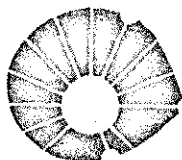


UM ALGORITMO DO TIPO PARTIÇÃO-LIMI-
TAÇÃO PARA O PROBLEMA DE REPRE-
SENTAÇÃO DE CONJUNTOS

VALDOMIRO NEVES LIMA



UNICAMP

UNIVERSIDADE ESTADUAL DE CAMPINAS

INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E CIÊNCIA DA COMPUTAÇÃO

CAMPINAS - SÃO PAULO
BRASIL

UM ALGORITMO DO TIPO PARTIÇÃO-LIMITAÇÃO PARA O PROBLEMA
DE REPRESENTAÇÃO DE CONJUNTOS.

Este exemplar corresponde à redação
final da tese devidamente corrigida e
defendida por VALDOMIRO NEVES LIMA e
aprovada pela comissão julgadora.

Campinas, 21 de Dezembro de 1987


Prof. Dr. CLOVIS PERIN FILHO

Orientador

Dissertação apresentada ao Instituto
de Matemática, Estatística e Ciência
da Computação da UNICAMP, como requi-
sito parcial para obtenção do título
de Mestre em Matemática Aplicada.

"Toda vida existe pra iluminar

o caminho de outras vidas que a gente encontrar"

Milton Nascimento / Fernando Brant

À Salu: luz mãe

À Aurea: luz companheira

À Juliana: luz filha

À Carolina: luz filha

AGRADECIMENTOS

À Universidade Federal Rural do Rio de Janeiro e ao PJC/D/CAPES pelas facilidades e bolsas oferecidas;

Aos colegas, professores e funcionários do Instituto de Matemática, Estatística e Ciência da Computação da UNICAMP, pela colaboração.

A todos que iluminaram esta caminhada, pelo incentivo.

E em especial, ao Professor Clóvis Perin Filho, pela confiança, paciência, orientação segura, constante e amiga ao longo destes anos de trabalho.

SUMÁRIO

	PG.
CAPÍTULO 1: INTRODUÇÃO	
1.1 - Problemas Estudados.....	1
1.1.1 - O Problema de Representação de Conjuntos (PRC)....	1
1.1.2 - O Problema de Cobertura com Conjuntos (PCC).....	4
1.1.3 - O problema de Partição com Conjuntos (PP).....	6
1.1.4 - O Problema de Empacotamento com Conjuntos (PEC)...	7
1.1.5 - O Problema de Representação com Cardinalidade Mínima (PRCM).....	7
1.1.6 - O Problema de Emparelhamento Máximo com Ares- tas (PEMA).....	9
1.1.7 - O Problema de Emparelhamento Perfeito (PEP).....	10
1.1.8 - O Problema de Cobertura com Arestas (PCA).....	11
1.1.9 - O Problema de Cobertura com Nós (PCN).....	11
1.2 -Relação entre os Problemas Formulados.....	12
1.3 -Reduções no Problema de Partição (PP) e no Proble- ma de Cobertura com Conjuntos (PCC).....	14
1.4 -Algumas Aplicações.....	15
1.4.1 - Problema de Entrega de Mercadorias.....	16
1.4.2 - Localização de Hidrantes de Bombeiro.....	16
1.4.3 - Recuperação de Informações.....	17
1.4.4 - Programação de Tripulação em Linhas Aéreas.....	17
1.4.5 - Localização de Serviços Emergenciais em uma Co- munidade.....	18
CAPÍTULO 2: ASPECTOS TEÓRICOS DO ALGORITMO UTILIZADO	
2.1 - Resultados Teóricos.....	20
2.1.1 - Algoritmo para o (PRCM).....	20
2.1.2 - O Método de Partição-Limitação.....	28
2.1.3 - Relaxação Lagrangeana.....	29
2.1.4 - Otimização com Subgradiente.....	35

CAPÍTULO 3: PROBLEMA DE COBERTURA COM ARESTAS

<i>3.1 - Formulação Matemática.....</i>	<i>44</i>
<i>3.2 - Conceitos.....</i>	<i>46</i>
<i>3.3 - Etapas do Algoritmo.....</i>	<i>49</i>
<i>3.4 - O Algoritmo.....</i>	<i>53</i>

CAPÍTULO 4: ALGORITMO PARA RESOLUÇÃO DO PROBLEMA DE REPRESENTAÇÃO DE CONJUNTOS (PRC).....

<i>4.1 - Relação entre o (PRC) e o (PCA).....</i>	<i>56</i>
<i>4.2 - O Algoritmo.....</i>	<i>57</i>
<i>4.2.1 - Estratégia de Limitação.....</i>	<i>59</i>
<i>4.2.2 - Estratégia de Partição.....</i>	<i>61</i>
<i>4.2.3 - Execução.....</i>	<i>63</i>

CAPÍTULO 5: RESULTADOS COMPUTACIONAIS.....

REFERÊNCIAS BIBLIOGRÁFICAS.....

112

CAPÍTULO 1

INTRODUÇÃO

A motivação principal da realização deste trabalho foi a observação de que um grande número de problemas práticos de otimização exigem soluções inteiras. Mais especificamente, soluções inteiras do tipo 0-1.

A partir de um levantamento de aplicações de programação 0-1 foi escolhido como tema do presente trabalho o estudo do *problema de Representação de Conjuntos (PRC)* e outros problemas relacionados.

Estes problemas apresentam um número grande de aplicações e estas aplicações são provenientes de áreas distintas. O problema em si apresenta uma estrutura simples, com um conjunto de restrições lineares com coeficientes 0-1.

1.1 - Problemas Estudados

1.1.1 - O Problema de Representação de Conjuntos (PRC)

Considere o problema de desconexão de uma rede de distribuição de energia elétrica. Deseja-se analisar o rompimento simultâneo de todos os caminhos que conectam as estações ① e ⑤ da rede $G = (N, A)$ abaixo, onde o conjunto de estações é $N = \{1, 2, 3, 4, 5\}$ e o conjunto de linhas de transmissão é $A = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$

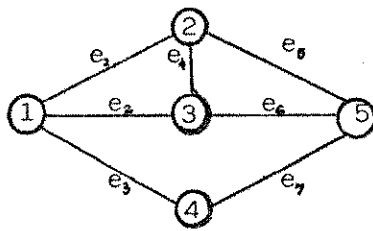


FIGURA I

Identificando os caminhos que conectam as estações (1) e (5) através de sequências de linhas de A , tem-se:

caminho	sequência de linhas
P_1	e_1, e_5
P_2	e_1, e_4, e_6
P_3	e_2, e_6
P_4	e_2, e_4, e_5
P_5	e_3, e_7

Seja $P = \{P_1, P_2, P_3, P_4, P_5\}$ a família de caminhos de (1) à (5), sem repetição de linhas.

Observe que o corte da linha e_1 desconecta os caminhos P_1 e P_2 e que os caminhos P_3, P_4 e P_5 continuam conectando as estações (1) e (5). Entretanto, o corte das linhas e_1, e_2, e_3 desconecta todos os caminhos entre essas estações. Supondo que à linha e_i esteja associado o custo c_i , deseja-se determinar um conjunto de linhas cujo corte desconecte as estações (1) e (5) e que apresente o mínimo custo total.

Este problema é um caso típico do *problema de Representação de Conjuntos (PRC)* que será, agora, formulado:

Considere um conjunto $\tau = \{1, 2, 3, \dots, n\}$

Considere uma família de m subconjuntos, não-vazios de τ , denotada por $F = \{F_1, F_2, F_3, \dots, F_m\}$

O problema de Representação de Conjuntos (PRC) consiste em selecionar um subconjunto S tal que

(a) $S \subseteq r$

(b) $S \cap F_i \neq \emptyset ; i = 1, 2, 3, \dots, m$

(c) $\sum_{j \in S} c_j$ é minimizada para todo S que satisfaz (a) e (b), onde c_j é o custo de inclusão do elemento j no conjunto S e $j = 1, 2, 3, \dots, n$

Definições:

-Se S satisfaz (a) e (b) acima, diz-se que S é um Representante.

-Se $S \cap F_i \neq \emptyset$, diz-se que S representa o subconjunto F_i na família F .

-Se S é um Representante e se o elemento $j \in S \cap F_i$, diz-se que o subconjunto F_i é representado por j em S .

Existindo elementos $j \in r$ tais que $c_j \leq 0$, pode-se incluir todos eles em S sem perder otimalidade. Assim, sem perda de generalidade, pode-se assumir que $c_j > 0; j=1, 2, 3, \dots, n$. No caso do PRC apresentar elementos com custos não positivos, este problema pode ser reduzido a um PRC onde:

a) r está restrito às variáveis originais j tais que $c_j > 0$

b) F está restrita aos subconjuntos F_i que ainda não foram representados por aqueles elementos incluídos em S , por terem custos não positivos.

O Problema de Representação de Conjuntos pode ser formulado como um Programa Linear Inteiro, cuja forma matricial-

al é a seguinte:

$$\begin{aligned} \min z(x) &= c^T \cdot x \\ \text{sujeito a } M \cdot x &\geq 1 \\ x &\in \{0,1\}^n \end{aligned}$$

onde $c^T = (c_1, c_2, \dots, c_n)$

$x^T = (x_1, x_2, \dots, x_n)$

$1^T = (1, 1, 1, \dots, 1) \in \mathbb{R}^m$

Observa-se que:

(1) $M = (m_{ij})$ é uma matriz de incidência ($m \times n$). Assim, denotando por M^j , a j -ésima coluna de M e por M_i , a i -ésima linha de M , a linha M_i indica a incidência do subconjunto F_i e a coluna M^j indica a incidência do elemento $j \in V$ nos subconjuntos F_i .

(2) As correspondências um a um entre os subconjuntos F_i e as linhas M_i de M e, entre os elementos $j \in V$ e as colunas M^j de M serão utilizadas sempre que necessário.

Pode-se notar que o PRC é um tipo muito especial de Programa Linear Inteiro 0-1 pois M é uma matriz binária (seus elementos são 0's e 1's) e o lado direito das desigualdades lineares, que representam as restrições, é um vetor de 1's pertencente a $\mathbb{R}^m$.

1.1.2 - O Problema de Cobertura com Conjuntos (PCC)

Considere o problema de coleta de lixo de 8 ruas $e_1, e_2, \dots, e_8$ de uma determinada região. O caminhão de lixo deve seguir alguma das rotas conhecidas que partem do depósito em ③, seguem por algumas ruas e retornam ao depósito em ③. Tais rotas são apresentadas a seguir:

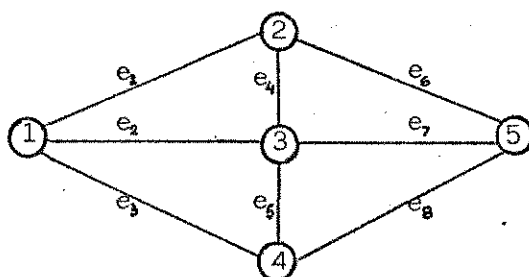


FIGURA II

rota	sequência de ruas
R_1	e_4, e_6, e_7
R_2	e_4, e_1, e_2
R_3	e_2, e_3, e_5
R_4	e_7, e_6, e_1, e_2
R_5	e_5, e_3, e_1, e_4

À cada rota está associado um custo d_i ; $i = 1, 2, \dots, 5$. Observe que a rua e_4 pode ter seu lixo coletado através da rota R_1, R_2 ou R_5 . Deseja-se determinar um conjunto de rotas que permita a coleta do lixo de todas as ruas, mesmo que eventualmente uma delas venha a ser percorrida mais de uma vez. Tal conjunto de rotas deve apresentar o menor custo total, que é a soma dos custos das rotas escolhidas.

Definição:

-Considerando-se os conjuntos $\mathcal{r}$ e F como no PRC, define-se uma Cobertura T , para o conjunto $\mathcal{r}$, pela família F , como sendo uma subfamília $T \subseteq F$ tal que $\bigcup_{i/F_i \in T} F_i = \mathcal{r}$, ou seja, a união

de todos os subconjuntos, não-vazios, F_i de $T \subseteq F$ é o conjunto $\mathcal{r}$.

Designando por d_i ; $i = 1, 2, \dots, m$; o custo de

inclusão do subconjunto F_i na cobertura T para r , obtém-se o *problema de Cobertura com Conjuntos (PCC)*. Este problema busca encontrar uma cobertura T , para r , de custo mínimo e sua formulação como Programa Linear Inteiro é:

$$\begin{aligned} \text{mín } w(y) &= d^T \cdot y \\ [2] \quad \text{suj. } & M^T \cdot y \geq \mathbb{1} \\ & y \in \{0,1\}^m \end{aligned}$$

onde: $d^T = (d_1, d_2, \dots, d_m)$

$y^T = (y_1, y_2, \dots, y_m)$

$\mathbb{1}^T = (1, 1, \dots, 1) \in \mathbb{R}^n$

$M = (m_{ij})$ é a mesma do *PRC*.

1.1.3 - O Problema de Partição com Conjuntos (PP)

Mantendo-se os mesmos significados para $c, M, \mathbb{1}$, r e F utilizados no *PRC*, a formulação do *problema de Partição com Conjuntos (PP)*, como Programa Linear Inteiro 0-1 é:

$$\begin{aligned} \text{mín } z(x) &= c^T \cdot x \\ [3] \quad \text{suj. } & M \cdot x = \mathbb{1} \\ & x \in \{0,1\}^n \end{aligned}$$

[GN1] e [Sal] mostram que o *problema de Partição (PP)* apresenta o mesmo conjunto de soluções ótimas que o *problema de Representação de Conjuntos (PRC)*, desde que o (PP) tenha uma solução viável e que os custos referentes ao (PRC) sejam alterados em função dos coeficientes da matriz M .

Observa-se que se x é uma solução viável para o (PP) esta solução particiona o conjunto de índices das linhas $\{1, 2, 3, \dots, m\}$ da matriz M pois, denotando por $E_1 = \{j/x_j^1 = 1 \text{ e } j \in r\}$ e por $I(j) = \{i/m_{ij} = 1 \text{ para todo } j \in r\}$ tem-se $I(r) \cap I(s) = \emptyset$, para todo $r \in E_1$, para todo $s \in E_1$ e $r \neq s$; e

$$\bigcup_{r \in E_1} I(r) = \{1, 2, 3, \dots, m\}$$

1.1.4 - O Problema de Empacotamento com Conjuntos (PEC)

Definição:

-Define-se um pacote D , em F , como sendo uma subfamília de F cujos elementos F_i são mutuamente disjuntos.

O problema de Empacotamento com Conjuntos (PEC) consiste em encontrar um pacote D que maximize o lucro total $\sum_{i/F_i \in D} d_i$; onde d_i é o lucro obtido pela inclusão de F_i em D . A formulação desse problema em termos de Programa Linear Inteiro 0-1 é:

$$\begin{aligned} \text{MÁX } w(y) &= d^T \cdot y \\ [4] \quad \text{suj. } M^T \cdot y &\geq \mathbb{1} \\ y &\in \{0,1\}^m \end{aligned}$$

onde d^T, y, M e $\mathbb{1}$ são os mesmos definidos no (PCC).

Observa-se que a inclusão de variáveis de folga (que são do tipo 0-1) no problema de Empacotamento, transforma-o num problema de partição (PP).

1.1.5 - O Problema de Representação com Cardinalidade Mínima (PRCM)

Este é um caso especial do problema de Representação de Conjuntos (PRC) obtido ao considerar todos os custos $c_j = 1$, para todo $j \in r$. Sua formulação como Programa Linear Inteiro 0-1 é:

$$\min z(x) = \mathbb{1}^T \cdot x$$

$$[5] \quad \text{su}j. \quad M \cdot x \geq \mathbb{1}$$

$x \in \{0,1\}^n$, que é equivalente a escrevê-

lo como:

$$\min |S|$$

$$\text{su}j. \quad S \cap F_i \neq \emptyset, \text{ para } i=1,2,\dots,m$$

$S \subseteq V$, onde $|S|$ = cardinalidade da
solução S .

Casos particulares dos problemas citados anteriormente e de grande utilidade, por possuírem técnicas eficientes de resolução, são os problemas de Cobertura e Emparelhamento com Arestas num Grafo [Ed1.].

Antes de analisar tais problemas, algumas definições preliminares serão dadas.

Definições:

-Seja $V = \{i/i=1,2,\dots,m\}$ um conjunto finito qualquer. Seja S o conjunto de todos os pares, não ordenados e distintos $\{i,j\}$, de elementos de V , isto é, $S = \{(i;j)/i \in V, j \in V\}$, onde $(i;j)$ e $(j;i)$ representam o mesmo elemento. O par $G = (V,A)$; $A \subseteq S$ é chamado GRAFO NÃO ORIENTADO.

-Os elementos de V são chamados Vértices de G .

-Os elementos de A são chamados Arestas de G .

-Dado um grafo não orientado $G = (V,A)$ e um subconjunto $K \subseteq A$, define-se o Grau do Vértice i com relação ao subconjunto de arestas K como: $g_K(i)$ = número de arestas de K incidentes no vértice i .

-Um subconjunto de arestas $E \subseteq A$ é definido como um Emparelhamento para o grafo G se $g_E(i) \leq 1$, para todo $i \in V$.

-Se E^* é tal que $|E^*| = \text{Máx } \{|E|\}$, $E = \text{emparelhamento para } G$, então E^* é chamado Emparelhamento Máximo para o grafo G , onde $|E| = \text{cardinalidade}(n^\circ \text{ de arestas})$ do subconjunto E .

-Se c_j é o custo de inclusão da aresta a_j no emparelhamento E , define-se o custo do emparelhamento E , como $c(E) = \sum_{j/a_j \in E} c_j$.

Observa-se que todo grafo contém, trivialmente, um emparelhamento pois $E = \emptyset \subseteq A$ e $g_\emptyset(i) = 0 \leq 1$, para todo $i \in V$.

Observa-se, também, que um emparelhamento $\bar{E}$ tem custo máximo se $c(\bar{E}) \geq c(E)$, para todos os emparelhamentos E , para G .

1.1.6 - O Problema do Emparelhamento Máximo com Arestas (PEMA)

Considerando-se $\underline{c}$ e $\underline{x}$ vetores $|A|$ -dimensionais, onde c_j é o custo de inclusão da aresta a_j em E e,

$$x_j = \begin{cases} 1, & \text{se } a_j \in E \\ 0, & \text{caso contrário;} \end{cases}$$

o problema do Emparelhamento Máximo com Arestas é formulado por:

$$\begin{aligned} & \text{máx } \underline{c}^T \cdot \underline{x} \\ [6] \quad & \text{suj. } M \cdot \underline{x} \leq \underline{1} \\ & \underline{x} \in \{0,1\}^{|A|} \end{aligned}$$

onde M é a matriz de incidência (nó $\times$ aresta) do grafo G , ou seja, $m_{ij} = \begin{cases} 1, & \text{se a aresta } a_j \text{ incide no nó } i \\ 0, & \text{caso contrário} \end{cases}$

1.1.7 - O Problema do Emparelhamento Perfeito (PEP)

Uma alteração do emparelhamento máximo com arestas pode ser feita obrigando as restrições serem satisfeitas com igualdade, enquanto procura-se encontrar o menor número de arestas que solucione este problema modificado. Isto fornece a formulação do problema do Emparelhamento Perfeito:

$$\begin{aligned}
 & \min c^T \cdot x \\
 [7] \quad & \text{sujeito a } M \cdot x = \mathbb{1} \\
 & x \in \{0,1\}^{|A|}
 \end{aligned}$$

Observa-se que a estrutura do problema do Emparelhamento Máximo com arestas é a mesma que a do problema do Empacotamento (PEC), enquanto a do problema do Emparelhamento Perfeito coincide com a do problema de Partição.

Definições:

-Um subconjunto de arestas C é uma Cobertura para o grafo $G = (V, A)$ se $g_C(i) \geq 1$, para todos os vértices $i \in V$.

- C^* é uma Cobertura mínima para o grafo $G = (V, A)$ se:
 $|C^*| = \min \{|C| ; C = \text{cobertura para } G\}$

-Define-se o custo da cobertura C , como sendo $c(C) = \sum_{j/a_j \in C} c_j$,

onde a_j é uma aresta de A e c_j é o custo de inclusão da aresta a_j em C .

- C^* tem custo mínimo se $c(C^*) \leq c(C)$, para todas as coberturas C , para G .

1.1.8 - O Problema de Cobertura com Arestas.
(PCA)

Com as últimas definições, formula-se o *problema de Cobertura com Arestas* como o seguinte Programa Linear Inteiro 0-1 :

$$\begin{aligned} & \text{mín } c^T \cdot x \\ [8] \quad & \text{sujeito a } M \cdot x \geq \mathbb{1} \\ & x \in \{0,1\}^{|A|} \end{aligned}$$

onde M é a matriz de incidência (nó-aresta) do grafo.

1.1.9 - O Problema de Cobertura com Nós
(PCN)

Definição:

-Observando o problema de cobertura, tendo por referência os nós (ou vértices) do grafo $G = (V, A)$, pode-se definir Cobertura com Nós como sendo um subconjunto (N) de nós do grafo tal que cada aresta a_j de G , tem ponto extremo em, pelo menos, um nó do subconjunto (N) .

Associando a cada nó i um custo d_i (custo de inclusão do nó i na cobertura N), o *problema de Cobertura com nós*, para o grafo G , será formulado como o seguinte Programa Linear Inteiro 0-1 :

$$\begin{aligned} & \text{mín } w(y) = d^T \cdot y \\ [9] \quad & \text{sujeito a } M^T \cdot y \geq \mathbb{1} \\ & y \in \{0,1\}^{|V|} \end{aligned}$$

onde $y_i = \begin{cases} 1, & \text{se o } i\text{-ésimo nó está na cobertura } N \\ 0, & \text{caso contrário} \end{cases}$

e M é a matriz de incidência (nó-aresta) do grafo $G = (V, A)$

A figura abaixo representa uma das situações anteriormente descritas: grafo $G=(V,A)$ para $V=\{1,2,3,4,5,6\}$ e $A=\{\{1,2\} ; \{1,5\} ; \{2,3\} ; \{2,5\} ; \{3,5\} ; \{3,6\} ; \{3,4\} ; \{4,6\}\}$

Uma cobertura com arestas em G :

$C=\{\{1,2\} ; \{2,5\} ; \{3,5\} ; \{4,6\}\}$

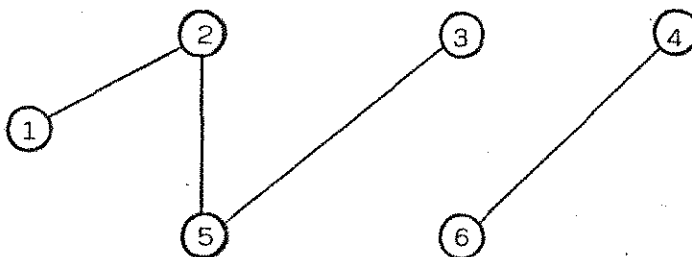


FIGURA III

1.2 - Relação entre os Problemas Formulados

Observando as estruturas dos problemas formulados, em 1.1, vê-se que o *problema de Cobertura com nós (PCN)* possui a estrutura do *problema de Cobertura com Conjuntos (PCC)*, enquanto o *problema de Cobertura com Arestas (PCA)* apresenta a estrutura do *problema de Representação de Conjuntos (PRC)*.

[Mul] aponta para uma equivalência entre os problemas de *Representação de Conjuntos (PRC)* e de *Cobertura com Conjuntos (PCC)* e [Lal.] mostra uma maneira de fazer esta equivalência utilizando grafo bipartido e a semelhança das estruturas desses problemas observadas no parágrafo anterior.

Definição:

-Um grafo $G=(V,A)$ é chamado grafo bipartido se existem subconjuntos V_1 e V_2 de V , tais que $V_1 \cap V_2 = \emptyset$ e $V_1 \cup V_2 = V$, e toda aresta de G é incidente com um vértice de V_1 e um vértice de V_2 .

Com esta definição e considerando o (PRC) formulado em 1.1.1, pode-se construir um grafo bipartido para este problema da seguinte maneira: os nós em V_1 são identificados com os elementos $j \in \mathcal{V}$, os nós em V_2 são identificados com os subconjuntos $F_i \in F$ e existe uma aresta $\{j, i\}$ se e sómente se o elemento j pertence ao conjunto F_i .

Assim, considerando o exemplo que introduziu o (PRC), tem-se:

$$V_1 = \{e_1, e_2, e_3, \dots, e_7\} \text{ e } V_2 = \{P_1, P_2, P_3, \dots, P_5\}$$

$$A = \{(e_1, P_1); (e_1, P_2); (e_2, P_3); (e_2, P_4); (e_3, P_5); (e_4, P_2); (e_4, P_4); (e_5, P_1); (e_5, P_4); (e_6, P_2); (e_6, P_3); (e_7, P_5)\}$$

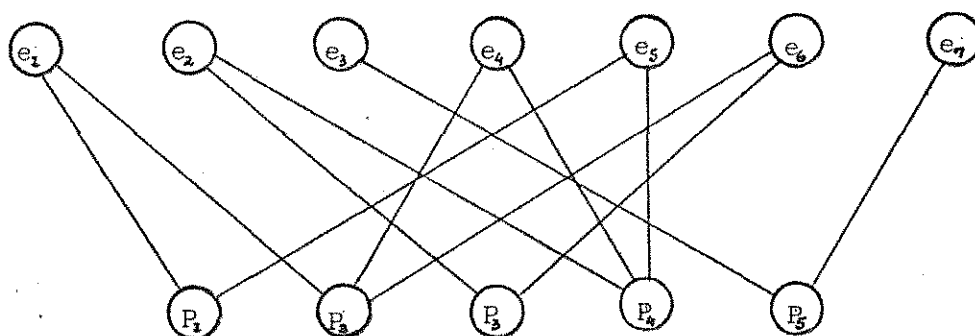


FIGURA IV
(Grafo bipartido associado ao PRC)

Associado a este problema de Representação de Conjuntos (PRC) está o seguinte problema de Cobertura com Conjuntos (PCC):

Considere o conjunto $I = \{1, 2, \dots, m\}$, onde m é o número de subconjuntos de F .

