\frac{x}{l_m})) = \varphi(r(\frac{x}{l_m})) \end{aligned}$$

[C.Q.D.]

Se $x \geq Nl_m$, as condições do Lema I.2 acima são satisfeitas. Assim:

$$\begin{aligned} f(x) &= \max\{f(x - l_i) + \pi'_i - [\frac{r(\frac{x-l_i}{l_m}) + l'_i}{l_m}] \cdot \pi_m; \\ &\quad i \geq l(x - l_i) \text{ e } x \geq l_i\} \text{ desde que } \\ &\quad x - l_i > x_0 \forall i; i \geq l(x - l_i) \text{ e } x \geq 0 \end{aligned}$$

Se também tivermos $x - l_i \geq Nl_m$ podemos por, pelo lema I.3, $f(x - l_i) = \varphi(r(\frac{x-l_i}{l_m}))$ e $f(x) = \varphi(r(\frac{x}{l_m}))$. Assim, finalmente chegamos a:

$$\begin{aligned} \varphi(r(\frac{x}{l_m})) &= \max\{\varphi(r(\frac{x-l_i}{l_m})) + \pi'_i - [\frac{r(\frac{x-l_i}{l_m}) + l'_i}{l_m}] \cdot \pi_m; \quad (I.17) \\ &\quad i \geq l(x - l_i) \text{ e } x \geq \dots\} \text{ desde que } x - l_i \geq Nl_m \forall i; i \geq l(x - l_i) \text{ e } x \geq l_i. \end{aligned}$$

As equações I.15, I.16 e I.17 foram obtidas através da análise do caso $r = m - i$. Daqui para frente, nós refaremos essa análise redefinindo de uma outra forma (um pouco diferente) as funções anteriores.

$$\bar{f}(x) = F(x) - (x/l_m)\pi_m, \quad \forall x, x > x_0 \quad (I.18)$$

Como $F(x) = F(x - l_m) + \pi_m, \forall x, x > x_0$, teremos:

$$\bar{f}(x) = F(x - l_m) + \pi_m - (\frac{x}{l_m}) \cdot \pi_m = \bar{f}(x - l_m)$$

Assim: $\bar{f} : (x_0, \infty) \mapsto R$, dada por $\bar{f}(x) = F(x) - (x/l_m)\pi_m$ é periódica de período igual a l_m . Observação: $x_0 = x_0(m-1)$ (x_0 é função de $m-1$).

Seja $\bar{p}(x) = \max\{-(x/l_m)\pi_m, F(x-l_i) + \pi_i - (x/l_m)\pi_m; i \geq l(x-l_i), x \geq l_i\}$. Queremos mostrar que $\bar{p}(x) = \bar{f}(x), \forall x, x > x_0$.

Seja x qualquer fim. Temos duas possibilidades:

1. $\nexists i; i \geq l(x-l_i), e \geq l_i$. Assim, pela equação I.10, $F(x) = 0$ e, pelo definido acima, $\bar{p}(x) = -(x/l_m)\pi_m$. Conseqüentemente, temos:

$$\bar{f}(x) = F(x) - (x/l_m)\pi_m = 0 - (x/l_m)\pi_m = -(x/l_m)\pi_m = \bar{p}(x)$$

2. $\{i; i \geq l(x-l_i) \wedge x \geq l_i\} \neq \emptyset$. Nesse caso, por I.10, $\exists i_0$ satisfazendo $F(x) = F(x-l_{i_0}) + \pi_{i_0}$. Assim, de acordo com a equação I.10, temos:

$$F(x-l_{i_0}) + \pi_{i_0} - (x/l_m)\pi_m \geq F(x-l_i) + \pi_i - (x/l_m)\pi_m \\ \forall i; i \geq l(x-l_i), x \geq l_i$$

$$\text{Assim: } \bar{p}(x) = F(x-l_{i_0}) + \pi_{i_0} - (x/l_m)\pi_m = F(x) - (x/l_m)\pi_m = \bar{f}(x)$$

Isto conclui a prova. Então:

$$\left\{ \begin{array}{l} \bar{f}(x) = \max\{-(x/l_m)\pi_m, F(x-l_i) + \pi_i - (x/l_m)\pi_m; \\ i \geq l(x-l_i), x \geq l_i\} \\ \bar{f}(x-l_i) = F(x-l_i) - ((x-l_i)/l_m)\pi_m \\ \text{desde que } x-l_i > x_0 \end{array} \right. \Rightarrow$$

$$\Rightarrow \bar{f}(x) = \max\left\{-\left(\frac{x}{l_m}\right) \cdot \pi_m, \bar{f}(x-l_i) + \left(\frac{x-l_i}{l_m}\right) \cdot \pi_m + \pi_i - \left(\frac{x}{l_m}\right) \cdot \pi_m; i \geq l(x-l_i), x \geq l_i\right\} = \\ \max\left\{-\left(\frac{x}{l_m}\right) \cdot \pi_m, \bar{f}(x-l_i) - \left(\frac{l_i}{l_m}\right) \cdot \pi_m + \pi_i; i \geq l(x-l_i), x \geq l_i\right\} \\ \text{desde que } x-l_i > x_0, \forall i; i \geq l(x-l_i) \text{ e } x \geq l_i.$$

Se chamarmos $\pi'_i = \pi_i - \left(\frac{l_i}{l_m}\right) \cdot \pi_m$, poderemos reescrever:

$$\bar{f}(x) = \max\left\{-\left(\frac{x}{l_m}\right) \cdot \pi_m, \bar{f}(x-l_i) + \pi'_i; i \geq l(x-l_i), x \geq l_i\right\} \\ \text{desde que } x-l_i > x_0, \forall i; i \geq l(x-l_i) \text{ e } x \geq l_i \quad (\text{I.19})$$

Definição:

$\bar{\varphi}(l) = \bar{f}(Nl_m + l) \forall l \in [0, l_m)$, onde R é um inteiro satisfazendo $Nl_m > x_0$.

Vamos supor, como fizemos anteriormente, que $x-l_i \geq Nl_m$. Assim, devemos ter: $\left[\frac{x-l_i}{l_m}\right] \geq N$ e $\left[\frac{x}{l_m}\right] \geq N$. Donde retiramos:

$$\bar{f}(x-l_i) = \bar{f}\left(\left[\frac{x-l_i}{l_m}\right] \cdot l_m + r\left(\frac{x-l_i}{l_m}\right)\right) = \bar{f}\left(\left(\left[\frac{x-l_i}{l_m}\right] - N\right) \cdot l_m + Nl_m + r\left(\frac{x-l_i}{l_m}\right)\right) \stackrel{\text{período}=l_m}{=} \\ = \bar{f}(Nl_m + r\left(\frac{x-l_i}{l_m}\right)) = \bar{\varphi}\left(r\left(\frac{x-l_i}{l_m}\right)\right)$$

Analogamente: $\bar{f}(x) = \bar{\varphi}(r(x/l_m))$. Por fim, fazendo as substituições, concluímos:

$$\bar{\varphi}(r(\frac{x}{l_m})) = \max\{-\frac{x}{l_m}\pi_m, \bar{\varphi}(r(\frac{x-l_i}{l_m})) + \pi'_i; i \geq l(x-l_i), x \geq l_i\},$$

desde que $x \geq Nl_m + l_i \forall i; i \geq l(x-l_i)$ e $x \geq l_i$

A seguir, acompanhemos o raciocínio abaixo:

$$\begin{aligned} F(x-l_i) + \pi_i \geq 0 \forall i &\Rightarrow F(x-l_i) - \frac{x}{l_m} \cdot \pi_m + \frac{l_i}{l_m} \cdot \pi_m + \pi_i - \frac{l_i}{l_m} \cdot \pi_m \geq \\ &\geq -\frac{x}{l_m} \cdot \pi_m \Rightarrow F(x-l_i) - (\frac{x-l_i}{l_m}) \cdot \pi_m + \pi_i - (\frac{l_i}{l_m}) \cdot \pi_m \geq -(\frac{x}{l_m}) \cdot \pi_m \Rightarrow \\ &\Rightarrow \bar{f}(x-l_i) + \pi'_i \geq -(\frac{x}{l_m}) \cdot \pi_m \Rightarrow \bar{\varphi}(r(\frac{x-l_i}{l_m})) + \pi'_i \geq -(\frac{x}{l_m}) \cdot \pi_m \\ &\text{desde que } x-l_i \geq Nl_m \end{aligned}$$

Donde, finalmente, chegamos à fórmula final:

$$\bar{\varphi}(r(\frac{x}{l_m})) = \max\{\bar{\varphi}(r(\frac{x-l_i}{l_m})) + \pi'_i; i \geq l(x-l_i) \text{ e } x \geq l_i\} \quad (I.20)$$

desde que as condições abaixo sejam todas satisfeitas:

- $x \geq Nl_m + l_i \forall i; i \geq l(x-l_i)$ e $x \geq l_i$;
- $\{i; i \geq l(x-l_i) \text{ e } x \geq l_i\} \neq \emptyset$;

Vamos supor, agora, que a seguinte restrição seja acrescida à equação L7:

$$\sum_{i=1}^m z_i \leq k \quad (I.21)$$

Dessa forma, podemos redefinir a função mochila como sendo:

$$G(k, x) = \max\{z_1\pi_1 + z_2\pi_2 + \dots + z_m\pi_m; z_1l_1 + z_2l_2 + \dots + z_ml_m \leq x, 0 \leq z_i \in Z \text{ e } \sum_{i=1}^m z_i \leq k\}$$

É possível que a condição 1.2, para esse tipo de função, não seja satisfeita. Exemplo:

$$\begin{aligned} l_1 = 5 \text{ e } \pi_1 = 2.5 & \quad G(3, 20) = 3 \cdot 3 = 9 \quad (3 \text{ pedaços de } l_2) \\ l_2 = 6 \text{ e } \pi_2 = 3 & \quad G(3, 12) = 3 \cdot 2 = 6 \quad (3 \text{ pedaços de } l_3) \\ l_3 = 4 \text{ e } \pi_3 = 2 & \quad G(3, 8) = 2 \cdot 2 = 4 \quad (2 \text{ pedaços de } l_3) \\ k = 3 & \end{aligned}$$

$$G(3, 12) + G(3, 8) = 6 + 4 = 10 > 9 = G(3, 20)$$

Desenvolvemos, a seguir, uma expressão que nas permita calcular $G(k, 4)$. Para isso, refaremos algumas passagens anteriores, só que adaptadas para o novo caso.

Definição

(z_1, z_2, \dots, z_m) é uma partição realizando $G(k, x)$ se

$$\begin{cases} z_1 l_1 + z_2 l_2 + \dots + z_m l_m \leq x \\ \pi_1 z_1 + \pi_2 z_2 + \dots + \pi_m z_m = G(k, x) \\ z_1 + z_2 + \dots + z_m \leq k \\ 0 \leq z_i \in Z, \forall i, i = 1, 2, \dots, m \end{cases}$$

Definição

$(z_1^*, z_2^*, \dots, z_m^*)$ é a partição associada a $G(k, x)$ se da é uma partição realizando $G(k, x)$ e para toda partição (z_1, z_2, \dots, z_m) realizando $G(k, x)$, tem-se $z_m \leq z_m^*$ e:

se $z_m = z_m^*, z_{m-1} = z_{m-1}^*, \dots, z_{r+1} = z_{r+1}^*$ então $z_r \leq z_r^* \forall r \in \{1, 2, \dots, m-1\}$

Definição

$$l_1(k, x) = \begin{cases} 1 & \text{se } (x \leq 0 \text{ ou } k = 0) \\ \max\{1, i; z_i \neq 0\} & \text{se } (x > 0 \text{ e } k > 0), \text{ onde } (z_1, z_2, \dots, z_m) \text{ é a partição} \\ & \text{associada a } G(k, x) \end{cases}$$

(I.22)

$$l_2(k, x) = \max\{1, i; G(k, x) = G(k-1, x-l_i) + \pi_i \wedge x \geq l_i, k > 0\}$$

Lema I.4 $l_1(k, x) = l_2(k, x), \forall k, \forall x$

Demonstração

Se $x < \min\{l_i; i = 1, 2, \dots, m\}$ então $l_1(k, x) = 1$ e $l_2(k, x) = 1$.
Suponha $x \geq \min\{l_i; i = 1, 2, \dots, m\}$. Se $k = 0$ então $l_1(k, x) = l_2(k, x) = 1$. Assim, suponhamos também $k > 0$. Consideremos (z_1, z_2, \dots, z_m) a partição associada a $G(k, x)$.

$$\begin{cases} i_1 = \max\{i; z_i \neq 0\} \\ i_2 = \max\{i; G(k, x) = G(k-1, x-l_i) + \pi_i \text{ e } x \geq l_i\} \end{cases}$$

Seja $(z'_1, z'_2, \dots, z'_m)$ a partição associada a $G(k-1, x-l_{i_2})$. Ela satisfaz as seguintes restrições:

$$\begin{cases} z'_1 l_1 + z'_2 l_2 + \dots + z'_m l_m \leq x - l_{i_2} \\ z'_1 + z'_2 + \dots + z'_m \leq k - 1 \\ \pi_1 z'_1 + \pi_2 z'_2 + \dots + \pi_m z'_m = G(k-1, x-l_{i_2}) \\ G(k, x) = G(k-1, x-l_{i_2}) + \pi_{i_2} \end{cases} \Rightarrow$$

$$\Rightarrow \begin{cases} z'_1 l_1 + z'_2 l_2 + \dots + (z'_{i_2} + 1) l_{i_2} + \dots + z'_m l_m \leq x \\ z'_1 + z'_2 + \dots + (z'_{i_2} + 1) + \dots + z'_m \leq k \\ \pi_1 z'_1 + \pi_2 z'_2 + \dots + \pi_{i_2} (z'_{i_2} + 1) + \dots + \pi_m z'_m = G(k-1, x - l_{i_2}) + \pi_{i_2} \\ = G(k, x) \end{cases}$$

Assim, $(z'_1, z'_2, \dots, z'_{i_2} + 1, \dots, z'_m)$ é uma partição realizando $G(k, x)$.

$$\begin{cases} i_1 < i_2 \\ (z_1, z_2, \dots, z_m) \text{ é a partição} \\ \text{associada a } G(k, a) \end{cases} \Rightarrow z'_{i_2} + 1 = 0 \Rightarrow z'_{i_2} = -1 < 0 \text{ (absurdo!)}$$

Assuma $i_1 > i_2$. Se $x < l_{i_1}$ então $z_{i_1} = 0$, o que é absurdo pela definição de i_1 . Suponha, então, que $x \geq l_{i_1}$. Temos duas possibilidades:

1. $G(k-1, x - l_{i_1}) + \pi_{i_1} > G(k, x)$, absurdo pela definição de $G(k, x)$;
2. $G(k, x) > G(k-1, x - l_{i_1}) + \pi_{i_1} \Rightarrow z_{i_1} = 0$, pois $z_{i_1} \geq 1 \Rightarrow G(k, x) = G(k-1, x - l_{i_1}) + \pi_{i_1}$. Absurdo, pois contradiz a definição de i_1 .

Logo, $G(k, x) = G(k-1, x - l_{i_1}) + \pi_{i_1} \Rightarrow i_1 \in (i; G(k, x) = G(k-1, x - l_i) + \pi_i \wedge r \geq i) \Rightarrow i_1 \leq i_2$ (absurdo!).

Dessa forma, devemos ter $i_1 = i_2$ e, assim, $l_1(k, x) = l_2(k, x)$.

[C.Q.D.]

Doravante, chamaremos simplesmente $l(i, x) = l_1(k, x) = l_2(k, x)$.

Teorema I.6 A função $G(k, x)$ é definida pela equação:

$$\begin{aligned} G(0, x) &= 0, \quad \forall x \\ G(k+1, x) &= \max\{0, G(k, x - l_i) + \pi_i; x \geq l_i, i \geq l(k, x - l_i)\} \end{aligned} \quad (\text{I.23})$$

Demonstração

$G(0, x) = 0, \forall x$. É óbvio. Para $k = 0$, é fácil ver que a equação I.23 é satisfeita. Suponhamos, então, $k > 0$.

Suponha $x < \min\{l_i; i = 1, 2, \dots, m\}$ então $\{i; x \geq l_i \wedge i \geq l(k, x - l_i)\} = \emptyset$ e o segundo membro da equação I.23 é zero. Ocorre que $G(k+1, x) = 0$ pois $x < \min\{l_i; i = 1, 2, \dots, m\}$. O resultado fica, então, provado para este caso.

Suponha $r \geq \min\{l_i; i = 1, 2, \dots, m\}$. Seja (z_1, z_2, \dots, z_m) a partição associada a $G(k+1, x)$ e seja $r = l(k+1, r)$. Assim, usando o lema L4 acima, temos:

$$\begin{cases} G(k+1, x) = \pi_1 z_1 + \pi_2 z_2 + \dots + \pi_r z_r, \\ \text{onde } z_r > 0 \text{ e } z_{r+1} = \dots = z_m = 0 \\ G(k+1, x) = G(k, x - l_r) + \pi_r \end{cases} \Rightarrow \begin{aligned} G(k, x - l_r) &= \pi_1 z_1 + \\ &+ \dots + \pi_r (z_r - 1) \end{aligned}$$

Afirmação:

$z_1, z_2, \dots, (z_r - 1)$ é a partição associada a $G(k, x - l_r)$. De fato, nós já temos:

$$\begin{aligned} z_1 l_1 + \dots + z_r l_r \leq x &\Rightarrow z_1 l_1 + \dots + z_{r-1} l_{r-1} + (z_r - 1) l_r \leq x - l_r \\ z_1 + z_2 + \dots + z_r \leq k + 1 &\Rightarrow z_1 + z_2 + \dots + (z_r - 1) \leq k \end{aligned}$$

Assim, só nos resta provar apenas que se $(z'_1, z'_2, \dots, z'_m)$ é uma outra partição realizando $G(k, r - i)$, ela satisfaz as condições abaixo:

- (i) $z'_m = z'_{m-1} = \dots = z'_{r+1} = 0$;
- (ii) $z'_r \leq z_r - 1$;
- (iii) Se $z'_r = z_r - 1, z'_{r-1} = z_{r-1}, \dots, z'_{s+1} = z_{s+1}$ então $z'_s \leq z_s \forall s \in \{1, 2, \dots, r-1\}$.

Consideremos, então, a partição dada por $(z'_1, z'_2, \dots, z'_r + 1, \dots, z'_m)$. Ela satisfaz as relações abaixo:

- $z'_1 l_1 + z'_2 l_2 + \dots + (z'_r + 1) l_r + \dots + z'_m l_m = (z'_1 l_1 + \dots + z'_m l_m) + l_r \leq (x - l_r) + l_r = x$;
- $z'_1 \pi_1 + z'_2 \pi_2 + \dots + (z'_r + 1) \pi_r + \dots + z'_m \pi_m = G(k, x - l_r) + \pi_r = G(k + 1, x)$;
- $z'_1 + z'_2 + \dots + (z'_r + 1) + \dots + z'_m \leq k + 1$;

Assim, a partição $(z'_1, z'_2, \dots, (z'_r + 1), \dots, z'_m)$ realiza $G(k + 1, x)$. Pela definição de partição associada a $G(k + 1, x)$ e, levando-se em conta que $(z_1, z_2, \dots, z_r, 0, \dots, 0)$ é a partição associada a $G(k + 1, x)$, podemos concluir (i), (ii) e (iii).

Portanto, $l(k, x - l_r) \leq r$ e $x \geq l_r$. Logo,

$$\max\{G(k, x - l_i) + \pi_i; x \geq l_i, i \geq l(k, x - l_i)\} \geq G(k, x - l_r) + \pi_r = G(k + 1, x)$$

Portanto, $\max\{G(k, x - l_i) + \pi_i; x \geq l_i, i \geq l(k, x - l_i)\} = G(k + 1, x)$.

[C.Q.D.]

Capítulo II

O Problema da Mochila em Duas Dimensões sem Restrições

II.1 Definição do Problema

Este capítulo está baseado no paper escrito por J. C. Herz referenciado em [21]. Os principais resultados e teoremas podem aí ser encontrados.

Dado um retângulo R e um conjunto S de retângulos "pequenos", queremos encontrar uma forma de se extrair alguns elementos de S do retângulo R , de modo que o valor retornado seja máximo. A cada elemento de S é atribuído um valor. Os cortes realizados em R são do tipo guilhotina. Estamos supondo que não haja limites para o número de ocorrências de um elemento qualquer de S geradas por um tipo de corte sobre R .

II.2 Discretização

Sejam a e b as dimensões da retângulo R . Representemos por $S = \{(p_i, q_i); i = 1, 2, \dots, n\}$. Tome como sendo v_i o valor associado ao "pedaço" (p_i, q_i) . Façamos:

$$p = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix} \quad q = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} \quad z = [z_1 \ z_2 \ \dots \ z_n], \quad 0 \leq z_i \in \mathbb{Z}$$

Se representarmos por $\#A$, o número de elementos de um conjunto

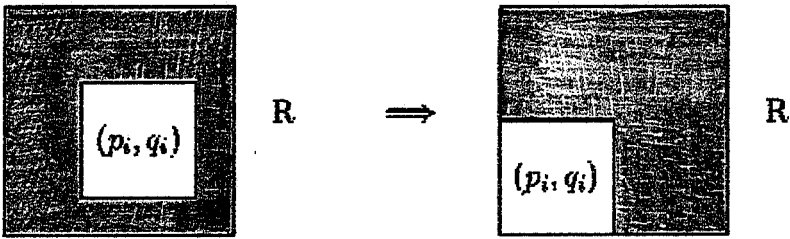


Figura II.1: uma pequena translação resolve o problema

A, podemos concluir, de acordo com a notação acima:

$$\#\{z; zp \leq a\} < \infty \quad e \quad \#\{z; zq \leq b\} < \infty$$

Representemos $P = \{zp; zp \leq a\}$ e $Q = \{zq; zq \leq b\}$.

Chamamos de **dissecção** do retângulo R o resultado final de uma série de cortes nele realizados. Uma abscissa de uma dissecção é a distância entre a linha vertical que representa um corte no ponto da abscissa e o lado do retângulo R mais à esquerda. Analogamente, temos também uma ordenada da dissecção.

Teorema II.1 *A cada dissecção do retângulo R podemos associar uma outra de valor retornado pelo menos igual, tal que todas as suas abscissas e ordenadas estão em P e Q , respectivamente.*

Demonstração

A demonstração será feita por indução.

O caso inicial é aquele em que o retângulo R tem a seguinte propriedade, que chamaremos de propriedade A:

"A melhor dissecção do retângulo R é aquela que produz um único pedaço do conjunto S ."

Para um retângulo que tem a propriedade h , é fácil ver que o teorema é válido, conforme a figura II.1.

Seja o caso geral. Assumiremos um retângulo R e uma dissecção D sobre R . Assumiremos, sem perda de generalidade, que a primeira linha de corte L é vertical. Sejam R_1 e R_2 os retângulos resultantes desse corte.

Admitamos que para R_1 e R_2 o teorema esteja provado. Ou seja, existem dissecções "melhores" D_1 de R_1 e D_2 de R_2 , em que todas as abscissas estão em P e todas as ordenadas em Q . As abscissas de D_2 são medidas a partir de L .

Se a abscissa de L pertence a P , a união de D_1 e D_2 é uma dissecção de R satisfazendo o teorema:

$$\text{Se } x \in P, \text{ e } \exists P \text{ e } c + c \leq a \text{ então } x + x \in P$$

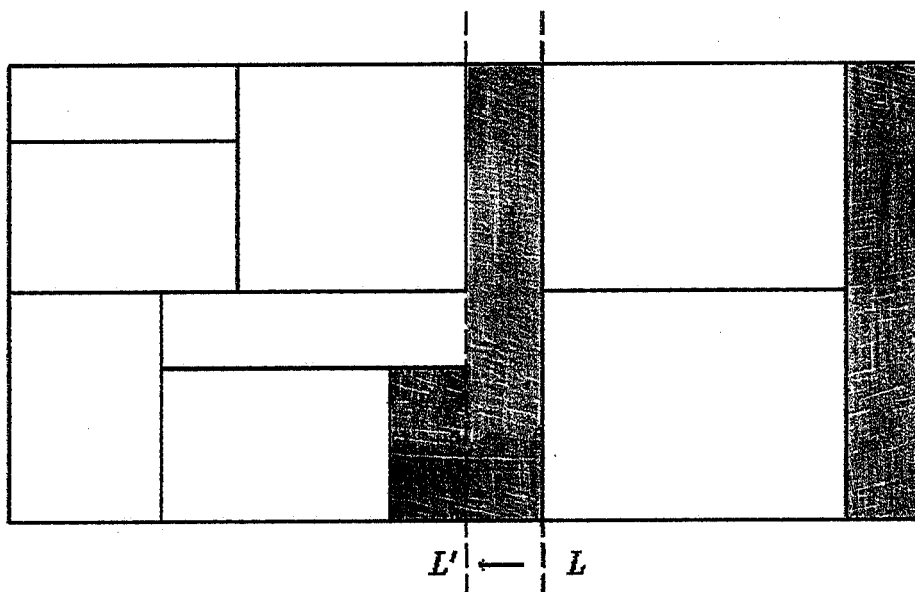


Figura II.2: basta deslocarmos L

Se a abscissa de L não pertence a P , então consideremos $L' = \max\{x \in P; x \text{ é abscissa de } D_1\}$. Obtemos D por construção. Basta deslocarmos o ponto de corte correspondente à linha vertical L para o local onde foi feito o corte L' . veja a figura II.2.

O teorema fica, assim, provado. De fato, se R_1 ou R_2 não satisfizerem a propriedade A, basta repetirmos a raciocínio empregado no caso geral para esses dois retângulos. Até que, finalmente, teremos chegado a subretângulos para os quais é válida a propriedade 4, em que o teorema já foi demonstrado.

[C.Q.D.]

Uma dissecção em que todas as abscissas estão em P e todas as ordenadas em Q h dita ser uma dissecção canônica.

Nós denominamos de dissecção homogênea aquela que produz retângulos de iguais dimensões (veja figura II.3). As abscissas são $p_i, 2p_i, 3p_i, \dots$, etc e as ordenadas são $q_i, 2q_i, 3q_i, \dots$, etc para algum i .