Para $j \in \mathcal{V}$ ($j=1, 2, \dots, n$), seja $I_j = \{i / i \in I \text{ e } j \in F_i\}$.

Como cada $F_i \neq \emptyset$, então $\bigcup_{j=1}^n I_j = I$. Seja $\theta = \{I_1, I_2, \dots, I_n\}$

Dessas relações, evidencia-se que um subconjunto

$\tilde{\theta}$ de θ é uma cobertura de cardinalidade mínima para I se e sómente se $R = \{j/I_j \in \tilde{\theta}\}$ é um representante de cardinalidade mínima para F .

No exemplo citado $I = P = \{P_1, P_2, \dots, P_5\}$

$Ie_1 = \{P_1, P_2\}; Ie_2 = \{P_3, P_4\}; Ie_3 = \{P_5\}; Ie_4 = \{P_2, P_4\}$

$Ie_5 = \{P_1, P_4\}; Ie_6 = \{P_2, P_3\}; Ie_7 = \{P_5\}$

e $\theta = \{Ie_1, Ie_2, \dots, Ie_7\}$

Redefinindo os nós de V_1 por $Ie_1, Ie_2, \dots, Ie_7$ e mantendo os mesmos nós para V_2 , obtém-se o mesmo grafo bipartido apresentado na figura IV e que corresponde ao (PCC).

1.3 - Reduções no problema de Partição (PP) e no problema de Cobertura com Conjuntos (PCC)

Algumas reduções no (PP) e no (PCC) são possíveis por observações na estrutura da matriz M , [Sa1] e [GN2]. Entre esses resultados, destacam-se os seguintes:

1.3.1 - se qualquer linha M_i de M for nula, tanto o (PP) como o (PCC) não apresentam solução viável, pois a i -ésima restrição não está satisfeita.

1.3.2 - se em uma linha M_i de M existe somente um valor 1 e, se este ocorre na coluna M^k então $x_k = 1$ para qualquer solução $x = \{0, 1\}^n$ e portanto, a i -ésima restrição e as demais que são satisfeitas por $x_k = 1$ podem ser apagadas. Complementando esta observação para o caso do (PP), toda coluna $M^r \neq M^k$ tal que $m_{tr} = m_{tk} = 1$ para algum $t \neq i$ deve ser apagada, pois $x_k = 1$ e fazendo $x_r = 1$ para alguma coluna resulta

$\sum_{j=1}^n m_{tj} \cdot x_j \geq 2 \neq 1$. Isto quer dizer que as variáveis diferentes de x_k com coeficientes diferentes de zero, em qualquer restrição onde x_k também tem coeficiente diferente de zero, de

vem assumir valor zero.

1.3.3 - (Linha Dominante): suponha M_s e M_r , duas linhas da matriz M tais que $M_r \geq M_s$ (isto é, se x_k tem coeficiente diferente de zero na s -ésima restrição, também o terá na r -ésima restrição). Assim, uma variável que satisfaz a s -ésima restrição, satisfaz a r -ésima restrição que pode, então, ser a apagada. Diz-se que M_r é dominada pela linha M_s . Considerando o caso do (PP), ao apagar M_r , toda coluna M^k tal que $m_{rk} = 1$ e $m_{sk} = 0$ deve ser apagada. Isto se deve ao fato que, se alguma coluna M^p é tal que $m_{sp} = m_{rp} = 1$ para poder satisfazer M_s e, se $x_k = 1$, tem-se:

$$\sum_{j=1}^n m_{rj} \cdot x_j \geq 2 + 1.$$

1.3.4 - (Coluna Dominante): suponha que para alguma coluna M^j de M no (PCC) exista um subconjunto S de outras colunas de M cuja soma é maior ou igual à M^j , isto é, $\sum_{k \in S} M^k \geq M^j$ e, suponha que o custo de x_j seja menor ou igual que a soma dos custos das variáveis correspondentes às colunas de S , isto é, $\sum_{k \in S} c_k \leq c_j$. Então, M^j pode ser apagada. Isto porque as restrições que podem ser satisfeitas por $x_j = 1$, também se satisfazem fazendo $x_t = 1$ para $t \in S$ e com custo menor.

Complementando essa situação para o (PP), se $\sum_{k \in S} M^k = M^j$ e se $\sum_{k \in S} c_k \leq c_j$, então a coluna M^j pode ser apagada.

1.4 - Algumas Aplicações

Além das aplicações introdutórias, problema de desconexão de rede elétrica (caso do PRC) e do problema de co

leta de lixo (caso do PCC), outras aparecem utilizando a estrutura do (PRC) e dos demais problemas formulados.

1.4.1 - Problema de Entrega de Mercadorias

Para Etcheberry [Et1] ,este problema aparece como um (PRC) e tem a seguinte descrição: um armazém deve fazer entrega de mercadorias, através de um carregador, para m clientes em destinos diferentes de uma região. O carregador pode combinar um número de clientes, mantendo-os juntos, para formar uma rota.

Um cliente pode receber mercadoria, via um número diferente de rotas. Uma listagem com as várias rotas viáveis é possível ser elaborada. Suponha esta lista contendo n rotas ($1, 2, 3, \dots, n$). Se c_j é o custo de operação da j -ésima rota, o problema (PRC) busca encontrar um subconjunto de rotas de custo mínimo tal que cada cliente esteja em, pelo menos, uma dessas rotas.

$$A \text{ matriz } M = (m_{ij}) = \begin{cases} 1, \text{ se a } i\text{-ésima rota contém o } j\text{-ésimo cliente} \\ 0, \text{ caso contrário} \end{cases}$$

O mesmo problema é estudado por Balinski e Quandt [BQ1] . Nesse estudo, o que se procura é um conjunto de atividades que satisfaça todos os clientes e minimize o custo total. Este custo total, por sua vez, é uma função do custo dos pedidos e de um custo fixo. O problema é então, apresentado como um (PP) .

1.4.2 - Localização de Hidrantes de Bombeiro [Et1]

Dada uma rede de ruas (arcos) e cruzamentos (nós)

o problema consiste em encontrar um subconjunto de nós para localização de hidrantes de maneira que cada rua contenha, pelo menos, um hidrante.

A minimização do número de elementos desse subconjunto é um $(PRCM)$.

Aquí, $m_{ij} = \begin{cases} 1, & \text{se o cruzamento (nó) } j \text{ está na rua (arco) } i \\ 0, & \text{caso contrário} \end{cases}$

1.4.3 - Recuperação de Informações [Et1]

Dados n arquivos, onde o comprimento do j -ésimo arquivo é c_j , suponha que m pedidos de informação sejam recebidos. Em geral, cada unidade de informação será armazenada em mais de um arquivo. O (PRC) produz um subconjunto de arquivos a serem pesquisados, em ordem, para recuperar todas as informações pedidas e de maneira que esta busca minimize o comprimento total dos arquivos utilizados.

Nesse caso, $m_{ij} = \begin{cases} 1, & \text{se a unidade de informação } i \text{ está no arquivo } j \\ 0, & \text{caso contrário} \end{cases},$

para $i=1,2,\dots,m$ e $j=1,2,3,\dots,n$

1.4.4 - Programação de Tripulação em Linhas Aéreas

Em [MS1], Marsten e Shepardson mostram que a mais frequente aplicação discutida para o (PP) tem sido no contexto de tripulações de linhas aéreas. A interpretação padrão do (PP) para este problema é a seguinte:

As linhas M_i representam segmentos de vôos (exemplo, vôo 740, São Paulo-Recife) que devem ser realizados ($i=1, 2, 3, \dots, n$). As colunas M^j representam possíveis combinações de

vôos que cada tripulação pode voar (feito por um programa de computação) . Seja c_j o custo associado a cada combinação j , então $m_{ij} = \begin{cases} 1, & \text{se o segmento de vôo } i \text{ está na combinação } j \\ 0, & \text{caso contrário} \end{cases}$

Existe uma variável de decisão, binária, x_j para cada combinação j e as equações do (PP) afirmam que cada segmento de vôo deve ser coberto por, exatamente, uma combinação selecionada ($x_j = 1$)

O problema de otimização é, então, selecionar um conjunto de combinações de vôos, de custo mínimo, que satisfaça as equações do (PP) . Ainda com relação a esse problema, Marsten e Shepardson [MS1.] dizem da incerteza de sua utilização e mostram que, para pequenas companhias aéreas ou para combinações de vôos que durem um tempo pequeno, a aplicação é satisfatória. Em [Et1] , uma alteração nas restrições coloca o problema como um (PRC) .

1.4.5 - Localização de Serviços Emergenciais em uma Comunidade [TS1]

Este problema consiste em construir, em uma comunidade, um número mínimo de serviços emergenciais (corpo de bombeiros, hospitais, escolas, centros de saúde, etc...) de maneira que todo morador da comunidade esteja, no máximo, à T minutos de cada um desses serviços. O (PRC) é aplicado interpretando V como um conjunto de locais com possibilidade de instalação dos serviços e F_i é o conjunto de locais acessíveis a partir do local i .

Em [Et1] , outras aplicações do (PRC) e problemas relacionados são apresentadas como: Balanceamento de li-

nha de montagem, designação de congressistas para comissões ,
programação de enfermeiras para um hospital, etc...

CAPÍTULO 2

ASPECTOS TEÓRICOS DO ALGORITMO UTILIZADO

Este capítulo apresenta alguns resultados teóricos que servem de base para o algoritmo a ser desenvolvido no presente trabalho. Esses resultados fundamentam o algoritmo desenvolvido por Murty [Mu1] para um $PRCM$.

Lawler [La1] sugere que o procedimento usado por Murty é útil apenas para problemas onde $|V|$ é grande, se comparada com $|F|$. Estas observações foram analisadas, sob o ponto de vista computacional, por Etcheberry [Et1] que introduz um método de enumeração implícita ou Partição-Limitação, para resolver um PRC genérico.

Os conceitos e observações apresentados a seguir são necessários para a discussão do algoritmo implementado.

2.1 - Resultados Teóricos

2.1.1 - Algoritmo para $PRCM$ [Mu1]

Definição:

$-S \subset V$ é um representante minimal para F , se:

- a) S representa F ,
- b) nenhum subconjunto próprio de S é representante para F .

Seja RM o conjunto de todos os representantes minimais de F e RCM o representante de cardinalidade mínima, para F . Observe que $RCM \subset RM$.

O algoritmo utiliza apenas operações sobre conjuntos (diferença, interseção, união) e é indutivo sobre $m = |F|$.

Seja $RM(k)$ o conjunto de todos os RM do $PRCM$ com $\mathcal{V} = \{1, 2, \dots, n\}$ e $F = \{F_1, F_2, \dots, F_k\}$. Inicialmente determina-se $RM(1)$ e a seguir $RM(2), RM(3), \dots, RM(m)$ sucessivamente, sempre obtendo $RM(k)$ a partir de $RM(k-1)$ considerando a inclusão da restrição k na formação do novo $PRCM$.

Finalmente, procura-se as soluções de menor cardinalidade em $RM(m)$.

As proposições que seguem provam que o processo indutivo utilizado é válido [Et1].

PROPOSIÇÃO 1: Se $S \in RM(k)$ e $S \cap F_{k+1} \neq \emptyset$ então $S \in RM(k+1)$

Prova:

Por hipótese, $S \in RM(k)$ $\rightarrow$ S é um representante minimal para $F = \{F_1, F_2, \dots, F_k\}$.

Por definição de $RM(k)$, nenhum elemento pode ser retirado do conjunto S sem violar uma ou mais entre as k primeiras restrições do problema.

Como, por hipótese, $S \cap F_{k+1} \neq \emptyset$, a afirmação continua válida quando a $(k+1)$ -ésima restrição é adicionada ao problema, ou seja $S \in RM(k+1)$.

PROPOSIÇÃO 2: Se $S \in RM(k)$ e $S \cap F_{k+1} = \emptyset$, então

(i) $S \notin R(k+1)$

(ii) $S \cup \{j\} \in R(k+1)$; para todo $j \in F_{k+1}$

Prova:

(i) claro, pois $S \cap F_{k+1} = \emptyset$, por hipótese.

(ii) como j representa F_{k+1} , para todo $j \in F_{k+1}$

e $S \in RM(k)$ mas $S \cap F_{k+1} = \emptyset$, então $S \cup \{j\} \in R_{k+1}$ para todo

$j \in F_{k+1}$.

Observa-se que $S \cup \{j\}$; para todo $j \in F_{k+1}$ não é

necessariamente um $RM(k+1)$, pois a inclusão do elemento j pertencente a F_{k+1} pode tornar algum dos elementos de S em Redun

dante, como no seguinte exemplo:

$\mathcal{U} = \{1, 2, 3, 4, 5, 6\}$ e $F = \{F_1, F_2\}$ onde $F_1 = \{1, 2, 3\}$ e $F_2 = \{3, 4, 5, 6\}$

Se $k = 1$, claramente $S = \{2\} \in RM(1)$

Agora, $S \cup \{3\} = \{2, 3\} \in R(2)$, mas não pertence à $RM(2)$, pois 2 torna-se Redundante, podendo ser apagado e $\{3\} \in RM(2)$.

PROPOSIÇÃO 3: Se $S \in R(k)$ e $S \cap F_{k+1} \neq \emptyset$, então $S \in R(k+1)$

Prova: análoga à da proposição 1

No contexto dessas proposições enunciadas, observa-se que S poderia, também, estar em $RM(k+1)$ se:

- (i) $S \in RM(k)$ (proposição 1)
- (ii) $|S \cap F_{k+1}| = 1$ e $[S - (S \cap F_{k+1})] \in RM(k)$

PROPOSIÇÃO 4: Se $S \in R(k)$ e $S \cap F_{k+1} = \emptyset$, então

- (i) $S \notin R(k+1)$
- (ii) $S \cup \{j\} \in R(k+1)$; para todo $j \in F_{k+1}$

Prova: Análoga à da proposição 2

Observa-se, novamente, que S poderia pertencer a $RM(k+1)$ se e somente se $S \in RM(k)$ e se não existe um elemento $r \in S$, tal que $\{j\} \cup (S - \{r\}) \in RM(k)$, ou seja, a inclusão de j não deixa qualquer elemento de S redundante.

Define-se, a seguir, o *produto cartesiano não ordenado* (p.c.n.o.) que representa de forma compacta os RM 's gera

dos nas etapas do procedimento. Esta forma torna o método para encontrar todos os representantes (método de Hakimi, segundo [Mul] ou de Petrick, segundo [Lal]) mais eficiente.

Definições:

-Sejam $D_1, D_2, \dots, D_r$ uma família de subconjuntos, não-vazios, mutuamente disjuntos de $\mathcal{V}$, define-se uma outra família de subconjuntos de $\mathcal{V}$, chamada produto cartesiano não ordenado (p.c.n.o.), por:

$$D = D_1 \times D_2 \times D_3 \times D_4 \times \dots \times D_r = \{ \{j_1, j_2, \dots, j_r\} / j_t \in D_t, t=1, \dots, r \}$$

-Os conjuntos $D_1, D_2, D_3, \dots, D_r$ são chamados Fatores do (p.c.n.o.)

Observações:

(1) O termo, não ordenado, surge do fato que o mesmo conjunto D é obtido trocando a ordem dos fatores.

(2) Em D , qualquer conjunto tem cardinalidade r , pois exatamente um elemento de cada fator D_t ; $t=1, 2, 3, \dots, r$ é considerado para a obtenção de um elemento de D . Assim, se algum $D_t = \emptyset$, então $D = \emptyset$.

(3) Em geral, $|D| = |D_1| \cdot |D_2| \cdot |D_3| \cdot \dots \cdot |D_r|$

O (p.c.n.o.) é utilizado para armazenar conjuntos de representantes e a proposição que segue mostra a construção desses conjuntos.

PROPOSIÇÃO 5: Seja $D = D_1 \times D_2 \times \dots \times D_r \in R(k)$. Os seguintes conjuntos são definidos sempre que seus fatores forem não-vazios:

$$D^1 = (D_1 \cap F_{k+1}) \times D_2 \times D_3 \times \dots \times D_r$$

$$D^2 = D_1^c \times (D_2 \cap F_{k+1}) \times D_3 \times \dots \times D_r$$

$$D^3 = D_1^c \times D_2^c \times (D_3 \cap F_{k+1}) \times D_4 \times \dots \times D_r$$

$$D^r = D_1^c \times D_2^c \times D_3^c \times \dots \times D_{r-1}^c \times (D_r \cap F_{k+1})$$

$$D^{r+1} = D_1^c \times D_2^c \times D_3^c \times \dots \times D_r^c \times (F_{k+1} - \bigcup_{t=1}^r (D_t \cap F_{k+1}));$$

onde $D_t^c = D_t - F_{k+1}$, $t=1, 2, 3, \dots, r$ e, então

os subconjuntos assim definidos pertencem à $R(k+1)$.

Prova:

Para todo $l=1, 2, 3, \dots, r$; tem-se que $D^l \in D \in R(k)$ e, como existe um elemento de $\mathcal{V}$, para cada representante S em D^l , que também está contido em F_{k+1} então, pela proposição 3, $D^l \in R(k+1)$; $l=1, 2, 3, \dots, r$

Se D^{r+1} está definido então, pela proposição 4 (ii), $D^{r+1} \in R(k+1)$.

Portanto $D^l \in R(k+1)$ para todo $l=1, 2, 3, \dots, r+1$

Por construção, observa-se que $D^{l_1} \cap D^{l_2} = \emptyset$, para quaisquer l_1 e $l_2 \in \{1, 2, \dots, r+1\}$ e $l_1 \neq l_2$, sempre que sejam definidos os conjuntos D^{l_1} e D^{l_2} .

Com esta forma compacta de representação, a apli-

cação das proposições, enunciadas anteriormente, propicia a busca dos representantes minimais. Esta busca é enunciada na proposição a seguir:

PROPOSIÇÃO 6: Com a notação usada na proposição 5, se $D \subseteq RM(k)$, então

- (i) $D^1 \subseteq RM(k+1)$, para todo $l=1, 2, \dots, r$
- (ii) $D^{r+1} \subseteq R(k+1)$, (alguns R 's de D^{r+1} poderão estar em $RM(k+1)$).

Prova: aplicação direta das proposições 1 e 2 aos conjunto D^l , $l=1, 2, \dots, r+1$.

Observa-se que (1) $|D^1| = r$

$$(2) |D^{r+1}| = r+1$$

As proposições, a seguir, sugerem possibilidades de cortes no conjunto solução do $PRCM$:

PROPOSIÇÃO 7: Para $S \in R(k)$, $S \notin RM(k) \leftrightarrow$ existe $u \in S$, tal que

$$I_k(u) \subseteq \bigcup_{j \in S - \{u\}} I_k(j),$$

onde $I_k(j) = I(j) \cap \{1, 2, \dots, k\}$; para todo $j \in S$ e

$I(j) = \{i/j \in F_i; i=1, 2, \dots, k\}$ (índices das restrições das restrições que contém j).

Prova:

(+) como, por hipótese, $S \in R(k)$ mas $S \notin RM(k) \rightarrow$ existe um subconjunto de S que representa $F = \{F_1, F_2, \dots, F_k\}$. Seja $S - \{u\}$

para $u \in S$, este representante. Como $u \in S$, $S \in R(k)$ e $S - \{u\} \in R(k)$

$$\text{então } I_k(u) \subseteq \bigcup_{j \in S - \{u\}} I_k(j)$$

$$(+)\text{ como } I_k(u) \subseteq \bigcup_{j \in S - \{u\}} I_k(j), \text{ por hipótese, e}$$

$S \in R(k) \rightarrow S - \{u\}$ pertence à $R(k)$ e então $S \notin RM(k)$, por definição de $RM(k)$.

PROPOSIÇÃO 8: Seja $V = \{S_i / S_i \in RM(k), S_i \cap F_{k+1} \neq \emptyset\}$

Seja $W = \{S_i \cup \{j\} / S_i \in RM(k); S_i \notin RM(k+1); j \in F_{k+1} \text{ e}$

$S_i \cup \{j\} \in RM(k+1)\}$, então V e W é uma partição de $RM(k+1)$

Prova:

Pela proposição 1, se $S \in V \rightarrow S \in RM(k+1)$

Também, se $S \in W$, pela definição de $W \rightarrow S \in RM(k+1)$.

Das definições de V e de W , tem-se que $V \cap W = \emptyset$. Para verificação que V e W formam uma partição de $RM(k+1)$, tem-se que provar que para qualquer $S \in RM(k+1) \rightarrow S \in V$ ou $S \in W$.

Seja $S \in RM(k+1) \rightarrow S$ pode estar em $Rm(k)$ ou em $[R(k) - RM(k)]$

Se $S \in RM(k) \rightarrow S \in V$

Se $S \in [R(k) - RM(k)] \rightarrow$ existe um subconjunto S' de S tal que $S' \in RM(k)$.

Como $S \in RM(k+1) \rightarrow$ não existe um subconjunto S' de S tal que $S' \in RM(k+1)$.

Nestas condições, pela proposição 3, S pode ser obtido de S' adicionando, no máximo, um elemento $j \in F_{k+1}$, ou seja,

$$|S \cap F_{k+1}| = 1 \text{ e } [S - (S \cap F_{k+1})] = S' \text{ pertencente à } RM(k) \text{ e}$$

$S' \notin RM(k+1) \rightarrow S \in W$. Isto prova que V e W formam uma partição de $RM(k+1)$.

PROPOSIÇÃO 9: Qualquer representante S , tal que

$$S \in \bar{W} = \{ S_i \cup \{j\} / S_i \in RM(k), S_i \notin RM(k+1), j \in F_{k+1} \text{ e } S_i \cup \{j\} \notin RM(k+1) \}$$

conterá um subconjunto próprio de menor cardinalidade que está em V .

Prova:

Para qualquer $S \in \bar{W}$, existe $S' \subset S$ tal que $S' \in RM(k+1)$ + pela prova da proposição 8 + $S' \in V$.

PROPOSIÇÃO 10: Se $S \in RM(k)$, $S \notin RM(k+1)$ e se $S \cup \{j\} \in [R(k+1) - RM(k+1)]$, então existe um ou mais subconjuntos de $S \cup \{j\}$ que estão em $RM(k+1)$.

Prova:

Como $S \in RM(k)$, pela proposição 7, para todo $u \in S$, existe uma ou mais das k primeiras linhas da matriz M que são representadas exclusivamente por u .

Seja $\bar{I}_k(u) = \{I_k(u) - [\bigcup_{t \in S - \{u\}} I_k(t)]\}$, onde $I_k(u) = I(u) \cap \{1,$

$2, 3, \dots, k\}$ para todo $u \in S$.

Como $S \notin RM(k+1)$ e $S \cup \{j\} \in R(k+1)$, por hipótese + j não é redundante.

Os elementos $u / \bar{I}_k(u) \subseteq I_k(j)$ podem ser apagados de S pois são redundantes ($\bar{I}_k(u)$ é atualizado cada vez que um elemento de S é apagado). Assim obtém-se $S' \subset S$, em $R(k+1)$, que em geral não é único pois a ordem de eliminação dos elementos redundantes é importante.

As proposições enunciadas a seguir dizem respeito às restrições do problema e suas demonstrações [Etl.] não se fa

zêm necessárias no presente trabalho.

PROPOSIÇÃO 11: Se $D = D_1 \times D_2 \times D_3 \times \dots \times D_r \subseteq RM(k)$, $D \neq \emptyset$ e os subconjuntos D_t são subconjuntos disjuntos de $\mathcal{V}$, para $t=1,2,\dots,r$; então:

$$J_1 = \bigcap_{j \in D} I_k(j) \neq \emptyset$$

$$J_t = \left\{ \bigcap_{j \in D_t} I_k(j) - \bigcup_{s=1}^{t-1} J_s \right\} \neq \emptyset, \text{ para } t=2,3,\dots,r$$

PROPOSIÇÃO 12: $J_1, J_2, \dots, J_r$ é uma partição de $\{1,2,3,\dots,k\}$

Com esta base teórica, Murty [Mul] apresenta o algoritmo cujo esboço é o seguinte:

Passo 1: Faça $k=1$ e $RM(1) = \{F_1\}$

Passo 2: Obtenha $RM(k+1)$ a partir de $RM(k)$, utilizando a proposição 6 (os representantes não minimais de D^{r+1} serão eliminados, diretamente, pela proposição 7 ou por investigação, sobre o restante dos $RM(k+1)$, pelo uso das proposições 8 e 9)

Passo 3: Se $k = m$, PARE

Passo 4: Faça $k = k+1$

Passo 5: Volte ao passo 2.

Este algoritmo é denominado Algoritmo Indutivo Cartesiano não Ordenado (a.i.c.n.o.), [Et1] e [Mul]

2.1.2 - O Método de Partição-Limitação

Essencialmente, o método de Partição-Limitação é um esquema enumerativo para resolver problemas de otimização [Mil]. Esse método garante a obtenção de uma solução ótima, caso exista, para um problema de otimização discreta.

Sua utilidade vem do fato de que, em geral, somente uma parte das soluções possíveis necessitam ser enumeradas. As demais são eliminadas pela aplicação de limitantes que estabelecem a impossibilidade de tais soluções serem ótimas. Em geral, é mantida uma lista de subproblemas a resolver que, inicialmente, contém apenas o problema original. Para cada subproblema (problema candidato (PC)) da lista são aplicadas as rotinas abaixo:

Limitação- consiste no estabelecimento de um valor limitante da função objetivo. Caso este valor seja maior (para problema de minimização) do que o valor da solução viável já conhecida, o (PC) é descartado da lista. Caso contrário, executa-se a rotina de partição para este (PC).

Partição- consiste na partição do conjunto de soluções viáveis em subconjuntos gerando novos problemas candidatos que serão incluídos na lista.

Para a obtenção do limitante, é comum a relaxação do problema candidato, isto é, algumas restrições são relaxadas podendo ou não serem incorporadas na função objetivo.

2.1.3 - Relaxação Lagrangeana

Definição:

-Diz-se que um problema de minimização P_n é uma relaxação do problema de minimização P se:

(a) $s(P_n) \supseteq s(P)$, onde $s(.)$ é o conjunto de soluções viáveis do problema $(.)$.

(b) $v(P_n) \geq v(P)$, onde $v(.)$ é o valor ótimo do problema $(.)$

Considere o programa linear inteiro referente ao PRC , cuja formulação matricial (1.1.1) é:

$$\begin{aligned} \min z(x) &= c^T \cdot x \\ \text{sujeito a } M \cdot x &\geq \mathbf{1} \\ x &\in \{0,1\}^n \end{aligned}$$

Suponha que a matriz $M(m \times n)$ seja particionada, segundo suas linhas M_i , em duas matrizes $M^0(m_1 \times n)$ e $\bar{M}(m_2 \times n)$, onde $m_1 + m_2 = m$, e que o vetor coluna $\mathbf{1}(m \times 1)$ seja igualmente particionado em $\mathbf{1}^0(m_1 \times 1)$ e $\bar{\mathbf{1}}(m_2 \times 1)$, respectivamente. Com estas operações, o PRC acima pode ser reescrito como:

$$\begin{aligned} [10] \quad \min z(x) &= c^T \cdot x \\ \text{sujeito a } M^0 \cdot x &\geq \mathbf{1}^0 \\ \bar{M} \cdot x &\geq \bar{\mathbf{1}} \\ x &\in \{0,1\}^n \end{aligned}$$

Se as restrições $M^0 \cdot x \geq \mathbf{1}^0$ possuem uma estrutura especial, define-se a relaxação Lagrangeana de [10] relativa às restrições $\bar{M} \cdot x \geq \bar{\mathbf{1}}$ e à um vetor não negativo $\lambda \in R^{m_2}$, como:

$$\begin{aligned} [11] \quad w(\lambda) &= \min [c^T \cdot x + \lambda^T \cdot (\bar{\mathbf{1}} - \bar{M} \cdot x)] \\ \text{sujeito a } M^0 \cdot x &\geq \mathbf{1}^0 \\ x &\in \{0,1\}^n \end{aligned}$$

ou,

$$\begin{aligned} w(\lambda) &= \min L(x, \lambda) \\ \text{sujeito a } M^0 \cdot x &\geq \mathbf{1}^0 \\ x &\in \{0,1\}^n, \end{aligned}$$

para $L(x, \lambda) = c^T \cdot x + \lambda^T \cdot (\bar{\mathbf{1}} - \bar{M} \cdot x)$

Na prática, dado $\lambda \geq 0$, $w(\lambda)$ não deve ser difícil obtenção.

Definições:

-O vetor λ é chamado vetor de multiplicadores de Lagrange, onde de cada coordenada $\lambda_i \geq 0$ é o multiplicador de Lagrange associado à i -ésima restrição de $\bar{M}.x \geq \bar{1}$, [Mi2].

-A função $L(x, \lambda)$ é chamada função de Lagrange associada ao problema [10].

Técnicas Lagrangeanas foram propostas para problemas de otimização discreta por volta de 1955 quando Loire e Savage sugeriram um método simples para resolver problemas de programação inteira 0-1, [Sh1]. A meta do uso dessas técnicas é estabelecer condições de otimalidade que permitam responder quando uma solução $x \in \{0,1\}^n$, ótima na relaxação Lagrangeana [11], será ótima no problema não relaxado.

Reescrevendo a função de Lagrange como:

$L(x, \lambda) = \lambda^T \cdot \bar{1} + (c^T - \lambda^T \cdot \bar{M}) \cdot x$, o problema [11] fica:

$$\begin{aligned}
 [12] \quad w(\lambda) = \min \quad & \lambda^T \cdot \bar{1} + (c^T - \lambda^T \cdot \bar{M}) \cdot x \\
 \text{sujeito a} \quad & \bar{M}^0 \cdot x \geq \bar{1}^0 \\
 & x \in \{0,1\}^n
 \end{aligned}$$

e, as seguintes condições de otimalidade podem ser estabelecidas: o conjunto $\{x^*, \lambda^*\}$, onde $x^* \in \{0,1\}^n$; $\bar{M}^0 \cdot x^* \geq \bar{1}^0$ e $\lambda^* \geq 0$, satisfaz as condições suficientes de otimalidade para o programa linear inteiro [10] se:

- (i) $w(\lambda) = \lambda^{*T} \cdot \bar{1} + (c^T - \lambda^{*T} \cdot \bar{M}) \cdot x^*$
- (ii) $\lambda^{*T} \cdot (\bar{1} - \bar{M} \cdot x^*) = 0$
- (iii) $\bar{M} \cdot x^* \geq \bar{1}$

Mostra-se que se a solução x^* satisfaz (i), (ii) e (iii), para algum λ^* , então x^* é solução ótima para [10]:

PROPOSIÇÃO 13: Se $\{x^*, \lambda^*\}$ satisfaz as condições de otimalidade para [10], então x^* é solução ótima para [10]

Prova:

Como $x^* \in \{0,1\}^n$ e por (iii) $\bar{M}.x^* \geq \bar{1}$, então x^* é solução viável de [10]. Seja x uma outra solução viável de [10]. Então, por (i), tem-se:

$$\begin{aligned} w(\lambda^*) &= \lambda^{*T} \bar{1} + (c^T - \lambda^{*T} \bar{M}).x^* \leq \lambda^{*T} \bar{1} + (c^T - \lambda^{*T} \bar{M}).x = \\ &= c^T.x + \lambda^{*T} .(\bar{1} - \bar{M}.x) \leq c^T.x \text{ (pois } \lambda^* \geq 0 \text{ e} \end{aligned}$$

$$(\bar{1} - \bar{M}.x) \leq 0 \text{ por (iii) e, então } [\lambda^{*T}(\bar{1} - \bar{M}.x)] \leq 0.$$

Portanto, $w(\lambda^*) \leq c^T.x$, para qualquer $x \in \{0,1\}^n$, viável.

Pela condição (ii) $w(\lambda^*) = c^T.x^*$.

Assim $c^T.x^* \leq c^T.x$ para todo x viável, ou seja, x^* é uma solução ótima de [10].

COROLÁRIO 1:(Dualidade Fraca) : Para todo $\lambda \geq 0$; $w(\lambda) \leq c^T.x^*$, onde x^* é solução ótima de [10].

Prova:

Imediata a partir da proposição anterior, pois $w(\lambda) \leq c^T.x$, para todo x viável de [10] e, em particular, para x^* (ótima), tem-se que $w(\lambda) \leq c^T.x^* = z(x^*)$.

O corolário acima mostra que o valor da função ob

jetivo da Lagrangeana [11] é um limitante inferior para o valor da função objetivo de [10]. Assim, a melhor escolha do vetor de multiplicadores de Lagrange $\lambda \geq 0$ será aquela que produza o maior limitante inferior, ou equivalentemente, qualquer λ que seja ótimo do problema dual, [Mi2] e [Sh1] :

$$\begin{aligned} & \text{Máx. } w(\lambda) \\ [13] \quad & \text{sujeito a } \lambda \geq 0 \end{aligned}$$

Usando as condições de otimalidade e o corolário 1, uma outra consequência da proposição 13 é a seguinte:

COROLÁRIO 2: Se $\{x^*, \lambda^*\}$ satisfaz as condições de otimalidade para [10], então λ^* é a solução ótima para o problema dual [13]

Prova:

Como $w(\lambda^*) = c^T \cdot x^*$, pela proposição 13 e $w(\lambda) \leq c^T \cdot x^* = w(\lambda^*)$, para todo $\lambda \geq 0$, pelo corolário 1, então:

$w(\lambda) \leq c^T \cdot x^* = w(\lambda^*)$, para todo $\lambda \geq 0$.

Portanto $w(\lambda) \leq w(\lambda^*)$, ou seja, λ^* é uma solução ótima para [13].

A estratégia indicada para a aplicação das técnicas Lagrangeanas é a seguinte:

Primeiro, encontrar uma solução ótima λ^* para [13].

Feito isso, encontrar $x^* \in \{0,1\}^n$ tal que as condições de otimalidade sejam válidas, calculando uma ou mais soluções $x \in \{0,1\}^n$, satisfazendo $w(\lambda^*) = \lambda^{*T} \cdot \bar{1} + (c^T - \lambda^{*T} \cdot \bar{M})x$.

Esta estratégia não fornece garantia de sucesso, pois pode ocorrer as seguintes possibilidades:

(a) pode não existir λ^* (ótimo), para [13], que dê condições de otimalidade para algum $x^* \in \{0,1\}^n$.

(b) o ótimo λ^* , para [13], não permite que as condições de otimalidade sejam satisfeitas para qualquer $x \in \{0,1\}^n$.

(c) o(s) específico(s) x^* (ou x^* 's) selecionado(s) pela minimização da Lagrangeana não satisfaz (ou satisfazem) as condições de otimalidade, embora algum x^* , que seja mínimo para a Lagrangeana, satisfaça.

O problema dual [13] será, então, tratado como um problema de otimização não diferenciável, isto é, [13] será resolvido com um algoritmo de busca de uma direção de subida, não diferenciável, usando o fato que a função $w(\lambda)$ é uma função côncava.

PROPOSIÇÃO 14: A função dual w é uma função côncava.

Prova:

Sejam λ^1 e λ^2 quaisquer e para $\theta \in [0,1]$ qualquer, define-se

$$\lambda = \theta \cdot \lambda^1 + (1 - \theta) \cdot \lambda^2$$

$w(\lambda) = \min [c^T \cdot x + \lambda^T \cdot (\bar{1} - \bar{M} \cdot x)]$. Então, existe um $\tilde{x}$ tal que

$w(\lambda) = c^T \cdot \tilde{x} + \lambda^T \cdot (\bar{1} - \bar{M} \cdot \tilde{x})$. Pela definição de $w(\lambda^1)$ e $w(\lambda^2)$,

$$\text{tem-se: } w(\lambda^1) \leq c^T \cdot \tilde{x} + \lambda^1{}^T \cdot (\bar{1} - \bar{M} \cdot \tilde{x}) \quad (\text{I})$$

$$w(\lambda^2) \leq c^T \cdot \tilde{x} + \lambda^2{}^T \cdot (\bar{1} - \bar{M} \cdot \tilde{x}) \quad (\text{II})$$

Multiplicando a desigualdade (I) por $\theta \geq 0$ e a desigualdade (II) por $(1 - \theta)$ e somando esses produtos, obtém-se:

$$\theta \cdot w(\lambda^1) + (1 - \theta) \cdot w(\lambda^2) \leq \theta \cdot [c^T \cdot \tilde{x} + \lambda^1{}^T \cdot (\bar{1} - \bar{M} \cdot \tilde{x})] + (1 - \theta) \cdot$$

$$\begin{aligned}
& \cdot [c^T \cdot \tilde{x} + \lambda^2 T \cdot (\bar{1} - \bar{M} \cdot \tilde{x})] = \\
& = \theta \cdot c^T \cdot \tilde{x} + \theta \cdot \lambda^1 T \cdot (\bar{1} - \bar{M} \cdot \tilde{x}) + c^T \cdot \tilde{x} + \lambda^2 T \cdot (\bar{1} - \bar{M} \cdot \tilde{x}) - \theta \cdot c^T \cdot \tilde{x} - \\
& - \theta \cdot \lambda^2 T \cdot (\bar{1} - \bar{M} \cdot \tilde{x}) = \\
& = c^T \cdot \tilde{x} + \theta \cdot \lambda^1 T \cdot (\bar{1} - \bar{M} \cdot \tilde{x}) + (1 - \theta) \cdot \lambda^2 T \cdot (\bar{1} - \bar{M} \cdot \tilde{x}) = \\
& = c^T \cdot \tilde{x} + [\theta \cdot \lambda^1 T + (1 - \theta) \cdot \lambda^2 T] (\bar{1} - \bar{M} \cdot \tilde{x}) = \\
& = c^T \cdot \tilde{x} + \lambda T \cdot (\bar{1} - \bar{M} \cdot \tilde{x}) = w(\lambda) .
\end{aligned}$$

Portanto, $\theta \cdot w(\lambda^1) + (1 - \theta) \cdot w(\lambda^2) \leq w(\lambda) = w[\theta \cdot \lambda^1 + (1 - \theta) \lambda^2]$,

o que mostra que w é côncava.

O uso de técnicas Lagrangeanas em otimização discreta é, em geral, baseado em duas características que são: a natureza combinatória dos algoritmos utilizados para o problema [13] e a não diferenciabilidade de w devido à discretização do conjunto de soluções.

Essa última característica permite a construção de métodos de busca de uma direção de subida, usando uma generalização do gradiente.

2.1.4 - Otimização com Subgradiente

Subgradiente é uma generalização da noção de gradiente para ser usada no tratamento de problemas de otimização de funções não diferenciáveis [Sal] .

Definição:

-Define-se subgradiente, da função côncava w , em um ponto λ^0 , como sendo um vetor $\gamma \in R^{m_2}$ que satisfaz:

$$w(\lambda) - w(\lambda^0) \leq \gamma^T(\lambda^0) \cdot (\lambda - \lambda^0), \text{ para todo } \lambda \geq 0.$$

A interpretação geométrica do subgradiente é a seguinte:

$w(\lambda^0) + \gamma^T(\lambda^0) \cdot (\lambda - \lambda^0)$ é a equação de um hiperplano suporte para o gráfico da função w , em λ^0 . Assim, se w é diferenciável em λ^0 , o único hiperplano suporte para o gráfico de w , em λ^0 , é o hiperplano tangente e $\gamma(\lambda^0) = \nabla w(\lambda^0)$, que é o gradiente de w , em λ^0 . Nesse sentido, o subgradiente é uma generalização do gradiente.

Quando o interesse é otimização, métodos usando gradiente diferem do método com subgradiente, pois a convergência de tais métodos tem como base critérios diferentes, a saber: no caso gradiente, para uma aproximação monotônica do valor de λ^* , onde $w(\lambda^*)$ assume o valor máximo, há um crescimento monotônico da função w , enquanto que, para este mesmo tipo de aproximação, os métodos subgradientes não implicam na monotonicidade do crescimento da função objetivo w , podendo ocorrer oscilação próximo do valor ótimo.

Embora aplicável a uma grande classe de problemas, o método subgradiente tem sido usado, principalmente, para encontrar solução de problemas do tipo que aqui se deseja resolver:

$$\begin{aligned} & \text{Máx. } w(\lambda) \\ [14] \quad & \text{sujeito a } \lambda \geq 0 \end{aligned}$$

$$\text{onde } w(\lambda) = \min_{x \in \{0,1\}^n} [c^T \cdot x + \lambda^T (\bar{1} - \bar{M} \cdot x)]$$

e w é côncava (como visto em 2.1.3) e linear por partes sobre o conjunto $\{\lambda / \lambda \geq 0\}$ [FS1] e [HW1].

A solução de [14] fornece o melhor limitante que

pode ser encontrado para o problema [10], considerando que o conjunto de restrições $M^0 \cdot x \geq 1^0$ de [10] dá ao problema uma estrutura tal que encontrar a solução do problema relaxado [11] não seja tarefa difícil.

A proposição seguinte indica uma maneira de encontrar um subgradiente para w [Mi2].

PROPOSIÇÃO 15: Para $\lambda \geq 0$, $\lambda \in R^{m_2}$,

Seja $X(\lambda^0) = \{x \in \{0,1\}^n / w(\lambda^0) = L(x, \lambda^0)\}$, então

para todo $x \in X(\lambda^0)$, tem-se $(\bar{1} - \bar{M} \cdot x)$ é um subgradiente de w , em λ^0 .

Prova:

Por definição de w : $w(\lambda) \leq c^T \cdot x + \lambda^T (\bar{1} - \bar{M} \cdot x)$ (1), para todo $\lambda \geq 0$, $\lambda \in R^{m_2}$ e $x \in \{0,1\}^n$.

Particularmente, para $x \in X(\lambda^0)$, $w(\lambda^0) = c^T \cdot x + \lambda^0{}^T (\bar{1} - \bar{M} \cdot x)$ (2)

Sutraindo (2) de (1), obtém-se:

$$\begin{aligned} w(\lambda) - w(\lambda^0) &\leq c^T \cdot x + \lambda^T (\bar{1} - \bar{M} \cdot x) - c^T \cdot x - \lambda^0{}^T (\bar{1} - \bar{M} \cdot x) = \\ &= (\lambda^T - \lambda^0{}^T) \cdot (\bar{1} - \bar{M} \cdot x). \end{aligned}$$

Portanto $w(\lambda) - w(\lambda^0) \leq (\bar{1} - \bar{M} \cdot x)^T \cdot (\lambda - \lambda^0)$, ou seja $(\bar{1} - \bar{M} \cdot x)$ é um subgradiente de w , em λ^0 .

Notação: Denota-se por $\gamma(\lambda^0) = (\bar{1} - \bar{M} \cdot x)$, $x \in X(\lambda^0)$, ao vetor subgradiente de w , em λ^0 .

$$\begin{aligned} \text{Seja } \Gamma(\lambda) &= \{\gamma^i / c^T \cdot x + (\gamma^i)^T \cdot \lambda = w(\lambda)\} = \\ &= \min_x \{c^T \cdot x + (\gamma^i)^T \cdot \lambda\}. \end{aligned}$$

Isto é, $\Gamma(\lambda)$ é o conjunto de todos os vetores γ^i pertencentes à R^{m_2} tais que o mínimo de $L(x, \lambda)$ é encontrado

para o índice i . A derivada direcional de w , na direção de um vetor unitário d (qualquer) e que fornece a taxa de crescimento de w ao longo da direção d , é igual ao comprimento da menor componente de $\gamma^i \in \Gamma(\lambda)$, na direção d .

Portanto $w'(\lambda, d) = \min \{ (\gamma^i)^T \cdot d / \gamma^i \in \Gamma(\lambda) \}$, onde $w'(\lambda, d)$ denota a derivada direcional de w , na direção d .

Definições:

- $\Gamma(\lambda)$ é chamado conjunto dos "gradientes ativos", em λ . [Sa1]
- O conjunto dos subgradiantes de w , em λ , é chamado subdiferencial de w , em λ , e é denotado por $\partial w(\lambda)$.

Uma generalização da observação feita anteriormente pode ser feita e, como consequência, obtém-se:

$$w'(\lambda, d) = \min \{ (\gamma)^T \cdot d / \gamma \in \Gamma(\lambda) \},$$

$$\text{onde } \Gamma(\lambda) = \{ \gamma / \gamma = \sum_{i \in I(\lambda)} \theta_i \cdot \gamma^i ; \sum_{i \in I(\lambda)} \theta_i = 1 \text{ e } \theta_i \geq 0 \} \text{ e}$$

$$I(\lambda) = \{ i / w(\lambda) = c^T \cdot x + (\gamma^i)^T \cdot \lambda \}.$$

Sandi, [Sa1] mostra que o conjunto dos subgradiantes de w , em λ , é igual ao conjunto das combinações convexas dos "gradientes ativos" de w , em λ , isto é $\partial w(\lambda) = \Gamma(\lambda)$.

Como $w'(\lambda, d) = \min \{ \gamma^T \cdot d / \gamma \in \partial w(\lambda) \}$, para um dado λ e d (qualquer), uma condição necessária e suficiente para que o vetor λ^* seja tal que $w(\lambda^*)$ assuma seu valor máximo, é que $w'(\lambda^*, d) \leq 0$, ou equivalentemente que $0 \in \partial w(\lambda)$, [HW1].

Este resultado é obtido da proposição que segue:

PROPOSIÇÃO 16: Seja w uma função côncava, λ^* maximiza w , se e somente se $0 \in \partial w(\lambda^*)$.

Prova:

De fato $0 \in w(\lambda^*)$ se e somente se $w(\lambda) \leq w(\lambda^*) + 0^T(\lambda - \lambda^*)$, para todo $\lambda \in \mathbb{R}^{m^2}$.

Portanto $w(\lambda) \leq w(\lambda^*)$, ou seja, λ^* maximiza w .

Com esta base teórica, um método de busca de uma direção de subida é elaborado no sentido de obter o melhor limitante, expresso como o máximo de uma função descrita como o mínimo de um número finito de funções lineares, [HK1], para [10].

Este método é iterativo relativamente ao método de relaxação para a solução de sistemas de desigualdades lineares e, cada iteração não necessariamente melhora o valor da função w mas, ao invés, reduz a distância euclidiana ao ponto λ^* , máximo para w .

Em cada etapa é calculada uma sequência de vetores $\{\lambda^k\}$, de acordo com a seguinte recursão:

Para inicializar, toma-se um vetor λ^0 (arbitrário).

A seguir, calcula-se $\lambda^{k+1} = \lambda^k + t_k \cdot \gamma^k$, onde

$\gamma^k \in r(\lambda^k)$ e $\{t_k\}$ é uma sequência de números positivos.

PROPOSIÇÃO 17: Sejam λ^k e λ vetores de $\mathbb{R}^{m^2}$ tais que $w(\lambda) \geq w(\lambda^k)$, então:

$$(\gamma^k)^T \cdot (\lambda - \lambda^k) \geq w(\lambda) - w(\lambda^k) \geq 0$$

Prova:

Seja $\gamma^k \in r(\lambda^k)$ então $w(\lambda^k) = c^T \cdot x + (\gamma^k)^T \cdot \lambda^k$ (1)

$$c^T \cdot x + (\gamma^k)^T \cdot \lambda \geq \min_{\lambda \in I(\lambda^k)} [c^T \cdot x + (\gamma^k)^T \cdot \lambda] = w(\lambda) \quad (2)$$

Subtraindo (1) de (2), obtém-se:

$$\begin{aligned} w(\lambda) - w(\lambda^k) &\leq c^T \cdot x + (\gamma^k)^T \cdot \lambda - c^T \cdot x - (\gamma^k)^T \cdot \lambda^k = \\ &= (\gamma^k)^T \cdot (\lambda - \lambda^k) . \end{aligned}$$

$$\text{Portanto } (\gamma^k)^T \cdot (\lambda - \lambda^k) \geq w(\lambda) - w(\lambda^k) \geq 0$$

O hiperplano, $(\gamma^k)^T \cdot (\lambda - \lambda^k) = 0$, passando por λ^k e tendo γ^k como vetor normal, determina um semi-espaço fechado contendo todos os pontos λ tais que

$(\gamma^k)^T \cdot (\lambda - \lambda^k) \geq w(\lambda) - w(\lambda^k) \geq 0$, ou seja, $w(\lambda) \geq w(\lambda^k)$ e, em cada iteração λ move-se para dentro desse semi-espaço na direção da normal γ^k . Em particular, esse semi-espaço inclui algum ponto λ^* , máximo para w .

A escolha dos números $t_k > 0$ que fazem com que o ponto λ^{k+1} esteja o mais próximo possível de λ^* é, portanto, de importância fundamental. O valor ideal do passo t_k é uma solução ótima para o problema:

$$\text{minimizar } \|\lambda^* - (\lambda^k + t_k \cdot \gamma^k)\|^2 .$$

Também, é fácil verificar que esta solução ótima é dada por um t_k que verifique $\|\lambda^* - (\lambda^k + t_k \cdot \gamma^k)\|^2 = 0$ e, isto ocorre se

$$t_k^* = \frac{(\lambda^* - \lambda^k) \cdot (\gamma^k)^T}{\|\gamma^k\|^2} .$$

Pela proposição 17, tem-se que:

$(\gamma^k)^T \cdot (\lambda^* - \lambda^k) \geq w(\lambda^*) - w(\lambda^k)$, o que implica em:

$$t_k^* \geq \frac{w(\lambda^*) - w(\lambda^k)}{\|\gamma^k\|^2}$$

Uma aproximação razoável para t_k^* é usar como tamanho do passo $t_k = \frac{\tilde{w} - w(\lambda^k)}{\|Y^k\|^2}$, para $\tilde{w} \geq w(\lambda^*)$.