II.3 Simetrização

Teorema II.2 Para cada dissecção não homogênea e canônica de R , existe uma dissecção canônica de igual valor, de modo que a coordenada do primeiro corte é na

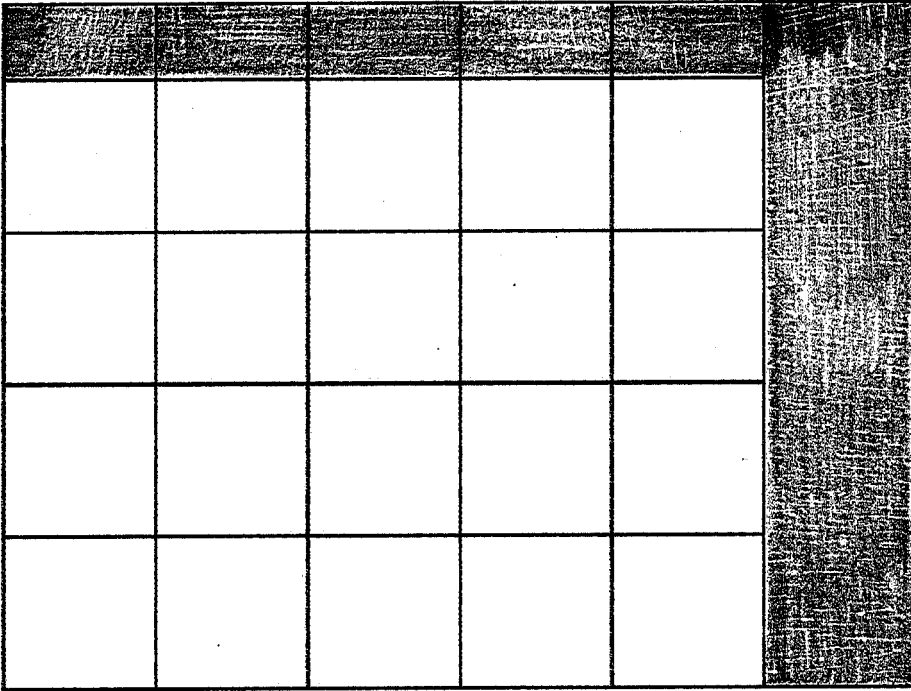


Figura II.3: uma dissecção homogênea canônica

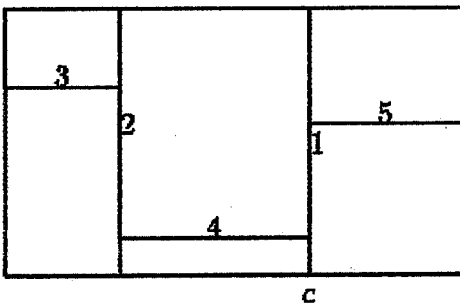


Figura II.4: Simetrização - Primeiro Caso

máximo metade da dimensão dos lados do retângulo R onde é realizado o corte.

Demonstração

Suponha que a primeira linha vertical de abscissa c é maior do que $\frac{a}{2}$. Se a , segunda linha vertical em uma abscissa menor do que ou igual a $\frac{a}{2}$, d e m os inverter a ordem e considerá-la como sendo o primeiro corte (figura II.4).

Se a segunda linha é vertical também em uma abscissa maior do que $\frac{a}{2}$, podemos permutar o retângulo gerado por esses dois cortes com o retângulo mais à esquerda (figura II.5). Observe que a permuta de retângulos preserva a canonicidade.

Se a segunda linha é horizontal, tome $c' = \max\{x; x \text{ é abscissa de algum retângulo utilizável gerado pelos cortes}\}$.

Se $c' > c$, basta desenharmos uma linha vertical em c' (desde que $c' < a$) e permutarmos o retângulo compreendido por c e c' com o que fica mais à

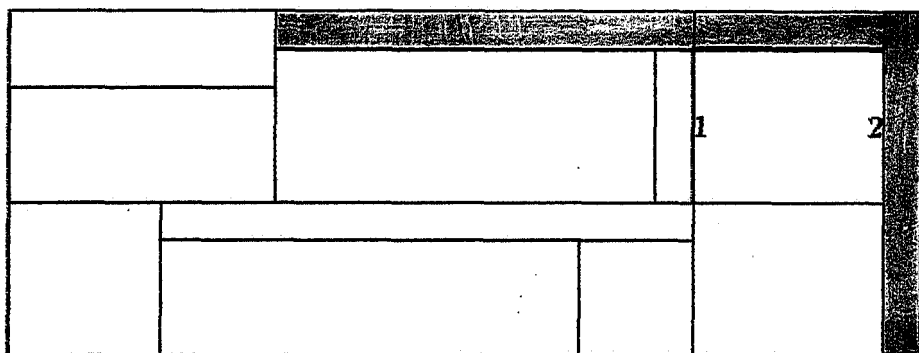
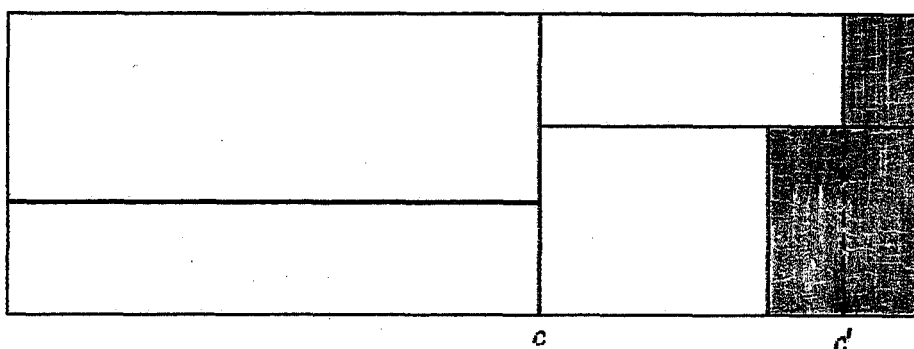


Figura 115: Simetrização - Segundo Caso

Figura II.6: Simetrização - Terceiro Caso: $c < c'$

esquerda (figura II.6).

Se $c = c'$, podemos estender a segunda linha até atingirmos o lado que fica mais à direita (figura II.7). Então, desprezamos o primeiro corte. Este só gera retângulos não utilizáveis mais à direita. Assim, o segundo corte fica sendo o primeiro. O processo se repete, só que o estudo se processa com os cortes na horizontal.

Finalmente, o último caso a ser estudado é o da figura II.8. Podemos atingir esse caso, quando sucessivamente desprezamos cortes horizontais e verticais pelo fato de gerarem sobras (simetrização - quarto caso). Porém, este último caso não satisfaz as hipóteses do teorema, pois é do tipo homogêneo.

[C.Q.D.]

II.4 Exclusão

Consideremos um retângulo de dimensões $\alpha \times \beta$. Definamos os conjuntos abaixo:

$$P_{\alpha\beta} = \{z\beta / (q_i > \beta \Rightarrow z_i = 0) \text{ e } (z\beta \leq \frac{\alpha}{2})\}$$

$$Q_{\alpha\beta} = \{z\alpha / (p_i > \alpha \Rightarrow z_i = 0) \text{ e } (z\alpha \leq \frac{\beta}{2})\}$$

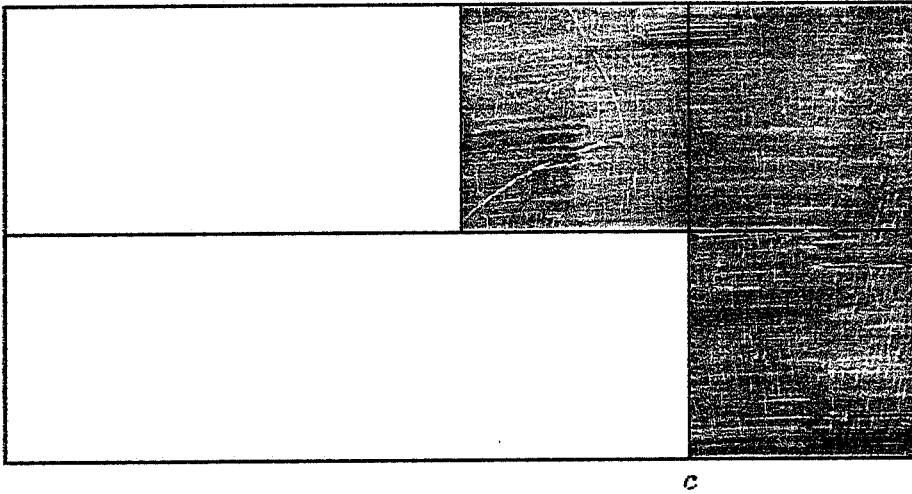


Figura II.7: Simetrização - Quarto Caso: $c = c'$

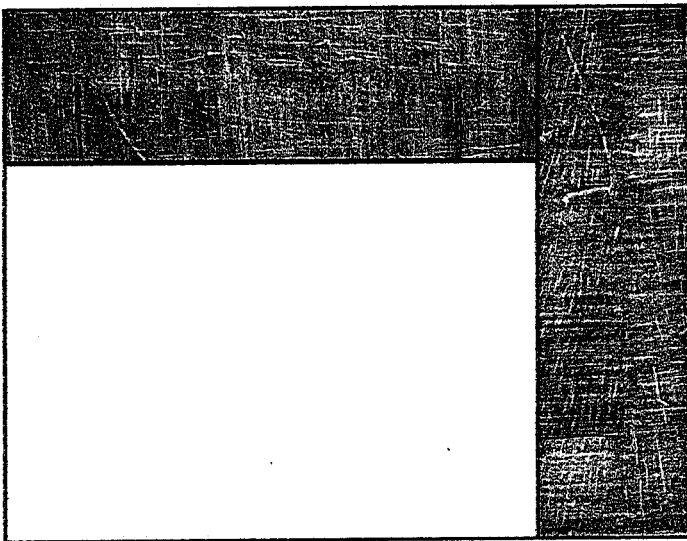


Figura II.8: Simetrização - Quinto Caso

Estes conjuntos permitem que limitemos a nossa busca aos pedaços (p_i, q_i) que estão dentro de (α, β) .

Lema II.1 *Seja o conjunto $\overline{P_{\alpha\beta}} = \{x \in P / x \leq \frac{\alpha}{2}, F(x) \leq \beta\}$, onde $f(x) = \max\{q_i / z_i > 0\}$ e $F(x) = \min\{f(z) / zp = x\} \forall x \in P$. Afirmamos que $P_{\alpha\beta} = \overline{P_{\alpha\beta}}$.*

Demonstração

Inicialmente, mostremos que $P_{\alpha\beta} \subset \overline{P_{\alpha\beta}}$. Seja $x \in P_{\alpha\beta}$, i.e., $x = zp$, onde r satisfaz $(q_i > \beta \Rightarrow z_i = 0)$ e $(zp \leq \frac{\alpha}{2})$. Temos, assim:

$$x \leq \frac{\alpha}{2} \quad (1)$$

Suponha $F(x) > \beta$. Então $\beta < f(z) \Rightarrow \exists i$ tal que $q_i > \beta, z_i > 0 \Rightarrow zp \notin P_{\alpha\beta} \Rightarrow x \notin P_{\alpha\beta}$. Absurdo.

$$\text{Logo, } F(x) \leq \beta. \quad (2)$$

Por (1) e (2), vem $x \in \overline{P_{\alpha\beta}}$.

Agora, provemos que $\overline{P_{\alpha\beta}} \subset P_{\alpha\beta}$. Seja $x \in \overline{P_{\alpha\beta}}$, i.e., $x \leq \frac{\alpha}{2}$, e $F(x) \leq \beta$. Como $x \in P$, seja $\bar{z}p = x$; $F(x) = f(\bar{z})$. Como $x \leq \frac{\alpha}{2}$, isto implica em $\bar{z}p \leq \frac{\alpha}{2}$. (3)

Admita que existe i com $q_i > \beta$ e $\bar{z}_i > 0$. Logo,

$$\begin{aligned} q_i > \beta \geq F(x) &\Rightarrow q_i > F(x) \\ \bar{z}_i > 0 \Rightarrow q_i \leq f(\bar{z}) &\Rightarrow F(x) \geq q_i. \end{aligned}$$

Pelas duas desigualdades acima vem o absurdo. Assim, se $q_i > \beta$ então $\bar{z}_i = 0$. (4)

Por (3) e (4), o resultado segue-se.

[C.Q.D.]

Analogamente, provamos também que:

$$\begin{aligned} Q_{\alpha\beta} = \overline{Q_{\alpha\beta}} &= \{y \in Q / y \leq \frac{\beta}{2}, G(y) \leq \alpha\}, \text{ onde} \\ G(y) &= \min\{g(z) / zq = y\} \text{ e} \\ g(z) &= \max\{p_i / z_i > 0\} \end{aligned}$$

A vantagem de trabalharmos com $\overline{P_{\alpha\beta}}$ e $\overline{Q_{\alpha\beta}}$ é que as funções F e G podem ser calculadas à parte. Elas não dependem de α e β .

II.5 Recursão Principal

A dissecção ótima de um retângulo com dimensões α e β será a dada por uma das três seguintes situações:

- (i) A dissecção ótima é homogênea;
- (ii) O primeiro corte da dissecção ótima é vertical e o valor dessa dissecção corresponde à soma dos valores obtidos dos retângulos gerados por esse corte;
- (iii) O primeiro corte da dissecção ótima é horizontal e a situação é análoga a anterior.

Sejam $\alpha_0 = \max\{x; x \in P \text{ e } x \leq \alpha\}$ e $\beta_0 = \max\{y; y \in Q \text{ e } y \leq \beta\}$. Assim, se representarmos por $v(\alpha, \beta)$ o valor máximo que podemos extrair do retângulo (α, β) então podemos dizer que $v(\alpha, \beta) = v(\alpha_0, \beta_0)$. Logo, basta guardarmos os valores de $v(\alpha, \beta)$, $\forall (\alpha, \beta) \in P \times Q$.

II.6 Uso de Cotas

Representaremos por $\mu(\alpha, \beta)$ uma cota superior para os valores possíveis de serem extraídos do retângulo (α, β) . Por exemplo, uma cota superior seria dada por:

$$\mu(\alpha, \beta) = \max\left\{\frac{v_i}{P_i q_i}; i = 1, 2, \dots, n\right\} \cdot \alpha \beta$$

Suponha que estamos estudando um retângulo (α, β) . De todos os cortes efetuados neste retângulo, dizemos que aquele que teve o melhor aproveitamento produziu o valor a . Suponha que estamos estudando um modo de cortar em γ e que o retângulo (γ, β) já tenha sido estudado e $v(\gamma, \beta) = w$. Assim, se tivermos $v \geq w + \mu(\alpha - \gamma, \beta)$ então é desnecessário estudarmos o retângulo $(\alpha - \gamma, \beta)$.

Dito de outra forma, se tivermos $\mu(\alpha, \beta) \geq \mu(\gamma, \beta) + \mu(\alpha - \gamma, \beta)$ então não há necessidade de estudarmos o corte em γ .

Em resumo, podemos dizer que existe um v_0 que é o menor valor que devemos extrair de um retângulo (α, β) que justifique seu corte. Assim, se $\mu(\alpha, \beta) \geq v_0$ então (α, β) poderá ser cortado e entramos na recursão. Como os cortes em (α, β) só são aceitáveis desde que produzam um valor superior ou igual a v_0 , este v_0 não pode ser perdido e é passado como parâmetro para a rotina recursiva, que é a realizadora dos cortes. Assim, dentro do procedimento recursivo, ao invés de trabalharmos com v , que é o maior valor obtido até o momento do retângulo

estudado na recursão, trabalhamos com $\max\{v, v_0\}$. Este último valor é o que serve como parâmetro para decidirmos sobre a realização ou não de um corte..

A rotina de recursão *fica* assim:

ROTINA RECURSIVA $r(x, y, v_0)$

Faça: $x_0 = \max\{x'/x' \leq x, x' \in P\}$
 $y_0 = \max\{y'/y' \leq y, y' \in Q\}$

Se o retângulo (x_0, y_0) foi dissecado

então retorne com o valor da dissecção

senão calcule a melhor dissecção homogênea de (x, y)

Se a melhor dissecção homogênea = $\mu(x, y)$

então retorne com o valor correspondente para a melhor dissecção homogênea

senão $v \leftarrow$ melhor dissecção homogênea

Para cada $c \in P_{xy}$, faça:

Se $\max\{v, v_0\} - \mu(x - c, y) \geq \mu(c, y)$

então salte para o c seguinte

senão $w = r(c, y, \max\{v, v_0\} - \mu(x - c, y))$

Se $\max\{v, v_0\} - w \geq \mu(x - c, y)$

então salte para o c seguinte

senão $v_1 = r(x - c, y, \max\{v, v_0\} - w) + w$

Se $v_1 = \mu(x, y)$

então retorne com v_1

senão $v \leftarrow \max\{v_1, v\}$

Para cada $s \in Q_{xy}$, faça:

Se $\max\{v, v_0\} - \mu(x, y - s) \geq \mu(x, s)$

então salte para o s seguinte

senão $w = r(x, s, \max\{v, v_0\} - \mu(x, y - s))$

Se $\max\{v, v_0\} - w \geq \mu(x, y - s)$

então salte para o s seguinte

senão $v_2 = r(x, y - s, \max\{v, v_0\} - w) + w$

Se $v_2 = \mu(x, y)$

então retorne com v_2

senão $v \leftarrow \max\{v_2, v\}$

Retorne com v

A rotina recursiva é chamada com parâmetros $a, b, 0$. A seguir, mostramos um procedimento iterativo que tem o mesmo efeito do recursivo. Denominemos por $v(x, y)$ o valor ótimo retomado por um retângulo (x, y) . Observe que a rotina iterativa calcula a melhor dissecção para o retângulo dado e todos os seus subretângulos.

Para cada $c \in P$ faça:

Para cada $d \in Q$ tal que $d \geq F(c)$, faça:

Encontre a melhor dissecção homogênea de (c, d)

Se a melhor dissecção homogênea de $(r, d) = \mu(c, d)$
então $v(c, d) \leftarrow \mu(c, d)$
 salte para o d seguinte
senão $v \leftarrow$ a melhor dissecção homogênea

Para cada $e \in P_{cd}$

$f = \max\{x' / x' \leq c - e, x' \in P\}$
 $v_1 = v(e, d) + v(f, d)$
Se $v_1 = \mu(c, d)$
então $v(c, d) \leftarrow v_1$
 salte para o d seguinte
senão $v \leftarrow \max\{v_1, v\}$

Para cada $r \in Q_{cd}$

$s = \max\{y' / y' \leq d - r, y' \in Q\}$
 $v_2 = v(c, r) + v(c, s)$
Se $v_2 = \mu(c, d)$
então $v(c, d) \leftarrow v_2$
 salte para o d seguinte
senão $v \leftarrow \max\{v_2, v\}$

$v(c, d) \leftarrow v$

O procedimento geral que nos fornece a melhor dissecção do retângulo (a, b) é dado por:

(1) Leia a, b, n, p_i, q_i, v_i ($1 \leq i \leq n$)

- (2) Construa P e Q
- (3) Aplique a rotina recursiva ou iterativa
- (4) Imprima a dissecção ótima

Capítulo III

O Problema do Corte em 1 Dimensão

11.1 Considerações sobre o Caso Geral

As seções III.1 e III.2 foram baseadas no paper escrito por Gilmore e Gomory referenciado em [16]. A seção III.3 foi baseada no paper também escrito por Gilmore e Gomory referenciado em [17]. As seções III.4 e III.5 foram baseadas no paper de Harald Dyckhoff referenciado em [12]. A seção III.6 foi baseada no capítulo 13 do livro de Chvátal referenciado em [8]. Os principais resultados podem aí ser encontrados.

Seja dado um estoque de peças com uma dimensão relevante. Os diversos tamanhos assumem valores no conjunto $\{L_1, L_2, \dots, L_K\}$.

Seja dada uma demanda por pedaços, de tamanhos variados, com uma dimensão relevante. Esses pedaços podem ser obtidos por meio de cortes realizada nas peças em estoque. A demanda consiste de N_i pedaços com dimensão l_i , $i = 1, \dots, m$.

Se tivermos para um dado j , $L_j \geq l_i \forall i$, o problema apresentará solução.

Há muitas formas de se cortar uma peça L_h , com $h \in \{1, 2, \dots, K\}$, de modo a gerarmos pedaços l_i com $i \in \{1, 2, \dots, m\}$. Sejam n a diversos modos de se cortar peças vindas do estoque, cada um deles caracterizado por uma peça L_h e uma série de pedaços l_i obtidos a partir do corte nesta peça.

A cada forma de corte j , com $j = 1, 2, \dots, n$, é associada uma variável x_j . Uma solução para o problema de cortes é o conjunto ordenado (x_1, x_2, \dots, x_n) , onde cada variável x_j nos indica, o número de vezes que a forma de cortar j é executada.

Em corte do tipo j , com $j \in \{1, 2, \dots, n\}$, são gerados pedaços com dimensão l_i , com $i \in \{1, 2, \dots, m\}$. Denotemos por a_{ij} o número desses pedaços.

O modo de cortar j **h executado** em cima de uma peça L_k , com $k \in \{1, 2, \dots, K\}$. Essa peça tem um valor. Seja c_j o custo associado ao modo de cortar j , que é igual ao valor da peça L_k onde o corte é realizado.

De acordo com a notação acima, o problema, pode ser escrito como:

$$\begin{aligned} &\text{minimizar } \sum_{j=1}^n c_j x_j \\ &\text{sujeito a } \sum_{j=1}^n a_{ij} x_j \geq N_i \quad (i = 1, \dots, m) \\ &0 \leq x_j \in Z \quad (j = 1, \dots, n) \end{aligned}$$

Introduzindo variáveis de folga, o problema pode ser reescrito como:

Obter os inteiros x_j que resolvam o PPL abaixo

$$\begin{aligned} &\text{minimizar } \sum_{j=1}^n c_j x_j \\ &\text{sujeito a } \sum_{j=1}^n a_{ij} x_j - x_{n+i} = N_i \quad (i = 1, \dots, m) \\ &x_j \geq 0 \quad (j = 1, \dots, n + m) \end{aligned}$$

Ao se resolver o PPL acima, podemos encontrar soluções não inteiras. Uma maneira de se contornar o problema, seria o de arredondarmos os valores obtidos "para cima", acrescentando um valor ao custo final. Outra forma seria a de arredondarmos os valores encontradas "para baixo", e a demanda não satisfeita seria resolvida com a aplicação de métodos muito particulares. Se os resultados obtidos no PPL forem fracionários muito grandes, então o arredondamento dado deverá causar um erro tolerável.

Vamos tentar resolver o PPL acima sem a restrição de integridade.

Lema III.1 *Para toda solução viável do PPL acima, existe uma solução em que todas as variáveis de folga assumem valores nulos com a seguinte característica: o valor nela calculado pela função objetivo é no máximo igual àquele encontrado com a primeira solução dada.*

Em outras palavras, para cada solução dada do PPL acima existe uma outra "pelo menos tão boa", em que todas as variáveis de folga possuem valor nulo.

Demonstração

Consideremos uma solução $(x_1, x_2, \dots, x_{n+m})$ em que a variável x_{n+p} tenha valor não nulo, com $1 \leq p \leq m$.

Suponha que não haja nenhum índice i para a qual $a_{pi}x_i \geq x_{n+p}$. Assim, sejam x_j , $j \in \{1, 2, \dots, n\}$, como sendo a primeira variável a assumir um valor não nulo em nossa solução, e k o modo de cortar idêntica a j exceto pelo fato de ele não gerar pedaços de comprimento l_p . Assim: $a_{pk} = 0$ e $a_{qj} = a_{qk} \forall q \neq p$.

Consideremos os elementos abaixo:

$$\begin{aligned} x'_i &= x_i \text{ para } i \neq j, k, n+p \\ x'_j &= 0 \\ x'_k &= x_k + x_j \\ x'_{n+p} &= x_{n+p} - a_{pj}x_j > 0 \text{ (devido à nossa hipótese)} \end{aligned}$$

Eles formam uma solução. De fato,

$$\begin{aligned} \sum_{j, k \neq i=1}^n a_{pi}x'_i + \overbrace{a_{pj}x'_j}^{=0} + \overbrace{a_{pk}x'_k}^{=0} - x'_{n+p} &= \\ \sum_{j, k \neq i=1}^n a_{pi}x_i + 0 + 0 - x_{n+p} + a_{pj}x_j &= \\ \sum_{j, k \neq i=1}^n a_{pi}x_i + \overbrace{a_{pk}x'_k}^{=0} + a_{pj}x_j - x_{n+p} &= N_p \\ \sum_{j, k \neq i=1}^n a_{qi}x'_i + \overbrace{a_{qj}x'_j}^{=0} + a_{qk}x'_k - x'_{n+q} &= \\ \sum_{j, k \neq i=1}^n a_{qi}x_i + 0 + a_{qk}x_k + a_{qk}x_j - x_{n+q} &= \\ \sum_{j, k \neq i=1}^n a_{qi}x_i + a_{qk}x_k + a_{qj}x_j - x_{n+q} &= N_q \quad (q = 1, 2, \dots, m \text{ e } q \neq p) \end{aligned}$$

Se ainda não tivermos para essa nova solução algum índice i satisfazendo $a_{pi}x_i \geq x_{n+p}$, então continuamos o processo acima até conseguirmos.

Seja, então, o índice \bar{j} tal que $a_{p\bar{j}}x_{\bar{j}} \geq x_{n+p}$. Considere \bar{k} o modo da cortar semelhante a \bar{j} exceto pelo fato de ele não gerar pedaços de comprimento l_p . Tome uma nova solução: (observe que $a_{p\bar{j}} > 0$, pois $a_{p\bar{j}} = 0 \Rightarrow x_{n+p} = 0$).

$$\begin{cases} x'_i = x_i \quad \forall i \neq \bar{j}, \bar{k}, n+p \\ x'_{\bar{j}} = x_{\bar{j}} - \left(\frac{x_{n+p}}{a_{p\bar{j}}}\right), \quad x'_{\bar{k}} = x_{\bar{k}} + \left(\frac{x_{n+p}}{a_{p\bar{j}}}\right) \text{ e } x'_{n+p} = 0 \end{cases}$$

Então:

$$\begin{aligned}
 & \sum_{\substack{j, k \\ j, k \neq i=1}}^n a_{pj} x_j' + a_{p\bar{j}} x_{\bar{j}}' + a_{p\bar{k}} x_{\bar{k}}' - x_{n+p}' = \\
 & = \sum_{\substack{j, k \\ j, k \neq i=1}}^n a_{pj} x_j + a_{p\bar{j}} x_{\bar{j}} - x_{n+p} + a_{p\bar{k}} x_{\bar{k}} + \frac{\overbrace{a_{p\bar{k}} x_{n+p}}^{=0}}{a_{p\bar{j}}} = N_p \\
 & \sum_{\substack{j, k \\ j, k \neq i=1}}^n a_{qj} x_j' + a_{q\bar{j}} x_{\bar{j}}' + a_{q\bar{k}} x_{\bar{k}}' - x_{n+q}' = \\
 & = \sum_{\substack{j, k \\ j, k \neq i=1}}^n a_{qj} x_j + a_{q\bar{j}} x_{\bar{j}} - \frac{a_{q\bar{j}} x_{n+p}}{a_{p\bar{j}}} + a_{q\bar{k}} x_{\bar{k}} + a_{q\bar{k}} \cdot \frac{x_{n+p}}{a_{p\bar{j}}} - x_{n+q} = N_q \text{ para } q = 1, 2, \dots, m \text{ e } q \neq p \\
 & \sum_{i=1}^m c_i x_i' = \sum_{i \neq j, k} c_i x_i' + c_j x_j' + c_{\bar{k}} x_{\bar{k}}' = \sum_{i \neq j, k} c_i x_i + c_j x_j - \frac{c_j x_{n+p}}{a_{p\bar{j}}} + c_{\bar{k}} x_{\bar{k}} + \frac{c_{\bar{k}} x_{n+p}}{a_{p\bar{j}}} = \\
 & = \left(\sum_{i \neq j, k} c_i x_i + c_j x_j + c_{\bar{k}} x_{\bar{k}} \right) + \frac{x_{n+p}}{a_{p\bar{j}}} \cdot \underbrace{(c_{\bar{k}} - c_j)}_{\leq 0} \leq \sum_{i=1}^n c_i x_i
 \end{aligned}$$

[C.Q.D.]