A utilização do sobreestimador $\tilde{w}$, para $w(\lambda^*)$, está intimamente ligada à análise de convergência do método proposto. Poljak, [Po1] mostra que as condições:

$$\lim_{k \rightarrow \infty} t_k = 0 \quad \text{e} \quad \sum_{k=0}^{\infty} t_k = \infty,$$

são suficientes para garantir que dado um n° arbitrário $\epsilon > 0$, existe k tal que $w(\lambda^*) - w(\lambda^k) < \epsilon$, onde

$$\lambda^* \in \bigwedge^* = \{\lambda^*/w(\lambda^*) \geq w(\lambda) \text{ para todo } \lambda \in R^{m_2}, \lambda \geq 0\} \neq \emptyset$$

Embora as condições acima sejam suficientes, nem todo tamanho de passo t_k que as satisfaça é desejável em virtude da lentidão com que se dá o processo de convergência. Para superar esta dificuldade, Poljak, [Po2], propõe a alternativa de selecionar, em cada etapa do método, um passo:

$$t_k = \delta_k \cdot \left[\frac{w(\lambda^*) - w(\lambda^k)}{\|Y^k\|^2} \right], \text{ para } \epsilon_1 < \delta_k < 2 - \epsilon_2;$$

$\epsilon_1 > 0$ e $\epsilon_2 > 0$, escolhidos de maneira adequada para que o intervalo $(\epsilon_1, 2 - \epsilon_2)$ seja não vazio.

A aplicação dessa alternativa, no entanto, torna-se difícil em muitos problemas pelo desconhecimento, à priori, do valor $w(\lambda^*)$.

Poljak em [Po2] sugere que $w(\lambda^*)$ seja substituído por um sobreestimador $\tilde{w} > w(\lambda^*)$ e prova que a sequência gerada $w(\lambda^k)$ é tal que $w(\lambda^k) \leq \tilde{w}$, para todo k ou $w(\lambda^k) > \tilde{w}$ para algum k e $\lim_{k \rightarrow \infty} w(\lambda^k) = \tilde{w}$, com taxa geométrica.

A proposição que segue indica a redução da distância euclídeana ao ponto $\tilde{\lambda}$ durante a iteração proposta:

PROPOSIÇÃO 18: Se $0 < t_k < 2 \cdot \frac{w(\tilde{\lambda}) - w(\lambda^k)}{\| \gamma^k \|^2}$, então

$$\| \tilde{\lambda} - (\lambda^k + t_k \cdot \gamma^k) \| \leq \| \tilde{\lambda} - \lambda^k \|$$

Prova:

Por hipótese: $t_k \cdot \| \gamma^k \|^2 < 2 \cdot [w(\tilde{\lambda}) - w(\lambda^k)]$

$$\begin{aligned} \| \tilde{\lambda} - (\lambda^k + t_k \cdot \gamma^k) \|^2 &= \| (\tilde{\lambda} - \lambda^k) - t_k \cdot \gamma^k \|^2 = \\ &= [(\tilde{\lambda} - \lambda^k) - t_k \cdot \gamma^k] \cdot [(\tilde{\lambda} - \lambda^k) - t_k \cdot \gamma^k]^T = \\ &= (\tilde{\lambda} - \lambda^k) \cdot (\tilde{\lambda} - \lambda^k)^T - t_k \cdot \gamma^k \cdot (\tilde{\lambda} - \lambda^k)^T - (\tilde{\lambda} - \lambda^k) \cdot t_k \cdot (\gamma^k)^T + \\ &+ t_k (t_k \cdot \gamma^k (\gamma^k)^T) = \| \tilde{\lambda} - \lambda^k \|^2 - 2 \cdot t_k (\gamma^k)^T \cdot (\tilde{\lambda} - \lambda^k) + \\ &+ t_k (t_k \| \gamma^k \|^2) = \| \tilde{\lambda} - \lambda^k \|^2 + t_k [t_k \| \gamma^k \|^2 - 2 (\gamma^k)^T \cdot (\tilde{\lambda} - \lambda^k)] \leq \\ &\leq \| \tilde{\lambda} - \lambda^k \|^2 + t_k [t_k \| \gamma^k \|^2 - 2 [w(\tilde{\lambda}) - w(\lambda^k)]] \leq \| \tilde{\lambda} - \lambda^k \|^2 \end{aligned}$$

$$\text{Portanto } \| \tilde{\lambda} - \lambda^{k+1} \| \leq \| \tilde{\lambda} - \lambda^k \|$$

Assim se t_k está no intervalo indicado, o novo ponto λ^{k+1} está mais próximo de $\tilde{\lambda}$ do que estava o ponto λ^k .

Iteração do tipo aqui citada tinha sido investigada anteriormente junto com o método de relaxação para a solução de um sistema de desigualdades lineares [Ag1] e [MS1] e, uma interpretação geométrica desta situação é dada em [HK1].

Em virtude da concavidade da função w , a diferença $[w(\lambda^*) - w(\lambda^k)]$ é maior quando λ^k está mais distante de $\lambda^* \in \bigwedge^*$. Desta maneira, Bazaraa e Sherali, [BS1] propõem, ao invés da escolha de um sobreestimador $\tilde{w}$ fixo, uma atualização periódica

de $\tilde{w}$ de modo que $\left[\frac{\tilde{w} - w(\lambda^*)}{\| \gamma^k \|^2} \right]$ esteja diretamente relacionada com o intervalo $\left[\frac{w(\lambda^*) - w(\lambda^k)}{\| \gamma^k \|^2}, t_k^* \right]$.

Esta atualização periódica é feita por:

$$\tilde{w} = \alpha_k \cdot w^0 + (1 - \alpha_k) w^p, \text{ onde:}$$

w^0 é fixo e tal que $w^0 > w(\lambda^*)$.

w^p é o melhor valor presente para a função objetivo w e, a sequência decrescente $\{\alpha_k\}$ tenha as propriedades seguintes:

$$\alpha_0 = 1 \text{ e } \lim_{k \rightarrow \infty} \alpha_k = \epsilon_0 > 0. \text{ Procura-se, nesse processo, evi-}$$

tar que $\tilde{w}$ se torne um subestimador de $w(\lambda^*)$.

Nesse sentido, a sequência $\{\alpha_k\}$ selecionada deve dar um peso maior para w^0 nas etapas iniciais do processo e, gradualmente, este peso é transferido para w^p à medida que w^p aproxima-se de $w(\lambda^*)$.

CAPÍTULO 3

PROBLEMA DE COBERTURA COM ARESTAS

3.1 - Formulação Matemática

Definição:

-Define-se uma semi-aresta para um grafo $G = (N, A \cup S)$ como sendo um elemento de $S \subset N$ que não forma, com outro nó distinto de N , uma aresta para o grafo G .

Seja $G = (N, A \cup S)$ um grafo não-orientado, onde: $N = \{1, 2, \dots, n\}$ é um conjunto de nós, $A = \{(i;j)/i \in N, j \in N \text{ e } i \neq j\}$ é um conjunto de arestas (pares não ordenados de nós distintos) e $S \subset N$ é um conjunto de semi-arestas.

Seja $c = (c_{ij})$ um vetor de custos das arestas e $d = (d_i)$ um vetor de custos das semi-arestas.

Associa-se à cada aresta $(i;j) \in A$ uma variável x_{ij} e à cada semi-aresta $i \in S$, uma variável y_i . Observa-se que x_{ij} e x_{ji} denotam a mesma variável.

O problema de cobertura com arestas consiste em encontrar vetores binários $x \in \{0,1\}^{|A|}$ e $y \in \{0,1\}^{|S|}$ que sejam solução ótima do problema seguinte:

$$\begin{aligned}
 & \min \sum [\bar{c}_{ij} \cdot x_{ij} / (i;j) \in A] + \sum [\bar{d}_i \cdot y_i / i \in S] \\
 [15] \quad & \text{sujeito a:} \quad \sum_j [x_{ij} / (i;j) \in A] + y_i \geq 1, \text{ para todo } i \in S \\
 & \quad \sum_j [x_{ij} / (i;j) \in A] \geq 1, \text{ para todo } i \in (N-S) \\
 & \quad x_{ij} \in \{0,1\}, \text{ para todo } (i;j) \in A \\
 & \quad y_i \in \{0,1\}, \text{ para todo } i \in S
 \end{aligned}$$

onde $\bar{c}_{ij} = c_{ij} - \sum_{i \in (N-S)} \lambda_i \bar{M}^j$ e $\bar{d}_i = d_i - \sum_{i \in S} \lambda_i \cdot \bar{M}^j$ são os

custos relaxados, conforme visto no capítulo anterior.

Considerando que a matriz de restrições do problema [15] é totalmente unimodular, pode-se concluir que uma solução ótima para esse programa inteiro 0-1 é dada por uma solução ótima básica do seguinte programa linear [Mil], [Lo1]:

$$\begin{aligned}
 & \min \sum [\bar{c}_{ij} \cdot x_{ij} / (i;j) \in A] + \sum [\bar{d}_i \cdot y_i / i \in S] \\
 [16] \quad \text{sujeito a:} \quad & \sum_j [x_{ij} / (i;j) \in A] + y_i \geq 1, \text{ para todo } i \in S \\
 & \sum_j [x_{ij} / (i;j) \in A] \geq 1, \text{ para todo } i \in (N-S) \\
 & 0 \leq x_{ij} \leq 1, \text{ para todo } (i;j) \in A \\
 & 0 \leq y_i \leq 1, \text{ para todo } i \in S
 \end{aligned}$$

O problema primal [16] apresenta seu problema dual associado, cuja formulação é:

$$\begin{aligned}
 & \max \sum [\pi_i / i \in N] + \sum [\omega_{ij} / (i;j) \in A] + \sum [\theta_i / i \in S] \\
 [17] \quad \text{sujeito a:} \quad & \pi_i + \pi_j + \omega_{ij} \leq \bar{c}_{ij}, \text{ para todo } (i;j) \in A \\
 & \pi_i + \theta_i \leq \bar{d}_i, \text{ para todo } i \in S \\
 & \pi_i \geq 0, \text{ para todo } i \in N \\
 & \omega_{ij} \leq 0, \text{ para todo } (i;j) \in A \\
 & \theta_i \leq 0, \text{ para todo } i \in S
 \end{aligned}$$

As condições de folgas complementares para o par de problemas [15] e [16], são as seguintes:

$$\begin{aligned}
 & \pi_i \cdot [1 - \sum_j (x_{ij} / (i;j) \in A) - y_i] = 0, \text{ para todo } i \in S \\
 & \pi_i \cdot [1 - \sum_j (x_{ij} / (i;j) \in A)] = 0, \text{ para todo } (i;j) \in (N-S) \\
 & \omega_{ij} \cdot (1 - x_{ij}) = 0, \text{ para todo } (i;j) \in A
 \end{aligned}$$

$$\begin{aligned} \theta_i \cdot (1 - y_i) &= 0, \quad \text{para todo } i \in S \\ (\bar{c}_{ij} - \pi_i - \pi_j - \omega_{ij}) \cdot x_{ij} &= 0, \quad \text{para todo } (i;j) \in A \\ (d_i - \pi_i - \theta_i) \cdot y_i &= 0, \quad \text{para todo } i \in S \end{aligned}$$

Definindo o custo relativo da aresta $(i;j)$ como sendo $e_{ij} = \bar{c}_{ij} - \pi_i - \pi_j$ e o custo relativo para a semi-aresta (i) por $e_i = \bar{d}_i - \pi_i$, então para toda solução ótima do problema de cobertura, tem-se $\omega_{ij} = \min\{0, e_{ij}\}$ e $\theta_i = \min\{0, e_i\}$

Desta forma:

$$e_{ij} < 0 \rightarrow \omega_{ij} = e_{ij} \text{ e } x_{ij} = 1, \quad \text{para todo } (i;j) \in A$$

$$e_i < 0 \rightarrow \theta_i = e_i \text{ e } y_i = 1, \quad \text{para todo } i \in S$$

$$e_{ij} > 0 \rightarrow \omega_{ij} = 0 \text{ e } x_{ij} = 0, \quad \text{para todo } (i;j) \in A$$

$$e_i > 0 \rightarrow \theta_i = 0 \text{ e } y_i = 0, \quad \text{para todo } i \in S$$

As variáveis duais π_i estão associadas a cada nó $i \in N$, as duais ω_{ij} à cada aresta $(i;j) \in A$ e as duais θ_i à cada semi-aresta $i \in S$

3.2 - Conceitos

Antes do algoritmo para cobertura com arestas ser apresentado, alguns conceitos são importantes:

Definições:

-Dados os vetores x e π , define-se o conjunto de arestas-soluções e semi-arestas-soluções como sendo o conjunto:

$$A^* \cup S^* = \{(i;j) \in A / x_{ij} = 1\} \cup \{i \in S / y_i = 1\}$$

-Define-se o conjunto $A^- \cup S^- = \{(i;j) \in A / c_{ij} \leq 0\} \cup \{i \in S / d_i \leq 0\}$

-Dada uma solução dual viável de [17], define-se o conjunto das arestas-de-igualdade e semi-arestas de igualdade, como o conjunto: $(A^* \cup S^*) = \{ (i;j) \in A / \bar{e}_{ij} = 0 \} \cup \{ i \in S / \bar{e}_i = 0 \}$

Observa-se que o conjunto $A^* \cup S^*$ fornece uma cobertura de custo mínimo ao final do algoritmo.

Dependendo do conjunto onde estão as arestas-soluções e semi-arestas-soluções uma classificação é utilizada para arestas e nós [MP1]:

(a) arestas-emparelhadas: correspondem aos elementos de A^* que são únicos para cada nó i e que não estão em A^- .

(b) arestas de cobertura e semi-arestas de cobertura: são todas as arestas e semi-arestas de $A^* \cup S^*$ que não estão definidas em (a).

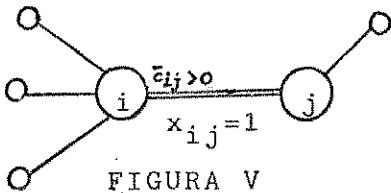


FIGURA V
(aresta emparelhada)

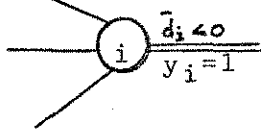


FIGURA VI
(semi-aresta de cobertura)

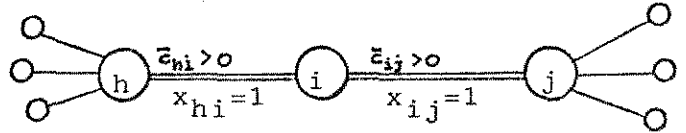


FIGURA VII
((i;j) e (h;i) são arestas de cobertura)

h é nó emparelhado

i é nó tipo 2

j é nó tipo 1

(c) nós emparelhados: são nós incidentes com uma aresta emparelhada

(d) nós tipo 1: nós incidentes com exatamente uma aresta ou uma semi-aresta de cobertura

(e) nós tipo 2: nós incidentes com duas ou mais arestas ou semi-arestas de cobertura

(f) nós expostos: nós que apresentam todas as arestas ou semi-arestas incidentes não pertencentes ao conjunto $A^* \cup S^*$.

Observa-se, portanto, que se o nó i é exposto, tem-se $\sum_j [x_{ij} / (i;j) \in A] = 0$ e $y_i = 0$, caso $i \in S$.

Definições:

-Caminho Alternante é um caminho constituído por elementos de $A \cup S$ que se alternam, ao longo desse caminho, em elementos não pertencentes a $A^* \cup S^*$ e elementos de $A^* \cup S^*$,

-Caminhos Aumentantes são caminhos alternantes que liga um nó exposto a outro nó exposto, ou a um nó tipo 2 ou a um nó tipo 1. (Neste último caso o caminho deve ter um número ímpar de arestas).

Como para todo nó i exposto $\sum_j [x_{ij} / (i;j) \in A] = 0$ e $y_i = 0$, caso $i \in S$, a aresta incidente com a raiz dessa árvore ou caminho alternante não pertence a $A^* \cup S^*$, pois as árvores são plantadas com raízes em nós expostos e crescem pela incorporação de novas arestas de $A \cup S$ [Lo1].

A manutenção de árvores alternantes durante o algoritmo é realizada pela associação, para cada nó i da árvore, de um rótulo com duas componentes (tipo, predecessor), a saber:

i) se o número de arestas da árvore alternante, até a sua raiz, a partir do nó é ímpar ou par, o tipo é interno(-) ou externo(+), respectivamente.

ii) o número da aresta incidente no nó que pertence ao caminho

do nó até a raiz da árvore alternante é a componente predecessora.

A utilização de árvores alternantes objetiva a determinação de caminhos aumentantes, afim de que seja possível a realização de ampliações.

Definição:

-Uma ampliação é a troca do papel solução/não solução das arestas no caminho aumentante encontrado.

Neste algoritmo cada vez que uma ampliação é efetuada pelo menos um nó exposto torna-se não exposto. A busca de caminhos aumentantes é importante e uma cobertura mínima é obtida quando não se consegue novos caminhos aumentantes [Gal] [GN1].

As raízes das árvores sempre recebem rótulos tipo externo (+) e, em todo caminho predecessor, os nós alternam-se em rótulos tipo externo (+) e interno (-), a partir das raízes.

O algoritmo considera P_i como sendo o caminho predecessor do nó i até a raiz de sua correspondente árvore alternante.

3.3 - Etapas do Algoritmo

As seguintes operações são realizadas pelo algoritmo:

-Plantação da árvore: realizada na inicialização do algoritmo e após cada ampliação. Consiste na atribuição de rótulo (externo, sem predecessor) = $(+, \emptyset)$ para todo nó exposto.

-Crescimento da árvore (Rotulação dos nós): executada quando uma aresta-de-igualdade não solução conecta um nó externo i com um nó emparelhado j .

-Eliminação das árvores: realizada após cada ampliação. Todos os nós em cada árvore alternante perdem os rótulos.

-Ampliação: executada quando uma das situações abaixo ocorre:

1) ampliação dupla- quando existe uma aresta-de-igualdade não solução unindo dois nós externos i e j . A operação consiste na troca do papel solução/não solução das arestas do caminho aumentante $P_i \cup \{(i;j)\} \cup P_j$.

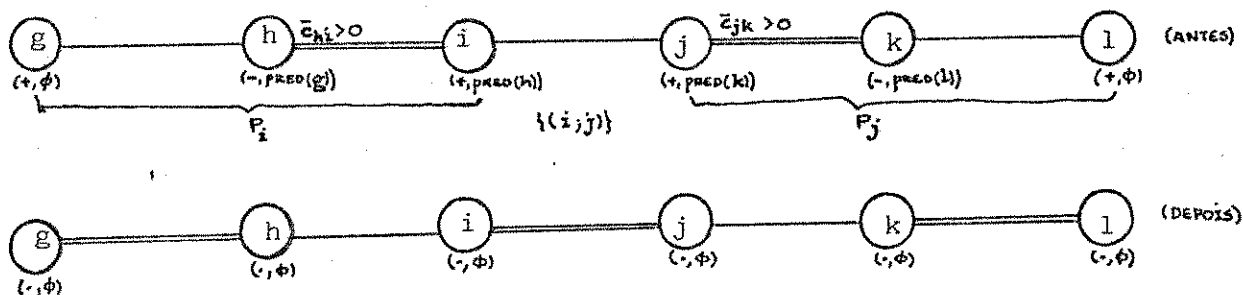


FIGURA VIII : (Dupla ampliação)

2) ampliação do tipo 1 - quando existe uma aresta-de-igualdade não solução unindo um nó externo i com um nó do tipo 1 ou com um nó emparelhado com $\pi_j = 0$.

2a) se a aresta $(j;k) \in A^* - A^-$ a operação consiste na troca do papel solução/não solução das arestas do caminho aumentante $P_i \cup \{(i;j), (j;k)\}$.

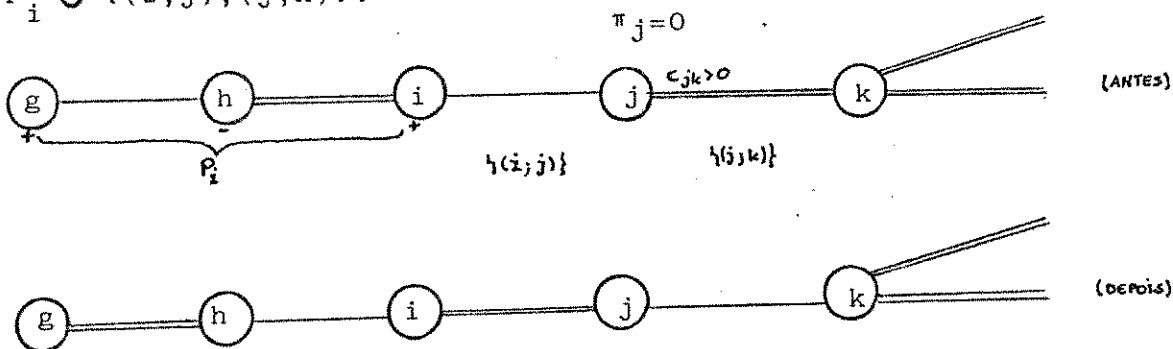


FIGURA IX : (Ampliação do tipo 1)

2b) se a aresta $(j;k) \in A^-$ o caminho aumentante sobre o qual se faz a troca é $P_i \cup \{(i;j)\}$

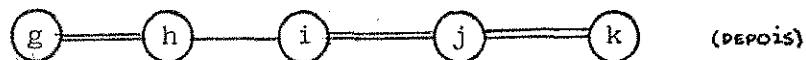
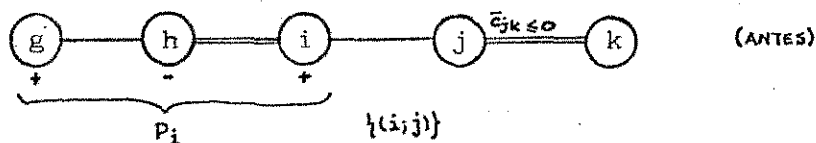


FIGURA X : (Ampliação do tipo 1)

2c) se $j \in S^* - S^-$ o caminho aumentante utilizado é $P_i \cup \{j\} \cup \{(i;j)\}$

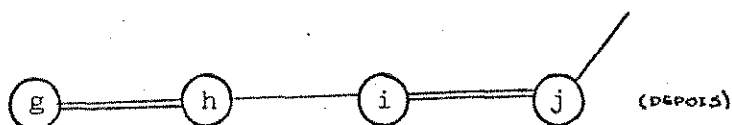
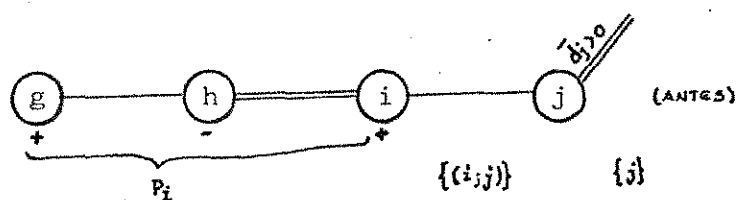


FIGURA XI : (Ampliação do tipo 1)

2d) se $j \in S^-$, a troca do papel solução/não solução é realizada sobre o caminho aumentante $P_i \cup \{(i;j)\}$

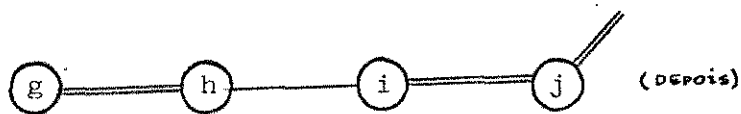
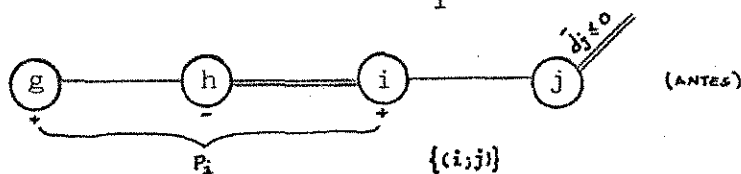


FIGURA XII : (Ampliação do tipo 1)

3) ampliação do tipo 2 - quando existe uma aresta-de-igualdade não solução

unindo um nó externo i com um nó j do tipo 2:

nesta situação, a troca do papel solução/não solução é realizada sobre o caminho aumentante $P_i \cup \{(i;j)\}$.

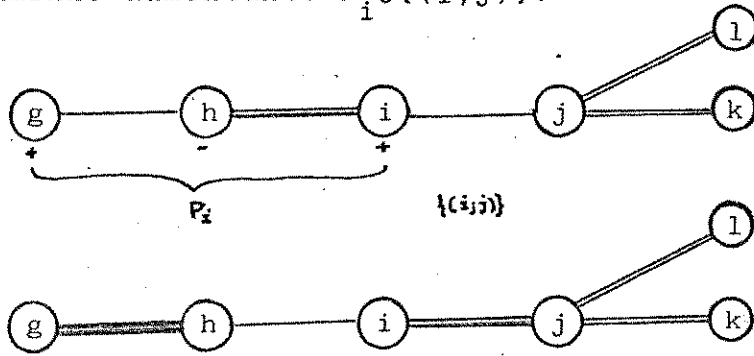


FIGURA XIII: (Ampliação do tipo 2)

Durante toda a execução do algoritmo um nó que se torna não exposto permanece nesta condição até o final do algoritmo, podendo, no entanto, ter sua classificação quanto ao tipo (emparelhado, tipo 1, tipo 2) alterada nas etapas do processo.