Daqui para frente, buscaremos nossa solução ótima apenas entre aquelas cujas variáveis de folga assumem valores nulos.

III.2 Um Procedimento para o Caso Geral de Cortes

Vamos determinar uma base inicial para a PPL. O objetivo dessa base inicial é dar início à computação pelo método simplex com uma solução básica viável inicial.

Cada coluna i da base corresponde a um modo de cortar. Para cada $i \in \{1, 2, \dots, m\}$, escolha uma peça de tamanho $L_j \geq l_i$ e defina o modo de cortar relativo à coluna i como consistindo apenas de a_{ji} peças l_i extraídas da peça L_j onde $a_{ji} = \lfloor \frac{L_j}{l_i} \rfloor$. Como foi dito, O custo associado a este modo de cortar é o da peça de comprimento L_j .

Assim, podemos escrever:

$$\text{Base } B = \begin{bmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{mm} \end{bmatrix} \quad X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n+m} \end{bmatrix}$$

$$C = [c_1 \ c_2 \ \dots \ c_n \ 0 \ \dots \ 0]$$

$X^B, C^B \longmapsto$ componentes associados às variáveis básicas

$N \longmapsto$ variáveis não-básicas

$$Q(X) = \sum_{j=1}^n c_j x_j \longmapsto \text{função objetivo}$$

$$\left. \begin{array}{l} Q(X) = C^B X^B + C^N X^N \\ BX^B + NX^N = b \end{array} \right\} \xrightarrow{X^N \geq 0} Q(X) = C^B (B^{-1}b) \quad b = \begin{bmatrix} N_1 \\ N_2 \\ \vdots \\ N_m \end{bmatrix}$$

Seja P uma coluna não básica, ie, $P \in N$. Seja x a variável correspondente. Então, se chamarmos de \bar{X} a nova solução obtida com a inclusão de P , teremos:

$$\bar{X} = B^{-1}(b - xP) \implies Q(\bar{X}) = C^B \bar{X}^B + C^N \bar{X}^N = C^B B^{-1}(b - xP) + cx,$$

onde c h o custo associado a P .

Logo,

$$Q(X) - Q(\bar{X}) = C^B B^{-1}b - C^B B^{-1}(b - xP) - cx = x(C^B B^{-1}P - c)$$

$$\begin{bmatrix} 1 & -C^B \\ 0 & B \end{bmatrix} \cdot \begin{bmatrix} 1 & C^B B^{-1} \\ 0 & B^{-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & I \end{bmatrix} \implies$$

$$\implies \begin{bmatrix} 1 & -C^B \\ 0 & B \end{bmatrix}^{-1} = \begin{bmatrix} 1 & C^B B^{-1} \\ 0 & B^{-1} \end{bmatrix}$$

$$\begin{bmatrix} 1 & -C^B \\ 0 & B \end{bmatrix}^{-1} \cdot \begin{bmatrix} -c \\ P \end{bmatrix} = \begin{bmatrix} 1 & C^B B^{-1} \\ 0 & B^{-1} \end{bmatrix} \cdot \begin{bmatrix} -c \\ P \end{bmatrix} = \begin{bmatrix} -c + C^B B^{-1}P \\ B^{-1}P \end{bmatrix}$$

Se o primeiro elemento da matriz

$$\begin{bmatrix} 1 & -C^B \\ 0 & B \end{bmatrix}^{-1} \cdot \begin{bmatrix} -c \\ P \end{bmatrix}$$

for positivo, então a nova solução \bar{X} que é obtida com a inclusão de P na base B é melhor que a anterior associada a B .

Suponha que os vetores $C^B B^{-1}$ e P tenham os coeficientes abaixo:

$$C^B \cdot B^{-1} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}^t \quad P = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix}$$

Assim:

$$-c + C^B B^{-1} P = -c + a_1 b_1 + a_2 b_2 + \dots + a_m b_m$$

$$-c + C^B B^{-1} P > 0 \iff a_1 b_1 + a_2 b_2 + \dots + a_m b_m > c$$

Conclusão:

Uma condição suficiente para que uma coluna não básica P possa ser colocada na base é que existam inteiros não negativos a_1, a_2, \dots, a_m satisfazendo:

$$a_1 b_1 + a_2 b_2 + \dots + a_m b_m > c$$

Uma coluna não básica tem duas possibilidades. Pode se referir a uma variável de folga ou não. Pelo lema III.1 basta tomarmos as colunas P que se referem a variáveis que não são de folga. Assim, consideraremos apenas aquelas colunas que se referem a modos de cortar uma barra da estoque de comprimento L .

É interessante notar que basta resolvermos o problema abaixo para uma barra de comprimento L qualquer escolhida do estoque:

$$a_1 l_1 + a_2 l_2 + \dots + a_m l_m \leq L \quad (\text{III.1})$$

$$a_1 b_1 + a_2 b_2 + \dots + a_m b_m > c \quad (\text{III.2})$$

Uma solução (a_1, a_2, \dots, a_m) satisfazendo as duas inequações acima poderia ser obtida da seguinte forma. Sejam i_1, i_2, \dots, i_m tais que

$$\frac{b_{i_1}}{l_{i_1}} \geq \frac{b_{i_2}}{l_{i_2}} \geq \dots \geq \frac{b_{i_m}}{l_{i_m}}$$

Escolha:

$$a_{i_1} = \left[\frac{L}{l_{i_1}} \right], \quad a_{i_2} = \left[\frac{L - l_{i_1} a_{i_1}}{l_{i_2}} \right], \quad a_{i_3} = \left[\frac{L - l_{i_1} a_{i_1} - l_{i_2} a_{i_2}}{l_{i_3}} \right], \dots, \quad a_{i_m} = \left[\frac{L - \sum_{j=1}^{m-1} l_{i_j} a_{i_j}}{l_{i_m}} \right],$$

onde $[y] =$ maior inteiro que não excede y .

Chamarem (a_1, a_2, \dots, a_m) de "a solução imediata". Se a solução (a_1, a_2, \dots, a_m) assim obtida não satisfizer III.2 (isto é, a solução imediata não funciona) então partiremos para resolver o do problema da mochila abaixo:

$$\text{maximizar } a_1 b_1 + a_2 b_2 + \dots + a_m b_m$$

$$\text{sujeito a } a_1 l_1 + a_2 l_2 + \dots + a_m l_m \leq L, \quad L \in \{L_1, L_2, \dots, L_k\}$$

$$Z \ni a_i \geq 0$$

Se existir uma solução para o problema da mochila e ela satisfizer III.2 então teremos gerado uma nova coluna P para a base. Caso contrário, a solução atual para o PPL não admite melhora e teremos chegado ao final. O procedimento completo é o seguinte:

1. Existe algum índice j para o qual $L_j \geq l_i$ para todo i , $i = 1, 2, \dots, n$? Se a resposta for sim vá para (3);
2. O PPL não admite solução. FIM;
3. Obtenha uma solução básica viável inicial;
4. A solução imediata funciona? Se a resposta for sim, vá para (8);
5. O problema da mochila admite solução? Se a resposta for não então vá para (7);
6. A solução para o problema da mochila que foi obtida satisfaz III.2? Se a resposta for sim então vá para (8);
7. A solução atual é ótima. FIM;
8. Obtenha uma nova base com a inclusão da coluna P . Vá para (4);

Para se resolver o problema da mochila podemos recorrer a programação dinâmica.

Notação:

$$F_s(L) = \max\{b_1 a_1 + \dots + b_s a_s; l_1 a_1 + l_2 a_2 + \dots + l_s a_s \leq L, a_i \in \mathbb{Z}\}$$

Vamos supor calculado $F_s(L)$, para algum s . A seguir, calculemos $F_{s+1}(L)$. Temos duas hipóteses a considerar:

1. $a_{s+1} = 0 \Rightarrow$ A peça de comprimento l_{s+1} não foi gerada $\Rightarrow F_s(L) = F_{s+1}(L)$;
2. $a_{s+1} > 0 \Rightarrow$ A peça de comprimento l_{s+1} foi gerada

$$\Rightarrow F_{s+1}(L) = F_s(L - a_{s+1} l_{s+1}) + a_{s+1} b_{s+1}$$

Donde, concluímos que:

$$F_{s+1}(L) = \max_r \{r b_{s+1} + F_s(L - r l_{s+1})\}, 0 \leq r \leq \lfloor \frac{L}{l_{s+1}} \rfloor$$

111.3 Um Procedimento para a Solução do Problema da Mochila em uma Dimensão

Desejamos resolver k diversos problemas do tipo

$$\text{maximizar } \sum_{i=1}^m b_i a_i$$

$$\text{sujeito a } \sum_{i=1}^m l_i a_i \leq L \text{ onde } L \in \{L_1, L_2, \dots, L_k\}, a_i \in Z_+$$

Este procedimento, segundo Gilmore e Gomory, é cerca de cinco vezes mais rápido que o mostrado no final da última seção baseado em programação dinâmica:

(1) Inicialização

(1.1) Reordene as variáveis a_1, a_2, \dots, a_m tal que

$$\frac{b_1}{l_1} \geq \frac{b_2}{l_2} \geq \dots \geq \frac{b_m}{l_m}$$

e reordene os tamanhos de estaque tal que $L_1 > L_2 > \dots > L_k$

Explicação: essa reordenação possibilita uma convergência mais rápida do algoritmo.

b_i/l_i = densidade de valor da peça de primento l_i

(1.2) Introduza a variável a_{m+1} com $b_{m+1} = 0$ e $l_{m+1} = 1$

Explicação: é para tornar possível a realização do passo (4)

(1.3) Faça:

$$\lambda_{m+1} = \begin{bmatrix} l_1 \\ l_2 \\ \vdots \\ l_m \end{bmatrix} \quad \beta_{m+1} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

(1.4) Faça $M_j = c_j$, $j = 1, 2, \dots, k$ e $t = 1$

Explicação: M_j = maior valor possível, por meio de cortes, que eu posso extrair da barra L_j , $j \in \{1, 2, \dots, k\}$

A barra L_j tem valor c_j . Logo, só fará sentido cortá-la caso seja viável extrair pequenas barras cujos valores somadas ultrapassem c_j . Assim:

$M_j = c_j$, caso não tenha sido feito nenhum corte

$M_j = \sum_{i=1}^m b_i a_i$, desde que $\sum_{i=1}^m b_i a_i > c_j$, onde

a_i = número de peças de comprim. l_i que foram geradas a partir da barra L_j

Portanto.. $M_j = \max\{c_j, \max \sum_{i=1}^m b_i a_i \text{ sujeito a } \sum_{i=1}^m l_i a_i \leq L_j\}$

A variável t fornece o menor índice para o qual é possível que o vetor (α) , a ser gerado nos conduza a uma melhora em M_j com $j \geq t$. O vetor $(\alpha)_m$ será gerado a partir de cortes em L_t

(1.5)

$$a_1 = \left[\frac{L_1}{l_1} \right], a_2 = \left[\frac{L_1 - l_1 a_1}{l_2} \right], \dots, a_m = \left[\frac{L_1 - (l_1 a_1 + \dots + l_{m-1} a_{m-1})}{l_m} \right] \quad (\alpha)_m = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix}$$

(2) $\forall j, t \leq j \leq k; L_j \geq \lambda \cdot (\alpha)_m$ e $\beta \cdot (\alpha)_m > M_j$, Faça $M_j = \beta \cdot (\alpha)_m$

Explicação: o que nós fazemos nesse passo é atualizar os valores de M_j , em relação à última solução viável obtida.

(3) $s = \max\{i; 1 \leq i \leq m, a_i \neq 0\}$

Explicação: determinamos o maior s para o qual é possível existir alguma extensão de $(\alpha)_{s-1}$ a nos conduzir a uma melhora em pelo menos um dos M_j 's.

(4) Faça $a_s = a_s - 1$

Se $\{j, 1 \leq j \leq k; L_j \geq \lambda \cdot (\alpha)_s$ e $(L_j - \lambda \cdot (\alpha)_s) b_{s+1} > (M_j - \beta \cdot (\alpha)_s) l_{s+1}\} \neq \emptyset$
então

$$\begin{cases} t = \min\{j, 1 \leq j \leq k; L_j \geq \lambda \cdot (\alpha)_s \text{ e } (L_j - \lambda \cdot (\alpha)_s) b_{s+1} > (M_j - \beta \cdot (\alpha)_s) l_{s+1}\} \\ \text{vá para (6)} \end{cases}$$

senão vá para (5)

Explicação: Temos uma solução $\hat{\alpha}$. Ao fazermos $a_s \leftarrow a_s - 1$ criamos as condições para saber se é possível ou não gerar melhores soluções a partir de $(\alpha)_{s-1}$. A variável t fornecerá o mínimo dos candidatos j para os quais é possível termos uma melhora em M_j . Se existir tais j 's iremos p a a (6) gerar a extensão de $(\alpha)_s$ que melhorará o resultado atual. A variável t é escolhida como o mínimo pelo fato de os comprimentos estarem ordenados na forma $L_1 > L_2 > \dots > L_k$ e gerarmos sempre a maior solução lexicográfica possível. É possível alguma extensão de $(\alpha)_s$ nos conduzir a uma melhora em pelo menos um dos M_j 's?

Se $L_j \geq \lambda \cdot (a_j)$, então podemos observar que:

$$\beta \cdot (\alpha)_s + b_{s+1} (L_j - \lambda \cdot (\alpha)_s) / l_{s+1} \leq M_j \implies \text{Não é possível melhorar-se em } M_j \text{ a partir de } (\alpha)_s.$$

Isto ocorre porque mesmo trabalhando com a melhor hipótese, que é atribuindo a a_{s+1} o seu valor máximo (observe que estamos trabalhando com a hipótese da ordenação $\frac{b_1}{l_1} \geq \frac{b_2}{l_2} \geq \dots \geq \frac{b_m}{l_m}$) não conseguimos melhorar o resultado até aqui obtido.

- (5) Se $\{i; 1 \leq i \leq s \text{ e } a_i \neq 0\} \neq \emptyset$
 então $s = \max\{i; 1 \leq i \leq s \text{ e } a_i \neq 0\}$
 ir para (4)
 senão os valores atuais de M_j correspondem ao ótimo

Explicação: Entramos em (5) para descobrir um novo s , pode ser até o anterior, tal que $(\alpha)_{s-1}$ poderá nos conduzir a uma solução $(\alpha)_m$ melhor. Caso não exista tal s é porque evidentemente atingimos o ótimo.

- (6) $a_{s+1} = \left[\frac{L_t - \lambda(\alpha)_s}{I_{s+1}} \right], \dots, a_m = \left[\frac{L_t - (\lambda(\alpha)_s + I_{s+1}a_{s+1} + \dots + I_{m-1}a_{m-1})}{I_m} \right]$
 vá para (2)

Explicação: nesse passo, calculamos a maior lexicográfica extensão de $(\alpha)_s$, a partir de L_t . Como sabemos que ela melhora a valor de M_j para alguns j 's então desviamos para (2) a fim de atualizarmos esses M_j 's.

Observação: Foi verificado que, ao se aproximar da solução ótima, o algoritmo requer um tempo considerável para se obter pequenas evoluções. Assim, Gilmore e Gomory adotaram uma certa "cutoff heuristic", que encerra a programação de aplicação do algoritmo quando dez consecutivos passos de pivoteamento são aplicados sem produzir uma melhora de pelo menos 0.1 % no resultado obtido.

11.4 O Problema dos Cortes com Reaproveitamento

Uma nova proposta para o problema de cortes é comparada com o modelo anterior para o problema de cortes em uma dimensão, pari o qual Gilmore e Gomory propuseram um método de geração de colunas. Esta nova tentativa leva em conta as perdas obtidas nos diversos cortes, pois elas podem ser reaproveitadas na geração de novos pedaços.

Um processo real que implementaria o modelo anterior pode ser descrito pela ação simultânea de diversos cortadores em número ilimitado, que produziriam os pedaços solicitados.

Agora, nós imaginamos um novo processo de geração das pedaços demandados: um único cortador efetua um número ilimitado de cortes, Este processo pode ser melhor detalhado assim:

- (A) O número de cortes é ilimitado;
- (B) Em cada corte, pela menos um dos dois pedaços produzidos faz parte da demanda;
- (C) Pedaços que são gerados em um corte, e que não fazem parte da demanda podem ser cortados para se produzir novos pedaços;

- (D) Dado um pedaço de tamanho mantido em estoque, podem ocorrer dois casos: ou o pedaço trata-se de um item de estoque; ou o pedaço foi gerado por meio de um corte realizado em outro maior;
- (E) O estoque mantém uma quantidade infinita de peças;
- (F) A demanda D deve ser satisfeita;
- (G) O custo de produção da demanda é linearmente dependente da quantidade de itens de estoque consumidos;
- (H) $l_i < L_{max} \quad \forall i, i = 1, 2, \dots, m$;
- (I) $L_{min} > l_{min}$ e $l_i \neq L_k \quad \forall i, k$.

Este modelo difere do anterior pelas hipóteses (A - C).

Representaremos um corte de uma peça de comprimento k por $[k; l]$ onde l e $k - l$ são as dimensões dos produtos deste corte. Em inglês, o termo técnico adotado para definirmos este tipo de corte é "one-cut-patterns" ou "one-cuts".

A representação de um corte como sendo $[k; l]$ ou $[k; k - l]$ é escolhida arbitrariamente, desde que $l \in D$ e $k - l \in D$. Caso contrário, optamos por aquela representação em que o segundo elemento pertence a D .

Podemos definir uma medida de comprimento em termos da qual todos os tamanhos envolvidos sejam números inteiros.

Sejam:

- $S =$ conjunto dos itens de estoque $L \in \{L_1, L_2, \dots, L_k\} \subset Z$;
- $D =$ conjunto formado pelos pedaços demandados com tamanhos $l \in \{l_1, l_2, \dots, l_m\} \subset Z$;
- $R =$ conjunto formado pelos tamanhos residuais cujas dimensões são pelo menos iguais a l_{min} .

$$A_l = \begin{cases} \{k \in S \cup R; k > l\}, & \text{se } l \in D \\ \emptyset, & \text{se } l \notin D \end{cases}$$

$$B_l = \{k \in D; k + l \in S \cup R\}$$

$$C_l = \{k \in D; k < l\}$$

Seja uma peça de comprimento l . Denotemos por N_l a demanda por peças desse tipo. Assim:

$$N_l = \begin{cases} 0 & \text{para } l \notin D \\ N_l & \text{para } l \in D, l = l_i \end{cases}$$

Denotemos por c_l o custo de utilização da peça em estoque $l \in S$. Definimos por $y_{k,l}$ o número de vezes que o modo de cortar $[k; l]$ é executado em uma determinada solução, onde $l \in D$ e $k - l$ é um resto residual.

$$Z \ni y_{k,l} \geq 0, \text{ onde } k \in S \cup R, l \in D \text{ e } l < k \quad (\text{III.3})$$

A seguir, temos, se $l \in (D \cup R) \setminus S$.

$$\sum_{k \in A_l} y_{k,l} = \text{quantidade de peças de dimensão } l \text{ geradas para satisfazer a demanda de peças do tipo } l$$

$$\sum_{k \in B_l} y_{k+l,k} - \text{quantidade de peças de dimensão } l \text{ geradas como sendo peças residuais;}$$

$$\sum_{k \in C_l} y_{l,k} - \text{quantidade de peças de dimensão } l \text{ que são residuais e utilizadas na geração de outras peças que fazem parte da demanda}$$

$$\sum_{k \in A_l} y_{k,l} + \sum_{k \in B_l} y_{k+l,k} \geq \sum_{k \in C_l} y_{l,k} + N_l \quad \forall l \in (D \cup R) \setminus S \quad (\text{III.4})$$

O novo modelo tem como objetivo resolver o seguinte problema:

$$Z = \sum_{l \in S} c_l \cdot \left(\sum_{k \in C_l} y_{l,k} - \sum_{k \in B_l} y_{k+l,k} \right) \rightarrow \min! \quad (\text{III.5})$$

sujeito a III.3 e III.4,

onde

$$\sum_{k \in C_l} y_{l,k} = \text{número de peças com comprimento padrão } l \text{ utilizadas para se satisfazer a demanda}$$

$$\sum_{k \in B_l} y_{k+l,k} - \text{número de peças com comprimento padrão } l \text{ geradas como peças residuais}$$

$$\sum_{k \in C_l} y_{l,k} - \sum_{k \in B_l} y_{k+l,k} = \text{número de peças vindas do estoque com comprimento padrão } l \text{ utilizadas para se satisfazer a demanda}$$

Pela equação acima, se tivermos para um dado $l \in S$, $\sum_{k \in B_l} y_{k+l,k} \geq$

$$\sum_{k \in C_l} y_{l,k}, \text{ então } c_l \cdot \left(\sum_{k \in C_l} y_{l,k} - \sum_{k \in B_l} y_{k+l,k} \right) \leq 0.$$

O que significa dizer que o retorno para o estoque das peças com comprimento padrão $l \in S$ é levado em conta. Este valor retornado ao estoque diminui o custo total. Isto está de acordo com as hipóteses D e G.

Se não levarmos em conta o retorno das peças residuais com comprimento padrão ao estoque, então a função a ser minimizada torna-se:

$$\tilde{Z} = \sum_{l \in S} c_l \cdot \max\{0, \sum_{k \in C_l} y_{l,k} - \sum_{k \in B_l} y_{k+1,k}\}$$

11.5 Um Exemplo

Sejam dados:

$$S = \{5, 6, 9\} \text{ com } c_5 = 6, c_6 = 7, c_9 = 10 \text{ e}$$

$$D = \{2, 3, 4\} \text{ com } N_2 = 20, N_3 = 10, N_4 = 20$$

Então o conjunto de peças residuais R é o dado por:

$$R = \{7, 6, 5, 4, 3, 2\}$$

Os tipos de corte $[7; 4]$, $[6; 4]$, e $[5; 3]$ são similares a $[7; 3]$, $[6; 2]$ e $[5; 2]$. Escolhemos os três primeiros para serem utilizados na formulação do novo modelo.

minimizar $Z = 10y_{9,4} + 10y_{9,3} + 10y_{9,2} + 7y_{6,4} + 7y_{6,3} + 6y_{5,4} + 6y_{5,3} - 6y_{9,4} - 7y_{9,3} - 6y_{7,2} =$
 $4y_{9,4} + 3y_{9,3} + 10y_{9,2} + 7y_{6,4} + 7y_{6,3} + 6y_{5,4} + 6y_{5,3} - 6y_{7,2}$

sujeito a $y_{9,2} - y_{7,4} - y_{7,2} \geq 0$
 $y_{9,4} + y_{7,4} + y_{6,4} + y_{5,4} - y_{4,3} - y_{4,2} \geq 20$
 $y_{9,3} + y_{7,4} + 2y_{6,3} + y_{5,3} + y_{4,3} - y_{3,2} \geq 10$
 $y_{9,2} + y_{7,2} + y_{6,4} + y_{5,3} + 2y_{4,2} + y_{3,2} \geq 20$
 $y_{k,l} \geq 0$ para todos os cortes $[k; l]$

Resolvendo-se o PPL acima, obtemos:

$$y_{9,2} = 10, y_{7,2} = 10, y_{9,4} = 20, y_{9,3} = 10, z = 150$$

A solução ótima para o problema acima de acordo com o modelo anterior nos daria:

r cortar cada um dos 10 pedaços de tamanho de estoque 6 em um pedaço de tamanho 4 e um de tamanho 2;

• cortar cada um dos 10 pedaços de tamanho de estoque 9 em um pedaço de tamanho 2, um pedaço de tamanho 3 e um pedaço de tamanho 4.

O custo total dessa solução é 170.

III.6 Uma Técnica para se Determinar uma Solução Inicial para o Problema de Cortes

Suponhamos que todas as barras do estoque sejam de mesmo comprimento L e que a demanda tenha sido ordenada assim: $N_1 > N_2 > \dots > N_m$.

Obteremos uma base inicial B e uma solução inicial X em m iterações.

A cada iteração, obtemos primeiro uma coluna de B composta dos elementos a_1, a_2, \dots, a_m . Em seguida, com base no vetor $(N'_i)_{i=1,2,\dots,m}$, que é inicializado como $N'_i = N_i \forall i$, obtemos uma componente do vetor solução.

Em cada iteração, seja R um conjunto contendo índices relativos às componentes do vetor solução que ainda não foram determinadas.

Seja a iteração j , $j \in \{1, 2, \dots, m\}$. Desejamos determinar a j ª componente da vetor solução. Até o momento foram encontradas $j - 1$ componentes desse vetor. Portanto, em R devemos ter $m - (j - 1) = m - j + 1$ elementos associados a componentes que devem ser determinadas. Para determinarmos a coluna de B , $a = [a_1, a_2, \dots, a_m]^t$, fazemos assim:

$$a_i = \begin{cases} 0 & \text{se } i \notin R \\ \left[\frac{(L - \sum_{k=1}^{j-1} l_k a_k)}{l_i} \right] & \text{se } i \in R \end{cases}$$

Seja H tal que:

$$\frac{N'_k}{a_k} = \min \left\{ \frac{N'_i}{a_i}; i \in R \text{ e } a_i > 0 \right\}$$

Em seguida, temos:

$$R = R - \{k\} \text{ e } x_j = \frac{N'_k}{a_k},$$

a nova componente do vetor solução.

Substituímos N'_i por $N'_i - x_j a_i \forall i$, $i = 1, \dots, m$, e passamos à nova iteração.