As seguintes propriedades são satisfeitas pelo algoritmo durante sua execução:

$$\pi_i \geq 0, \text{ para todo } i \in N$$

$$\sum_j [x_{ij} / ((i;j) \in A)] + y_i > 1, \text{ implica em } \pi_i = 0, \text{ p/cada } i \in S$$

$$\sum_j [x_{ij} / ((i;j) \in A)] > 1, \text{ implica em } \pi_i = 0, \text{ p/cada } i \in (N-S)$$

$$x_{ij} = 1, \text{ implica em } e_{ij} \leq 0, \text{ para todo } (i;j) \in A$$

$$y_i = 1, \text{ implica em } e_i \leq 0, \text{ para todo } i \in S$$

$$x_{ij} = 0, \text{ implica em } e_{ij} \geq 0, \text{ para todo } (i;j) \in A$$

$$y_i = 0, \text{ implica em } e_i \geq 0, \text{ para todo } i \in S$$

Definição:

-Durante o algoritmo, um conjunto de árvores alternantes é chamada uma floresta húngara sempre que as condições que seguem

são observadas:

a) $\pi_i > 0$, para cada nó interno $(i) \in N$.

b) todo nó interno $(-)$ é unido, por uma aresta emparelhada, com um nó externo $(+)$.

c) não existe aresta-de-igualdade unindo um nó externo $(+)$ com um nó externo $(+)$ ou sem rótulo. [Mil.]

Quando o algoritmo detecta uma floresta húngara, uma modificação na solução dual é realizada.

3.4 - O ALGORITMO

0 - INICIALIZAÇÃO

Atribua $\pi_i = 0$, para cada $i \in N$

$x_{ij} = 1$, para todo $(i;j) \in A^-$

$y_i = 1$, para todo $i \in S^-$

$x_{ij} = 0$, para todo $(i;j) \in A - A^-$

$y_i = 0$, para todo $i \in S - S^-$

I - INVESTIGAÇÃO DA SOLUÇÃO PRIMAL

1. Apagar rótulos existentes

2. Identificar todos os nós expostos e para cada um deles atribuir rótulos (tipo externo, sem predecessor) = $(+, \emptyset)$.

Tais nós são raízes de árvores alternantes. Elaborar uma lista com esses nós.

3. Se não existirem nós expostos, termine.

4. Se a lista de nós expostos está vazia, vá para II

5. Se a lista de nós expostos não está vazia, retirar o primeiro nó externo i da lista e examinar cada aresta $(i;j)$ e cada semi-aresta i , se existir, com custo relativo $e_{ij} = 0$ e custo relativo $e_i = 0$, respectivamente.

6. Se o nó j é externo (+), então existe caminho aumentante (dupla ampliação). Modificar a solução primal e retornar ao passo 1.

7. Se o nó j é interno (-), examinar a próxima aresta incidente com o nó i .

8. Se o nó j é sem rótulo e $\pi_j = 0$, então existe caminho aumentante (ampliação do tipo 1). Modificar a solução primal e retornar ao passo 1.

9. Se o nó j é um nó sem rótulo com $\pi_j > 0$ e ou uma aresta $(j;k)$ é uma aresta de cobertura ou $j \in S^*$, então existe caminho aumentante. Modificar a solução primal e retornar ao passo 1.

10. Se uma situação distinta daquelas citadas nos passos 6, 7, 8 e 9 é detectada, rotular o nó j como (interno, predecessor(i)) = (-, pred(i)) e o nó k , ligado ao nó j , como (externo, predecessor(j)) = (+, pred(j)). Incluir o nó k na lista para exame.

II - MUDANÇA NA SOLUÇÃO DUAL

11. Esta etapa é executada quando detecta-se uma floresta húngara após a realização das etapas anteriores:

Calcular:

$$\delta_i = \min \left\{ \frac{1}{2} e_{ij} \mid (i;j) \text{ é uma aresta não solução unindo dois nós externos} \right\}.$$

$\delta_2 = \min \{ e_{ij} / (i;j) \text{ é uma aresta não solução unindo um } \text{ nó externo } i \text{ com um nó sem rótulo na componen} \text{ te tipo (nó } j) \} .$

$$\delta_3 = \min \{ e_i / i \in S \}$$

$$\delta_4 = \min \{ \pi_i / i \text{ é um nó interno } (-) \}$$

$$\delta = \min \{ \delta_1, \delta_2, \delta_3, \delta_4 \}$$

Utiliza-se a convenção: $\min \emptyset = \infty$.

Após o cálculo de δ , tem-se: se $\delta = \infty$, o problema é inviável e o algoritmo termina, caso contrário, para cada nó $i \in N$, faz-se:

$$\pi_i = \begin{cases} \pi_i + \delta & , \text{ se o nó } i \text{ é do tipo externo } (+) \\ \pi_i & , \text{ se o nó } i \text{ é do tipo sem rótulo } (\emptyset) \\ \pi_i - \delta & , \text{ se o nó } i \text{ é do tipo interno } (-) \end{cases}$$

A seguir, volta-se para o passo 1.

CAPÍTULO 4

ALGORÍTMO PARA RESOLUÇÃO DO PROBLEMA DE REPRESENTAÇÃO DE CONJUNTOS (PRC)

4.1 - Relação entre o (PRC) e o (PCA)

No capítulo 1 apresentou-se a seguinte formulação para o problema de representação de conjuntos (PRC) :

$$\begin{aligned}
 & \min c^T \cdot z \\
 [18] \quad & \text{sujeito a } M \cdot z \geq \mathbf{1} \\
 & z \in \{0,1\}^n, \text{ onde } c \in \mathbb{R}^n; z \in \mathbb{R}^n \text{ e } \mathbf{1} \in \mathbb{R}^m.
 \end{aligned}$$

Na matriz $M_{m \times n}$, a linha M_k indica a incidência dos elementos de $V = \{1, 2, \dots, n\}$ no subconjunto F_k da família $F = \{F_1, F_2, \dots, F_m\}$ enquanto a coluna M^t indica os subconjuntos F_k que são representados pelo elemento $t \in V$.

O problema consiste em selecionar um subconjunto S , tal que:

$$a) S \subset V$$

$$b) S \cap F_k \neq \emptyset, \text{ para todo } k = 1, 2, \dots, m$$

$$c) \sum_{t \in S} c_t \text{ seja mínima, onde } c_t = \text{custo de inclusão do elemento } t \in V \text{ em } S.$$

Este problema é um caso muito especial de Programa Linear Inteiro 0-1, pois M é uma matriz binária ($m_{kt} \in \{0,1\}$ para $(k,t) \in \{1,2,\dots,m\} \times \{1,2,\dots,n\}$) e o lado direito das desigualdades lineares é um vetor de 1's ($\mathbf{1}_k = 1$, para $k=1,\dots,m$).

Murty [Mul] desenvolve um algoritmo, para este tipo de problema, usando apenas operações com conjuntos (união, interseção e diferença). Encontra-se, dessa forma, todas as soluções do problema de combinatória que busca a determinação de todos os representantes de cardinalidade mínima para F (PRCM).

No capítulo 2, um algoritmo indutivo sobre a cardinalidade de F ($|F| = m$) é resultado de Etcheberry [Et1]. Determina-se todos os representantes minimais $RM(1)$, considerando apenas a primeira restrição como F , isto é, $F = \{F_1\}$. Segue-se, a partir dessas soluções, os representantes minimais $RM(2)$, $RM(3)$, ..., $RM(m)$.

$RM(k+1)$ é sempre obtido de $RM(k)$ pela inclusão do vínculo $(k+1)$ na formação do novo PRC.

No capítulo 2 também, fez-se uma afirmação que, sob certas condições, o conjunto de restrições $M^0 \cdot x \geq 1^0$ apresenta uma estrutura especial que torna o problema relaxado [11] passível de ser resolvido de forma eficiente.

Murty e Perin [MP1] mostram que se a matriz M^0 apresenta em cada coluna exatamente dois elementos diferentes de zero, então o problema relaxado [11] é um problema de cobertura com arestas, de custo mínimo, num grafo. Neste caso M^0 é a matriz de incidência (nó $\times$ aresta) de um grafo não orientado. Esta situação permite utilizar a propriedade de unimodularidade total da matriz M^0 e, conseqüentemente conseguir soluções básicas inteiras para o programa linear obtido do programa linear inteiro do (PRC) pela relaxação das restrições de integrabilidade do problema [11] [Mi1] e [Lo1].

4.2 - O ALGORITMO

Desenvolveu-se neste trabalho um algoritmo do tipo Partição-Limitação ("Branch and Bound") na resolução do PRC associado ao método de relaxação Lagrangeana [Sh1], [Ge1] e [Et1].

O método consiste numa enumeração parcial das soluções viáveis e do corte das soluções não-ótimas pela utiliza

ção de valores que limitam inferiormente o valor da função objetivo, conforme visto no capítulo 2.

Utilizou-se durante todo o algoritmo, dois produtos cartesianos não ordenados (p.c.n.o.), como definido anteriormente.

Observa-se que as características de qualquer problema candidato (PC) são:

1. dois (p.c.n.os.): D e E

2. um conjunto de restrições não incluídas nestes produtos:

$$I = \{\text{índices das restrições não incluídas}\}$$

Em termos das desigualdades $M.z \geq 1$ do problema, tal caracterização indica uma partição nesse conjunto de restrições como $M^0.z \geq 1^0$ e $\bar{M}.z \geq \bar{1}$ vista no capítulo 2.

As restrições $M^0.z \geq 1^0$ correspondem aos vínculos F_k cujos elementos participam da formação dos produtos cartesianos não ordenados, enquanto as restrições $\bar{M}.z \geq \bar{1}$ correspondem às restrições F_k que ainda não participam da formação dos produtos cartesianos não ordenados (D ou E) e chamadas de restrições não incluídas: $k \in I$.

Desta forma, considerando um problema candidato (PC) cuja solução viável represente $F = \{F_1, F_2, \dots, F_k\}$ tem-se a seguinte formulação:

$$\begin{aligned}
 & \text{mín} \quad \sum_{t=1}^n c_t \cdot z_t \\
 [19] \quad & \text{suj} \quad \sum_{t \in D_i} z_t \geq 1 \quad ; i = 1, 2, \dots, r \\
 & \quad \sum_{t \in E_t} z_t \geq 1 \quad ; t = 1, 2, \dots, s \\
 & \quad \sum_{t \in F_k} z_t \geq 1 \quad ; k \in I
 \end{aligned}$$

$$z_t \in \{0,1\}$$

4.2.1 - Estratégia de Limitação

A estratégia de limitação apresentada corresponde ao método de relaxação Lagrangeana vista no capítulo 2. Relaxam-se as restrições não incluídas $k \in I$. Assim, o problema candidato relaxado oriundo do problema candidato [19], apresenta a formulação seguinte:

$$\begin{aligned}
 [20] \quad & \min \quad \sum_{t=1}^n c_t \cdot z_t + \sum_{k \in I} \lambda_k (1 - \sum_{t \in F_k} z_t) \\
 & \text{sujeito a} \quad \sum_{t \in D_i} z_t \geq 1 ; i=1,2,\dots,r \\
 & \quad \quad \quad \sum_{t \in E_j} z_t \geq 1 ; j=1,2,\dots,s \\
 & \quad \quad \quad z_t \in \{0,1\}
 \end{aligned}$$

ou,

$$\begin{aligned}
 [21] \quad & \min \quad \sum_{k \in I} \lambda_k + \sum_{t=1}^n (c_t - \sum_{k \in I} m_{kt} \cdot \lambda_k) \cdot z_t \\
 & \text{sujeito a} \quad \sum_{t \in D_i} z_t \geq 1 ; i=1,2,\dots,r \\
 & \quad \quad \quad \sum_{t \in E_j} z_t \geq 1 ; j=1,2,\dots,s \\
 & \quad \quad \quad z_t \in \{0,1\}
 \end{aligned}$$

$$\text{onde } m_{kt} = \begin{cases} 1, & \text{se } t \in F_k \\ 0, & \text{caso contrário} \end{cases}$$

Identificando como $\bar{c}_t = c_t - \sum_{k \in I} m_{kt} \cdot \lambda_k$, o cus

to relaxado de inclusão do elemento $t \in F$ na restrição F_k , tem-

$$\begin{aligned}
 & \text{-se:} \quad \sum_{k \in I} \lambda_k + \min_{z_t} \sum_t \bar{c}_t \cdot z_t \\
 [22] \quad & \text{sujeito a:} \quad \sum_{t \in D_i} z_t \geq 1; \quad i=1,2,\dots,r \\
 & \quad \quad \quad \sum_{t \in E_j} z_t \geq 1; \quad j=1,2,\dots,s \\
 & \quad \quad \quad z_t \in \{0,1\}
 \end{aligned}$$

As soluções dos problemas candidatos são obtidas pelo algoritmo de cobertura com arestas, com custo mínimo, num grafo bipartido descrito no capítulo anterior.

Para que o problema adquira uma estrutura de problema de cobertura citada, faz-se a seguinte identificação:

(a) se $t \in \left[\left(\bigcup_{i=1}^r D_i \right) \cap \left(\bigcup_{j=1}^s E_j \right) \right]$, então t é uma aresta.

(b) se $t \in S = \left[\left(\bigcup_{i=1}^r D_i \right) \Delta \left(\bigcup_{j=1}^s E_j \right) \right]$, então t é uma semi-aresta

Com tal identificação feita para cada problema candidato, associa-se a variável x_{ij} para a ocorrência de arestas e a variável y_i para a ocorrência de semi-arestas. Dessa forma a formulação do problema passa a ser dada por:

$$\begin{aligned}
 & \sum_{k \in I} \lambda_k + \min \sum [\bar{c}_{ij} \cdot x_{ij} / (i;j) \in A] + \sum [\bar{d}_i \cdot y_i / i \in S] \\
 [23] \quad & \text{sujeito a:} \quad \sum_j [x_{ij} / (i;j) \in A] + y_i \geq 1, \quad \text{para } i \in S \\
 & \quad \quad \quad \sum_j [x_{ij} / (i;j) \in A] \geq 1, \quad \text{para } i \in N - S \\
 & \quad \quad \quad (x_{ij}, y_i) \in \{0,1\} \times \{0,1\}
 \end{aligned}$$

onde $A = \{(i;j) \text{ tal que existe } t \in D_i \cap E_j\}$

Observe que cada variável x_{ij} ou y_i de [23] corresponde a uma única variável z_t em [22] e que cada variável z_t corresponde a, no máximo, uma variável x_{ij} ou y_i .

Associando aos fatores D_i e E_j dos produtos cartesianos não ordenados D e E , respectivamente, nós do grafo, observa-se que o problema [23] descrito é idêntico ao problema de cobertura para o grafo bipartido [15] do capítulo anterior.

A característica bipartida do grafo é resultante da propriedade disjunção mútua dos fatores D_i , para $i=1,2,\dots,r$ no p.c.n.o. D . A mesma propriedade vale para o p.c.n.o. E .

Conforme, já visto no capítulo 2, para qualquer vetor de multiplicadores, não negativos, de Lagrange $\lambda \in R^{|I|}$, associado às restrições não incluídas, o valor da função objetivo para o problema candidato relaxado [20] é um limitante inferior para o valor da função objetivo do referido problema.

Obtém-se bons multiplicadores de Lagrange, realizando-se o algoritmo de otimização subgradiente, conforme visto no capítulo 2.

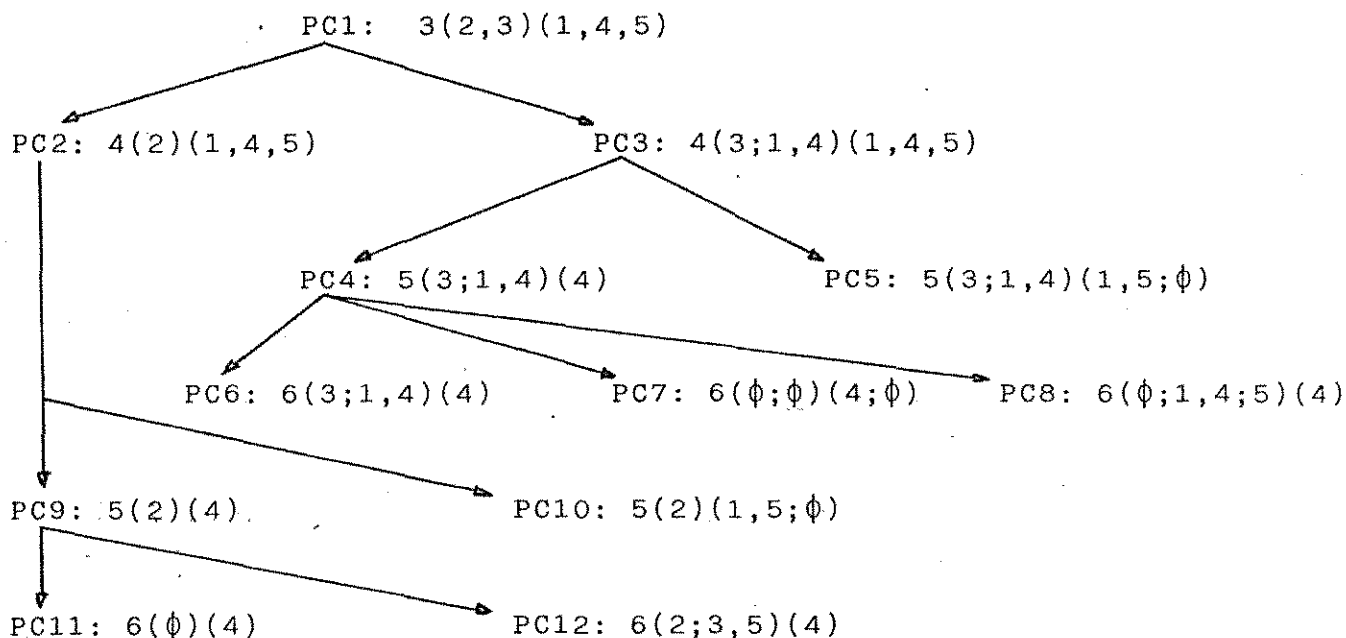
Após esta etapa, mantém-se como multiplicadores de Lagrange iniciais nos problemas candidatos, obtidos a partir de um determinado (PC), aqueles que estão associados a melhor solução viável do (PC).

4.2.2 - Estratégia de Partição

Considere como exemplo um problema de Representação de Conjuntos (PRC) com o seguinte conjunto de restrições:

$$\begin{array}{rcl}
 & x_2 + x_3 & \geq 1 \\
 x_1 + & & + x_4 + x_5 \geq 1 \\
 x_1 + x_2 & + x_4 & \geq 1 \\
 & x_4 & \geq 1 \\
 & x_3 + x_5 & \geq 1
 \end{array}$$

O algoritmo inicializa-se com $R(2)$ pois são utilizadas as duas primeiras restrições na formação dos p.c.n.os. Assim, define-se $D = D_1 = F_1$ (1ª restrição) e $E = E_1 = F_2$ (2ª restrição). Segue-se uma representação compacta da geração dos subproblemas realizada pelo algoritmo para este exemplo e na ordem desta geração:



Nesta representação o primeiro número corresponde à restrição F_{k+1} (próxima a entrar na formação dos (p.c.n.os.) . No primeiro entre parenteses está o produto cartesiano não ordenado D com seus fatores separados por (;) e no segundo entre parenteses está o (p.c.n.o.) E com idêntica representação.

A presença de fator ϕ indica inviabilidade do subproblema gerado a partir do problema candidato. Cada (p.c.n.o.) gerado nesse processo dá origem a um novo problema candidato.

A geração desses novos produtos cartesianos não ordenados segue a seguinte regra de partição:

1. se $(k+1)$ é ímpar particiona-se o p.c.n.o. $D \subset R(k)$
2. se $(k+1)$ é par particiona-se o p.c.n.o. $E \subset R(k)$.

Em cada caso citado o produto cartesiano não ordenado que não é utilizado na etapa de partição permanece inalterado, isto é, nos subproblemas gerados um dos (p.c.n.os.) é o mesmo do problema candidato gerador.

4.2.3 - Execução

Cada subproblema gerado é realizado e não necessáriamente resolvido, ou seja, é estudado com o objetivo de obtenção de uma solução ótima mas, com frequência, apenas são determinadas algumas soluções cujos valores relaxados fornecem limitantes inferiores para o valor da solução ótima do subproblema. Nestes casos, escolhe-se, dentre estas, aquela com o maior valor relaxado que é denominada melhor solução do subproblema.

Dado um subproblema, a resolução deste pode ser classificada como:

- i) viável ou não para o problema original.
- ii) viável ou não para o subproblema em questão.
- iii) viável ou não para o subproblema relaxado.

Todas as soluções encontradas no algoritmo são soluções viáveis para o subproblema relaxado. Em caso do subproblema relaxado (cobertura mínima) ser inviável, nenhuma solução é obtida.

Dada uma solução z , o valor desta solução é dada por $\text{obj} = c^T \cdot z$ e o valor relaxado é $\text{objr} = c^T \cdot z + \lambda^T \cdot (\bar{1} - \bar{M} \cdot z)$. Pode-se afirmar que $\text{objr} \leq \text{obj}$, como consequência do corolário 1 (capítulo 2).

Para cada solução obtida é determinado um subgradiente associado que indica uma direção de melhoria da solução. Um subgradiente nulo indica que a solução obtida é uma solução

ótima para o subproblema e uma solução viável para o problema original.

Os testes adotados para a aplicação das rotinas de partição, de corte e de atualização do incumbente (melhor solução viável para o subproblema) foram implementados com a seguinte sequência, após a realização de um subproblema:

- i) subproblema relaxado é inviável - cortar subproblema.
- ii) melhor valor relaxado é maior do que o valor do incumbente - cortar subproblema.
- iii) melhor solução com subgradiente nulo - atualizar incumbente e cortar subproblema.
- iv) melhor solução é viável para o problema original - atualizar incumbente e particionar subproblema.
- v) melhor solução é inviável para o problema original - particionar subproblema.

A prioridade na escolha dos subproblemas da lista utilizada é LIFO, que seleciona o último subproblema gerado. Tal armazenamento é eficiente e apresenta um menor número de subproblemas candidatos na lista alcançando, dessa maneira, soluções viáveis mais rapidamente.

CAPÍTULO 5

RESULTADOS COMPUTACIONAIS

Foi implementado o método de Partição-Limitação ("Branch and Bound") para resolver o Problema de Representação de Conjuntos. Ao método citado está associado o método de relaxação Lagrangeana e otimização com subgradiente.

A implementação foi realizada utilizando-se a linguagem FORTRAN 77 e os resultados que seguem foram obtidos no microcomputador MICROTEC PCXT2002-Master, tendo na sua configuração 2 drives de 5 1/4" e Winchester de 40Mb, do Laboratório de Matemática Aplicada da UNICAMP.

O método utiliza:

- (i) estratégia de limitação - corresponde ao método de relaxação Lagrangeana e as soluções do subproblema são obtidas pelo algoritmo de cobertura com arestas apresentado no capítulo 3.
- (ii) estratégia de partição - apresentada no capítulo 4.
- (iii) disciplina: LIFO.

Os problemas testados foram gerados aleatoriamente a partir das seguintes constantes para cada problema:

nl = número de linhas da matriz

nc = número de colunas da matriz

csto inf. = extremo inferior do intervalo de variação dos custos de cada coluna.

csto sup. = extremo superior do intervalo de variação dos custos de cada coluna

dens. = densidade da matriz = (nº de 1's)/(nl.nc)

Nas tabelas que seguem foi adotada a seguinte nota

ção:

li = número da linha da matriz

tl = total de elementos não zero na linha li

(cj/ m(li,cj) = 1) é um vetor de números, cujas componentes cor
respondem às colunas cj com elementos não ze
ro na posição (li,cj).

PR = número do problema

T = tipo de execução implementada

N = número da execução

maior = número de iterações realizadas para o primeiro subproblema
ma, considerando a 1ª restrição como o (p.c.n.o.):D e
a 2ª restrição como o (p.c.n.o.) : E.

menor = número de iterações para os demais subproblemas.

napo = número máximo de iterações realizadas com o mesmo fator
alfa do passo da otimização subgradiente sem ocorrer me-
lhoria no valor do limitante inferior para o primeiro subproblema
ma. Após esse número de iterações, o valor de alfa é dividido
por 2.

nasp = número correspondente ao napo e associado aos demais subpro
blemas.

custo - varia no intervalo [csto inf.,csto sup.] , podendo assumir
valor real ou inteiro

tempo de exec. = tempo de execução medido em segundos até:

1.subp. - encontrar a solução do 1º subproblema.

1.viav. - encontrar a 1ª solução viável para o problema original
nal.

otimo - encontrar a solução ótima do problema original.

final - encontrar o término da execução.

n.subp. = número de subproblemas (PCs) na lista (médio e máximo)

n.part.D= número de partições tipo D na lista (mínimo, médio e
máximo)

$n_{part.E}$ = número de partições tipo E na lista (mínimo, médio e máximo).
 tamanho da lista (tamanho médio e máximo).
 n_{ct} = número de subproblemas cortados.
 $n_{c\phi}$ = cortados pela inviabilidade do subproblema relaxado.
 n_{c5} = cortados por subgradiente nulo e não atualização do incumbente.
 n_{c6} = cortados por subgradiente nulo e incumbente atualizado.
 n_{c7} = cortados pelo fato do melhor valor relaxado ser maior do que o valor do incumbente.
 n_{pt} = número de subproblemas particionados.
 n_{p1} = particionados em virtude da melhor solução ser viável para o problema original e incumbente atualizado.
 n_{p2} = particionados em virtude da melhor solução ser viável para o problema original e não atualização do incumbente.
 n_{p3} = particionados pelo fato que a melhor solução do subproblema é inviável para o problema original.
 n_{p4} = particionados pelo fato do passo assumir valor zero.
 $n_{res.}$ = número total de subproblemas resolvidos.
 $n_{ng\phi}$ = número de subproblemas que apresentaram fator vazio na estratégia de partição.
 $1.subp./sol.ot.$ = qualidade da solução do 1º subproblema com relação à solução ótima obtida.
 $1.viav/sol.ot.$ = qualidade da 1ª solução viável com relação à solução ótima obtida.
 $1.subp.$ = número de subproblemas examinados até solução 1º subp.
 $1.viav.$ = número de subproblemas examinados até 1ª solução viável.
 $otimo$ = número de subproblemas examinados até solução ótima.
 $total$ = número de subproblemas examinados até final da execução.