Para o exemplo abaixo:

$$\begin{aligned} L &= 100 \\ l_1 &= 45 & N_1 &= 97 \\ l_2 &= 36 & N_2 &= 610 \\ l_3 &= 31 & N_3 &= 395 \\ l_4 &= 14 & N_4 &= 211 \end{aligned}$$

A solução inicial encontrada é a seguinte:

$$E_0 = \begin{bmatrix} a & \acute{u} & 0 & 0 \\ 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 3 \\ 0 & 2 & 0 & 0 \end{bmatrix} \quad X_0 = \begin{bmatrix} 48.5 \\ 105.5 \\ 199.5 \\ 131.67 \end{bmatrix}$$

Capítulo IV

Problemas de Corte em Duas e Três Dimensões

IV.1 O Problema Geral de Cortes em Duas Dimensões

Este capítulo foi baseado no trabalho de Gilmore e Gomory [18].

Neste problema, o estoque dispõe de uma quantidade infinita de objetos em forma de paralelepípedo de comprimento L e largura W . Queremos satisfazer uma demanda por N_i objetos em forma de paralelepípedo com a mesma altura que aqueles do estoque mas com comprimento l_i e largura w_i . O índice i toma valores no conjunto $(1, 2, \dots, 412)$.

O problema é modelado como segue. São efetuados cortes tipo guilhotina nas peças em estoque a fim de produzirmos a demanda. Dada uma peça estocada, existem diversas formas de se cortá-la. Sejam n os modos de cortar, onde cada modo de cortar é função dos pedaços que ele gera e do objeto em estoque sobre o qual ele atua. Associe a cada modo de cortar j , onde $j = 1, 2, \dots, n$, uma variável x_j que denota o número de vezes que esse tipo de corte é realizado em uma determinada solução do problema.

O objetivo consiste em se obter as variáveis x_1, x_2, \dots, x_n de tal modo que a demanda seja satisfeita com a utilização da menor quantidade possível de objetos estocados. Ou seja, se denotarmos

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad N = \begin{bmatrix} N_1 \\ N_2 \\ \vdots \\ N_m \end{bmatrix}$$

O problema pode ser escrito como:

minimizar $1 \cdot X$

sujeito a $AX = N$,

onde A é uma matriz $m \times n$ que cada coluna A_i representa um modo de cortar um objeto do estoque, da seguinte forma

$$A = [A_1 \ A_2 \ \dots \ A_n] \quad A_i = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix},$$

onde a_k é o número de pedaços com dimensões $l_k \times w_k$ gerados pelo i -ésimo modo de cortar, $1 \leq k \leq m$.

Se nós aplicarmos o método de geração de colunas da mesma forma como no caso de 1 dimensão, surge a necessidade de resolvermos um problema da mochila:

maximizar $\pi_1 a_1 + \dots + \pi_m a_m$

sujeito a $[a_1, a_2, \dots, a_m]$ corresponde a um modo de se cortar o objeto $L \times W$ do estoque onde a_i representa a quantidade de pedaços w_i com dimensões $l_i \times w_i$ obtidos

Seja $G(x, y)$ o valor obtido pela função mochila definida como acima para um estoque composto de peças w_i comprimento x e largura y . Se denotarmos por $g(x, y) = \max\{0, \pi_i; x \geq l_i \text{ e } y \geq w_i\}$ então poderemos reescrever:

$$G(x, y) = \max_{(x_0, y_0) \in \Gamma} \{g(x, y), G(x_0, y) + G(x - x_0, y), G(x, y_0) + G(x, y - y_0)\}$$

, onde $\Gamma = [0, \frac{x}{2}] \times [0, \frac{y}{2}]$

A demonstração do fato acima pode ser obtida estudando-se as três situações mutuamente excludentes:

(i) O primeiro corte realizado no objeto de dimensões x e y é perpendicular ao comprimento. Ele produz dois retângulos de comprimentos x_0 e $x - x_0$.

Conseqüentemente, $G(x, y) = G(x_0, y) + G(x - x_0, y)$.

(ii) O primeiro corte realizado no objeto de dimensões x e y é perpendicular à largura. Ele produz dois retângulos de largura y_0 e $y - y_0$.

Conseqüentemente, $G(x, y) = G(x, y_0) + G(x, y - y_0)$.

(iii) Não foi realizado nenhum corte. Logo: $G(x, y) = \max\{0, \pi_i; x \geq l_i \text{ e } y \geq w_i\}$.

IV.2 O Problema de Cortes Tipo Guilhotina em Dois Estágios

O corte tipo guilhotina em dois estágios pode ser descrito como segue. Uma série de cortes paralelos ao comprimento de um retângulo são produzidos. O resultado são tiras de um mesmo comprimento (igual ao do retângulo cortado). Este foi o primeiro estágio dos cortes. No segundo estágio, estas tiras são cortadas em uma direção paralela à largura. Os pedaços produzidos são utilizados para satisfazer a demanda.

Analogamente aos casos anteriores, para resolvermos este tipo de problema de cortes, basta resolvermos o correspondente problema da mochila descrito abaixo:

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

sujeito a $[a_1, a_2, \dots, a_m]$ corresponde a um modo de cortar tipo guilhotina em dois estágios

Para resolvermos a PPL acima, começamos reordenando os pedidos de modo a termos $u_1 \leq u_2 < \dots < u_m$. Neste método de corte por dois estágios, primeiro você corta a barra $L \times W$, de modo a obtermos tiras de dimensões $L \times W_{i_1}, L \times W_{i_2}, \dots, L \times W_{i_r}$ onde $W_{i_1}, W_{i_2}, \dots, W_{i_r} \in \{W_1, W_2, \dots, W_m\}$. No segundo estágio, você irá cortar cada uma dessas tiras na direção perpendicular ao comprimento de modo a termos a_1 quantidades de comprimento l_1 e largura maior do que ou igual a w_1 , a_2 quantidades de comprimento l_2 e largura maior do que ou igual a w_2 , até finalmente termos a_m quantidades de comprimento l e largura maior do que ou igual a w_m , satisfazendo:

$$F(L, W) = \pi_1 a_1 + \dots + \pi_m a_m \text{ onde } F \text{ é a função mochila}$$

Vamos supor que no primeiro estágio tivéssemos gerado alguma tira $L \times W_i$. Qual seria o melhor resultado a ser obtido dessa tira no segundo estágio? Se $w_j \leq w_i$ e $w_{j+1} > w_i$ então, devido à ordenação $w_1 \leq w_2 \leq \dots \leq w_m$, a resposta seria $\max\{x_1 \pi_1 + x_2 \pi_2 + \dots + x_j \pi_j; l_1 x_1 + l_2 x_2 + \dots + l_j x_j \leq L \text{ e } x_1, x_2, \dots, x_j \in Z\}$. Chamaremos esse valor de π_i^* .

A solução do problema abaixo nos deve dizer quais tiras foram geradas no primeiro estágio:

$$\text{maximizar } \pi_1^* b_1 + \pi_2^* b_2 + \dots + \pi_m^* b_m$$

$$\text{sujeito a } w_1 b_1 + w_2 b_2 + \dots + w_m b_m \leq W$$

Se (b_1, b_2, \dots, b_m) é a solução do problema acima então:

$$b_i > 0 \iff \text{a tira } L \times W_i \text{ foi gerada no primeiro estágio}$$

Sendo assim, fica claro como resolvermos o problema da mochila para cortes tipo guilhotina em dois estágios. E só realizarmos os dois passos abaixo:

1. Para cada i obtenha $\pi_i^* = \max\{x_1 \pi_1 + \dots + x_j \pi_j; l_1 x_1 + l_2 x_2 + \dots + l_j x_j \leq L \text{ e } x_1, x_2, \dots, x_j \in Z\}$ onde $w_j \leq w_i$ e $w_{j+1} > w_i$.
2. A solução do PPL acima fica dada por:

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

$$\text{sujeito a } w_1 b_1 + \dots + w_m b_m \leq W$$

IV.3 Formulação do Problema como um PPL em Estágios

Este novo método se destina a modelar o problema como um único PPL.

Seja A uma matriz contendo $2m$ linhas e particionada verticalmente em $m+2$ matrizes A_i . Seja \bar{X} um vetor coluna particionado horizontalmente em correspondência a A , em $\bar{x}_0, \bar{x}_1, \dots, \bar{x}_m, \bar{u}$, onde \bar{u} são as variáveis de folga. Seja N um $2m$ -vetor coluna particionado em um m -vetor de zeros e um m -vetor \bar{N} contendo os valores N_i solicitados na demanda, em ordem crescente das larguras dos retângulos solicitados.

Assim, podemos escrever:

$$A = [A_0 \ A_1 \ \dots \ A_m \ A_{m+1}] \quad \bar{X} = \begin{bmatrix} \bar{x}_0 \\ \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_m \\ \bar{u} \end{bmatrix} \quad N = \begin{bmatrix} 0 \\ \bar{N} \end{bmatrix}$$

As colunas de A_0 correspondem aos modos de cortar os retângulos em estoque de modo a termos as tiras geradas no primeiro estágio. Ou seja, a j -ésima coluna de A_0 é um $2m$ -vetor cujas m primeiras componentes b_1, b_2, \dots, b_m satisfazem $\sum_{i=1}^m b_i w_i \leq W$, e as demais são nulas.

As matrizes restantes A_s , ($1 \leq s \leq m$) que compõem A podem ser vistas como segue. Fixado um s , a matriz A_s se compõe de colunas que representam modos de cortar o retângulo de largura w_s , gerado no primeiro estágio. As colunas de A_s contém zeros nas m primeiras linhas, exceto a s -ésima linha que contém -1 . Há inteiros não-negativos a_{is} nas linhas $m+i$, $i = 1, 2, \dots, s$ (observe que estamos supondo $w_1 < w_2 < \dots < w_m$) e zeros abaixo, satisfazendo $\sum_{i=1}^s a_{is} l_i \leq L$.

Finalmente, A_{m+1} é uma matriz $2m \times m$, onde suas m primeiras linhas são todas nulas e as restantes formam a matriz $-I$. Esta é uma matriz de coeficientes para as variáveis de folga das últimas m equações no PPL abaixo.

O problema pode ser colocado então:

$$\text{minimizar} \quad 1 \cdot \bar{x}_0$$

$$\text{sujeito a} \quad A \cdot \bar{X} = N$$

As primeiras m equações estabelecem que apenas as tiras resultantes do corte no primeiro estágio são usadas no segundo. As restantes m equações estabelecem que a demanda é satisfeita a partir de cortes nestas tiras realizados no segundo estágio.

IV.4 O Problema de Cortes em 3 Dimensões a 3 Estágios

Este problema consiste em se satisfazer uma demanda de paralelepípedos de dimensões $l_i \times w_i \times h_i$, $i = 1, 2, \dots, m$, a partir de cortes em um paralelepípedo de dimensões $L \times W \times H$. Os cortes são feitos de modo semelhante ao anterior: apenas que são realizados em 3 estágios.

Analogamente aos casos anteriores, para resolvermos este tipo de problema de cortes, basta resolvermos o correspondente problema da mochila, descrito abaixo:

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

sujeito a $[a_1, a_2, \dots, a_m]$ corresponde a um modo de cortar tipo guilhotina em 3 estágios

O problema acima pode ser resolvido da seguinte forma:

- (i) Para cada i , considerem um paralelepípedo de dimensões $l_i \times w_i \times H$. Denotemos por $v(l_i, w_i, h_i)$ o valor retornado pelo paralelepípedo de dimensões $l_i \times w_i \times h_i$. Denotemos também por A_i o conjunto $\{j; \text{ o retângulo de dimensões } l_j \text{ e } w_j \text{ está contido no retângulo de dimensões } l_i \text{ e } w_i\}$. Seja $v(l_i, w_i)$ a solução do problema abaixo:

$$\begin{aligned} \text{maximizar} \quad & v(l_i, w_i) = \sum_{j \in A_i} v(l_j, w_j, h_j) \cdot z_j \\ \text{sujeito e} \quad & \sum_{j \in A_i} z_j \cdot h_j \leq H \\ & 0 \leq z_j \in Z, \forall j \in A_i \end{aligned}$$

- (ii) Resolva o problema da mochila em duas dimensões: extrair do retângulo $L \times W$ pedaços $l_i \times w_i$, $i = 1, 2, \dots, m$, w m valores dadas $v(l_i, w_i)$ de modo que a soma dos valores dos pedaços obtidos seja máxima. Como o corte será realizado por estágios, a solução desse problema é a dada na seção anterior.

Capítulo V

O Problema da Mochila em Duas Dimensões com Restrições

V.1 Definição do Problema

Este capítulo foi baseado no trabalho de Nicos Christofides e Charles Whitlock [5].

Consideremos um retângulo $A_0 = (L_0, W_0)$ e um conjunto de pedaços $R = \{(l_1, w_1), (l_2, w_2), \dots, (l_m, w_m)\}$, onde cada pedaço (l_i, w_i) em R tem um valor v_i . O problema consiste em:

$$\text{maximizar } z = \sum_{i=1}^m \xi_i \cdot v_i$$

sujeito a:

- $\xi_i \leq b_i, \forall i, i = 1, 2, \dots, m;$
- existem uma série de cortes tipo guilhotina sobre A_0 que produzem ξ_i pedaços de dimensões l_i e $w_i;$
- $0 \leq \xi_i \in \mathbb{Z}$

Daqui para frente, chamaremos os elementos do conjunto R de pedaços. Os elementos produzidos por cortes em A_0 chamaremos de retângulos. Suporemos que L_0, W_0, l_i e w_i sejam inteiros, qualquer que seja i .

A orientação dos pedaços será considerada fixa, isto é, um pedaço de comprimento l e largura w não é o mesmo pedaço de comprimento w e largura l .

V.2 Procedimento Enumerativo

Os tipos possíveis de corte poderiam ser representados por uma árvore, onde os ramos dessa árvore indicam os tipos de corte e os nós finais desses ramos representam o retângulo gerado por esse corte.

Um tipo de corte que é muito importante no algoritmo que será desenvolvido é aquele que deixa o retângulo intacto. Chama-lo-emos de O-corte. Assim, um retângulo que sofreu a ação de corte O-corte não será, candidato a sofrer futuros cortes, e deverá ser considerado fixo a partir do nó em que se encontra em diante.

V.3 Efeitos de Simetria

Chamamos de um x -corte um corte transversal ao eixo dos x 's. Analogamente temos um y -corte.

Dado um retângulo (p, q) , um corte $x = a$ produz dois retângulos (a, q) e $(p - a, q)$. Estes mesmos retângulos poderiam ter sido produzidos por um corte em $x = p - a$. Assim, limitamos os x -corte para $1 \leq x \leq [p/2]$ e os y -corte para $1 \leq y \leq [q/2]$.

V.4 Efeitos na Ordenação dos Cortes

Suponha um corte no retângulo (p, q) em $x = a$, produzindo os retângulos (a, q) e $(p - a, q)$. Peguemos este último retângulo $(p - a, q)$ e realizemos nele um corte em $x = a + b$ com $0 < b < [?]$. Assim, passamos a ter os seguintes retângulos (a, q) , (b, q) , e $(p - a - b, q)$.

Realizemos agora em (p, q) uma outra seqüência de cortes. Primeiro em $x = b$, produzindo (b, q) e $(p - b, q)$. Segundo, em $x = b + a$, sobre $(p - b, q)$, produzindo (a, q) e $(p - b - a, q)$. Ao todo, produzimos os mesmos retângulos que a seqüência anterior: (a, q) , (b, q) e $(p - a - b, q)$.

Para evitar que tal ocorra, estabeleçamos a seguinte ordenação nos cortes que são realizados: dado um retângulo (p, q) , se foi realizado um corte em $x = \alpha$, os próximos cortes a serem feitos nos retângulos remanescentes (a, q) e $(p - \alpha, q)$ deverão ocorrer para $x \geq 2\alpha$. Ou seja, o próximo retângulo a ser cortado só poderá ser $(p - \alpha, q)$, desde que $\alpha \leq [p/2]$, em um ponto x tal que $2\alpha \leq x \leq [\alpha + p/2]$.

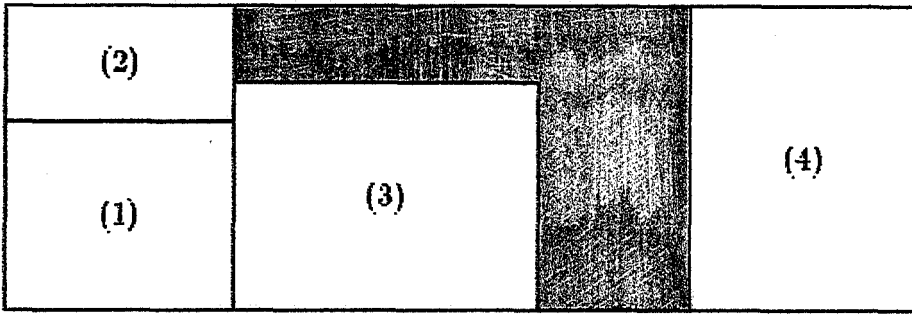
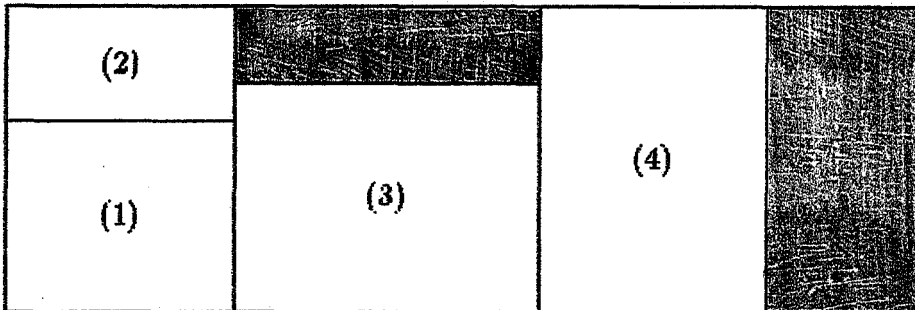


Figura V.1: Tipo de Corte Não Normalizado



Figura, V.2: Forma Normalizada do Tipo de Corte Acima

V.5 Cortes Normais

Um corte a em (p, q) é dito normal quando existe uma combinação de peças (l_j, w_j) , com $j = 1, 2, \dots, r$, satisfazendo $w_j \leq q \forall j, j = 1, 2, \dots, r$, e $a = \sum_{j=1}^r l_j$. Todo tipo de corte não normalizado possui uma forma equivalente normalizada (ver figura V.2). Daqui para frente, restringiremos os cortes a serem feitos no eixo dos x 's e dos y 's apenas aos cortes normais.

Notação:

$S^q =$ conjunto dos pontos no eixo dos x 's onde os cortes realizados têm a característica de serem normais.

$T^p =$ análogo ao anterior, só que para o eixo dos y 's

Coma nós obtemos o conjunto S^q ? Assuma que $w_1 \leq w_2 \leq \dots \leq w_m$. Seja $1 \leq r \leq m$. Definimos:

$$f_r(x) = \begin{cases} w_{i^*} & \text{se existem } \bar{\xi}_i \text{ satisfazendo:} \\ & x = \sum_{i=1}^r \bar{\xi}_i l_i, \quad 0 \leq \bar{\xi}_i \leq b_i \text{ e } \bar{\xi}_i \in Z \text{ onde } i^* = \min_r [\max\{i; \bar{\xi}_i \neq 0\}] \\ \infty & \text{se não existem tais } \bar{\xi}_i \end{cases}$$

Lema V.1 A tabela $f_i(x)$, $1 \leq i \leq m$, $0 \leq x \leq L_0$ pode ser gerado pela recursão:

$$f_i(x) = \begin{cases} \min\{f_{i-1}(x), \max\{w_i, \min_k(f_{i-1}(x - kl_i))\}\}, & \text{onde} \\ \quad 1 \leq k \leq \min\{b_i, [\frac{x}{l_i}]\}, k \in Z \text{ se } x \geq l_i \\ f_{i-1}(x) \text{ se } x < l_i \end{cases}$$

$$f_0(x) = \infty$$

Demonstração

Suponha $x \geq 4$. Considere as afirmações abaixo:

(i) $f_i(x) \leq f_{i-1}(x)$

Se $f_{i-1}(x) = \infty$, o resultado é óbvio.

Se $f_{i-1}(x) < \infty$ então $f_{i-1}(x) \leq w_{i-1}$ o que implica em $f_i(x) \leq w_{i-1}$ e, portanto, $f_i(x) = f_{i-1}(x)$.

(ii) $f_i(x) \leq \max\{w_i, \min_k(f_{i-1}(x - kl_i))\}$

O que ocorre se e só se $f_i(x) \leq w_i$ ou $f_i(x) \leq \min_k(f_{i-1}(x - kl_i))$

Suponha $f_i(x) > w_i$. Então $f_i(x) = \infty$. Logo, $f_{i-1}(x - kl_i) = \infty, \forall k$.

Portanto, $f_i(x) = \min_k(f_{i-1}(x - kl_i)) = \infty$.

(iii) A ocorrência simultânea de

$$\begin{cases} f_i(x) < f_{i-1}(x) \\ \text{e} \\ f_i(x) < \max\{w_i, \min_k(f_{i-1}(x - kl_i))\} \end{cases} \quad (*)$$

não é possível. A demonstração será feita por absurdo. De fato, temos:

$$\begin{cases} f_i(x) < f_{i-1}(x) \Rightarrow f_i(x) < \infty \Rightarrow f_i(x) \leq w_i \\ f_i(x) < f_{i-1}(x) \Rightarrow f_{i-1}(x) = \infty \end{cases}$$

Se $f_i(x) \leq w_i$ e $f_{i-1}(x) = \infty$ então $f_i(x) = w_i$. Assim, por (*),

$$\begin{aligned} \min_k(f_{i-1}(x - kl_i)) > w_i &\Rightarrow f_{i-1}(x - kl_i) > w_i \quad \forall k \Rightarrow \\ \Rightarrow f_{i-1}(x - kl_i) = \infty \quad \forall k, 1 \leq k \leq \min\{b_i, [\frac{x}{l_i}]\} &\quad (**) \end{aligned}$$

Tome $k \in Z$ satisfazendo $x = \sum_{j=1}^{i-1} \bar{\xi}_j l_j + kl_i$ para alguns $\bar{\xi}_j$ tal que $0 \leq \bar{\xi}_j \leq b_j$,

$\bar{\xi}_j \in Z, j = 1, \dots, i-1$ e $1 \leq k \leq b_i$. Isso é possível, pois $f_i(x) = w_i$.

Por (**), para esse k deveríamos ter $f_{i-1}(x - kl_i) = \infty$, o que é absurdo.

Por (i), (ii) e (iii) fica provado o lema.

[C.Q.D.]

Segue-se da definição acima, que se $f_m(x) \leq q$ então x se escreve como $x = \sum_{i=1}^m \bar{\xi}_i l_i$, satisfazendo $0 \leq \bar{\xi}_i \leq b_i$, $\bar{\xi}_i \in Z$ e $w_i \leq q$ se $\bar{\xi}_i > 0$. Logo, nesse caso, teremos $x \in S^q$.

Assim: $S^q = \{x; f_m(x) \leq q\}$.

Analogamente, calculamos T^p .

V.6 Descrição do Algoritmo Enumerativo

A cada nó n em nosso algoritmo estará associado um conjunto L de retângulos. Cada elemento de L , será representado por um conjunto ordenado (p, q, x, y) onde p é o comprimento e q a largura do retângulo em questão e x e y são inteiros tomando valores nos intervalos $0 \leq x \leq [p/2] + 1$, $1 \leq y \leq [q/2] + 1$.

$1 \leq x \leq [p/2]$	—	se esse retângulo foi escolhido para ser cortado, o próximo corte será um x -corte.
$x = [p/2] + 1$ e $1 \leq y \leq [q/2]$	—	o próximo corte a ser feito será um y -corte
$x = [p/2] + 1$ e $y = [q/2] + 1$	—	todos os x -cortes e y -cortes viáveis foram realizados. O próximo corte a ser realizado é um 0 -corte.
$x = 0$	—	(um 0 -corte foi realizado) o retângulo (p, q) em questão não será mais cortado

V.7 Algoritmo de Busca em Árvore

Dado um nó de nossa árvore, seja H_0 o conjunto dos retângulos de L para os quais já foi realizado um 0 -corte. Na solução a ser obtida seguindo-se este nó, teremos um pedaço de a de a a cada retângulo em H_0 . Para obtermos uma cota superior dos valores que podem ser obtidos das retângulos que compõem H_0 , procuramos preencher os pedaços nos retângulos de uma forma ótima.

Forme uma matriz $[a_{ik}]$ com m linhas correspondentes aos pedaços em R e u colunas correspondentes aos retângulos (p_k, q_k) em H_0 . Faça $a_{ik} = v_i$, se $(l_i \leq p_k$ e $w_i \leq q_k)$ e $a_{ik} = -\infty$, caso contrário.

A melhor forma de alocar pedaços aos retângulos é dado pela solução do problema dos transportes:

$$\begin{aligned} \text{maximizar } z &= \sum_{k=1}^u \sum_{i=1}^m \alpha_{ik} x_{ik} \\ \text{sujeito a } \sum_{i=1}^m x_{ik} &\leq 1 \\ \sum_{k=1}^u x_{ik} &\leq b_i \\ Z \ni x_{ik} &\geq 0 \end{aligned}$$

Uma cota superior para cada um dos retângulos que compõem $L - H_0$ pode ser obtida resolvendo-se um problema de cortes de duas dimensões sem restrições utilizando-se o procedimento em programação dinâmica dado por Gilmore e Gomory [16].

Chamemos z^* a cota superior obtida do conjunto $L = \{ \text{solução do problema dos transportes} + \text{solução dos problemas de programação dinâmica} \}$. Convencionemos como sendo \bar{z} o maior valor obtido até o momento de (L_0, W_0) , respeitando-se as restrições. Temos as seguintes situações possíveis:

- (i) $z^* \leq \bar{z}$. Se a melhor forma de cortar partindo-se dos retângulos atuais não consegue produzir um valor maior do que o melhor obtido até o momento, então não há necessidade de se efetuar novos cortes. Ou seja, devemos retornar aos nós anteriores (backtracking).
- (ii) $z^* > \bar{z}$ e as restrições b_i estão sendo respeitadas nos cortes que originam z^* . Se a melhor forma de cortar é viável e supera \bar{z} , então ela deve passar a ser o nosso ótimo atual. Isto é, devemos fazer $\bar{z} \leftarrow z^*$. O modo de cortar que origina z^* deve também ser guardado como o melhor modo de cortar obtido até o momento. Devemos retornar (backtracking).
- (iii) $z^* > \bar{z}$ e as restrições b_i não estão sendo respeitadas nos cortes que originam z^* . A melhor forma de cortar não é viável. Nesse caso, ela não nos interessa. Porém, como ela supera \bar{z} , devemos continuar fazendo cortes a partir do nó atual, pois é possível que ainda encontremos uma solução que, mesmo não sendo tão boa quanto a nossa atual cota superior z^* , ainda assim supere \bar{z} e tenha a vantagem de ser viável.