PROBLEMA 1 : (nl,nc) = (10,10) - dens.:0.6

custo das colunas

2.1132	1.2931	3.9972	1.9566	5.1059
1.5824	7.7269	8.0831	5.3979	0.6212

li tl (cj / m(li,cj) = 1)

1	4	1	4	6	9				
2	6	1	3	4	6	7	9		
3	8	1	3	4	5	6	8	9	10
4	8	1	3	4	5	6	7	8	10
5	5	4	5	6	7	8			
6	3	4	6	10					
7	7	2	3	5	6	7	8	9	
8	6	1	4	5	6	7	8		
9	5	1	2	5	7	8			
10	6	2	5	6	8	9	10		

PROBLEMA 2: (nl,nc) = (10,10) - dens.:0.65

custo das colunas

388.2944	79.9929	634.9089	991.7936	909.7107
968.0364	566.4595	924.9812	83.9151	903.8876

li tl (cj / m(li,cj) = 1)

1	6	1	3	4	5	6	9		
2	6	1	5	6	7	8	9		
3	7	2	3	5	6	7	8	10	
4	5	3	4	5	7	9			
5	7	1	3	4	5	7	8	9	
6	6	2	3	5	6	7	8		
7	8	1	4	5	6	7	8	9	10
8	6	1	2	3	4	7	9		
9	4	2	4	5	9				
10	10	1	2	3	4	5	6	7	8

PROBLEMA 3 : $(n1,nc) = (10,10)$ - dens.:0.4

custo das colunas

21.1325	12.9315	39.9719	19.5664	51.0587
15.8242	77.2690	80.8309	53.9787	6.2125

li	tl	(cj / m(li,cj) = 1)							
1	4	1	4	6	9				
2	5	3	4	6	7	9			
3	7	1	3	4	5	6	8	10	
4	4	1	6	7	10				
5	3	4	6	7					
6	1	10							
7	5	5	6	7	8	9			
8	3	4	6	8					
9	4	1	2	7	8				
10	6	2	5	6	8	9	10		

PROBLEMA 4 : $(n1,nc) = (10,10)$ - dens.:0.28

custo das colunas

21	12	39	19	51
15	77	80	53	6

li	tl	(cj / m(li,cj) = 1)							
1	1	4							
2	4	3	4	7	9				
3	5	4	5	6	8	10			
4	4	1	6	7	10				
5	1	8							
6	1	9							
7	4	5	6	7	8				
8	2	7	10						
9	3	1	6	7					
10	3	1	8	9					

PROBLEMA 4A : (nl,nc) = (10,10) - dens.:0.28

custo das colunas

21.1325	12.9315	39.9719	19.5664	51.0587
15.8242	77.2690	80.8309	53.9787	6.2125

li tl (cj / m(li,cj) = 1)

1	1	4				
2	4	3	4	7	9	
3	5	4	5	6	8	10
4	4	1	6	7	10	
5	1	8				
6	1	9				
7	4	5	6	7	8	
8	2	7	10			
9	3	1	6	7		
10	3	1	8	9		

PROBLEMA 5 : (nl,nc) = (10,20) - dens.:0.24

custo das colunas

55	13	11	65	76
25	74	70	40	93
21	98	78	0	84
31	24	30	96	53

li tl (cj / m(li,cj) = 1)

1	6	1	2	6	10	13	18
2	2	15	18				
3	6	1	2	3	4	16	18
4	4	8	14	15	19		
5	4	5	13	18	19		
6	2	16	20				
7	7	2	6	7	14	17	18
8	4	2	12	13	17		
9	6	3	9	11	12	15	20
10	6	2	10	12	15	18	20

PROBLEMA 5A : (n1,nc) = (10,20) - dens.:0.24

custo das colunas

55.0307	13.3504	11.2084	65.9173	76.8210
25.2946	74.7445	70.6083	40.3549	93.4844
21.5282	98.0489	78.6403	0.9277	84.5673
31.2468	24.6850	30.6322	96.8695	53.2078

li tl (cj / m(li,cj) = 1)

1	6	1	2	6	10	13	18
2	2	15	18				
3	6	1	2	3	4	16	18
4	4	8	14	15	19		
5	4	5	13	18	19		
6	2	16	20				
7	7	2	6	7	14	17	18
8	4	2	12	13	17		
9	6	3	9	11	12	15	20
10	6	2	10	12	15	18	20

PROBLEMA 6 : (n1,nc) = (10,20) - dens.:0.2

custo das colunas

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

li tl (cj / m(li,cj) = 1)

1	4	7	9	10	13
2	4	2	8	9	13
3	4	3	9	10	12
4	4	4	5	8	9
5	4	3	6	8	11
6	4	3	6	7	10
7	4	2	4	5	12
8	4	4	5	6	13
9	4	1	2	4	11
10	4	1	5	7	12



PROBLEMA 7 : (n1,nc) = (10,50) - dens.:0.17

custo das colunas

21	12	39	19	51
15	77	80	53	6
29	96	62	67	28
48	10	81	8	20
30	32	1	95	55
2	41	57	18	75
60	87	52	34	92
70	90	61	70	87
66	40	26	23	11
53	93	59	59	84

li t1 (cj / m(li,cj) = 1)

1	11	1	2	8	19	20	23	26	37	39	40	44	
2	7	6	7	8	24	26	30	49					
3	5	14	20	27	42	44							
4	10	1	3	4	7	20	23	24	35	47	50		
5	7	9	10	20	22	27	36	42					
6	8	1	6	11	16	19	29	36	48				
7	6	9	13	22	24	35	43						
8	12	10	11	13	15	17	22	28	36	37	44	46	49
9	8	6	10	20	25	27	29	34	43				
10	10	10	12	14	15	22	26	40	42	46	49		



PROBLEMA 8 : (n1,nc) = (10,50) - dens.:0.27

custo das colunas

21.1325	12.9315	39.9719	19.5664	51.0587
15.8242	77.2690	80.8309	53.9787	6.2125
29.0637	96.0949	62.9001	67.1042	28.3637
48.3663	10.8307	81.5072	8.1770	20.6528
30.7785	32.4292	1.5113	95.9623	55.7512
2.8781	41.3743	57.3025	18.7566	75.8616
60.7740	87.6814	52.8025	34.3877	92.7182
70.1066	90.8968	61.4640	70.2146	87.5860
66.0473	40.0985	26.2710	23.1275	11.2617
53.2989	93.8953	59.7337	59.5479	84.0730

li t1 (cj / m(li,cj) = 1)

1	14	1	2	7	8	12	19	20	23	26	37	38	39	40	44
2	14	1	2	6	7	8	11	13	24	26	29	30	37	39	49
3	8	4	8	14	20	27	37	42	44						
4	16	1	3	4	7	9	19	20	23	24	25	29	35	41	45 47 50
5	13	3	9	10	20	22	27	28	33	36	41	42	43	50	
6	13	1	6	10	11	12	16	17	19	23	29	34	36	48	
7	13	5	9	12	13	18	22	24	31	33	35	41	43	48	
8	14	10	11	13	15	17	18	21	22	28	36	37	44	46	49
9	15	1	6	10	12	20	22	25	27	29	31	34	40	43	48 49
10	14	1	10	12	14	15	16	22	26	37	39	40	42	46	49



PROBLEMA 9 : (nl,nc) = (20,10) - dens.:0.37

custo das colunas

56.9466	96.3970	89.1643	29.9049	95.3066
44.0881	16.9996	36.8224	8.3692	18.4838

li t1 (cj / m(li,cj) = 1)

1	4	3	8	9	10			
2	3	2	4	8				
3	3	1	6	8				
4	2	2	4					
5	5	1	2	5	6	9		
6	2	6	9					
7	2	7	8					
8	6	3	5	6	7	8	10	
9	6	1	2	5	8	9	10	
10	2	3	5					
11	4	6	7	9	10			
12	4	5	6	7	8			
13	2	1	5					
14	4	1	2	4	9			
15	3	2	5	10				
16	5	1	2	3	8	10		
17	5	1	2	7	9	10		
18	6	1	3	5	6	8	9	
19	2	3	6					
20	3	2	7	9				

PROBLEMA 10 : (n1,nc) = (20,15) dens.: 0.25

custo das colunas

272.0625	864.7872	395.5279	451.4733	440.1248
589.2960	947.1131	382.0149	582.7526	58.3899
138.3030	972.1806	530.0240	602.1469	799.1755

li t1 (cj / m(li,cj) = 1)

1	9	2	3	6	8	9	10	11	13	14
2	5	2	3	9	14	15				
3	7	3	5	6	8	9	10	14		
4	9	2	4	6	7	8	9	10	11	12
5	8	1	2	4	5	7	9	10	12	
6	8	1	6	7	8	9	11	12	15	
7	7	2	3	4	6	10	12	15		
8	9	2	3	5	6	8	9	12	13	15
9	7	1	3	6	11	13	14	15		
10	9	1	4	5	6	8	11	12	14	15
11	5	7	10	11	13	14				
12	6	1	2	6	7	10	13			
13	7	5	6	8	9	10	12	14		
14	9	1	3	5	6	7	10	12	14	15
15	8	1	4	5	6	7	11	12	13	
16	8	2	5	6	9	10	11	13	14	
17	8	1	5	7	8	9	10	14	15	
18	9	4	5	9	10	11	12	13	14	15
19	12	1	2	4	5	6	7	8	10	11 13 14 15
20	8	1	3	4	5	6	7	12	15	



PROBLEMA 11 : (n1,nc) = (20,20) - dens.:0.3

custo das colunas

36.4526	90.1962	26.6398	34.5539	62.9433
41.0472	98.3392	14.5272	99.5536	93.5950
47.0626	-9.3983	45.0922	91.6403	-8.3814
26.5859	18.3677	99.0156	50.8157	76.2885

li t1 (cj / m(li,cj) = 1)

1	3	5	13	14						
2	8	1	4	10	12	14	16	19	20	
3	8	2	5	12	13	15	17	19	20	
4	2	7	11							
5	8	1	3	5	9	14	15	18	20	
6	4	5	11	14	17					
7	9	1	2	4	5	8	10	11	16	17
8	5	2	5	6	9	18				
9	6	1	3	6	13	16	19			
10	4	5	7	11	20					
11	4	2	6	8	12					
12	9	2	3	4	7	8	10	12	18	20
13	7	5	6	11	13	14	15	16		
14	8	1	2	3	4	10	11	12	17	
15	5	2	4	6	14	17				
16	5	5	7	9	10	20				
17	2	5	19							
18	3	10	18	19						
19	5	4	10	11	14	15				
20	9	2	4	6	7	8	9	14	15	19

PROBLEMA 12 : (n1,nc) = (10,100) - dens.:0.25

custo das colunas

19.5002	82.8381	55.8235	5.7770	82.0425
15.9951	21.0411	92.6183	8.1954	27.2726
91.3813	24.0841	30.9748	99.6787	99.8519
83.1823	83.3886	16.2209	79.4607	52.9206
98.4444	59.9072	63.2155	16.3068	9.1501
13.7454	84.1241	56.3489	48.5135	60.7432
75.1943	56.0545	49.5425	83.9086	69.4340
98.4315	45.2448	66.2492	99.8509	66.4877
68.6283	90.7729	99.4529	43.2309	52.7008
40.4876	1.2478	11.6690	30.4700	98.4010
38.2433	54.8097	34.9438	47.5210	43.4474
55.6608	99.1301	70.8553	49.8678	97.2303
56.0365	51.4118	90.5123	62.5258	38.6711
72.1983	98.3888	34.5552	80.0414	16.2225
6.5050	96.3263	13.1554	66.4924	47.8687
30.8469	71.0209	6.0158	55.4079	21.1249
86.6457	17.8003	62.3702	50.1680	93.0375
84.8537	1.5236	0.9841	23.2043	90.1844
5.9557	56.5809	10.1538	16.7996	8.6745
99.9663	86.8938	47.7072	42.6878	14.1481

11 t1 (cj / m(li,cj) = 1)

1	33	4	6	8	10	18	23	24	26	28	30	34	42	49	50	59
		61	62	64	67	69	71	72	78	79	80	83	85	86	88	90
		93	95	99												
2	26	3	6	7	8	10	24	27	28	29	35	36	43	44	45	53
		62	68	70	71	73	74	88	91	95	97	98				
3	30	6	9	11	13	17	18	21	22	29	30	34	44	45	51	60
		72	73	74	75	77	78	80	81	89	91	94	95	98	99	100
4	22	13	23	32	37	39	42	44	45	46	47	52	62	63	67	69
		72	74	77	83	84	96	97								
5	25	1	2	5	16	17	26	32	36	37	38	42	43	51	58	59
		62	67	68	71	74	75	77	78	93	97					
6	30	3	5	10	11	13	16	17	18	26	27	31	33	34	37	38
		41	42	43	46	52	59	65	79	81	86	92	95	96	97	100
7	22	4	6	19	24	25	33	34	36	39	45	47	48	51	55	65
		72	74	83	84	96	97	100								
8	18	11	13	14	19	23	34	39	43	51	72	76	80	83	87	91
		92	97	98												
9	18	1	6	10	11	15	21	23	26	29	31	32	37	43	51	54
		70	76	88												
10	27	9	10	14	17	21	24	28	30	31	33	34	40	45	49	50
		53	54	59	62	73	77	79	80	81	85	97	100			



PROBL.!		DADOS DE ENTRADA				custo		tempo(seg.) de exec.ate:			
PR!	T!N!	maior!	menor!	napo!	nasp!	real	int.	i.subp.	i.viav.	otimo	final
1	A 1	0	0	0	0	[0,10]		<1	1	9	10
1	A 2	5	1	4	1	[0,10]		<1	1	6	7
1	A 3	100	1	4	1	[0,10]		<1	1	6	7
1	A 4	60	10	5	2	[0,10]		<1	1	21	22
1	A 5	100	20	6	3	[0,10]		<1	1	38	41

LISTA

		n.subp.			n.part.D			n.part.E			!tamanho da lista	
PR	T	N	medio	max.	min.	medio	max.	min.	medio	max.	medio	max.
1	A	1	2,15	5	1	1,67	2	1	1,73	2	48,09	119
1	A	2	1,83	5	1	1,67	2	2	2,00	2	39,17	117
1	A	3	1,83	5	1	1,67	2	2	2,00	2	39,17	117
1	A	4	1,83	5	1	1,67	2	2	2,00	2	39,17	117
1	A	5	1,83	5	1	1,67	2	2	2,00	2	39,17	117

SUBPROBLEMAS

		cortados					particionados							
PR	T	N	nct	nc0	nc5	nc6	nc7	npt	np1	np2	np3	np4	n.res.	n.ng0
1	A	1	18	0	7	1	10	34	8	10	16	0	52	42
1	A	2	15	0	1	1	13	19	2	12	4	3	36	14
1	A	3	15	0	1	1	13	19	2	12	4	3	36	14
1	A	4	15	0	1	1	13	19	2	129	22	3	171	14
1	A	5	15	0	1	1	13	19	2	259	42	3	321	14

QUALIDADE DAS SOLUCOES

NUMERO DE SUBPROBLEMAS

PR	T	N	i.subp./sol.ot.	i.viav/sol.ot.	i.subp.	i.viav.	otimo	total
1	A	1	13,1725	4,2032	1	5	50	52
1	A	2	13,1725	1,2852	1	1	31	33
1	A	3	13,1725	1,2852	1	1	31	33
1	A	4	13,1725	1,2852	1	1	31	33
1	A	5	13,1725	1,2852	1	1	31	33



PROBL.!		DADOS DE ENTRADA				custo		tempo(seg.) de exec.ate:						
PR!	T!	N!	maior!	menor!	napo!	nasp!	real	int.	i.subp.	i.viav.	otimo	final		
2	A	1	0	0	0	0	[0,1000]		<1	1	1	4		
2	A	2	5	1	4	1	[0,1000]		<1	1	1	3		
2	A	3	100	1	4	1	[0,1000]		<1	1	15	20		
2	A	4	60	10	5	2	[0,1000]		<1	1	10	36		
2	A	5	100	20	6	3	[0,1000]		<1	<1	9	62		
LISTA														
			n.subp.		n.part.D			n.part.E		tamanho da lista				
PR!	T!	N!	medio	max.	min.	medio	max.	min.	medio	max.	medio	max.		
2	A	1	1,58	5	2	2,25	3	2	2,00	2	36,10	124		
2	A	2	1,65	5	2	2,25	3	2	2,00	2	37,93	124		
2	A	3	1,83	5	2	2,29	3	2	2,00	2	41,22	124		
2	A	4	1,83	5	2	2,29	3	2	2,00	2	41,22	124		
2	A	5	1,84	5	2	2,29	3	2,	2,00	2	41,28	124		
SUBPROBLEMAS														
			cortados				particionados							
PR!	T!	N!	nct!	nc0!	nc5!	nc6!	nc7!	npt!	np1!	np2!	np3!	np4!	n.res.	n.ng0!
2	A	1	14	0	2	0	12	15	1	9	2	0	26	10
2	A	2	14	0	3	0	11	16	3	4	4	6	31	12
2	A	3	19	0	3	0	16	23	7	106	5	1	138	18
2	A	4	19	0	3	0	16	23	7	219	14	1	260	18
2	A	5	19	0	5	0	14	25	7	424	24	2	476	21
			QUALIDADE DAS SOLUCOES					NUMERO DE SUBPROBLEMAS						
PR!	T!	N!	i.subp./sol.ot.!			i.viav/sol.ot.!		i.subp.!		i.viav.		otimo	total	
2	A	1	39,3634			1,0000		1		2		2	26	
2	A	2	39,3634			24,9409		1		1		2	27	
2	A	3	39,3634			24,9409		1		1		2	39	
2	A	4	39,3634			24,9409		1		1		2	39	
2	A	5	39,3634			24,9409		1		1		2	41	

PROBL. DADOS DE ENTRADA custo tempo(seg.) de exec.ate:											
PR	T	N	maior	menor	napo	nasp	real	int.	i.subp.	i.viav.	otimo final
3	A	1	0	0	0	0	[0,100]		<1	1	8 9
3	A	2	5	1	4	1	[0,100]		<1	1	6 7
3	A	3	100	1	4	1	[0,100]		<1	15	27 28
3	A	4	60	10	5	2	[0,100]		<1	10	75 76
3	A	5	100	20	6	3	[0,100]		<1	16	138 139
LISTA											
n.subp. n.part.D n.part.E tamanho da lista											
PR	T	N	medio	max.	min.	medio	max.	min.	medio	max.	medio max.
3	A	1	2,26	7	1	1,86	3	1	1,78	2	51,36 189
3	A	2	1,70	4	1	1,56	2	1	1,83	2	35,34 89
3	A	3	2,51	7	1	1,88	3	1	1,80	2	57,77 189
3	A	4	2,51	7	1	1,88	3	1	1,80	2	57,77 189
3	A	5	2,51	7	1	1,88	3	1	1,80	2	57,77 189
SUBPROBLEMAS											
cortados particionados											
PR	T	N	nct	nc0	nc5	nc6	nc7	npt	np1	np2	np3 np4 n.res. n.ng0
3	A	1	22	0	5	0	17	32	4	1	27 0 54 38
3	A	2	13	0	4	0	9	23	4	4	18 0 39 25
3	A	3	29	0	27	0	2	53	5	23	122 1 180 81
3	A	4	29	0	27	0	2	53	6	257	270 4 566 81
3	A	5	29	0	27	0	2	53	7	521	515 4 1076 81
QUALIDADE DAS SOLUCOES NUMERO DE SUBPROBLEMAS											
PR	T	N	i.subp./sol.ot.		i.viav/sol.ot.		i.subp.		i.viav.		otimo total
3	A	1	10,8320		2,5437		1		8		53 54
3	A	2	10,8320		2,7782		1		4		34 35
3	A	3	10,8320		2,7782		1		4		80 81
3	A	4	10,8320		2,8852		1		3		80 81
3	A	5	10,8320		4,3835		1		2		80 81

PROBL.		DADOS DE ENTRADA				custo	tempo(seg.) de exec.ate:				
PR	ITN	maior	menor	napo	naspl	real	int.	i.subp.	i.viav.	otimo	final
4	A 1	0	0	0	0	[0,100]	<1	2	3	4	
4	A 2	5	1	4	1	[0,100]	<1	2	3	4	
4	A 3	100	1	4	1	[0,100]	<1	14	15	17	
4	A 4	60	10	5	2	[0,100]	<1	11	20	27	
4	A 5	100	20	6	3	[0,100]	1	19	23	49	

LISTA

		n.subp.			n.part.D			n.part.E			!tamanho da lista	
PR	TIN	medio	!max.	!min.	medio	!max.	!min.	medio	!max.	medio	!max.	
4	A 1	1,18	3	1	1,00	1	1	1,67	2	22,50	63	
4	A 2	1,03	3	1	1,00	1	1	1,67	2	19,32	59	
4	A 3	1,18	3	1	1,00	1	1	1,67	2	22,50	63	
4	A 4	1,18	3	1	1,00	1	1	1,67	2	22,50	63	
4	A 5	1,18	3	1	1,00	1	1	1,67	2	22,50	63	

SUBPROBLEMAS

		cortados					particionados						
PR	TIN	nct	nc0	nc5	nc6	nc7	npt	np1	np2	np3	np4	n.res.	n.ng0
4	A 1	5	0	3	0	2	16	2	4	8	0	19	26
4	A 2	6	0	0	1	5	14	2	3	10	0	21	21
4	A 3	4	0	2	1	1	16	2	8	104	0	118	26
4	A 4	4	0	2	1	1	16	2	89	109	0	204	26
4	A 5	4	0	3	0	1	16	2	180	198	0	384	26

QUALIDADE DAS SOLUCOES

NUMERO DE SUBPROBLEMAS

PR	TIN	1.subp./sol.ot.	1.viav/sol.ot.	1.subp.	1.viav.	otimo	total
4	A 1	2,1561	1,4104	1	7	11	19
4	A 2	2,1561	1,1214	1	5	10	17
4	A 3	2,1561	1,1214	1	4	12	19
4	A 4	2,1561	1,1214	1	4	12	19
4	A 5	2,1561	1,1214	1	4	5	19



PROBL.		DADOS DE ENTRADA				custo		tempo(seg.) de exec.ate:						
PR	IT	N	maior	menor	napo	nasp	real	int.	i.subp.	i.viav.	otimo	final		
4A	A	1	0	0	0	0	0,100		<1	1	2	3		
4A	A	2	5	1	4	1	0,100		<1	1	2	4		
4A	A	3	100	1	4	1	0,100		<1	14	16	17		
4A	A	4	60	10	5	2	0,100		<1	10	20	26		
4A	A	5	100	20	6	3	0,100		<1	20	25	37		
LISTA														
		n.subp.		n.part.D			n.part.E			tamanho da lista				
PR	IT	N	medio	max.	min.	medio	max.	min.	medio	max.	medio	max.		
4A	A	1	1,18	3	1	1,00	1	1	1,67	2	19,32	59		
4A	A	2	1,03	3	1	1,00	1	1	1,67	2	22,50	63		
4A	A	3	1,18	3	1	1,00	1	1	1,67	2	22,50	63		
4A	A	4	1,18	3	1	1,00	1	1	1,67	2	22,50	63		
4A	A	5	1,18	3	1	1,00	1	1	1,67	2	22,50	63		
SUBPROBLEMAS														
		cortados					particionados							
PR	IT	N	nct	nc0	nc5	nc6	nc7	npt	np1	np2	np3	np4	n.res.	n.ng0
4A	A	1	5	0	3	0	2	16	2	4	8	0	19	26
4A	A	2	7	0	2	1	6	14	2	2	10	0	21	21
4A	A	3	4	0	2	1	1	16	2	8	104	0	118	26
4A	A	4	4	0	2	1	1	16	2	89	109	0	204	26
4A	A	5	4	0	3	0	1	16	2	180	198	0	384	26
		QUALIDADE DAS SOLUCOES					NUMERO DE SUBPROBLEMAS							
PR	IT	N	i.subp./sol.ot.		i.viav/sol.ot.		i.subp.		i.viav.		otimo	total		
4A	A	1	2,1471		1,4028		1		7		11	19		
4A	A	2	2,1471		1,1198		1		5		10	17		
4A	A	3	2,1471		1,1198		1		4		12	19		
4A	A	4	2,1471		1,1198		1		4		12	19		
4A	A	5	2,1471		1,1198		1		4		5	19		

PROBL.		DADOS DE ENTRADA				custo		tempo(seg.) de exec.ate:						
PR	T	N	maior	menor	napo	naspl	real	int.	i.subp.	i.viav.	otimo	final		
5	A	1	0	0	0	0		[0,100]	1	2	8	10		
5	A	2	5	1	4	1		[0,100]	<1	1	12	13		
5	A	3	100	1	4	1		[0,100]	<1	15	28	30		
5	A	4	60	10	5	2		[0,100]	<1	10	80	88		
5	A	5	100	20	6	3		[0,100]	<1	16	149	164		
LISTA														
			n.subp.		n.part.D			n.part.E		tamanho da lista				
PR	T	N	medio	max.	min.	medio	max.	min.	medio	max.	medio	max.		
5	A	1	2,88	7	1	1,90	3	1	1,20	2	99,88	277		
5	A	2	2,57	6	1	1,88	3	1	1,13	2	86,52	229		
5	A	3	2,99	7	1	1,89	3	1	1,11	2	102,90	277		
5	A	4	2,99	7	1	1,89	3	1	1,11	2	102,90	277		
5	A	5	2,99	7	1	1,89	3	1	1,11	2	102,90	277		
SUBPROBLEMAS														
			cortados				particionados							
PR	T	N	nct	nc0	nc5	nc6	nc7	npt	np1	np2	np3	np4	n.res.	n.ng0
5	A	1	16	0	8	0	8	30	7	1	18	0	42	63
5	A	2	20	0	12	0	8	43	6	15	24	0	65	82
5	A	3	21	0	18	0	3	53	7	25	119	0	172	113
5	A	4	21	0	18	0	3	53	7	266	261	4	559	113
5	A	5	21	0	18	0	3	53	7	528	509	4	1069	113
			QUALIDADE DAS SOLUCOES					NUMERO DE SUBPROBLEMAS						
PR	T	N	i.subp./sol.ot.			i.viav/sol.ot.		i.subp.		i.viav.		otimo	total	
5	A	1	12,2000			3,1059		1		8		33	42	
5	A	2	12,2000			1,3636		1		3		52	61	
5	A	3	12,2000			2,1273		1		2		64	73	
5	A	4	12,2000			2,1273		1		2		64	73	
5	A	5	12,2000			2,1273		1		2		64	73	

PROBL.	DADOS DE ENTRADA					custo	tempo(seg.) de exec.ate:					
PR	T	N	maior	menor	napo	nasp	real	int.	i.subp.	i.viav.	otimo	final
5A	A	1	0	0	0	0	[0,100]		1	2	9	11
5A	A	2	5	1	4	1	[0,100]		1	2	13	14
5A	A	3	100	1	4	1	[0,100]		<1	16	30	31
5A	A	4	60	10	5	2	[0,100]		<1	10	84	92
5A	A	5	100	20	6	3	[0,100]		<1	16	156	171

LISTA

PRITIN	n.subp. n.part.D n.part.E tamanho da lista
PRITIN	medio max. min. medio max. min. medio max. medio max.
5A A 1	2,88 7 1 1,90 3 1 1,20 2 99,88 277
5A A 2	2,57 6 1 1,88 3 1 1,13 2 86,52 229
5A A 3	2,99 7 1 1,89 3 1 1,11 2 102,90 277
5A A 4	2,99 7 1 1,89 3 1 1,11 2 102,90 277
5A A 5	2,99 7 1 1,89 3 1 1,11 2 102,90 277

SUBPROBLEMAS

PRITIN	cortados particionados
PRITIN	nct nc0 nc5 nc6 nc7 npt np1 np2 np3 np4 n.res. n.ng0
5A A 1	16 0 8 0 8 30 7 1 18 0 42 63
5A A 2	19 0 12 0 7 43 6 15 25 0 65 82
5A A 3	21 0 18 0 3 53 7 25 119 0 172 113
5A A 4	21 0 18 0 3 53 7 275 261 3 567 113
5A A 5	21 0 18 0 3 53 7 547 509 3 1087 113

QUALIDADE DAS SOLUCOES

NUMERO DE SUBPROBLEMAS

PRITIN	1.subp./sol.ot. i.viav/sol.ot. 1.subp. i.viav. otimo total
5A A 1	11,9861 3,0675 1 8 33 42
5A A 2	11,9861 1,3550 1 3 52 61
5A A 3	11,9861 2,0974 1 2 64 73
5A A 4	11,9861 2,0974 1 2 64 73
5A A 5	11,9861 2,0974 1 2 64 73

PROBL.!		DADOS DE ENTRADA					custo	tempo(seg.) de exec.ate:				

PR	T	N	maior	menor	napo	nasp	real	int.	i.subp.	i.viav.	otimo	final
6	A	1	0	0	0	0		1,1	1	2	2	17
6	A	2	5	1	4	1		1,1	<1	1	5	21
6	A	3	100	1	4	1		1,1	1	2	19	37
6	A	4	60	10	5	2		1,1	1	2	28	105
6	A	5	100	20	6	3		1,1	<1	1	20	394

LISTA

			n.subp.			n.part.D			n.part.E			!tamanho da lista	
PR	T	N	medio	!max.	!min.	medio	!max.	!min.	medio	!max.	!min.	medio	!max.
6	A	1	2,77	7	1	1,82	3	1	2,04	3		99,23	267
6	A	2	3,34	8	1	1,94	3	1	2,24	3		119,99	302
6	A	3	3,42	8	1	1,94	3	1	2,27	3		123,83	302
6	A	4	3,42	8	1	1,94	3	1	2,27	3		123,83	302
6	A	5	3,42	8	1	1,94	3	1	2,27	3		123,83	302

SUBPROBLEMAS

			cortados					particionados						
PR	T	N	nct	nc0	nc5	nc6	nc7	npt	np1	np2	np3	np4	n.res.	n.ng0
6	A	1	43	0	20	0	23	46	1	7	38	0	89	65
6	A	2	60	0	25	1	34	52	1	35	16	1	113	72
6	A	3	65	0	38	1	26	56	2	91	60	0	218	77
6	A	4	65	0	38	1	26	56	2	430	158	0	655	77
6	A	5	65	0	34	0	31	56	3	859	298	0	1225	77

QUALIDADE DAS SOLUCOES

NUMERO DE SUBPROBLEMAS

PR	T	N	1.subp./sol.ot.	1.viav/sol.ot.	1.subp.	1.viav.	otimo	total
6	A	1	5,0000	1,0000	1	5	5	89
6	A	2	5,0000	1,2500	1	2	18	109
6	A	3	5,0000	1,5000	1	1	18	119
6	A	4	5,0000	1,5000	1	1	18	119
6	A	5	5,0000	1,5000	1	1	3	119

PROBL.!		DADOS DE ENTRADA				custo		tempo(seg.) de exec.ate:						
PR!	T!	N!	maior	menor	napo	nasp	real	int.	i.subp.	i.viav.	otimo	final		
7	A	1	0	0	0	0	[0,100]	1	3	19	30			
7	A	2	5	1	4	1	[0,100]	<1	1	27	60			
7	A	3	100	1	4	1	[0,100]	<1	1	9	41			
7	A	4	60	10	5	2	[0,100]	<1	1	16	152			
7	A	5	100	20	6	3	[0,100]	<1	1	22	274			
LISTA														
		n.subp.			n.part.D			n.part.E			!tamanho da lista			
PR!	T!	N!	medio	max.	min.	medio	max.	min.	medio	max.	medio	max.		
7	A	1	4,85	11	1	1,76	3	1	2,48	3	363,43	952		
7	A	2	5,42	12	1	1,92	3	1	2,24	3	398,56	954		
7	A	3	5,00	12	1	1,86	3	1	2,16	3	354,27	954		
7	A	4	5,00	12	1	1,86	3	1	2,16	3	354,27	954		
7	A	5	5,00	12	1	1,86	3	1	2,16	3	354,27	954		
SUBPROBLEMAS														
		cortados					particionados							
PR!	T!	N!	nct	nc0	nc5	nc6	nc7	npt	np1	np2	np3	np4	n.res.	n.ng0
7	A	1	52	0	4	0	48	43	4	5	30	0	91	50
7	A	2	103	0	44	0	59	100	5	73	21	0	202	162
7	A	3	62	0	34	0	28	62	5	63	14	1	145	93
7	A	4	62	0	34	0	28	62	5	477	104	1	649	93
7	A	5	62	0	34	0	28	62	5	931	210	1	1209	93
QUALIDADE DAS SOLUCOES													NUMERO DE SUBPROBLEMAS	
PR!	T!	N!	1.subp./sol.ot.			1.viav/sol.ot.		1.subp.		1.viav.		otimo	total	
7	A	1	37,0455			1,6667		1		6		51	91	
7	A	2	37,0455			2,7879		1		1		27	60	
7	A	3	37,0455			2,7879		1		1		8	119	
7	A	4	37,0455			2,7879		1		1		7	119	
7	A	5	37,0455			2,7879		1		1		7	119	

PROBL.!		DADOS DE ENTRADA					custo		tempo(seg.) de exec.ate:			
PR	T	N	maior	menor	napo	nasp	real	int.	i.subp.	i.viav.	otimo	final
8	A	1	0	0	0	0	[0,100]		<1	3	22	30
8	A	2	5	1	4	1	[0,100]		<1	1	22	30
8	A	5	100	20	6	3	[0,100]		<1	1	104	209

LISTA

n.subp.			n.part.D			n.part.E			tamanho da lista			
PR	T	N	medio	max.	min.	medio	max.	min.	medio	max.	medio	max.
8	A	1	4,36	12	1	2,15	4	1	2,47	3	361,94	1110
8	A	2	4,36	12	1	2,15	4	1	2,47	3	361,94	1110
8	A	5	4,20	10	1	2,18	3	1	2,45	4	328,45	825

SUBPROBLEMAS

cortados							particionados							
PR	T	N	nct	nc0	nc5	nc6	nc7	npt	np1	np2	np3	np4	n.res.	n.ng0
8	A	1	51	0	3	1	47	38	2	5	29	0	87	26
8	A	2	51	0	3	1	47	38	3	1	32	4	91	26
8	A	5	56	0	17	1	38	40	7	695	53	1	812	31

QUALIDADE DAS SOLUCOES

NUMERO DE SUBPROBLEMAS

PR	T	N	1.subp./sol.ot.	1.viav/sol.ot.	1.subp.	1.viav.	otimo	total
8	A	1	45,9292	1,4820	1	6	60	87
8	A	2	45,9292	6,6332	1	1	60	87
8	A	5	45,9292	6,6332	1	1	5	67

PROBL.!		DADOS DE ENTRADA					custo		tempo(seg.) de exec.ate:			
PR	T	N	maior	menor	napo	nasp	real	int.	i.subp.	i.viav.	otimo	final
9	A	1	0	0	0	0	[0,100]		1	3	19	35
9	A	2	5	1	4	1	[0,100]		<1	3	21	38
9	A	3	100	1	4	1	[0,100]		<1	17	78	139
9	A	4	60	10	5	2	[0,100]		<1	16	301	624
9	A	5	100	20	6	3	[0,100]		<1	18	791	1557

LISTA

			n.subp.			n.part.D			n.part.E			!tamanho da lista	
PR	T	N	medio	max.	min.	medio	max.	min.	medio	max.	medio	max.	
9	A	1	3,91	11	1	1,75	4	1	1,81	3	97,22	317	
9	A	2	4,22	11	1	1,83	4	1	1,85	3	106,09	317	
9	A	3	4,67	11	1	1,46	4	1	1,95	3	118,46	317	
9	A	4	4,41	11	1	1,53	4	1	1,91	3	110,96	317	
9	A	5	4,67	11	1	1,46	4	1	1,95	3	118,35	317	

SUBPROBLEMAS

			cortados					particionados						
PR	T	N	nct	nc0	nc5	nc6	nc7	npt	np1	np2	np3	np4	n.res.	n.ng0
9	A	1	57	0	8	0	49	138	5	31	99	0	192	373
9	A	2	62	0	5	0	57	145	5	45	95	1	208	400
9	A	3	129	0	83	1	45	571	4	428	234	3	798	1927
9	A	4	114	0	43	1	70	433	4	3094	1244	3	4459	1403
9	A	5	129	0	83	1	45	570	5	8687	2710	3	11534	1923

QUALIDADE DAS SOLUCOES

NUMERO DE SUBPROBLEMAS

PR	T	N	1.subp./sol.ot.	1.viav/sol.ot.	1.subp.	1.viav.	otimo	total
9	A	1	2,4049	1,2890	1	14	97	192
9	A	2	2,4049	1,2890	1	14	107	204
9	A	3	2,4049	1,2491	1	6	348	699
9	A	4	2,4049	1,2481	1	6	257	546
9	A	5	2,4049	1,3858	1	2	347	698

PROBL.		DADOS DE ENTRADA					custo	tempo(seg.) de exec.ate:						
PR	T	N	maior	menor	napo	nasp	real	int.	i.subp.	i.viav.	otimo	final		
10	A	1	0	0	0	0	0,1000		<1	1	17	28		
10	A	2	5	1	4	1	0,1000		<1	1	1	17		
10	A	3	100	1	4	1	0,1000		<1	1	1	35		
10	A	4	60	10	5	2	0,1000		<1	1	1	77		
10	A	5	100	20	6	3	0,1000		<1	<1	1	139		
LISTA														
			n.subp.		n.part.D			n.part.E		tamanho da lista				
PR	T	N	medio	max.	min.	medio	max.	min.	medio	max.	medio	max.		
10	A	1	3,93	10	1	1,94	4	1	2,55	3	124,99	335		
10	A	2	2,95	7	1	1,85	3	1	2,50	3	91,98	247		
10	A	3	2,95	7	1	1,85	3	1	2,50	3	91,98	247		
10	A	4	2,95	7	1	1,85	3	1	2,50	3	91,98	247		
10	A	5	2,95	7	1	1,85	3	1	2,50	3	91,98	247		
SUBPROBLEMAS														
			cortados					particionados						
PR	T	N	nct	nc0	nc5	nc6	nc7	npt	np1	np2	np3	np4	n.res.	n.ng0
10	A	1	58	0	6	0	52	87	3	55	25	0	141	142
10	A	2	41	0	3	0	38	47	1	29	8	5	84	58
10	A	3	41	0	3	0	38	47	1	124	8	5	179	58
10	A	4	41	0	3	0	38	47	1	327	62	5	436	58
10	A	5	41	0	3	0	38	47	1	637	122	5	806	58
			QUALIDADE DAS SOLUCOES					NUMERO DE SUBPROBLEMAS						
PR	T	N	i.subp./sol.ot.		i.viav/sol.ot.		i.subp.		i.viav.		otimo	total		
10	A	1	13,5513		1,7432		1		3		84	141		
10	A	2	13,5513		1,0000		1		1		1	80		
10	A	3	13,5513		1,0000		1		1		1	80		
10	A	4	13,5513		1,0000		1		1		1	80		
10	A	5	13,5513		1,0000		1		1		1	80		

PROBL.!		DADOS DE ENTRADA				custo		tempo(seg.) de exec.ate:						
PR!	T!	N!	maior!	menor!	napo!	nasp!	real	int.	i.subp.	i.viav.	otimo	final		
11	A	1	0	0	0	0	-10,100		<1	4	278	290		
11	A	2	5	1	4	1	-10,100		<1	3	303	312		
11	A	3	100	1	4	1	-10,100		<1	21	1985	2061		
11	A	4	60	10	5	2	-10,100		<1	18	10825	11325		
LISTA														
		n.subp.		n.part.D			n.part.E			tamanho da lista				
PR!	T!	N!	medio	max.	min.	medio	max.	min.	medio	max.	medio	max.		
11	A	1	7,64	17	1	1,93	4	1	2,23	4	323,25	788		
11	A	2	8,08	16	1	1,99	4	1	2,31	4	344,18	720		
11	A	3	8,45	18	1	1,88	4	1	2,18	4	360,92	886		
11	A	4	8,45	18	1	1,88	4	1	2,18	4	360,87	886		
SUBPROBLEMAS														
		cortados					particionados							
PR!	T!	N!	nct	nc0	nc5	nc6	nc7	npt	np1	np2	np3	np4	n.res.	n.ng0
11	A	1	539	0	28	1	510	836	19	72	739	0	1369	2423
11	A	2	559	0	18	0	541	889	15	333	537	0	1444	2699
11	A	3	2998	0	2998	0	0	6579	14	4819	1817	28	9676	23541
11	A	4	2998	0	2995	0	3	6576	6	48895	16891	2	68792	23527
QUALIDADE DAS SOLUCOES													NUMERO DE SUBPROBLEMAS	
PR!	T!	N!	i.subp./sol.ot.		i.viav/sol.ot.		i.subp.		i.viav.		otimo	total		
11	A	1	6,4119		2,3687		1		16		1307	1369		
11	A	2	6,4119		1,9639		1		7		1393	1440		
11	A	3	6,4119		1,9639		1		7		9204	9577		
11	A	4	6,4119		2,0502		1		5		9201	9574		



PROBL.			DADOS DE ENTRADA				custo		tempo(seg.) de exec.ate:					
PR	T	N	maior	menor	napo	nasp	real	int.	i.subp.	i.viav.	otimo	final		
12	A	1	0	0	0	0	[0,100]		1	6	22	78		
12	A	2	5	1	4	1	[0,100]		<1	1	24	80		
12	A	3	100	1	4	1	[0,100]		<1	1	255	295		
12	A	4	60	10	5	2	[0,100]		<1	1	746	959		
12	A	5	100	20	6	3	[0,100]		<1	1	1381	1647		
LISTA														
			n.subp.		n.part.D			n.part.E		!tamanho da lista				
PR	T	N	medio	max.	min.	medio	max.	min.	medio	max.	medio	max.		
12	A	1	6,64	17	1	2,71	5	2	3,43	5	1075,15	3092		
12	A	2	6,64	17	1	2,71	5	2	3,43	5	1075,15	3092		
12	A	3	6,25	16	1	2,77	5	1	2,74	5	948,95	2826		
12	A	4	6,42	16	1	1,76	5	1	2,81	5	979,22	2826		
12	A	5	6,25	16	1	2,77	5	1	2,73	5	949,43	2826		
SUBPROBLEMAS														
cortados						particionados								
PR	T	N	nct	nc0	nc5	nc6	nc7	npt	np1	np2	np3	np4	n.res.	n.ng0
12	A	1	96	0	2	5	89	46	0	0	43	0	139	14
12	A	2	96	0	2	5	89	46	3	1	43	0	143	14
12	A	3	314	0	147	5	162	178	27	69	176	0	586	98
12	A	4	333	0	156	4	173	184	28	234	1558	0	2153	102
12	A	5	316	0	149	3	164	179	29	493	3018	0	3856	99
QUALIDADE DAS SOLUCOES								NUMERO DE SUBPROBLEMAS						
PR	T	N	1.subp./sol.ot.			1.viav/sol.ot.		1.subp.	1.viav.	otimo	total			
12	A	1	223,1524			2,5000		1	8	31	139			
12	A	2	223,1524			70,1375		1	1	31	139			
12	A	3	223,1524			70,1375		1	1	404	487			
12	A	4	223,1524			70,1375		1	1	427	510			
12	A	5	223,1524			70,1375		1	1	406	489			

O número da execução (N) varia com as mudanças dos dados de entrada: maior, menor, napo e nasp.

Observando os resultados apresentados para as execuções do tipo A realizadas, verifica-se:

PR	N	(% sol.ótima/t. subp.)	(% sol.ótima/t. viav.)	nº de sol.viáveis
1	1	7,59%	23,79%	9
	2	"	77,80%	3
	3	"	"	"
	4	"	"	"
	5	"	"	"
2	1	2,54%	100,00%	1
	2	"	4,00%	3
	3	"	"	7
	4	"	"	7
	5	"	"	"
3	1	9,23%	39,31%	4
	2	"	35,99%	"
	3	"	"	5
	4	"	34,66%	6
	5	"	22,81%	7
4	1	46,38%	70,90%	2
	2	"	89,17%	3
	3	"	"	"
	4	"	"	"
	5	"	"	2
4A	1	46,57%	71,29%	2
	2	"	89,30%	3
	3	"	"	"
	4	"	"	"
	5	"	"	2
5	1	8,20%	32,20%	7
	2	"	73,33%	6
	3	"	47,00%	7
	4	"	"	"
	5	"	"	"

PR	N	(% sol.ótima/ 1.subp.)	(% sol.ótima/ 1.viav.)	nº de sol.viáveis
5A	1	8,34%	32,60%	7
	2	"	73,33%	6
	3	"	47,68%	7
	4	"	"	"
	5	"	"	"
6	1	20,00%	100,00%	1
	2	"	80,00%	2
	3	"	66,67%	3
	4	"	"	"
	5	"	"	"
7	1	2,70%	60,00%	4
	2	"	35,87%	5
	3	"	"	"
	4	"	"	"
	5	"	"	6
8	1	2,18%	67,47%	3
	2	"	15,07%	4
	5	"	"	8
9	1	41,58%	77,58%	5
	2	"	"	"
	3	"	80,12%	"
	4	"	"	"
	5	"	72,16%	6
10	1	37,07%	57,36%	1
	2	"	100,00%	1
	3	"	"	"
	4	"	"	"
	5	"	"	"
11	1	15,60%	42,22%	20
	2	"	50,92%	15
	3	"	"	14
	4	"	48,78%	6
12	1	0,45%	39,90%	5
	2	"	1,42%	8
	3	"	"	32
	4	"	"	"
	5	"	"	"

As execuções de número 1 não utilizam o método de otimização subgradiente, pois nesses casos os dados de entrada são $(\text{maior}, \text{menor}, \text{napo}, \text{nasp}) = (0, 0, 0, 0)$.

Uma redução do número de problemas examinados é observada quando as execuções realizadas apresentam como dados de entrada, em geral, $(\text{maior}, \text{menor}, \text{napo}, \text{nasp}) = (60, 10, 5, 2)$ ou $(100, 80, 6, 3)$. Essa redução é vista comparativamente com aquelas execuções de número 1.

Observa-se, também, em todos os problemas analisados que o tempo gasto para a obtenção da 1ª solução viável é menor quando $N = 2$ (PROBLEMAS: 1, 2, 3, 4, 4A, 5, 5A, 6, 7, 8, 10, 11 e 12), $N = 3$ (PROBLEMAS: 1, 2, 6, 7, 10, e 12), $N = 4$ (PROBLEMAS: 1, 2, 7, 10 e 12), $N = 5$ (PROBLEMAS: 1, 2, 6, 7, 8, 10 e 12).

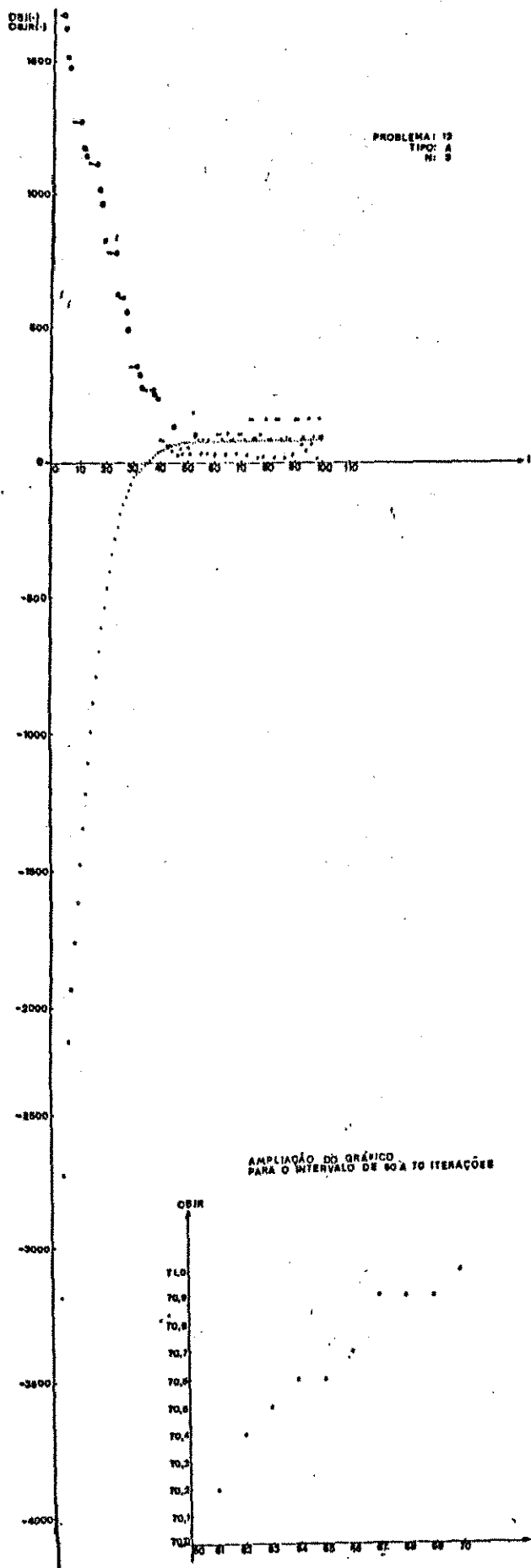
O tempo gasto para a obtenção da solução ótima é menor quando $N = 2$ (PROBLEMAS: 1, 2, 3, 4, 4A, 8), $N = 3$ (PROBLEMAS: 7 e 10).

O valor da solução do 1º subproblema é dado pela soma dos custos das colunas e está associado a uma solução onde todos os elementos (representados pelas colunas) estão presentes.

A seguir é apresentado um gráfico do Valor da solução (obj.) e do valor relaxado (objr.) em função do número de iterações (I), para o problema 12, execução tipo A e número da execução $N = 3$.