V.8 Métodos de Ramificação

O processo de escolha de um retângulo de $L - H_0$ para ser cortado pode ser feito de diversas formas:

- (i) Tomamos a retângulo (p_n, q_n) com a menor dimensão p_n no eixo dos x 's. Em caso de empate, tomamos aquele com a menor dimensão q_n . Analogamente, poderíamos tomar com a maior dimensão.
- (ii) A cada retângulo é associado um valor, que R o dado pela solução do problema de programação dinâmica. Escolhemos aquele retângulo com valor máximo associado.
- (iii) Para cada $i = 1, 2, \dots, m$ sejam r'_i e r''_i a quantidade de pedaços i obtidas, respectivamente, da solução do problema dos transportes e dos problemas de programação dinâmica dos retângulos de $L = H_0$. Como estamos escolhendo um retângulo para ser cortado, devemos ter, para algum i , $r'_i + r''_i > b_i$. Do contrário, não haveria necessidade de se efetuar um novo corte. Seja i^* tal que $b_{i^*} - (r'_{i^*} + r''_{i^*})$ é mínimo. Tomemos o retângulo que contém a maior quantidade de pedaços (l_{i^*}, w_{i^*}) fornecidos pela solução do problema de programação dinâmica. Se d e for cortado, é de se esperar que a quantidade de pedaços (l_{i^*}, w_{i^*}) diminua quando formos analisar os retângulos oriundos desse corte (os cortes são feitos de modo que as cotas superiores b_i não sejam ultrapassadas).

V.9 Uma Rotina para Solução do Problema

A variável c_i contém o número de pedaços do tipo i obtidos do cálculo da cota superior para o conjunto L , isto é, é a soma dos pedaços obtidos da solução do problema dos transportes com aqueles obtidos da solução dos problemas de programação dinâmica.

INICIALIZAÇÃO

(1.1) Faça: $n = 1$, $L = ((L^*, W_0, 1, 1))$, $\bar{z} = 0$

CÁLCULO DA COTA SUPERIOR

(2.1) Calcule o valor de r^* e c_i , $i = 1, 2, \dots, m$ para a lista L .

(2.2) Se $z^* \leq 3$ ponha $k = 0$ e vá para (6.1); caso contrário, continue.

Explicação: Se o melhor valor obtido até o momento (\bar{z}) for superior ou igual à cota máxima r^* possível de ser obtida então não há nada que possamos fazer a partir do nó em que nos encontramos. Efetuamos um backtracking a partir de (6.1). O flag $k = 0$ nos diz que não será feita a recuperação de nenhum nó.

(2.3) Se $b_i \geq 3r_i$, $\forall i = 1, 2, \dots, m$, faça $\bar{z} = z^*$, $k = 0$ e vá para (6.1); caso contrário, continue.

Explicação: Se, no cálculo da cota máxima z^* , a quantidade c_j de pedaços (l_j, w_j) for inferior ou igual ao limite b_j , então o modo de cortar as peças para,

produzir z^* está de acordo com as condições impostas. Levando-se em conta que para se chegar a (2.3) devemos ter $z^* > \bar{z}$, então fazemos $\bar{z} = z^*$. Vamos iniciar o backtracking a partir de (6.1). Caso as condições mencionadas não se confirmem, como já temos $z^* > \bar{z}$, vamos iniciar os cortes para tentar obter um modo de cortar que também extrapole \bar{z} , mas que dessa vez respeite as restrições.

ESCOLHA UM RETÂNGULO PARA CORTAR

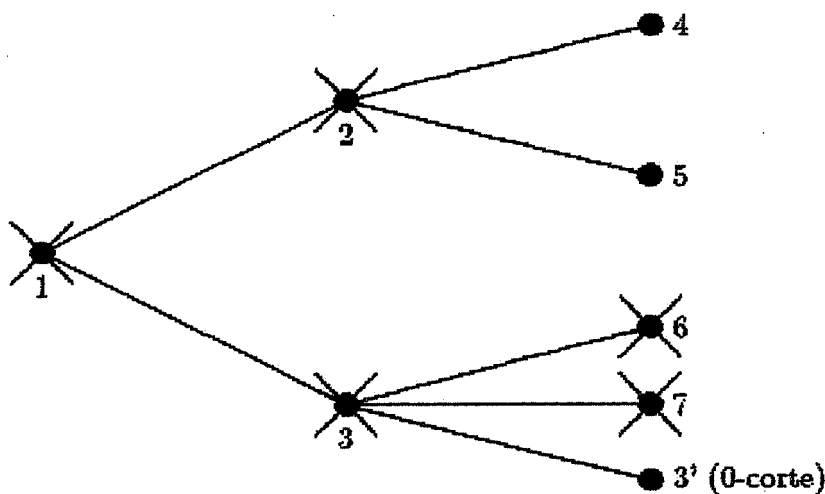
(2.4) Escolha um retângulo (p_n, q_n, x_n, y_n) da lista L com $x_n \neq 0$, e vá para (2.5); caso contrário, ponha $k = 0$ e vá para (6.1).

Explicação: Um retângulo presente em L com $x_n = 0$ é um que não mais será curtido, nem mesmo um 0-corte. Portanto, se todos os retângulos em L tiverem $x_n = 0$, não há nada mais a ser feito e retornamos (backtracking).

Caso obtenhamos um retângulo com $x_n \neq 0$, vamos cortá-lo.

(2.5) Faça $X_n = x_n$ e $Y_n = y_n$

Explicação: A finalidade de X_n e Y_n pode ser explicada pela animação abaixo. Suponha a seguinte situação:



O último nó cortado foi o nó 3. Ele gerou o nó 3', através de 0-corte. O nó 3' é retirada de L . A seguir, alguns passos executados pelo algoritmo e mostrados como atuariam na situação acima:

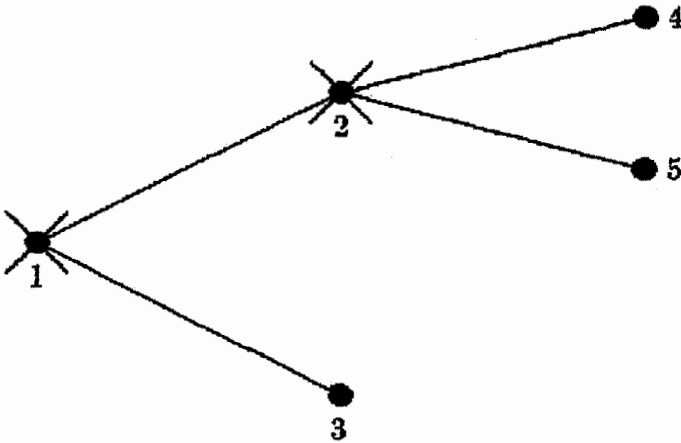
(6.5.3) Faça $L = L - \underbrace{\{(p_n, q_n, 0, y_n)\}}_{3'}$, $k = 1$ e vá para (6.1)

O valor n só pode se referir ao nó 3, que é o último que foi cortado.

(6.1) $n = 1$? Resposta: não.

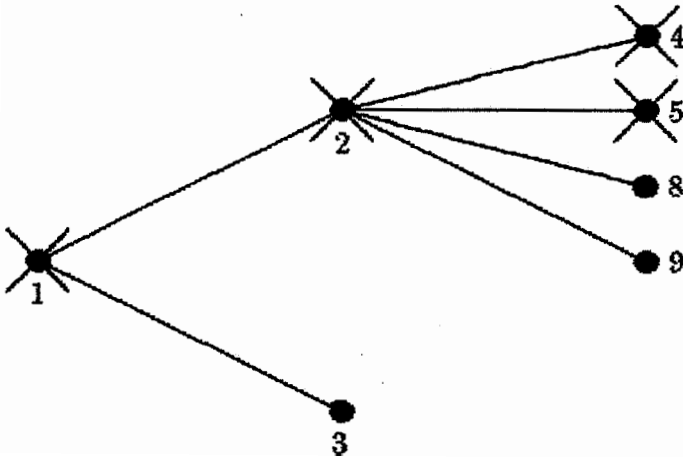
(6.2) $k = 0$? Resposta: não.

(6.3) $L = L \cup \{(p_n, q_n, X_n, Y_n)\}$. Você recuperou o nó 3 a partir do primeiro corte. Logo, você poderá cortá-lo de todas as formas possíveis novamente, caso haja necessidade, a partir de uma avaliação em (2.1). Ficamos com:



(6.4) Ponha $n = n - 1 =$ nó 2.

(6.5) Retiram-se os nós 4 e 5 e, eventualmente, faz-se um novo corte:



Voltamos para (2.1). Nesse ponto, pode-se chegar à conclusão de que são possíveis novos cortes para se melhorar o resultado até o momento. Inclusive em (3).

Portanto, $L = L \cup \{(p_n, q_n, X_n, Y_n)\}$ serve para recuperarmos um nó que seria útil no caso de realizarmos novos cortes.

(2.6) Faça $L = L - \{(p_n, q_n, x_n, y_n)\}$

Explicação: Se o retângulo (p_n, q_n, x_n, y_n) for cortado então ele deve desaparecer de L . É para isso que temos a declaração $L = L - \{(p_n, q_n, x_n, y_n)\}$. Em seu lugar, serão incluídos os produtos desse corte.

"FORWARD BRANCHING (X-CORTE)"

(3.1) Faça $x = x_n$.

(3.2) Se $r \geq \lfloor \frac{2x}{2} \rfloor + 1$ então faça $x_n = x + 1$ e vá para (4.1)

Explicação: Conforme dissemos, se $x \geq \lfloor \frac{2x}{2} \rfloor + 1$ então não será realizado nenhum r-corte. Sendo assim, nos deslocamos para o item (4.1) e verificamos se é possível ou não fazer-se um y-corte. O valor de x_n é incrementado de 1 para que em (6.5) se faça uma avaliação correta de qual foi o último corte realizado.

(3.3) Se $x \in S^{q_n}$ vá para (3.4): caso contrário, faça $a_i = a_i + 1$ e vá para (3.2).

Explicação: Se $x \notin S^{q_n}$ então o corte não é normal. Assim, este corte possui um equivalente normal diferente dele. Para se evitar duplicações, exigimos apenas que o corte normal seja feito.

(3.4) Faça $L = L \cup \{(x, q_n, x, 1), (p_n - x, q_n, x, 1)\}$

(3.5) Faça $x_n = x + 1$, $n = n + 1$ e vá para (2.1).

"FORWARD BRANCHING (Y-CORTE)"

(4.1) Faça $y = y_n$.

(4.2) Se $y \geq \lfloor \frac{2y}{2} \rfloor + 1$ vá para (5.1)

Explicação: De acordo com o que foi dito anteriormente, como já temos $x \geq \lfloor \frac{2x}{2} \rfloor + 1$, se $y \geq \lfloor \frac{2y}{2} \rfloor + 1$, o próximo corte será um 0-corte. Portanto, iremos para (5.1).

(4.3) Se $y \in T^{p_n}$ vá para (4.4); caso contrário, faça $y = y + 1$ e vá para (4.2).

(4.4) Faça $L = L \cup \{(p_n, y, 1, y), (p_n, q_n - y, 1, y)\}$.

(4.5) Faça $y_n = y + 1$, $n = n + 1$ e vá para (2.1).

"FORWARD BRANCHING (0-CORTE)"

(5.1) Faça $L = L \cup \{(p_n, q_n, 0, y_n)\}$.

Explicação: O valor $x_n = 0$ indica que eu realizei um 0-corte.

(5.2) Faça $x_n = 0$, $n = n + 1$ e vá para (2.1).

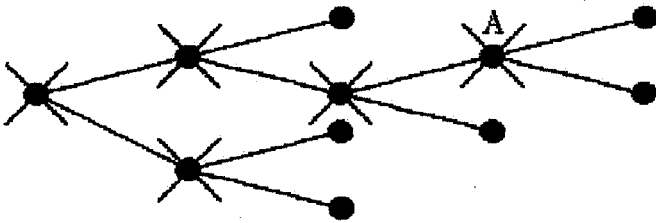
Explicação: Fazemos $e = 0$ para o caso de no item (2) haver um desvio para o item (6) e sabermos ai qual foi o último corte realizado.

"BACKTRACKING"

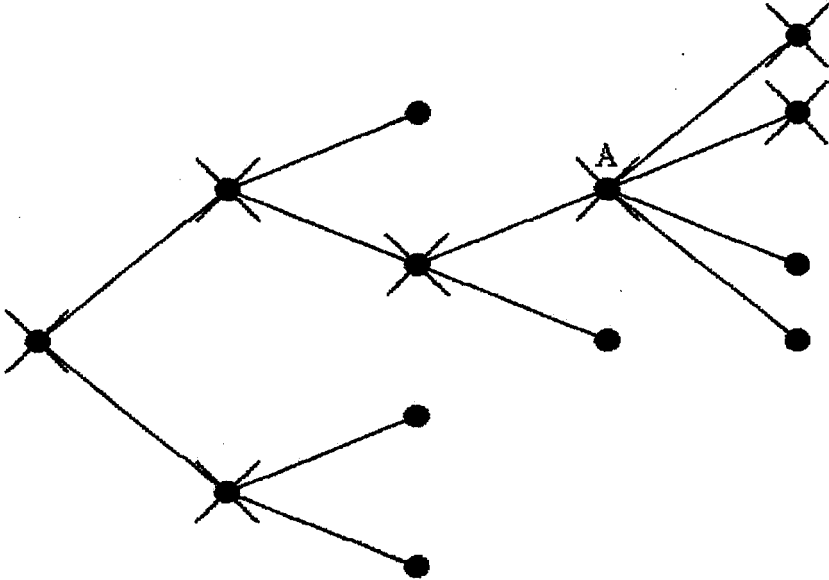
(6.1) Se $n = 1$, pare; todos os cortes normais foram gerados.

Explicação: Em linhas gerais, o algoritmo funciona assim:

Bom?primeiro ele gera uma árvore até que não haja mais sentido em se cortar. Antes do início do primeiro backtracking.

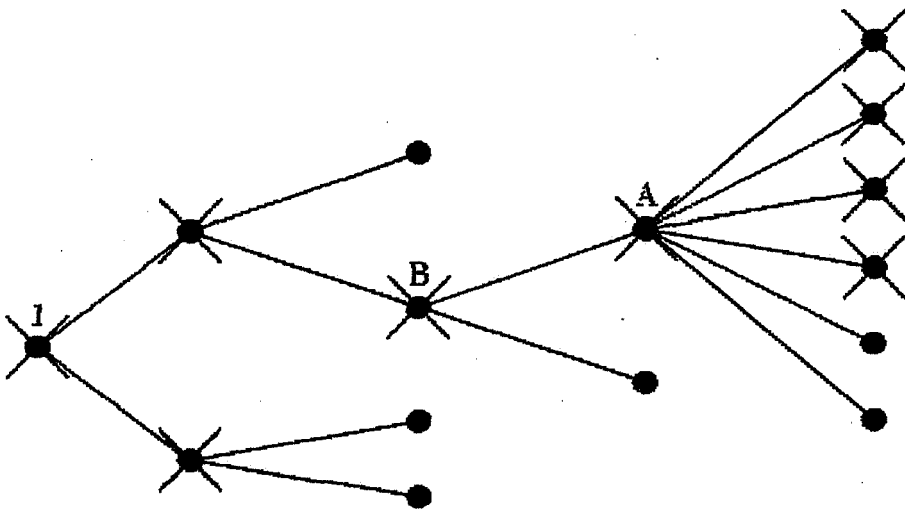


Suponhamos que o último nó cortado tenha sido o nó A. O próximo passo é fazermos um backtracking para A e depois gerarmos mais dois retângulos:



Analisamos o diagrama acima. Estudamos todas as possibilidades de corte a partir dele.

Após isso, novo backtracking para A e a geração de novos cortes:



Depois de termos esgotado todas as possibilidades para A, esgotaremos todas as possibilidades para o nó anterior a A (possivelmente B). E assim por diante, até chegarmos ao nó raiz 1.

(6.2) Se $k = 0$, vá para (6.4).

(6.3) $L = L \cup \{(p_n, q_n, X_n, Y_n)\}$.

(6.4) Faça $n = n - 1$

(6.5) Se $x_n > \lfloor \frac{p_n}{2} \rfloor + 1$, vá para (6.5.2). Se $x_n = 0$, vá para (6.5.3); caso contrário, continue.

O ÚLTIMO CORTE FEITO FOI UM X-CORTE

(6.5.1) Faça $L = L - \{(x_n - 1, q_n, x_n - 1, 1), (p_n - x_n + 1, q_n, x_n - 1, 1)\}$ e vá para (3.1).

O ÚLTIMO CORTE FEITO FOI UM Y-CORTE

(6.5.2) Faça $L = L - \{(p_n, y_n - 1, 1, y_n - 1), (p_n, q_n - y_n + 1, 1, y_n - 1)\}$ e vá para (4.1)

O ÚLTIMO CORTE FEITO FOI UM 0-CORTE

(6.5.3) Faça $L = L - \{(p_n, q_n, 0, y_n)\}$, $k = 1$ e vá para (6.1)

Explicação: Nós fazemos $k = 1$ para que possamos recuperar o nó que deu origem ao 0-corte.

V.10 RESUMO FINAL

O algoritmo anterior pode ser esquematizado assim:

(1) Podemos gerar uma melhora a partir das peças existentes em L ?

Se a resposta for não vá para (3).

(2) Efetua-se um novo corte. Vá para (1).

(3) BACKTRACKING

(3.1) $n = 1$? Se a resposta for não vá para (3.3).

(3.2) FIM DO ALGORITMO.

(3.3) Localize o nó pai do último corte realizado.

(3.4) O último corte foi um x -corte ou y -corte ?

Se a resposta for sim, vá para (3.7).

(3.5) Extraia o nó correspondente ao 0-corte.

(3.6) Recupere a nó da qual foi extraído a 0-corte.

Vá para (3.3).

(3.7) Extraia do conjunto L os 2 filhos do nó pai correspondentes ao último corte realizado

Vá para (2).

Capítulo VI

Dois Algoritmos para o Problema da Mochila em Duas Dimensões com Restrições

VI.1 Definições

Este capítulo foi baseado no paper escrito por P. Y. Wang [27]. A última seção foi baseada no trabalho de José Fernando Oliveira e José Soeiro Ferreira [15].

Sejam $L_0 \times W_0$ um tipo de retângulo em estoque e R um conjunto de retângulos R_1, R_2, \dots, R_m com dimensões $l_1 \times w_1, l_2 \times w_2, \dots, l_m \times w_m$.

Queremos determinar uma forma de se extrair retângulos R_i do item de estoque $L_0 \times W_0$ com o mínimo de sobras possível. O problema pode ser formulado assim:

$$\text{maximizar } \sum_{i=1}^m \xi_i \cdot l_i w_i$$

- sujeito a
- (i) $0 \leq \xi_i \leq b_i, \xi_i \in \mathbb{Z} \quad \forall i, i = 1, 2, \dots, m$
 - (ii) existem uma serie de cortes tipo guilhotina sobre $L_0 \times W_0$ que produzem ξ_i pedaços de dimensões l_i e w_i .

Definição:

Seja dado um modo de cortar G de um item de estoque. O seu correspondente *retângulo guilhotina* S é o menor retângulo que contém os pedaços R_i gerados pelo modo de cortar G , mantidas as posições relativas que os pedaços R_i guardam entre si (figuras VI.1 e VI.2).

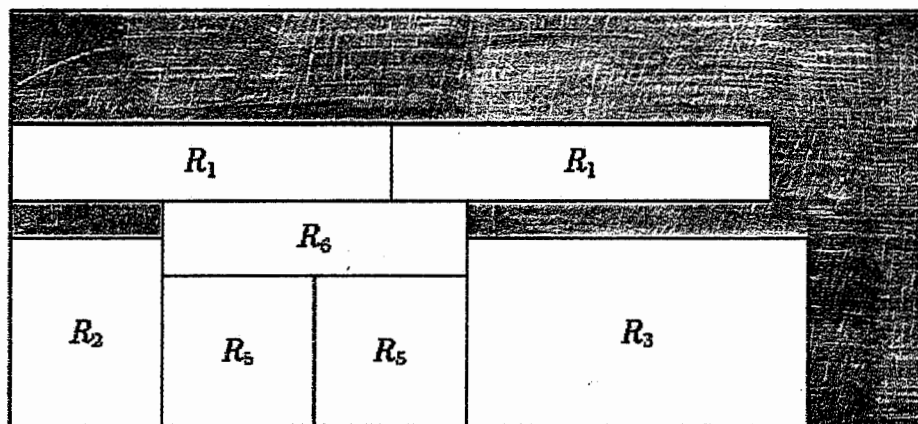


Figura VI.1: Um tipo de corte normalizado G

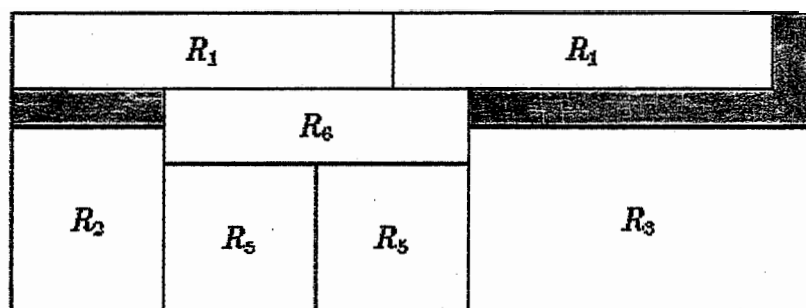


Figura VI.2: O retângulo guilhotina S associado a G

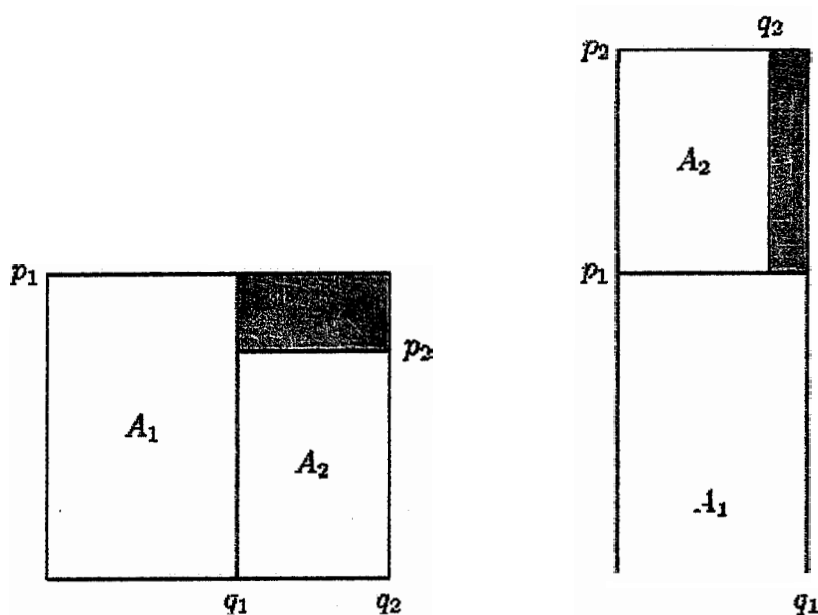


Figura VI.3: Uma construção horizontal e outra vertical de A_1 e A_2

Definição:

Uma *construção horizontal* de dois retângulos $A_1 = p_1 \times q_1$ e $A_2 = p_2 \times q_2$ é um retângulo S_h com dimensões $\max(p_1, p_2) \times (q_1 + q_2)$ e contendo A_1 e A_2 . Uma *construção vertical* é um retângulo S_v com dimensões $(p_1 + p_2) \times \max(q_1, q_2)$ também contendo A_1 e A_2 (veja a figura, VI.3).

Os tipos de corte no retângulo $L_0 \times W_0$ serão representados por um retângulo guilhotina S obtido a partir de construções horizontais e verticais dos pedaços R_1, R_2, \dots, R_n . Isso é possível, porque todos os cortes tipo guilhotina possuem uma forma normalizada equivalente.

Os retângulos guilhotina S gerados poderão ter sobras. As sobras devem satisfazer um teto máximo, que pode ser β_1 ou β_2 , dependendo do algoritmo utilizado. A percentagem das sobras do retângulo guilhotina S em relação a área do item de estoque $L_0 \times W_0$ chamamos de β_1 . A percentagem das sobras do retângulo guilhotina S em relação a área $a(S)$ do próprio retângulo S chamamos de β_2 .

Os métodos a seguir foram propostos por P. Y. Wang.

VI.2 Os Algoritmos

ALGORITMO UM

PASSO (1)

- 1.a Escolha um β_1 , $0 \leq \beta_1 \leq 1$.
- 1.b Tome $L^{(0)} = F^{(0)} = \{R_i; b_i \geq 1\}$ e faça $k = 1$.

PASSO (2)

- 2.a Obtenha o conjunto $F^{(k)}$ de todos os retângulos T satisfazendo:
 - (i) T é formado a partir da construção horizontal ou vertical de dois retângulos de $L^{(k-1)}$ e está contido no item de estoque $L_0 \times W_0$;
 - (ii) As perdas em T não excedem $\beta_1 L_0 W_0$;
 - (iii) Os retângulos R_i constituintes de T não extrapolam os limites b_i respectivos.
- 2.b Faça $L^{(k)} = L^{(k-1)} \cup F^{(k)}$. Retire do novo conjunto $L^{(k)}$ os elementos que aparecem repetida. Isto C , de dois retângulos guilhotina com as mesmas dimensões e os mesmos retângulos R_i constituintes, apenas um fica.

PASSO (3)

Se $F^{(k)}$ é não vazio, faça $R \leftarrow k + 1$ e volte ao passo (2). Caso contrário,

PASSO (4)

4.a Faça $M = k - 1$.4.b Escolha o retângulo de $L^{(M)}$ que gera o menor resto quando colocado dentro do ítem de estoque $L_0 \times W_0$.

ALGORITMO DOIS

Tome β_2 no lugar de β_1 no passo 1.a. O passo 2.a (ii) torna-se:(ii) As perdas em T não excedem $\beta_2 \cdot a(T)$;

VI.3 Limites para o Erro e Condições Ótimas

À medida que os valores de β_1 e β_2 são incrementados desde o valor 0 até 1, os retângulos T geradas pelos dois algoritmos crescem em número absurdo. Dessa forma, a solução ótima obtida a partir de $L^{(M)}$ é "pelo menos tão boa" quanto as anteriores obtidas de conjuntos $L^{(M)}$ menores, correspondentes a $\beta_{i,x}$ menores.

Chamamos de perda interna as sobras dentro da área compreendida por um determinado retângulo (veja figura VI.4). Seja μ a perda interna de um retângulo $S = h \times w$ gerado a partir de uma seqüência de construções horizontais e/ou verticais dos retângulos R_1, R_2, \dots, R_m sujeitos aos limites b_1, b_2, \dots, b_m e percentagem de sobras máxima β_i . Seja Ψ o valor total da perda quando o retângulo S é colocado dentro do ítem de estoque $L_0 \times W_0$. Assim, $\Psi = L_0 W_0 - hw + \mu$.

Denotaremos por Ψ^* a perda total correspondente à solução ótima da problema da mochila. Ou seja, é a menor perda de um retângulo obtido de conjuntos $L^{(M)}$ com β_i assumindo quaisquer valores possíveis. Consideremos o conjunto Ω como sendo constituído por todos os retângulos que possuem perda total Ψ^* .

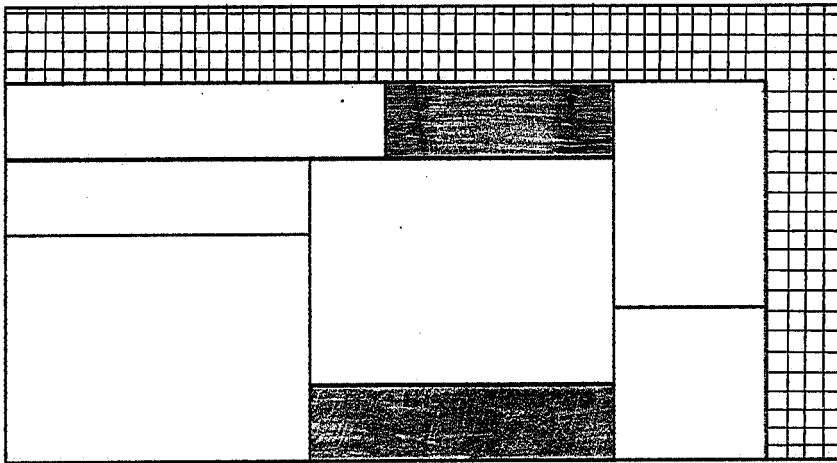
Denotemos por β_1^* o menor valor que β_1 deve assumir para que seja gerado pelo menos um elemento do conjunto Ω . Assim, existe um S_1^* retângulo de Ω com perda interna μ_1^* tal que $\Psi^* \geq \mu_1^* = \beta_1^* \cdot L_0 W_0$. Essa última igualdade vale para pelo menos um elemento de Ω . Mas pode não valer para todos, que embora tenham a mesma perda total Ψ^* , podem não ter a mesma perda interna μ^* .

Teorema VI.1 Se S é o retângulo guilhotina selecionado pelo algoritmo 1 de $L^{(M)}$ com perda Ψ então:

$$\Psi - \Psi^* \leq \|\Psi - \beta_1 \cdot L_0 W_0\|$$

Demonstração

Consideremos dois casos:



■ Interno

▣ Externo

Figura VI.4: Uma solução parcial e dois diferentes tipos de perdas

- (i) Se $\beta_1 < \beta_1^*$ então $S \notin \Omega$. Portanto, $\Psi > \Psi^* \geq \beta_1^* L_0 W_0 > \beta_1 L_0 W_0$. Assim, $0 < \Psi - \Psi^* < \Psi - \beta_1 L_0 W_0$.
- (ii) Se $\beta_1 \geq \beta_1^*$ então $S \in \Omega$ e $\Psi = \Psi^*$. Isto implica em $\Psi - \Psi^* = 0 \leq \|\Psi - \beta_1 L_0 W_0\|$.

[C.Q.D.]

Teorema VI.2 Se a perda Ψ de um retângulo guilhotina S obtido do algoritmo 1 com um valor β_1 fixo satisfaz $\Psi \leq \beta_1 L_0 W_0$, então S corresponde a um modo de cortar ótimo e $\Psi = \Psi^*$.

Demonstração

Suponha que $\Psi \leq \beta_1 L_0 W_0$. Se $S \notin \Omega$, então $\beta_1 < \beta_1^*$ e $\Psi^* \geq \beta_1^* L_0 W_0 > \beta_1 L_0 W_0 \geq \Psi$, contradizendo a otimalidade de Ψ^* . Assim, $S \in \Omega$.

[C.Q.D.]

Corolário VI.1 Se as dimensões de S são $L_0 \times W_0$, então S corresponde a um modo de cortar ótimo e $\Psi = \Psi^*$.

Demonstração

Se S tem dimensões $L_0 \times W_0$ então $\Psi = \mu \leq \beta_1 L_0 W_0$. Aplicando-se o teorema VI.3, obtemos o resultado desejado.

[C.Q.D.]

A extensão dos resultados acima para o algoritmo 2 pode ser facilmente obtida. Seja Ω o conjunto dos retângulos obtidos a partir da aplicação do algoritmo 2 com algum valor β_2 fixo que possuem custo total Ψ^* . Seja β_2^* o menor valor que devemos associar a β_2 para que a aplicação do algoritmo 2 gere pelo menos um elemento em Ω .

Dizemos que uma seqüência de retângulos S_1, S_2, \dots, S_t gera um retângulo guilhotina S se satisfazem:

- (i) S_{i+1} é obtido a partir de S_i pela construção, horizontal ou vertical, deste último com algum retângulo R_i , $i \in \{1, 2, \dots, m\}$;
- (ii) $S_t = S$.

Denotemos por $\mu_1, \mu_2, \dots, \mu_t$ as perdas internas dos retângulos S_1, S_2, \dots, S_t .

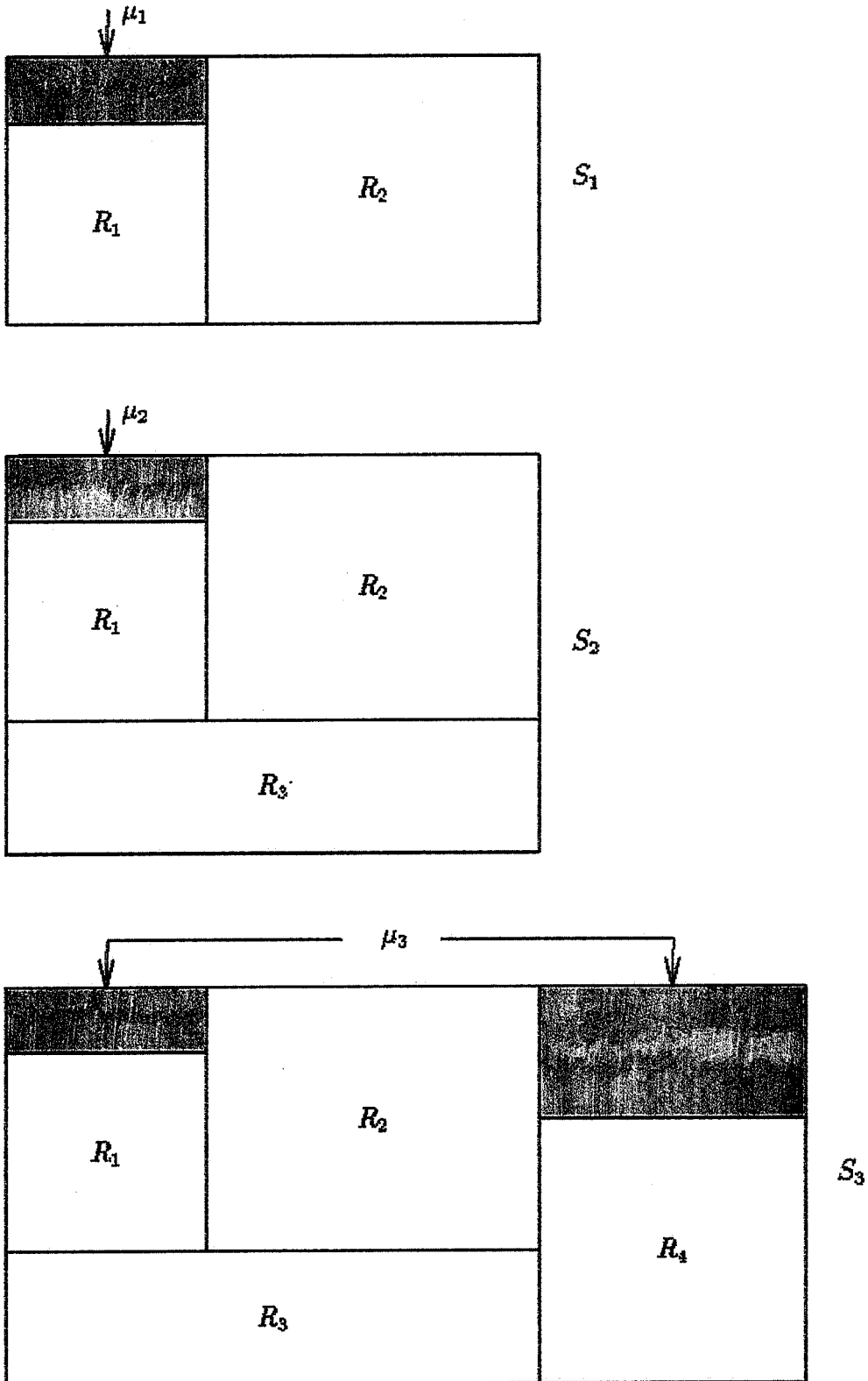


Figura VI.5: Retângulos parciais que geram S_2^*

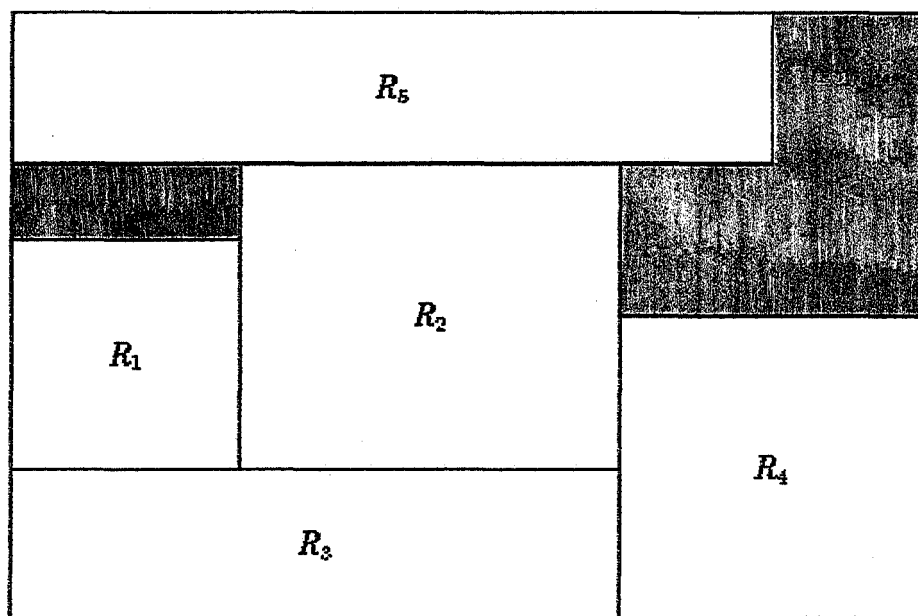


Figura VI.6: Um retângulo guilhotina ótimo $S_4 = S_2^*$

Assim, claramente, temos:

$$\beta_2^* = \min_{S \in \mathcal{R}} \left\{ \max_{1 \leq i \leq t} \left\{ \frac{\mu_i}{a(S_i)} \right\}, \text{ onde } S_1, S_2, \dots, S_t \text{ é a seqüência que gera } S \right\}$$

O cálculo de $\max_{1 \leq i \leq t} \left\{ \frac{\mu_i}{a(S_i)} \right\}$ se justifica, pois somente se β_2^* tiver pelo menos esse valor, será possível a geração de cada elemento da seqüência S_1, S_2, \dots, S_t a fim de chegarmos ao S . Veja figuras VI.5 e VI.6.

Seja ρ o retângulo de menor área, submetido às restrições b_i , obtido a partir da construção, horizontal ou vertical, de dois retângulos quaisquer R_k e R_j , $k \neq j$.

Seja S_p o retângulo parcial para o qual temos $\beta_2^* = \frac{\mu_p}{a(S_p)}$. Seja S_2^* o retângulo de R gerado pela seqüência de retângulos da qual S_p faz parte. Assim, $\mu_2^* \geq \mu_p$.

Então, supondo que S_2^* não é nenhum dos R_{i_s} , podemos concluir que:

$$\Psi^* \text{ e } \mu_2^* \geq \mu_p = \beta_2^* \cdot a(S_p) \geq \beta_2^* \cdot a(\rho)$$

Teorema VI.3 Se S é o retângulo guilhotina de $L^{(M)}$ selecionado do algoritmo 2 tendo custo total Ψ , e o algoritmo foi aplicado com um valor fixo β_2 , então

$$\Psi - \Psi^* \leq \|\Psi - \beta_2 \cdot a(\rho)\|$$

Demonstração

Suponha que S_2^* não é nenhum dos retângulos R_{i_2} . Assim sendo, temos dois casos a considerar:

$$(i) \beta_2 < \beta_2^* \Rightarrow \Psi > \Psi^* \geq \beta_2^* \cdot a(\rho) > \beta_2 \cdot a(\rho) \Rightarrow 0 < \Psi - \Psi^* < \Psi - \beta_2 \cdot a(\rho)$$

$$(ii) \beta_2^* \leq \beta_2 \Rightarrow \Psi - \Psi^* = 0 \leq \|\Psi - \beta_2 \cdot a(\rho)\|$$

Suponha que S_2^* é igual a algum dos R_{i_2} , então $\beta_2^* = 0$. Portanto, para qualquer valor de β_2 , teremos $\beta_2 \geq \beta_2^*$ e $\Psi - \Psi^* = 0$.

[C.Q.D.]

Teorema VI.4 *Seja S um retângulo obtido pela aplicação do algoritmo 2 a um valor fixo β_2 , com um custo total Ψ . Se $\Psi \leq \beta_2 \cdot a(\rho)$ e S_2^* não é nenhum dos R_{i_2} , então S é ótimo e $\Psi = \Psi^*$.*

Demonstração

Se $S \notin \Omega$ então $\beta_2 < \beta_2^*$ e $\Psi^* \geq \beta_2^* \cdot a(\rho) > \beta_2 \cdot a(\rho) \geq \Psi$, contradizendo a otimalidade de Ψ^* . Assim, $S \in \Omega$.

[C.Q.D.]

Teorema VI.5 $\beta_1^* \leq \frac{\Psi}{L_0 W_0}$ e, se S_2^* não é nenhum dos R_{i_2} , $\beta_2^* \leq \frac{\Psi}{a(\rho)}$ onde Ψ é a perda total de um modo de cortar que não excede os limites b_i para o retângulo R_i .

Demonstração

$$\begin{cases} \Psi^* \geq \beta_1^* \cdot L_0 W_0 \\ \Psi^* \geq \beta_2^* \cdot a(\rho) \\ \Psi^* \leq \Psi \end{cases} \Rightarrow \begin{cases} \Psi \geq \beta_1^* \cdot L_0 W_0 \Rightarrow \beta_1^* \leq \frac{\Psi}{L_0 W_0} \\ \Psi \geq \beta_2^* \cdot a(\rho) \Rightarrow \beta_2^* \leq \frac{\Psi}{a(\rho)} \end{cases}$$

[C.Q.D.]

O ideal é atribuímos a β_1 e β_2 valores um pouco acima do ideal β_1^* e β_2^* mas que permitam uma convergência suficientemente rápida do algoritmo.

O teorema VI.5 nos permite tirar vantagem de uma primeira aproximação feita à mão para o problema da modula. Ela nos daria um limite superior a β_1^* e β_2^* .

Se nós aplicarmos os dois algoritmos com $\beta_1 = \beta_2$, o conjunto $L^{(M_2)}$ gerado pelo algoritmo 3 será uni subconjunto de $L^{(M_1)}$ gerado pelo algoritmo 1. Uma idéia seria a de aplicarmos primeiro o algoritmo 2 e a perda total Ψ de sua solução ser utilizada (de acordo com o teorema VI.5) em uma cota superior para β_1^* . Aí, em seguida, aplicaríamos o algoritmo 1. Dessa forma, poderíamos chegar mais rapidamente a uma solução aceitável.

VI.4 O Problema do Corte de Chapas

Os algoritmos 1 e 2 podem ser utilizados na geração de soluções aproximadas para o problema do corte de chapas.

Representemos por N_i a demanda do retângulo R_i .

Suponha que ao todo tenhamos n modos de cortar associados aos retângulos em estoque. Cada modo de cortar será executado x_j vezes a um custo c_j , onde $j = 1, 2, \dots, n$.

Definamos uma matriz A coma tendo n colunas, cada uma delas associada a um modo de cortar. O elemento a_{ik} da A representa o número de pedaços $R_i = l_i \times w_i$ extraídos no modo de cortar k .

O problema pode então ser colocado da seguinte forma:

$$\begin{array}{l} \text{minimizar} \quad \sum_{j=1}^n c_j x_j \\ \text{sujeito a} \quad A \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} N_1 \\ N_2 \\ \vdots \\ N_m \end{bmatrix} \end{array}$$

$$0 \leq x_i \in Z \quad \forall i$$

O problema pode ser resolvido relaxando-se as restrições:

$$\begin{array}{l} \text{minimizar} \quad \sum_{j=1}^n c_j x_j \\ \text{sujeito a} \quad A \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \geq \begin{bmatrix} N_1 \\ N_2 \\ \vdots \\ N_m \end{bmatrix} \end{array}$$

$$x_i \geq 0, \quad \forall i$$

A idéia é reduzirmos o número de colunas de A a um valor aceitável, que permita a solução do PPL acima sem muita demora. O método é descrito abaixo.

Os algoritmos 1 ou 2 são aplicados no retângulo $L_0 \times W_0$. As restrições b_i podem ser determinadas como sendo $b_i = \left\lfloor \frac{L_0 W_0}{l_i w_i} \right\rfloor$, para $i = 1, 2, \dots, m$.

Do conjunto $L^{(M)}$ serão extraídos os retângulos guilhotina que correspondem aos modos de cortar. As colunas de A estarão associadas aos retângulos

guilhotina. O número de elementos de $L^{(M)}$ escolhidos serão responsáveis pela rapidez **com** que o PPL será resolvido. A precisão do resultado obtido é função dos tipos de retângulos guilhotina selecionados.

VI.5 Um a Nova Proposta Melhorada

Uma modificação nos algoritmos (1) e (2) pode ser feita, melhorando suas performances.

Consideremos o teorema VI.3. A condição dada no teorema é suficiente para garantirmos a otimalidade. Assim, buscamos a geração apenas de soluções que satisfazem a hipótese do teorema.

Seja um retângulo guilhotina T com perda interna T_{int} . Consideremos a parte não aproveitada $\& T$ quando este é colocado dentro de um ítem $\&$ estoque. Podemos gerar uma série de retângulos R_i a partir desta parte não aproveitada. Tomemos a menor perda possível na geração desses retângulos dentro da parte não aproveitada. Seja T_{min} o seu valor. Assim, a menor perda possível de retângulos guilhotina obtidos a partir de construções sobre T será $T_{int} \& T_{min}$.

A condição 2.a (ii) para o algoritmo 1 toma-se:

$$(ii) \quad T_{int} + T_{min} \leq \beta_1 L_0 W_0$$

Denominamos, novamente, β_1^* como sendo o menor valor que β_1 deva assumir para que seja gerado pelo menos um elemento do conjunto R . Assim, existe um S_1^* retângulo de Ω tal que $\Psi^* \geq (S_1)_{int} + \overbrace{(S_1)_{min}}^{=0} = \beta_1^* L_0 W_0$.

Os teoremas anteriores permanecem válidos. Suas demonstrações para o novo caso são idênticas.

Como se obtém T_{min} ? Consideremos a figura VI.7.

Denotemos por $P_{min}(h, w)$ a menor perda que se tem, extraindo-se do retângulo $h \times w$ pedaços R_i .

A fim de gerarmos o retângulo de dimensões H_T e W_T temos duas possibilidades:

- (1) efetuamos um corte horizontal produzindo o retângulo (a) e outro vertical. O custo mínimo resultante será: $T_{int} + P_{min}(L_0 - H_T, W_0) + P_{min}(H_T, W_0 - W_T)$
- (2) efetuam um corte vertical produzindo o retângulo (b) e outro horizontal. O custo mínimo resultante será: $T_{int} + P_{min}(L_0, W_0 - W_T) + P_{min}(L_0 - H_T, W_T)$.

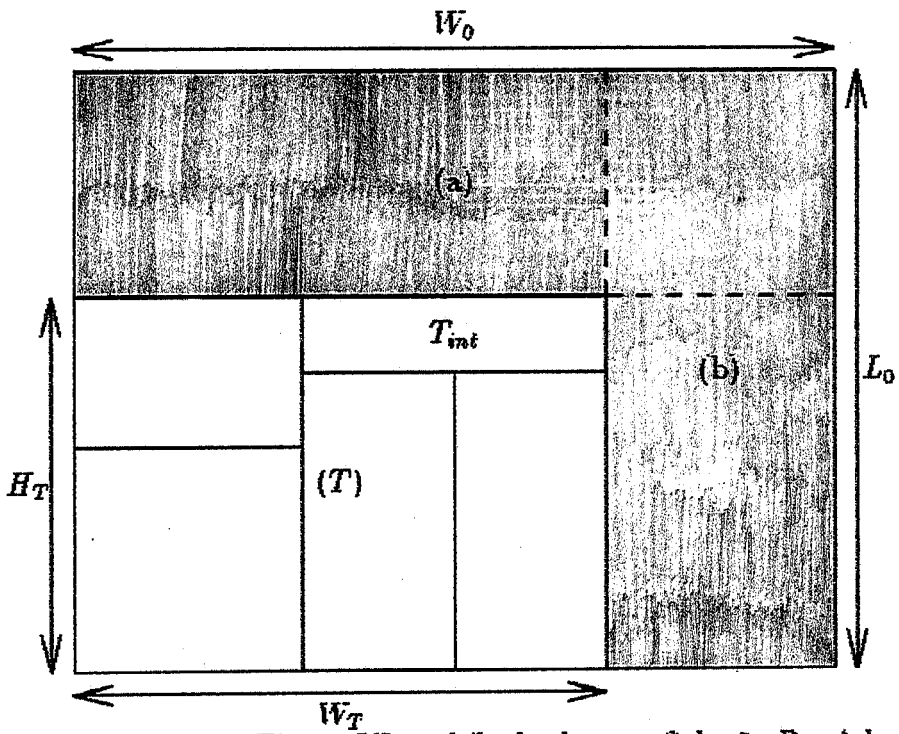


Figura VI.7: Cálculo de uma Solução Parcial

Assim, para o retângulo T de dimensões H_T e W_T , a perda mínima a ser obtida será:

$$T_{min} = \min(P_{min}(L_0 - H_T, W_0) + P_{min}(H_T, W_0 - W_T), \\ P_{min}(L_0, W_0 - W_T) + P_{min}(L_0 - H_T, W_T))$$

Como se obtém $P_{min}(h, w)$ para um dado retângulo $h \times w$? Resolvendo-se o problema da mochila em duas dimensões sem restrições para o retângulo $h \times w$ e pedaços R_i valorados pela sua área. Pode-se aplicar o método de programação dinâmica proposto por Gilmore e Gomory ou o proposto por J. C. Herz. Seja $V_M(h, w)$ a solução desse problema. Assim: $P_{min}(h, w) = hw - V_M(h, w)$.

Os valores $P_{min}(h, w)$ podem ser obtidos a priori, antes da aplicação do algoritmo para todo retângulo $h \times w$ contido na retângulo $L_0 \times W_0$, ie, $h \leq L_0$ e $w \leq W_0$. Com estes dados, formamos uma tabela.

Este novo algoritmo foi testado por José Fernando Oliveira e José Soeiro Ferreira [15] e os resultados se comprovaram melhores que os obtidos com os algoritmos de P. Y. Wang. [27].

Capítulo VII

O Problema de Alocação Discreta em Duas Dimensões

VII.1 Introdução

Este capítulo foi baseado no trabalho de David W. Pentico [23]. A seção VII.7 foi baseada em Zangwill [29]. A última seção foi baseada no paper de Wolfson [28].

Temos que satisfazer uma demanda de objetos dados por duas dimensões. Surge um problema: a variedade dos tipos de peças não permite que mantenhamos em estoque todos os tipos de objetos possíveis de serem solicitados (cada tipo é definido por um par que representa as dimensões da peça). Sendo assim, deveremos selecionar um conjunto de tipos de peças para ser mantido em estoque, que possam satisfazer a demanda, da seguinte forma: se um determinado objeto não tiver um similar em estoque, então um outro que esteja armazenado é escolhido para ser cortado, e originar o objeto com as dimensões requeridas.

Como selecionar este conjunto de tipos de peças a partir da conjunto de todos os tipos de peças que compõem a demanda? E qual a quantidade de cada elemento deste conjunto que será mantida em estoque? Os seguintes custos devem ser levados em conta:

1. Qualquer que seja o tipo da peça, existe um custo associado à peça para mantê-la em estoque (é um custo intrínseco; ele independe da quantidade da peça em estoque e do tipo da peça estocada);
2. Custo que depende do tipo da peça e da quantidade da peça de um determinado tipo mantida em estoque;
3. Custo das perdas resultantes em se cortar uma peça do estoque para se originar aquela que está sendo demandada.

Uma observação cabe aqui: quando nos referimos a peças que seriam cortadas e posteriormente utilizadas em lugar de outra, estávamos pensando em placas planas com duas dimensões (largura e altura) como sendo essas peças. As perdas mencionadas em (3) seriam os restos de material que seriam jogados fora após os cortes. Contudo, podemos dar outra interpretação ao problema, Ao invés de termos como dimensões de um objeto sua largura e sua altura teríamos duas outras qualidades quaisquer. A substituição de um objeto por outro só ocorreria se as qualidades do primeiro permitissem a este executar as tarefas do outro. Embora, daqui para frente, sempre façamos referência ao problema de corte de chapas, todo o raciocínio desenvolvido aqui também se aplicará a uma situação mais genérica.

Outra observação: a substituição se dá na razão de um objeto do estoque para um objeto solicitado na demanda.

VII.2 Formulação do Problema

Sejam os conjuntos abaixo:

$$\begin{aligned} N' &= \{(i, j); \text{ existe uma peça requisitada com dimensões } (i, j)\} \\ N_1 &= \{i; \exists j \text{ com } (i, j) \in N'\} = \{1, 2, \dots, n_1\} \\ N_2 &= \{j; \exists i \text{ com } (i, j) \in N'\} = \{1, 2, \dots, n_2\} \\ N &= N_1 \times N_2 \end{aligned}$$

Dizemos que a peça (i_1, i_2) pode substituir a peça (j_1, j_2) se $i_1 \geq j_1$ e $i_2 \geq j_2$. Notação:

x_{ik} → número de peças com dimensões (i, k) que serão mantidas em estoque;

d_{ik} demanda das peças com dimensões (i, k) ;

$U_{ik} = \{(j, m); j \leq i \text{ e } m \leq k\}$;

$S_{jm} = \{(i, k); i \geq j \text{ e } k \geq m\}$;

$y_{ik,jm}$ → número de peças (i, k) usadas para substituírem um igual número de peças (j, m) ¹;

$S_{ik,jm}(y)$ + custo da substituição de y peças com dimensões (j, m) por y peças com dimensões (i, k) ²;

$c_{ik}(x)$ + custo para se manter em estoque x peças com dimensões (i, k) ;

Hipóteses feitas sobre os custos:

$$1. c_{ik}(x) = \begin{cases} \text{côncava} \\ f_{ik} + \overbrace{v_{ik}(x)} & , \text{ se } x > 0 \\ 0, & \text{ se } x = 0 \end{cases}$$

¹Obs.: $y_{ik,ik} = d_{ik}$

²Obs.: $S_{ik,ik}(y) = 0 \forall y$

$$2. \begin{cases} s_{ik,jm}(y) = b_{ik,jm} \cdot y, \text{ onde} \\ b_{ik,jm} = b_{ik,pr} + b_{pr,jm} \quad \forall i, k, j, m, p, r; i \leq p \leq j \text{ e } k \leq r \leq m \end{cases}$$

Este é o problema de programação linear a ser resolvido:

$$(1) \begin{cases} \min \sum_N c_{ik}(x_{ik}) + \sum_N \sum_{U_{ik}} S_{ik,jm}(y_{ik,jm}) \\ \text{sujeito a } \begin{cases} \sum_{j,m} y_{ik,jm} = d_{jm} \\ \sum_{U_{ik}} y_{ik,jm} = x_{ik} \end{cases} \end{cases}$$

VII.3 Representação em Redes

Constrói-se uma matriz como mostrado na figura VII.1 (fonte é a representação simbólica do estoque). Daqui para frente, faremos referências a nós e peças, indistintamente, como sendo a mesma coisa.

Teorema VII.1 *Se o PPL (1) tiver solução ótima, ela deve ser tal que o consumo de cada nó ou é satisfeito por sua quantidade armazenada em estoque, ou é satisfeito pela quantidade armazenada em estoque de apenas um único outro nó (evidentemente, um cujas dimensões comportem o primeiro).*

Demonstração

Utilizaremos o resultado abaixo, provado por Zangwill (1968), e que está explicado na seção VII.7:

» Sejam N um conjunto de nós e A um conjunto de arcos. Consideremos o seguinte problema em programação linear:

$$\text{minimizar } \sum_{(e,f) \in A} c_{ef}(x_{ef})$$

$$\text{sujeito a } \sum_{f: (e,f) \in A} x_{ef} - \sum_{f: (f,e) \in A} x_{fe} = r_e, \quad e \in N$$

$$x_{ef} \geq 0, \quad (e, f) \in A$$

$$\sum_{e \in N} r_e = 0$$

$$c_{ef}(0) = 0$$

$$c_{ef}(x_{ef}), \text{ onde } (e, f) \in A, \text{ uma função côncava}$$

Se o problema acima tiver como solução ótima um vetor $x = (x_{ef}; (e, f) \in A)$ onde e é um nó qualquer, então:

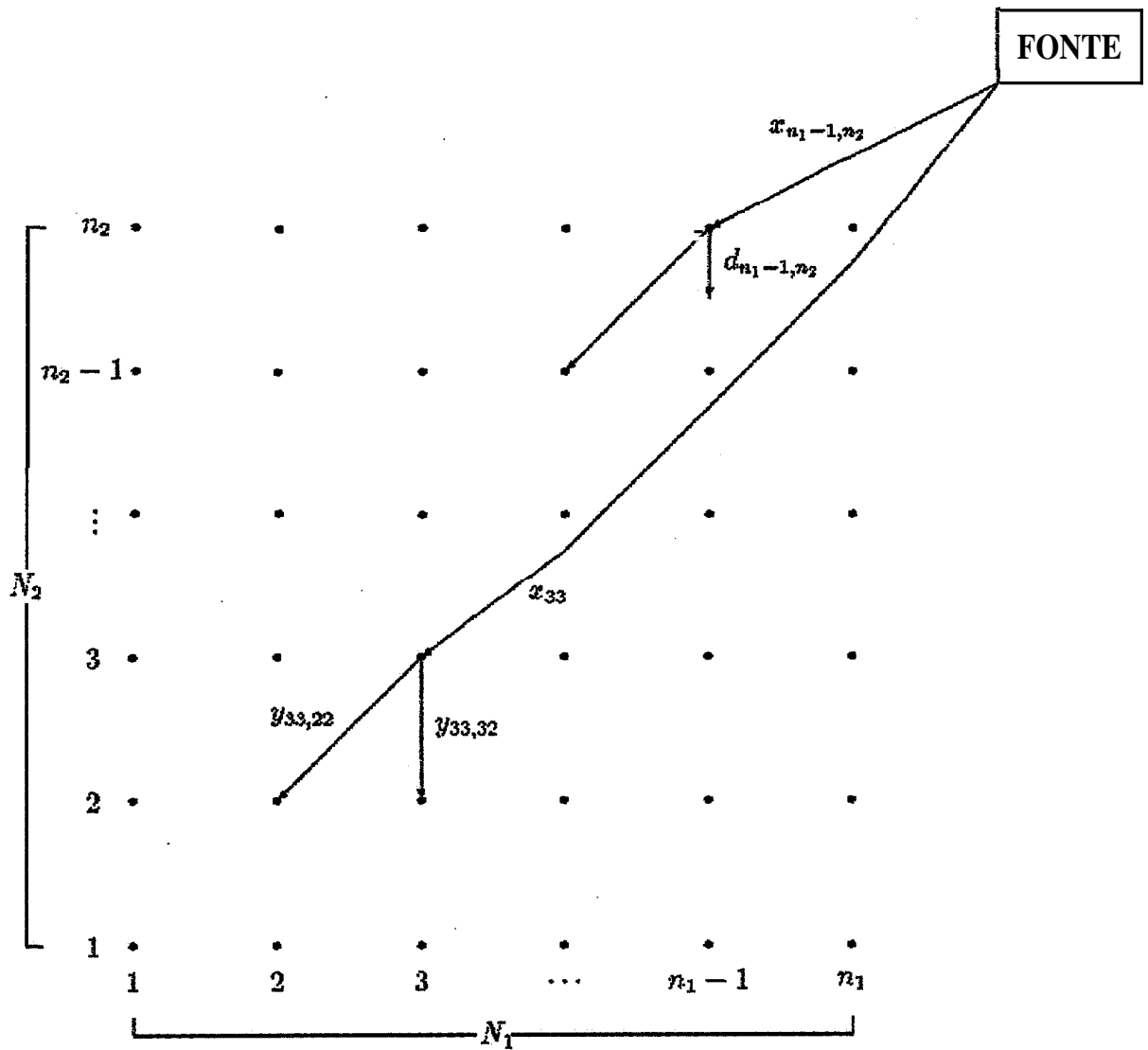


Figura VII.1: cada nó (i, j) representa um tipo de peça com dimensões i e j .

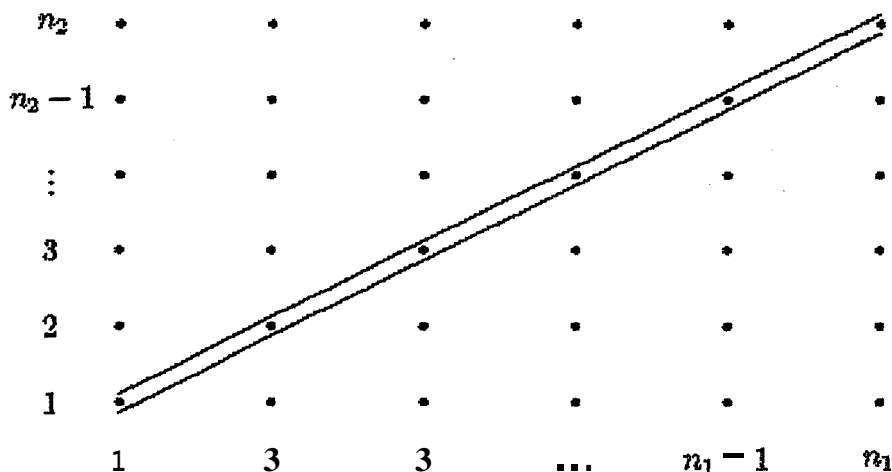


Figura VII.2: A maior parte da demanda fica na diagonal principal

N tem K fontes \implies e tem no máximo K entradas positivas "

O nosso PPL (1) tem uma forma similar ao do resultado acima. De acordo com sua representação, o PPL (1) só tem uma fonte. Assim, dado um nó (i, j) , ele só pode ter no máximo uma entrada positiva. O resultado segue-se.

[C.Q.D.]

VII.4 Padrões de Estocagem Aninhados

Um estudo feita por Diegel e Bocker [11] "on the demand for panes of glass" sugere que a hipótese de uma correlação positiva entre as duas dimensões que definem cada tipo de objeto é razoável de acontecer. Mas o que significa essa correlação positiva? Considere a figura VII.2. No caso de haver uma correlação positiva entre as duas dimensões, a maioria dos nós que compõem a demanda ficam próximos à diagonal principal. Esta correlação positiva sugere a seguinte definição:

Definição. Uma política de estoque é "aninhada" se, dados os tamanhos possíveis de serem estocados (i, k) , (j, m) , ..., (u, v) , nós tivermos $i < j < \dots < u$ e $k < m < \dots < v$. Isto é, a política é "aninhada" se cada elemento estocado é maior, em ambas as dimensões, do que o próximo tamanho (menor) a ser estocado,

Shier [25] propôs que restringíssemos nossa atenção para as políticas de estoque que são "aninhadas".

Voltemos ao nosso problema, supondo desta vez, que temos uma política "aninhada": Temos um conjunto N de peças com duas dimensões que compõem a demanda. Deste conjunto, vamos extrair um subconjunto daquelas

que ficarão em estoque. Mas como? A solução é obtida usando-se *programação dinâmica*.

Vamos supor que as peças com dimensões (1,1) vão para estoque na política de estocagem ótima. Então, nesse caso, dá para se saber quais *serão* as peças com dimensões menores do que (1,1) que irão para o estoque? Sim, a resposta é nenhuma: (1,1) é a menor dimensão possível para uma peça dada. Vamos usar a notação abaixo:

G_{ik}^m = custo para se satisfazer a demanda de todas as peças com dimensões menores do que ou igual a (i, k) , supondo que a política de estocagem utilizada é a ótima e a peça (i, k) fique armazenada em estoque nessa política.

Assim: $G_{11}^m = c_{11}(d_{11})$.

Notação:

$G_{ik}(j, m)$ = custo mínimo para se satisfazer a demanda de todas as peças com dimensões menores do que ou igual a (i, k) , supondo que a política ótima guarde (i, k) em estoque e a próxima peça a ser armazenada seja (j, m)

G_{ik} = igual ao anterior, só que, neste caso, (i, k) é a última peça mantida em estoque pela política de estocagem ótima

Vamos supor que a peça com dimensões (1,2) vá para o estoque. Temos duas possibilidades:

(1) A peça (1,1) também será estocada Logo:

$$G_{12}(1, 1) = c_{11}(d_{11}) + c_{12}(d_{12})$$

(2) A peça (1,1) não será estocada. Logo:

$$G_{12} = s_{12,11}(d_{11}) + c_{12}(d_{12} + d_{11})$$

Assim: $G_{12}^m = \min\{G_{12}(1, 1), G_{12}\}$. Se for o caso, fazemos o registro de (1,1) como sendo o sucessor de (1,2) na política ótima.

Suponha, agora, que a peça com dimensões (a, h) seja estocada em nossa solução ótima. Assim, podemos escrever:

$$G_{ik}(j, m) = c_{ik} \left(\sum_{(j,m) < (p,q) \leq (i,k)} d_{pq} \right) + \sum_{(i,k) \geq (p,q) > (j,m)} s_{ik,pq}(d_{pq}) + G_{jm}^m \quad \forall (j, m)$$

$$G_{ik} = c_{ik} \left(\sum_{(p,q) \leq (i,k)} d_{pq} \right) + \sum_{(i,k) \geq (p,q)} s_{ik,pq}(d_{pq})$$

$$G_{ik}^m = \min\{G_{ik}(j, m), (j, m) < (i, k); G_{ik}\}$$

Registramos o sucessor de (i, k) (se houver] na solução ótima.

O algoritmo apresentada acima nos permite chegar ao ótimo. Por quê? Seja (i_0, k_0) a peça de dimensões maiores que faz parte da demanda. Evidentemente, ela será estocada. Dado que (i_0, k_0) será estocado, seja (i_1, k_1) a próxima peça a ser mantida em estoque, obtida pelo algoritmo acima. E assim por diante, até chegarmos a (i_n, k_n) uma peça que não possui sucessor algum segundo a política ótima. Dessa forma, $\{(i_0, k_0), (i_1, k_1), \dots, (i_n, k_n)\}$ é o conjunto formado por todas as peças que irão para estoque utilizando-se a política ótima de estocagem.

$$x_{(i_j, k_j)} = \sum_{(i_j, k_j) \geq (p, q) > (i_{j+1}, k_{j+1})} d_{pq}, \quad 0 \leq j \leq n-1$$

$$x_{(i_n, k_n)} = \sum_{(i_n, k_n) \geq (p, q)} d_{pq}$$

VII.5 Características da Política Ótima

Definição. Uma política de estoque é retangularmente segmentada se, dado que um objeto (i, k) é estocado e supre a demanda de um outro objeto (j, m) , então deve suprir a demanda de todos os intermediários (p, r) , ie, $i \geq p \geq j$ e $k \geq r \geq m$.

Teorema VII.2 Se $c_{ik}(x)$ é uma função côncava e $s_{ik,jm}(y)$ é aditiva e proporcional então há uma política de estoque regularmente segmentada que é ótima

Demonstração

Defina $\Delta c_{ik}(x) = c_{ik}(x) - c_{ik}(x-1)$. Suponha a situação representada pela figura VII.3.

Qual deve ser o custo de se suprir por meio de 1 peça (i, k) a última unidade requisitada de objetos (j, m) ? Resposta: $c_{ik}(x_{ik}) - c_{ik}(x_{ik} - 1) + b_{ik,jm} \cdot 1 = \Delta c_{ik}(x_{ik}) + b_{ik,jm}$

Qual deve ser o custo de se suprir por meio de uma peça (h, n) a última unidade requisitada de objetos (j, m) ? Resposta: $c_{hn}(x_{hn} + 1) - c_{hn} \cdot x_{hn} + b_{hn,jm} \cdot 1 = \Delta c_{hn}(x_{hn} + 1) + b_{hn,jm}$

Como a fonte (i, k) é a que foi escolhida, devemos ter: $\Delta c_{ik}(x_{ik}) + b_{ik,jm} \leq \Delta c_{hn}(x_{hn} + 1) + b_{hn,jm}$

Com isso, chega-se à conclusão de que, se existe uma solução ótima na qual a peça (i, k) fornece material para (j, m) então haverá uma solução ótima na qual a peça (i, k) fornece material para toda peça (p, r) com $i \leq p \leq j$ e $k \leq r \leq m$.

[C.Q.D.]

VII.6 Resultados Computacionais

Definição. Seja (i, k) um nó dado.

$$D_{ik} \stackrel{\text{def}}{=} \sum_{(p,q) \in U_{ik}} d_{pq}$$

Teorema VII.3 Se $c_{ik}(x)$ é côncavo e $c_{ik}(D_{ik}) - c_{ik}(D_{ik} - d_{jm}) + b_{ik,jm} \cdot d_{jm} > c_{jm}(d_{jm})$ para algum $(j, m) \leq (i, k)$ então (i, k) não será fonte de (j, m) em nenhuma política de estoque ótima.

Demonstração

Vamos supor uma política de estocagem na qual se poria (i, k) para suprir a demanda de (j, m) . Proporemos algumas perguntas para orientarmos o nosso raciocínio:

- (1) Quanto gastaríamos caso a demanda de (j, m) fosse toda satisfeita por peças (j, m) mantidas em estoque? Resposta: $c_{jm}(d_{jm})$.
- (2) Quanto economizaríamos se deixássemos de atender a demanda (j, m) pelas peças (i, k) em estoque? Resposta: $c_{ik}(D_{ik}) - c_{ik}(D_{ik} - d_{jm}) + b_{ik,jm} \cdot d_{jm}$

Então, como saber se vale a pena ou não substituir o abastecimento de peças (j, m) via peças (i, k) pelo abastecimento direto de um estoque de (j, m) 's? É só comparar as respostas obtidas em (1) e (2):

Se $c_{jm}(d_{jm}) < c_{ik}(D_{ik}) - c_{ik}(D_{ik} - d_{jm}) + b_{ik,jm} \cdot d_{jm}$
então vale a pena a troca, ie, passar a ter (j, m) em estoque
senão não vale a pena.

[C.Q.D.]

Teorema VII.4 Se $c_{ik}(x)$ e $c_{pr}(x)$ são côncavas então, desde que exista algum $(p, r) \leq (i, k)$ com $c_{ik}(D_{ik}) - c_{ik}(D_{ik} - d_{pr}) + b_{ik,pr} \cdot d_{pr} > c_{pr}(d_{pr})$, em uma política de estoque ótima, (i, k) não será fonte para nenhum (j, m) tal que $(j, m) \leq (p, r)$.

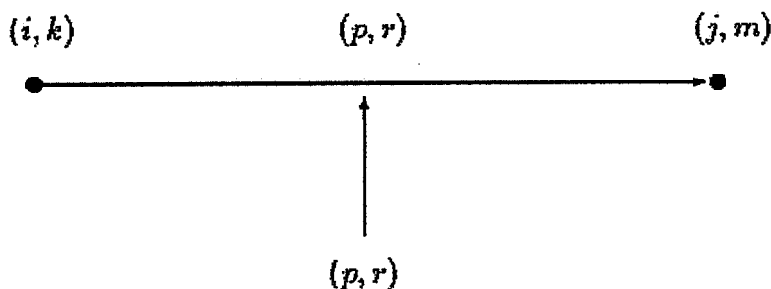


Figura VII.4: A peça (i, k) supre a demanda por (j, m)

Demonstração

- (1) Quanto gastaríamos caso a demanda de (p, r) fosse toda satisfeita por peças (p, r) mantidas em estoque? Resposta: $c_{pr}(d_{pr})$
- (2) Quanto economizaríamos se deixássemos de atender a demanda (p, r) pelas peças (i, k) em estoque? Resposta: $c_{ik}(D_{ik}) - c_{ik}(D_{ik} - d_{pr}) + b_{ik,pr} \cdot d_{pr}$

Como $c_{pr}(d_{pr}) < c_{ik}(D_{ik}) - c_{ik}(D_{ik} - d_{pr}) + b_{ik,pr} \cdot d_{pr}$ é mais interessante passarmos a ter (p, r) em estoque. Agora, suponha que exista $(j, m) < (p, r)$ tal que, na política ótima, (i, k) supre as deficiências de (j, m) . Veja a figura VII.4.

De acordo com o teorema VII.2 (i, k) deveria ser fonte para (p, r) na política ótima. Isto contradiz as nossas observações anteriores.

[C.Q.D.]

VII.7 O Problema do Custo Mínimo em Redes para uma Função Côncava

Nesta seção abordamos o resultado demonstrado por Zangwill (1968).

Uma rede $[N, A]$ consiste de um conjunto finito N de nós mais um conjunto de arcos A . Cada arco é suposto levar material de um nó para outro. Notação:

- $r_e > 0 \rightarrow$ o nó "e" é fonte de material
- $r_e < 0 \rightarrow$ o nó "e" é destino de material
- $x_{ef} \rightarrow$ quantidade de material enviada ao longo do arco (e, f) . Restrição: $x_{ef} \geq 0$
- $x \rightarrow$ vetor com componentes x_{ef}
- $c_{ef}(x_{ef}) \rightarrow$ o custo de se enviar x_{ef} unidades ao longo do arco (e, f)

Façamos a seguinte hipótese ao longo do texto: $c_{ef}(0) = 0$; caso contrário, definamos um novo custo $c'_{ef}(x_{ef}) = c_{ef}(x_{ef}) - c_{ef}(0)$, e trabalhemos com ele.

O problema consiste em determinarmos um fluxo, chamado ótimo, que minimiza a função $\sum_{(e,f) \in A} c_{ef}(x_{ef})$ sujeito às seguintes restrições:

$$\begin{aligned} \sum_{f:(e,f) \in A} x_{ef} - \sum_{f:(f,e) \in A} x_{fe} &= r_e, \quad e \in N \\ x_{ef} &\geq 0 \quad (e, f) \in A \\ \sum_{e \in N} r_e &= 0 \\ c_{ef}(0) &= 0 \\ c_{ef}(x_{ef}), \quad (e, f) \in A, &\text{ satisfaz uma função côncava} \end{aligned}$$

Seja x um fluxo viável, ou seja, satisfaz as restrições acima. Então, definamos:

x é um fluxo extremo $\iff \nexists x^1, x^2 \in$ região viável do PPL acima; $x = \frac{x^1 + x^2}{2}$

Eles correspondem às soluções básicas viáveis usando o método simplex.

Seja $E = \{e_1, e_2, \dots, e_n\}$, $n \geq 2$ uma coleção de nós. Se tivermos $(e_i, e_{i+1}) \in A$ para $i = 1, 2, \dots, n-1$ chamamos de cadeia a seqüência de nós e arcos formada por $e_1 \cup (e_1, e_2) \cup \dots \cup (e_{n-1}, e_n) \cup e_n$. Toda cadeia na qual tivermos $e_1 = e_n$ é chamada de ciclo.

Se na cadeia E tivermos ou (e_i, e_{i+1}) ou (e_{i+1}, e_i) em A , a seqüência de nós e arcos é chamada de caminho. Todo caminho no qual $e_1 = e_n$ é dito ser um laço ("loop").

Definição. Uma cadeia E é dita positiva com relação ao fluxo viável x , se para cada arco $(e_i, e_{i+1}) \in E$, tivermos $x_{e_i e_{i+1}} > 0$.

Definição. Duas cadeias E_1 e E_2 são ditas cadeias separadas se as duas possuem um único ponto em comum,

Definição. Um nó f é dito ser uma entrada positiva de um nó e e com relação a uma solução viável x , se $(f, e) \in A$ e $x_{fe} > 0$.

A seguir, um resultado que não vai ser demonstrado aqui, mas que pode ser obtido do texto abaixo:

Dantzig, G.B. *Linear Programming and Extensions*. Princeton University Press, Princeton, N.J., 1963.

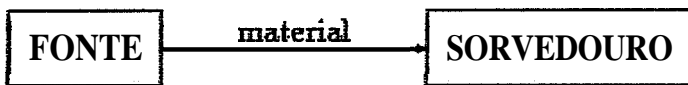


Figura VII.5: Um exemplo ilustrativo

Lema VII.1 *Seja dado x um fluxo viável de um PPL com as mesmas características do anterior, ie, funções custo côncavas. Então x é um fluxo extremo se e só se não há nenhum caminho fechado que seja positivo em relação a x .*

Teorema VII.5 *Sejam r um fluxo extremo em uma rede simples e e um nó dado. Se e tem k cadeias fonte separadas então e tem no máximo k entradas positivas.*

Demonstração

Para cada entrada positiva f , consideremos o conjunto $B(f) = \{p; (p, f) \in A\}$. Se existir algum $p_1 \in B(f)$; $x_{p_1 f} > 0$, façamos com o nó p_1 a mesma análise feita com o nó f , e deixemos o arco (p_1, f) registrado. Se existir $p_2 \in B(p_1)$; $x_{p_2 p_1} > 0$, repetamos a mesma análise, agora com p_2 . E assim por diante, até que tenhamos obtido a cadeia $\{P_n, P_{n-1}, \dots, P_3, P_2, P_1, f\}$ tal que $\{y \in B(p_n) \text{ e } x_{yp_n} > 0\} = \emptyset$. Nesse caso, p_n é uma fonte, pois nele entram apenas fluxos nulos ou nenhum e sai um fluxo positiva

Façamos o mesmo raciocínio para todas as entradas positivas de e . Afirmamos que as fontes p_n assim obtidas são todas distintas. De fato, caso contrário, teríamos duas cadeias positivas com dois nós em comum. Isto originaria um laço positivo. Contradição para o fato de r ser um fluxo extremo.

Como todas as fontes p_n de e são distintas? por hipótese, elas devem ser no máximo k . Mas cada p_n corresponde biunivocamente a uma entrada positiva. O resultado segue-se.

[C.Q.D.]

Corolário VII.1 *Se (N, A) é uma rede com K fontes, e um nó dado e e x um fluxo extremo então e terá no máximo k entradas positivas.*

Corolário VII.2 *Se às hipóteses do corolário acima adicionarmos que e é uma fonte de (N, A) então e terá no máximo $(k - 1)$ entradas positivas.*

Como aplicação do teorema e dos corolários anteriores consideremos uma fonte que fornece material para ser consumido por um sorvedouro, como mostrado na figura VII.5.

A fonte envia material para o sorvedouro em n períodos de tempo, que serão representados doravante pela letra i , com i variando desde 1 até n . Ocorre que em cada período i a fonte envia uma quantidade y_i de material e o sorvedouro

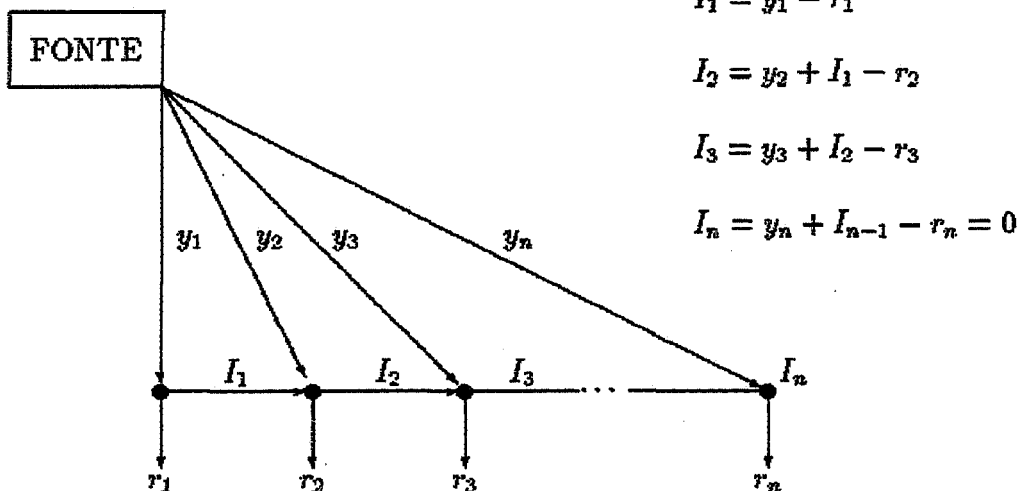


Figura VII.6: A fonte envia material para o sorvedouro

consome uma quantidade r_i . A diferença entre os dois $I_i = y_i - r_i$ é armazenada em estoque para posterior utilização no período de tempo $i + 1$. Veja a representação na figura VII.6 (cada nó é um período de tempo).

Como não nos interessa guardar mais nada depois da último período de tempo, devemos ter $I_n = 0$.

Sejam os custos abaixo:

$p_i(y_i)$ \longrightarrow custo para que a fonte forneça y_i unidades do material no período i .

$H_i(I_i)$ \longrightarrow custo para que sejam guardados em estoque I_i unidades do material em questão no instante i .

O PPL que será resolvido é o seguinte:

$$\text{minimizar } \sum_{i=1}^n p_i(y_i) + \sum_{i=1}^n H_i(I_i)$$

sujeito a P e H são funções côncavas

$$I_i = y_i + I_{i-1} - r_i, \quad 1 \leq i \leq n$$

$$\sum_{i=1}^n y_i = \sum_{i=1}^n r_i$$

$$I_0 = I_n = 0, \quad I_i \geq 0$$

$$r_i, y_i \geq 0, \quad i \in \{1, 2, 3, \dots, n\}$$

$$p_i(0) = H_i(0) = 0, \quad i = 1, 2, \dots, m$$

Vamos supor que tenhamos determinado a solução ótima X^* desse PPL. Essa solução, evidentemente, é um fluxo extremo. Consideremos um nó i , que representa o período i . Como o nosso PPL só possui uma única fonte, esse nó i

(pelo corolário) deve ter no máximo uma entrada positiva. Assim, se $y_i > 0$ então $I_{i-1} = 0$.

VII.8 O Método de Wolfson para o Problema Unidimensional

Suponha que desejemos satisfazer uma demanda N_I por "tamanhos" L_I onde $I = 1, 2, \dots, K$. Admita $N_1 \leq N_2 \leq \dots \leq N_K$. Evidentemente, qualquer que sejam os tipos de tamanhas que serão mantidos em estoque, o item de estoque $I = K$ será guardado. Quando uma peça solicitada não possui similar em estoque, escolhe-se uma peça do estoque cujo tamanho seja maior que o da peça dada. A peça maior é então cortada e a demanda satisfeita.

Seja o número $A_2(I, K)$ para um dado $I \leq K - 1$ que representa a perda obtida ao se estocar os tamanhos I e K em quantidade suficiente para se satisfazer a demanda. Esta perda é obtida da seguinte forma:

$$A_2(I, K) = \sum_{T < I} (L_I - L_T) \cdot N_T + \sum_{I < T < K} (L_K - L_T) \cdot N_T$$

Assim, se representarmos por J_S o custo mínimo caso nos decidíssemos por armazenar S tipos de itens de estoque diferentes, então:

$$J_2 = \min\{A_2(I, K), I \leq K - 1\}$$

Genericamente falando, seja o número $A_R(I, K)$ para um dado $I \leq K - R + 1$ que representa a perda obtida ao se estocas R tamanhos arbitrários entre I e K , inclusive. Assim:

$$A_{R+1}(I, K) = \min\{A_2(I, J) + A_R(J, K); J = I + 1, I + 2, \dots, K - R + 1\}$$

$$A_2(I, J) = \sum_{T < I} (L_I - L_T) \cdot N_T + \sum_{I < T < J} (L_J - L_T) \cdot N_T$$

Logo,

$$J_S = \min\{A_S(I, K), I \leq K - S + 1\}$$

A resposta para o problema é portanto:

$$J = \min\{J_S, 1 \leq S \leq K\}$$

Muitas vezes, opta-se por escolher entre diversas opções de número de elementos estocados, aquela que for a melhor levando-se em conta outros fatores que não apenas as sobras resultantes dos cortes.

Capítulo VIII

Um Algoritmo para o Problema de Alocação em duas Dimensões

VIII.1 Introdução

Este capítulo foi baseado no paper de J. E. Beasley [2].

O problema pode ser descrito da seguinte forma. Dispomos de um estoque formado por peças retangulares. Desejamos extrair um subconjunto de este estoque, cujas peças seriam cortadas para satisfazer uma demanda por pedaços retangulares. Temos, assim, dois subproblemas:

- (i) Qual é o subconjunto do estoque que deve ser extraído? Ou seja, que tipos de peças serão selecionadas e qual a quantidade de cada tipo? Existe um limite para a quantidade de tipos selecionados de peça.
- (ii) Qual é a melhor forma de se cortar uma dada peça do estoque? Existe um limite superior e inferior para a quantidade de pedaços de um determinado tipo ~~gerada~~ por todos a cortes.

VIII.2 Formulação do Problema

Notação. Temos para as definições abaixo: $i = 1, 2, \dots, n$ e $j = 1, 2, \dots, m$.

(1) Retângulos do estoque

n \mapsto número de diferentes tipos (tamanhos) de **retângulos** em estoque disponíveis,

L_i \mapsto comprimento de um item de **estoque** do tipo i ,

W_i \mapsto largura de um ítem de **estoque** do tipo i ,

f_i \mapsto custo fixo **associado à utilização** de um ítem de **estoque** do tipo i para corte,

C \mapsto **quantidade máxima** de tipos diferentes de Itens de estoque que são **utilizados nos cortes**, e

c_i \mapsto **custo** por unidade de **área não utilizada** de um ítem de estoque qualquer.

(2) Pedacos

m \mapsto número de tipos **diferentes** de **pedacos** que são **gerados nos** diversos cortes **para satisfazer a demanda**,

l_j \mapsto comprimento de um **pedaço** do tipo j gerado pelos diversos cortes,

w_j \mapsto largura de um **pedaço** do tipo j gerado pelos diversos cortes

v_j \mapsto valor associado a um **pedaço retangular** do tipo j ,

a_j \mapsto limite inferior para o número de **pedacos** do tipo j gerados pelos diversos cortes ($a_j \geq 0$), e

b_j \mapsto limite superior para o número de pedacos do tipo j gerados pelos diversas cortes ($b_j \geq a_j$).

(3) Modos de Cortar

$P(i)$ \mapsto conjunto formado por todas as maneiras de se cortar o item de estoque do tipo i , e

d_{jpi} \mapsto número de pedacos do tipo j que são gerados pelo modo de cortar $p \in P(i)$.

(4) Variáveis do Problema

x_{pi} \mapsto número de vezes em que o modo de cortar $p \in P(i)$ é executado em uma solução do problema, e

u_j \mapsto número de pedacos cortados do tipo j em uma solução do problema.

$$y_i = \begin{cases} 1, & \text{se o ítem de estoque } i \text{ for utilizado} \\ 0, & \text{caso contrário} \end{cases}$$

O problema pode ser escrito *como*:

$$\text{minimizar } \sum_{i=1}^n \sum_{p \in P(i)} (f_i + c_w L_i W_i) \cdot x_{pi} - \sum_{j=1}^m (v_j + c_w l_j w_j) \cdot u_j,$$

$$\begin{aligned} \text{sujeito a } \quad & a_j \leq u_j \leq b_j, \quad j = 1, 2, \dots, m, \\ & u_j \leq \sum_{i=1}^n \sum_{p \in P(i)} d_{jpi} \cdot x_{pi}, \quad j = 1, 2, \dots, m, \\ & \sum_{i=1}^n y_i \leq k, \\ & \sum_{p \in P(i)} x_{pi} \leq M y_i, \quad i = 1, 2, \dots, n, \\ & y_i \in (0, 1), \quad i = 1, 2, \dots, n, \\ & x_{pi} \geq 0, \quad i = 1, 2, \dots, n, \quad \forall p \in P(i) \\ & u_j \geq 0, \quad j = 1, 2, \dots, m. \end{aligned}$$

A desigualdade $u_j \leq \sum_{i=1}^n \sum_{p \in P(i)} d_{jpi} x_{pi}$, $j = 1, 2, \dots, m$, nos permite considerar uma gama maior de padrões de corte para se obter a solução de nosso problema, que não seria possível se fôssemos obrigados a aproveitar todos os pedaços (l_j, w_j) gerados.

A desigualdade $\sum_{p \in P(i)} x_{pi} \leq M y_i$, $i = 1, 2, \dots, n$, ocorre para garantir que os padrões de corte utilizados se referem apenas a itens de estoque também utilizados. Assim, tomamos um valor para M relativamente grande.

Podemos simplificar a função objetivo, definindo:

$$\begin{cases} F_i = f_i + c_w L_i W_i, & i = 1, 2, \dots, n, \\ V_j = v_j + c_w l_j w_j, & j = 1, 2, \dots, m. \end{cases}$$

A função objetivo torna-se:

$$\sum_{i=1}^n \sum_{p \in P(i)} F_i x_{pi} - \sum_{j=1}^m V_j u_j$$

Uma simplificação que pode ser feita no modelo acima é fazermos $f_i = v_j = 0$, $\forall i, j$ e $c_w = 1$.

O número de itens de estoque que serão utilizados em nossa solução ótima pode ser considerado k . Isso é possível, porque podemos fazer artificialmente $y_i = 1$ para os itens de estoque que não venham a ser utilizados, sem afetar a solução final. Em virtude disso, o problema pode ser dividido em dois problemas menores:

- (1) Seleção dos padrões de corte a serem utilizados. Dado um conjunto K de itens de estoque, qual é a melhor forma de cortá-los de modo que a demanda seja satisfeita.
- (2) Seleção dos itens de estoque. Das n itens estocados quais serão aqueles que comporão o nosso conjunto de tamanho k .

VIII.3 Os Algoritmos Utilizados

VIII.3.1 Seleção dos Padrões de Corte

Seja K o conjunto formado por todos os itens de estoque possíveis de serem cortados.

Geração Inicial dos Padrões

Para cada $i \in \{1, 2, \dots, n\}$ e $j \in \{1, 2, \dots, m\}$ resolva o problema de cortes em duas dimensões, definido por:

- (1) o retângulo estocado de dimensões $L_i \times W_i$;
- (2) o pedaço $l_j \times w_j$ com valor V_j ;
- (3) o pedaço $l_q \times w_q$ ($q \neq j$, $q = 1, 2, \dots, m$) com valor V_q/N , onde N é uma constante positiva. Fazendo N relativamente grande, o problema nos fornece a maneira de se obter a maior quantidade possível de pedaços $l_j \times w_j$ da item de estoque i .

Se utilizarmos para resolver o problema o método de programação dinâmica dado por Gilmore e Gomory então basta resolvermos a problema, para o retângulo $\max\{L_i; i = 1, 2, \dots, n\} \times \max\{W_i; i = 1, 2, \dots, n\}$. As soluções intermediárias, obtidas dos diversos retângulos menores, podem ser colocadas em uma tabela para consultas eventuais.

Procedimento Guloso

Este procedimento consiste em se calcular para algum retângulo i selecionado do estoque a melhor forma de cortá-lo que gera a quantidade de retângulos necessária para satisfazer a demanda de pelo menos um tipo de pedaço solicitado a um custo mínimo. Uma vez que a demanda de pelo menos um tipo de retângulo tenha sido satisfeita, o problema é resolvido novamente para se obter um modo de se cortar algum outro retângulo da estoque, agora dessa vez para satisfazer a demanda por uma quantidade menor de pedaços.

Seja $K \subset \{L_i \times W_i, i = 1, 2, \dots, n\}$.

Consideremos os conjuntos abaixo:

$$Q_1 = \{j/a_j \geq 1, j = 1, 2, \dots, m\}$$

$$Q_2 = \{j/a_j = 0, j = 1, 2, \dots, m\}$$

Se o conjunto Q_1 h vazio, então não há necessidade de se efetuar cortes nos retângulos do conjunto K .

Tome um retângulo i em K . Considere cada pedaço j em Q_1 como tendo valor V_j e cada pedaço j em Q_2 como tendo valor V_j/M com M grande.

Problema: Dado o retângulo $L_i \times W_i$, qual é o maior valor que eu posso extrair dele a partir dos pedaços em Q_1 e Q_2 com os valores associados acima. respeitadas os limites b_j ?

Solução: trata-se de um problema da mochila em duas dimensões com restrições. Algoritmos que podem ser utilizados são os propostos por "Nicos Christofides e Charles Whitlock" ou por "P. Y. Wang". Podemos simplificar o problema e considerar o retângulo $\max\{L_i; i = 1, 2, \dots, n\} \times \max\{W_i; i = 1, 2, \dots, n\}$. Para esse retângulo aplicamos o método de programação dinâmica proposto por Gilmore e Gomory. Ele nos dará a solução ideal (sem restrições) para todos os retângulos cujas dimensões caibam no retângulo considerado. Depois, desprezamos aqueles pedaços na solução dada que extrapolam os limites b_j e os consideramos como perdas.

Sejam A_i o maior valor possível de se extrair do i -ésimo retângulo de modo que os pedaços gerados não ultrapassem os limites b_j , e e_{ij} o número de pedaços $l_j \times w_j$ presentes nessa solução.

Tomemos t o retângulo para o qual seu valor ótimo A_t mais se aproxima de seu custo F_t , e que satisfaça a demanda de pelo menos um pedaço solicitado:

$$F_t - A_t = \min\{F_i - A_i / i \in K \text{ e } \exists j \in Q_1, e_{ij} \geq 1\}$$

(As colisões são resolvidas arbitrariamente)

Estamos supondo $F_t - A_t \geq 0$ que é o caso da maioria das situações.

Para o retângulo $L_t \times W_t$ escolhido anteriormente, já foi determinada a melhor forma de cortar. Com essa forma de cortar sendo executada, s vezes, temos que cada pedaço foi gerado um certo número de vezes, dada por: $s \cdot e_{t1}$ para o pedaço $l_1 \times w_1$; $s \cdot e_{t2}$ para o pedaço $l_2 \times w_2$, e assim por diante.

A idéia é tomarmos s como sendo:

$$s = \min\left\{\left\lceil \frac{a_{tj}}{e_{tj}} \right\rceil; e_{tj} \geq 1, j \in Q_1\right\}$$

O tipo de corte ideal para o retângulo t será executado um certo número s de vezes, de modo que a demanda, de pelo menos um pedaço é satisfeita.

A seguir, atualizamos:

$$\begin{aligned} a_j &\leftarrow a_j - \min\{a_j, e_{tj} \cdot s\} \quad j \in Q_1 \\ b_j &\leftarrow b_j - e_{tj} \cdot s \quad \forall j \text{ tal que } b_j \text{ está definido} \end{aligned}$$

Utilizamos o mínimo para o caso de algum $e_{tj} \cdot s$ ultrapassar a demanda estabelecida a_j .

Em seguida, atualizamos Q_1 e Q_2 e repetimos o problema. Como o novo Q_1 passa a ter no mínimo um elemento a menos então esse algoritmo deve convergir no máximo em m iterações.

Programa Linear

Os algoritmos das seções anteriores serão utilizados para gerarmos uma aproximação para $P(i)$, o conjunto formado por todos os modos de cortar o item de estoque $L_i \times W_i$.

Aplicamos o algoritmo que nos dá uma "Geração Inicial das Padrões". Ele nos fornece um padrão de corte para cada i e j , onde $i \in \{1, 2, \dots, n\}$ e $j \in \{1, 2, \dots, m\}$. Esses padrões *de corte* farão parte do conjunto $P(i)$.

Aplicamos depois o algoritmo "Guloso". Ele seleciona um máximo de m padrões de corte. Também estes serão incluídos em $P(i)$.

Uma vez determinada uma aproximação para $P(i)$, utilizaríamos o algoritmo simplex para resolvermos o PPL abaixo:

$$\text{minimizar} \quad \sum_{i \in K} \sum_{p \in P(i)} F_i \cdot x_{pi} - \sum_{j=1}^m V_j \cdot u_j,$$

$$\text{sujeito a} \quad a_j \leq u_j \leq b_j, \quad j = 1, 2, \dots, m,$$

$$u_j \leq \sum_{i \in K} \sum_{p \in P(i)} d_{jpi} \cdot x_{pi}, \quad j = 1, 2, \dots, m,$$

$$x_{pi} \geq 0, \quad i \in K, \quad \forall p \in P(i)$$

$$u_j \geq 0, \quad j = 1, 2, \dots, m.$$

A solução encontrada na maioria dos casos não é inteira. Sendo assim, faz-se necessário obter uma solução inteira relativamente próxima do ótimo.

A seguir, descrevemos o que chamamos de "método de simples arredondamento".

Procure diminuir de uma unidade os valores das variáveis x_{pi} , de modo que para cada j tenhamos $\sum_{i \in K} \sum_{p \in P(i)} d_{jpi} \cdot x_{pi} \geq a_j$ e $x_{pi} \geq 0 \quad \forall p \in P(i), \forall i \in K$.

Dessa forma, como deveremos ter $u_j \leq \sum_{i \in K_p \in P(i)} d_{jpi} \cdot x_{pi}$, $j = 1, 2, \dots, m$, é possível que u_j também diminua de uma unidade. Porém, como fizemos de modo que $\sum_{i \in K_p \in P(i)} d_{jpi} \cdot x_{pi} \geq a_j$, ainda teremos $a_j \leq u_j \leq b_j$. Ao diminuirmos os x_{pi} 's de uma unidade, nós nos aproximamos da solução ótima anteriormente obtida, que era fracionária.

Uma vez os x_{pi} 's sendo inteiros, os u_j 's também o serão, em virtude de que temos um número inteiro de pedaços l_j, w_j) presentes em cada modo de cortar $p \in P(i)$.

Uma vez definido o que chamamos de método de simples arredondamento, para obtermos as soluções inteiras (x_{pi}) e (u_j) do PPL anterior, aplicamos o seguinte algoritmo:

- (1) Resolva o PPL relaxado, isto é, sem as restrições inteiras. Faça em paralelo (2) e (3);
- (2)
 - (2.1) Arredonde a solução encontrada para cima;
 - (2.2) Procure reduzir alguns valores inteiros obtidos de uma unidade, como no "método de simples arredondamento". Vá para (4);
- (3)
 - (3.1) "Gomory f-cuts":
 - * Escolha a variável x_{pi} , cuja parte fracionária mais se aproxima de 0,5;
 - * Resolva o novo PPL com o método dual simplex;
 - (3.2) A solução obtida é inteira? Se a resposta for sim vá para (3.3). Se a resposta for não vá para (3.4);
 - (3.3) Procure reduzir alguns valores inteiros obtidos de uma unidade, como no "método de simples arredondamento". Vá para (4);
 - (3.4) O número de "f-cuts" já acrescentados ao PPL é muito grande? Se a resposta for sim vá para (3.5). Se a resposta for não vá para (3.1);
 - (3.5) Seja X_{pi} o valor associado à variável x_{pi} . Adicione $x_{pi} \geq [X_{pi}]$ ao PPL, onde x_{pi} é a que contém a maior parte fracionária;
 - (3.6) Resolva o PPL usando o método dual simplex;
 - (3.7) A solução obtida é inteira? Se a resposta for sim vá para (3.3). Se a resposta for não vá para (3.5);
- (4) Compare os dois resultados obtidos: o primeiro em (2); o segundo em (3). Depois escolha o melhor. FIM.

VIII.9.2 Seleção dos Retângulos do Estoque

A fim de se selecionar o melhor subconjunto K dos n retângulos em estoque, deveríamos aplicar o algoritmo que seleciona os padrões de corte em cada um dos possíveis subconjuntos $K \subset \{1, 2, \dots, n\}$ e então comparar os resultados obtidos. Ao todo, estudaríamos $\frac{n!}{k!(n-k)!}$ subconjuntos. Aquele subconjunto K cujo modo de cortar ótimo satisfaz a demanda com o menor custo entre todos os demais subconjuntos é o que seria escolhido.

Como o número de subconjuntos a ser estudado é, em geral, muito grande, optamos por uma heurística capaz de oferecer uma solução aceitável com um menor tempo utilizado:

- (1) Aplique o algoritmo que seleciona os padrões de corte em $K = \{1, 2, \dots, n\}$: obtenha a melhor forma de se cortar os retângulos $(L_1, W_1), (L_2, W_2), \dots, (L_n, W_n)$ de modo a se gerar u_j pedaços (l_j, w_j) satisfazendo $a_j \leq u_j \leq b_j$. Nessa solução ótima, cada retângulo i do estoque é utilizado um certo número U_i de vezes.
- (2) Reordene os índices i de modo que $U_1 \geq U_2 \geq \dots \geq U_n$.
- (3) Defina $K = \{1, 2, \dots, k-1, g\}$, onde $U_g = \max\{U_i / i \geq k\}$ e cada pedaço (l_j, w_j) , $j = 1, 2, \dots, m$, pode ser obtido a partir de cortes em pelo menos um dos retângulos $(L_1, W_1), (L_2, W_2), \dots, (L_{k-1}, W_{k-1}), (L_i, W_i)$. Se tal g não existe, então o problema não admite solução.
Se $g = k$ e $U_{k+1} = 0$ então o problema está terminado.
- (4) Considere um T , $T \leq n-k$. Construa o conjunto $S_T = \{i / U_i \text{ está no conjunto dos } T \text{ maiores } U_i\}$ em valor que não estão presentes em K .
- (5) Para cada par $(i_1, i_2) \in K \times S_T$, faça:
 - (i) Aplique o algoritmo que seleciona os padrões de corte para $K = K - \{i_1\} + \{i_2\}$ e avalie o resultado.
 - (ii) Se houve uma melhora no resultado obtido então faça $K = K - \{i_1\} + \{i_2\}$ e $S_T = S_T + \{i_1\} - \{i_2\}$. Quando não houver mais pares $(i_1, i_2) \in K \times S_T$ que produzam uma melhora ou o tempo computacional tiver ido além do limite, o problema, acaba.

Capítulo IX

CONCLUSÕES

De acordo com A. I. Hinxman, o problema tratado pela tese **não** está definido adotando-se um padrão. Isto é, existem diversas denominações desta classe de problemas, e dada uma delas, o mesmo pode ser posto de formas diversas.

Os métodos apresentados são, conforme o trabalho mostra, claramente convergentes. A diversidade de problemas para os quais eles se aplicam torna difícil realizar-se comparações, pois as problemas são diferentes.

O autor da tese acredita numa tendência a se utilizar processamento paralelo na solução desses problemas, visto que a maioria deles se utiliza, para sua solução, do cálculo de funções mochila, que é paralelizável.

A tese contribuiu no sentido de mostrar um tipo de problema e m diversas perspectivas numa forma didática, e seu relacionamento com diversos outros. Por exemplo, o estudante encontrará nesse material uma ligação entre o problema de cortes e o de estocagem, nem sempre evidente. Em virtude da método didático, os algoritmos apresentados serão mais facilmente entendidos e, portanto, implementados.

BIBLIOGRAFIA

- [1] Barnett, S. e G. J. Kynch (1967) *Exact Solution of a Simple Gutting Problem* Operations Research
- [2] Beasley, J. E. (1985) *An Algorithm for the Two-Dimensional Assortment Problem* European Journal of Operational Research 19
- [3] Bellman, Richard *Some Applications of the Theory of Dynamic Programming - A Review* Operations Research 2
- [4] Gani, Philip de *A Note on the Two-Dimensional Rectangular Cutting-Stock Problem* Journal Operational Research Society 29
- [5] Christofides, Nicos e Charles Whitlock *An Algorithm for Two-Dimensional Cutting Problems* Operations Research 25
- [6] Chvátal, Vásek *Linear Programming* W. H. Freeman and Company
- [7] Dągli, Gihan H. (1990) *Knowledge-Based Systems for Cutting Stock Problems* European Journal of Operational Research 44
- [8] Dantzig, George B. e Philip Wolfe *Decomposition Principle for Linear Programming* Operations Research 8
- [9] Dantzig, George B. *Discrete Variable Extremum Problems* Operations Research 5, pág. 266 a 277
- [10] Dantzig, G. B. (1963) *Linear Programming and Extensions* Princeton University Press, Princeton, New Jersey
- [11] Diegel, A., e H. J. Bocker (1984) *Optimal Dimensions of Virgin Stock in Cutting Glass to Order* Dec. Sci. 15, pág. 260 a 274
- C121 Dyckhoff, Harald *A New Linear Programming Approach to the Gutting Stock Problem* Operations Research 29
- C133 Farley, Alan A. (1990) *The Cutting Stock Problem in the Canvas Industry* European Journal of Operational Research 44
- C141 Gemmill, Douglas D. e Jerry L. Sanders (1990) *Approximate Solutions for the Gutting Stock 'Portfolio' Problem* European Journal of Operational Research 44

- C153 Ferreira, J. Soeiro e José Fernando Oliveira (1990) *An Improved Version of Wang's Algorithm for Two-Dimensional Cutting Problems* European Journal of Operational Research 44
- C163 Gilmore, P. C. e R. E. Gomory *A Linear Programming Approach to the Cutting Stock Problem* Operations Research 9, pág. 849 a 859
- C173 Gilmore, P. C. e R. E. Gomory *A Linear Programming Approach to the Cutting Stock Problem - Part II* Operations Research 11, pág. 863 a 888
- [18] Gilmore, P. C. e R. E. Gomory *Multi-Stage Cutting Stock Problems of Two and More Dimensions* Operations Research 13, pág. 94 a 120
- C191 Gilmore, P. C. e R. E. Gomory *The Theory and Computation of Knapsack Functions* Operations Research 15, pág. 1045 a 1075
- C203 Goulimis, Constantine (1990) *Optimal Solutions for the Cutting Stock Problem* European Journal of Operational Research 44
- C213 Herz, J. C. (September, 1972) *Recursive Computational Procedure for Two-Dimensional Stock Cutting* IBM Journal Research Developing
- C223 Hinxman, A. I. (1980) *The Trim Loss and Assortment Problem: A Survey* European Journal of Operational Research 5
- C233 Pentico, David W. *The Discrete Two-Dimensional Assortment Problem* Operations Research 36 número 2
- C243 Salkin, Harvey M. e Cornelis A. de Kluyver *The Knapsack Problem: A Survey* Nav. Res. Log. Quart. 22
- C253 Shier, D. R. (1978) *Mathematical Models for Selecting Catalogs of Standard-Sized Items* National Bureau of Standards Report NBSIR-78-1539
- C263 Stadtler, Hartmut (1990) *A One-Dimensional Cutting Stock Problem in the Aluminium Industry and its Solution* European Journal of Operational Research 44
- C273 Wang, P. Y. *Two Algorithms for Constrained Two-Dimensional Cutting Stock Problems* Operations Research 31 número 3
- C283 Wolfson, M. L. (1965) *Selecting The Best Lengths to Stock* Operations Research 13
- C293 Zangwill, Willard I. *Minimum Concave Cost Flows in Certain Networks* Management Science 14 número 7