Os resultados apresentados no gráfico referem-se aos dados de entrada ($\text{maior} = 100$ e $\text{napo} = 4$) e correspondem as iterações efetuadas para o 1º subproblema examinado na resolução do problema 12.

As soluções viáveis obtidas para o problema 12 nessas iterações encontram-se representadas por um círculo em torno do valor de obj($\otimes$).



OBJ(*)
OBJR(-)

1500

1000

500

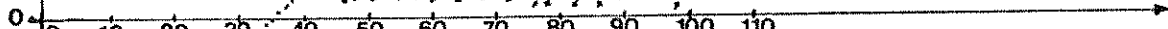
0

-500

-1000

PROBLEMA: 12
TIPO: A
N: 3

0 10 20 30 40 50 60 70 80 90 100 110



-1500

-2000

-2500

-3000

-3500

-4000

AMPLIAÇÃO DO GRÁFICO
PARA O INTERVALO DE 60 A 70 ITERAÇÕES

OBJR

71,0

70,9

70,8

70,7

70,6

70,5

70,4

70,3

70,2

70,1

70,0

60

61

62

63

64

65

66

67

68

69

70

A seguir são apresentados alguns resultados obtidos da execução tipo B que, ao contrário da execução tipo A, realiza uma atualização periódica do sobreestimador utilizado no cálculo do passo (vide capítulo 2).

Nesse caso, o sobreestimador é obtido fazendo uma combinação convexa ($\beta \cdot \text{obj0} + \delta \cdot \text{objr}$), onde β é igual ao erro relativo cometido no cálculo do vetor de multiplicadores de Lagrange para cada iteração realizada pelo algoritmo.

$$\text{Dessa forma } \beta = \frac{\|\lambda - \lambda^i\|}{\|\lambda\|} \quad \text{e } (0 \leq \beta \leq 1)$$

$\delta = (1 - \beta)$ e então ($0 \leq \delta \leq 1$). Considerando obji como sendo a melhor solução presente e objr como a solução obtida para o subproblema (problema relaxado), observa-se que se objr é maior que obji , então $\lambda^i = \lambda$ e decorre $\beta = 0$ e $\delta = 1$ e todo o peso no cálculo do sobreestimador recai sobre objr .

Por outro lado, se objr é menor ou igual que obji tem-se que $\lambda^i \neq \lambda$ e então o maior peso na combinação convexa que fornece o valor do sobreestimador será de obj0 ou de objr , dependendo de λ^i estar distante ou não de λ , respectivamente.

Procurou-se, nesse tipo de execução, evitar que o resultado da combinação convexa fosse um subestimador para a função $w(\lambda^*)$.

O valor obj0 utilizado nesse processo é dado pela soma dos custos de todos os elementos representados pelas colunas da matriz do problema.

PROBL. ! DADOS DE ENTRADA ! custo ! tempo(seg.) de exec.ate:											
PR ! T ! N !	maior !	menor !	napo !	nasp !	real !	int. !	i.subp. !	i.viav. !	otimo !	final !	
1 ! B ! 2 !	5 !	1 !	4 !	1 !	[0,10] !		<1 !	1 !	6 !	7 !	
1 ! B ! 3 !	100 !	1 !	4 !	1 !	[0,10] !		<1 !	1 !	6 !	7 !	
1 ! B ! 4 !	60 !	10 !	5 !	2 !	[0,10] !		<1 !	1 !	19 !	20 !	
1 ! B ! 5 !	100 !	20 !	6 !	3 !	[0,10] !		<1 !	1 !	15 !	18 !	
LISTA											
n.subp. ! n.part.D ! n.part.E ! tamanho da lista											
PR ! T ! N !	medio !	max. !	min. !	medio !	max. !	min. !	medio !	max. !	medio !	max. !	
1 ! B ! 2 !	1,83 !	5 !	1 !	1,67 !	2 !	2 !	2,00 !	2 !	39,17 !	117 !	
1 ! B ! 3 !	1,83 !	5 !	1 !	1,67 !	2 !	2 !	2,00 !	2 !	39,17 !	117 !	
1 ! B ! 4 !	1,85 !	5 !	1 !	1,80 !	2 !	2 !	2,00 !	2 !	39,73 !	117 !	
1 ! B ! 5 !	1,11 !	3 !	1 !	1,67 !	2 !	2 !	2,00 !	2 !	23,75 !	68 !	
SUBPROBLEMAS											
cortados ! particionados !											
PR ! T ! N !	nct !	nc0 !	nc5 !	nc6 !	nc7 !	npt !	np1 !	np2 !	np3 !	np4 !	n.res. ! n.ng0 !
1 ! B ! 2 !	15 !	0 !	1 !	1 !	13 !	19 !	2 !	12 !	2 !	5 !	36 ! 14 !
1 ! B ! 3 !	15 !	0 !	1 !	1 !	13 !	19 !	2 !	12 !	2 !	5 !	36 ! 14 !
1 ! B ! 4 !	15 !	0 !	1 !	1 !	13 !	17 !	2 !	117 !	2 !	5 !	141 ! 11 !
1 ! B ! 5 !	9 !	0 !	1 !	1 !	7 !	10 !	2 !	106 !	2 !	5 !	124 ! 4 !
QUALIDADE DAS SOLUCOES ! NUMERO DE SUBPROBLEMAS											
PR ! T ! N !	i.subp./sol.ot. !	i.viav/sol.ot. !	i.subp. !	i.viav. !	otimo !	total !					
1 ! B ! 2 !	13,1725 !	1,2852 !	1 !	1 !	31 !	33 !					
1 ! B ! 3 !	13,1725 !	1,2852 !	1 !	1 !	31 !	33 !					
1 ! B ! 4 !	13,1725 !	1,2852 !	1 !	1 !	29 !	31 !					
1 ! B ! 5 !	13,1725 !	1,2851 !	1 !	1 !	15 !	18 !					

PROBL.	DADOS DE ENTRADA	custo	tempo(seg.) de exec.	ate:
PR T N	maior menor napo nasp real int.	1.subp.	1.viav.	otimo final
2 B 2	5 1 4 1 0,1000	<1	1	1 5
2 B 3	100 1 4 1 0,1000	<1	1	16 21
2 B 4	60 10 5 2 0,1000	<1	1	11 24
2 B 5	100 20 6 3 0,1000	<1	1	17 38

LISTA

n.subp.	n.part.D	n.part.E	tamanho da lista
PR T N	medio max. min.	medio max. min.	medio max.
2 B 2	1,65 5 2	2,25 3 2	2,00 2 37,93 124
2 B 3	1,83 5 2	2,29 3 2	2,00 2 41,22 124
2 B 4	1,83 5 2	2,29 3 2	2,00 2 41,22 124
2 B 5	1,83 5 2	2,29 3 2	2,00 2 41,22 124

SUBPROBLEMAS

cortados	particionados
PR T N	nct nc0 nc5 nc6 nc7 npt np1 np2 np3 np4 n.res. n.ng0
2 B 2	14 0 3 0 11 16 3 0 3 11 31 12
2 B 3	19 0 3 0 16 23 7 95 4 13 138 18
2 B 4	19 0 3 0 16 23 7 109 4 13 152 18
2 B 5	19 0 3 0 16 23 7 209 4 13 252 18

QUALIDADE DAS SOLUCOES

NUMERO DE SUBPROBLEMAS

1.subp./sol.ot.	1.viav/sol.ot.	1.subp.	1.viav.	otimo	total
2 B 2	39,3634	24,9409	1	1	2 27
2 B 3	39,3634	24,9409	1	1	2 39
2 B 4	39,3634	24,9409	1	1	2 39
2 B 5	39,3634	24,9409	1	1	2 39

PROBL. ! DADOS DE ENTRADA ! custo ! tempo(seg.) de exec.ate: !											
PR !	T !	N !	maior !	menor !	napo !	nasp !	real !	int. !	i.subp. !	i.viav. !	otimo ! final !
3 !	B !	2 !	5 !	1 !	4 !	1 !	[0,100] !	! !	<1 !	2 !	6 ! 7 !
3 !	B !	3 !	100 !	1 !	4 !	1 !	[0,100] !	! !	<1 !	16 !	28 ! 29 !
3 !	B !	4 !	60 !	10 !	5 !	2 !	[0,100] !	! !	1 !	11 !	30 ! 31 !
3 !	B !	5 !	100 !	20 !	6 !	3 !	[0,100] !	! !	<1 !	17 !	44 ! 45 !
LISTA											
n.subp. ! n.part.D ! n.part.E ! tamanho da lista !											
PR !	T !	N !	medio !	max. !	min. !	medio !	max. !	min. !	medio !	max. !	medio ! max. !
3 !	B !	2 !	1,70 !	4 !	1 !	1,56 !	2 !	1 !	1,83 !	2 !	35,34 ! 89 !
3 !	B !	3 !	2,51 !	7 !	1 !	1,88 !	3 !	1 !	1,80 !	2 !	57,77 ! 189 !
3 !	B !	4 !	2,51 !	7 !	1 !	1,88 !	3 !	1 !	1,80 !	2 !	57,77 ! 189 !
3 !	B !	5 !	2,51 !	7 !	1 !	1,88 !	3 !	1 !	1,80 !	2 !	57,77 ! 189 !
SUBPROBLEMAS											
cortados ! particionados !											
PR !	T !	N !	nct !	nc0 !	nc5 !	nc6 !	nc7 !	npt !	np1 !	np2 !	np3 ! np4 ! n.res. ! n.ng0 !
3 !	B !	2 !	13 !	0 !	4 !	0 !	9 !	23 !	4 !	4 !	13 ! 5 ! 39 ! 25 !
3 !	B !	3 !	29 !	0 !	27 !	0 !	2 !	53 !	5 !	0 !	106 ! 40 ! 180 ! 81 !
3 !	B !	4 !	29 !	0 !	27 !	0 !	2 !	53 !	6 !	17 !	100 ! 45 ! 197 ! 81 !
3 !	B !	5 !	29 !	0 !	27 !	0 !	2 !	53 !	7 !	42 !	174 ! 45 ! 297 ! 81 !
QUALIDADE DAS SOLUCOES ! NUMERO DE SUBPROBLEMAS											
PR !	T !	N !	1.subp./sol.ot. !		1.viav/sol.ot. !		1.subp. !		1.viav. !		otimo ! total !
3 !	B !	2 !	10,8320 !		2,7782 !		1 !		4 !		34 ! 35 !
3 !	B !	3 !	10,8320 !		2,7782 !		1 !		4 !		80 ! 81 !
3 !	B !	4 !	10,8320 !		2,8852 !		1 !		3 !		80 ! 81 !
3 !	B !	5 !	10,8320 !		4,3835 !		1 !		2 !		80 ! 81 !

PROBL.!		DADOS DE ENTRADA				custo		tempo(seg.) de exec.ate:				
PR!	T!	N!	maior	menor	n.p.o	n.a.s	real	int.	i.subp.	i.viav.	otimo	final
4	B	2	5	1	4	1		[0,100]	<1	2	3	4
4	B	3	100	1	4	1		[0,100]	<1	14	16	17
4	B	4	60	10	5	2		[0,100]	<1	12	20	22
4	B	5	100	20	6	3		[0,100]	<1	19	24	37

LISTA

			n.subp.			n.part.D			n.part.E			tamanho da lista	
PR	T	N	medio	max.	min.	medio	max.	min.	medio	max.	medio	max.	
4	B	2	1,03	3	1	1,00	1	1	1,67	2	19,32	59	
4	B	3	1,18	3	1	1,00	1	1	1,67	2	22,50	63	
4	B	4	1,18	3	1	1,00	1	1	1,67	2	22,50	63	
4	B	5	1,18	3	1	1,00	1	1	1,67	2	22,50	63	

SUBPROBLEMAS

		cortados					particionados							
PR	T	N	nct	nc0	nc5	nc6	nc7	npt	np1	np2	np3	np4	n.res.	n.ng0
4	B	2	6	0	0	1	5	14	2	1	10	2	21	21
4	B	3	4	0	2	1	1	16	2	3	103	6	118	26
4	B	4	4	0	2	1	1	16	2	39	99	6	150	26
4	B	5	4	0	3	0	1	16	2	80	178	6	270	26

QUALIDADE DAS SOLUCOES

NUMERO DE SUBPROBLEMAS

PR.T.N		1.subp./sol.ot.	1.viav/sol.ot.	1.subp.	1.viav.	otimo	total
4	B 2	2,1561	1,1214	1	5	10	17
4	B 3	2,1561	1,1214	1	4	12	19
4	B 4	2,1561	1,1214	1	4	12	19
4	B 5	2,1561	1,1214	1	4	5	19

PROBL.		DADOS DE ENTRADA					custo		tempo(seg.) de exec.ate:					
PR	T	N	maior	menor	napo	nasp	real	int.	i.subp.	i.viav.	otimo	final		
4A	B	2	5	1	4	1	[0,100]		<1	2	3	4		
4A	B	3	100	1	4	1	[0,100]		1	15	17	18		
4A	B	4	60	10	5	2	[0,100]		<1	12	20	21		
4A	B	5	100	20	6	3	[0,100]		<1	20	25	37		
LISTA														
			n.subp.		n.part.D			n.part.E		tamanho da lista				
PR	T	N	medio	max.	min.	medio	max.	min.	medio	max.	medio	max.		
4A	B	2	1,03	3	1	1,00	1	1	1,67	2	19,32	59		
4A	B	3	1,18	3	1	1,00	1	1	1,67	2	22,50	63		
4A	B	4	1,18	3	1	1,00	1	1	1,67	2	22,50	63		
4A	B	5	1,18	3	1	1,00	1	1	1,67	2	22,50	63		
SUBPROBLEMAS														
			cortados					particionados						
PR	T	N	nct	nc0	nc5	nc6	nc7	npt	np1	np2	np3	np4	n.res.	n.ng0
4A	B	2	7	0	0	1	6	14	2	1	10	1	21	21
4A	B	3	4	0	2	1	1	16	2	4	103	5	118	26
4A	B	4	4	0	2	1	1	16	2	39	99	6	150	26
4A	B	5	4	0	3	0	1	16	2	80	178	6	270	26
QUALIDADE DAS SOLUCOES														
NUMERO DE SUBPROBLEMAS														
PR	T	N	1.subp./sol.ot.		1.viav/sol.ot.		1.subp.		1.viav.		otimo	total		
4A	B	2	2,1471		1,1198		1		5		10	17		
4A	B	3	2,1471		1,1198		1		4		12	19		
4A	B	4	2,1471		1,1198		1		4		12	19		
4A	B	5	2,1471		1,1198		1		4		5	19		

PROBL.		DADOS DE ENTRADA					custo		tempo(seg.) de exec.ate:					
PR	T	N	maior	menor	napo	nasp	real	int.	i.subp.	i.viav.	otimo	final		
5	B	2	5	1	4	1		[0,100]	1	2	12	14		
5	B	3	100	1	4	1		[0,100]	<1	16	30	31		
5	B	4	60	10	5	2		[0,100]	<1	10	40	42		
5	B	5	100	20	6	3		[0,100]	<1	16	63	65		
LISTA														
			n.subp.		n.part.D			n.part.E		tamanho da lista				
PR	T	N	medio	max.	min.	medio	max.	min.	medio	max.	medio	max.		
5	B	2	2,57	6	1	1,88	3	1	1,13	2	86,52	229		
5	B	3	2,99	7	1	1,89	3	1	1,11	2	102,90	277		
5	B	4	2,99	7	1	1,89	3	1	1,11	2	102,90	277		
5	B	5	2,99	7	1	1,89	3	1	1,11	2	102,90	277		
SUBPROBLEMAS														
			cortados				particionados							
PR	T	N	nct	nc0	nc5	nc6	nc7	npt	np1	np2	np3	np4	n.res.	n.ng0
5	B	2	20	0	12	0	8	43	6	4	11	24	65	82
5	B	3	21	0	18	0	3	53	7	2	106	36	172	113
5	B	4	21	0	18	0	3	53	7	36	131	40	235	113
5	B	5	21	0	18	0	3	53	7	68	249	40	385	113
QUALIDADE DAS SOLUCOES														
			i.subp./sol.ot.				i.viav/sol.ot.				NUMERO DE SUBPROBLEMAS			
PR	T	N	i.subp./sol.ot.				i.viav/sol.ot.				i.subp.	i.viav.	otimo	total
5	B	2	12,2000				1,3636				1	3	52	61
5	B	3	12,2000				2,1273				1	2	64	73
5	B	4	12,2000				2,1273				1	2	64	73
5	B	5	12,2000				2,1273				1	2	64	73

PROBL.		DADOS DE ENTRADA					custo	tempo(seg.) de exec.ate:				
PR	T	N	maior	menor	napo	nasp	real	int.	i.subp.	i.viav.	otimo	final
5A	B	2	5	1	4	1	[0,100]		<1	2	12	14
5A	B	3	100	1	4	1	[0,100]		<1	16	30	32
5A	B	4	60	10	5	2	[0,100]		<1	10	45	47
5A	B	5	100	20	6	3	[0,100]		1	17	76	78

LISTA

			n.subp.			n.part.D			n.part.E			tamanho da lista	
PR	T	N	medio	max.	min.	medio	max.	min.	medio	max.	medio	max.	
5A	B	2	2,57	6	1	1,88	3	1	1,13	2	86,52	229	
5A	B	3	2,99	7	1	1,89	3	1	1,11	2	102,90	277	
5A	B	4	2,99	7	1	1,89	3	1	1,11	2	102,90	277	
5A	B	5	2,99	7	1	1,89	3	1	1,11	2	102,90	277	

SUBPROBLEMAS

			cortados					particionados						
PR	T	N	nct	nc0	nc5	nc6	nc7	npt	np1	np2	np3	np4	n.res.	n.ng0
5A	B	2	19	0	12	0	7	43	6	4	11	25	65	82
5A	B	3	21	0	18	0	3	53	7	2	108	34	172	113
5A	B	4	21	0	18	0	3	53	7	45	151	37	261	113
5A	B	5	21	0	18	0	3	53	7	87	289	37	441	113

QUALIDADE DAS SOLUCOES

NUMERO DE SUBPROBLEMAS

PR	T	N	i.subp./sol.ot.	i.viav/sol.ot.	i.subp.	i.viav.	otimo	total
5A	B	2	11,9861	1,3550	1	3	52	61
5A	B	3	11,9861	2,0974	1	2	64	73
5A	B	4	11,9861	2,0974	1	2	64	73
5A	B	5	11,9861	2,0974	1	2	64	73

PROBL. ! DADOS DE ENTRADA ! custo ! tempo(seg.) de exec.ate:														
PR !	T !	N !	maior !	menor !	napo !	naspl !	real !	int. !	i.subp. !	i.viav. !	otimo ! final			
6	B	2	5	1	4	1		(1,1)	<1	1	4 21			
6	B	3	100	1	4	1		(1,1)	1	2	20 39			
6	B	4	60	10	5	2		(1,1)	<1	1	29 104			
6	B	5	100	20	6	3		(1,1)	1	2	22 154			
LISTA														
n.subp. ! n.part.D ! n.part.E ! tamanho da lista														
PR !	T !	N !	medio !	max. !	min. !	medio !	max. !	min. !	medio !	max. !	medio ! max.			
6	B	2	3,34	8	1	1,94	3	1	2,24	3	119,99 302			
6	B	3	3,42	8	1	1,94	3	1	2,27	3	123,83 302			
6	B	4	3,42	8	1	1,94	3	1	2,27	3	123,83 302			
6	B	5	2,86	7	1	1,86	3	1	2,05	3	103,51 267			
SUBPROBLEMAS														
cortados ! particionados														
PR !	T !	N !	nct !	nc0 !	nc5 !	nc6 !	nc7 !	npt !	np1 !	np2 !	np3 !	np4 !	n.res. !	n.ng0
6	B	2	60	0	25	1	34	52	1	30	16	6	113	72
6	B	3	65	0	38	1	26	56	2	92	55	4	218	77
6	B	4	66	0	39	1	26	56	2	418	129	4	619	77
6	B	5	41	0	27	0	14	40	3	579	276	4	903	56
QUALIDADE DAS SOLUCOES ! NUMERO DE SUBPROBLEMAS														
PR !	T !	N !	1.subp./sol.ot. !				1.viav/sol.ot. !				1.subp. !	1.viav. !	otimo !	total
6	B	2	5,0000				1,2500				1	2	18	109
6	B	3	5,0000				1,5000				1	1	18	119
6	B	4	5,0000				1,5000				1	1	18	119
6	B	5	5,0000				1,5000				1	1	3	78



PROBL. ! DADOS DE ENTRADA ! custo ! tempo(seg.) de exec.ate:												
PR ! T ! N !	maior	menor	napo	nasp	real	int.	i.subp.	i.viav.	otimo	final		
7 - B 2	5	1	4	1		[0,100]	<1	1	26	59		
7 - B 3	100	1	4	1		[0,100]	<1	1	9	41		
7 - B 4	60	10	5	2		[0,100]	<1	1	14	96		
7 - B 5	100	20	6	3		[0,100]	<1	1	19	136		
LISTA												
n.subp. ! n.part.D ! n.part.E ! tamanho da lista												
PR ! T ! N !	medio	max.	min.	medio	max.	min.	medio	max.	medio	max.		
7 - B 2	5,42	12	1	1,92	3	1	2,24	3	398,56	954		
7 - B 3	5,00	12	1	1,86	3	1	2,16	3	354,27	954		
7 - B 4	5,04	12	1	1,86	3	1	2,16	3	356,71	954		
7 - B 5	5,46	12	1	2,12	3	1	2,25	3	389,23	954		
SUBPROBLEMAS												
cortados ! particionados												
PR ! T ! N !	nct	nc0	nc5	nc6	nc7	npt	np1	np2	np3	np4	n.res.	n.ng0
7 - B 2	103	0	44	0	59	100	5	47	13	34	202	162
7 - B 3	62	0	34	0	28	62	5	40	8	30	145	93
7 - B 4	61	0	33	0	28	61	6	239	47	30	383	91
7 - B 5	54	0	26	0	28	47	6	396	56	22	534	69
QUALIDADE DAS SOLUCOES ! NUMERO DE SUBPROBLEMAS												
PR ! T ! N !	1.subp./sol.ot.		1.viav/sol.ot.		1.subp.		1.viav.		otimo	total		
7 - B 2	37,0455		2,7879		1		1		84	198		
7 - B 3	37,0455		2,7879		1		1		8	119		
7 - B 4	37,0455		2,7879		1		1		7	117		
7 - B 5	37,0455		2,7879		1		1		7	96		



PROBL. ! DADOS DE ENTRADA ! custo ! tempo(seg.) de exec.ate:														
PR	T	N	maior	menor	napo	nasp	real	int.	i.subp.	i.viav.	otimo	final		
9	B	2	5	1	4	1	[0,100]		<1	4	22	41		
9	B	3	100	1	4	1	[0,100]		<1	18	83	148		
9	B	4	60	10	5	2	[0,100]		<1	18	197	300		
9	B	5	100	20	6	3	[0,100]		<1	21	372	546		
LISTA														
n.subp. ! n.part.D ! n.part.E ! tamanho da lista														
PR	T	N	medio	max.	min.	medio	max.	min.	medio	max.	medio	max.		
9	B	2	4,22	11	1	1,83	4	1	1,85	3	106,09	317		
9	B	3	4,67	11	1	1,46	4	1	1,95	3	118,46	317		
9	B	4	4,41	11	1	1,53	4	1	1,91	3	110,96	317		
9	B	5	4,67	11	1	1,46	4	1	1,95	3	118,35	317		
SUBPROBLEMAS														
cortados ! particionados !														
PR	T	N	nct	nc0	nc5	nc6	nc7	npt	np1	np2	np3	np4	n.res.	n.ng0
9	B	2	62	0	5	0	57	145	5	42	91	8	208	400
9	B	3	129	0	83	1	45	571	4	93	136	436	798	1927
9	B	4	114	0	43	1	70	433	4	948	420	300	1786	1403
9	B	5	129	0	83	1	45	570	5	1926	811	436	3307	1923
QUALIDADE DAS SOLUCOES ! NUMERO DE SUBPROBLEMAS														
PR	T	N	1.subp./sol.ot.				1.viav/sol.ot.				1.subp. ! 1.viav. ! otimo		total	
9	B	2	2,4049				1,2890				1 ! 14 ! 107		204	
9	B	3	2,4049				1,2481				1 ! 6 ! 348		699	
9	B	4	2,4049				1,2484				1 ! 6 ! 257		546	
9	B	5	2,4049				1,3858				1 ! 2 ! 347		698	

PROBL.		DADOS DE ENTRADA					custo		tempo(seg.) de exec.ate:					
PR	T	N	maior	menor	napo	nasp	real	int.	1.subp.	1.viav.	otimo	final		
10	B	2	5	1	4	1	[0,1000]		<1	1	1	17		
10	B	3	100	1	4	1	[0,1000]		<1	1	1	37		
10	B	4	60	10	5	2	[0,1000]		<1	1	1	47		
10	B	5	100	20	6	3	[0,1000]		<1	1	1	72		
LISTA														
			n.subp.		n.part.D			n.part.E		tamanho da lista				
PR	T	N	medio	max.	min.	medio	max.	min.	medio	max.	medio	max.		
10	B	2	2,95	7	1	1,85	3	1	2,50	3	91,98	247		
10	B	3	2,95	7	1	1,85	3	1	2,50	3	91,98	247		
10	B	4	2,95	7	1	1,85	3	1	2,50	3	91,98	247		
10	B	5	2,39	6	1	1,73	2	1	2,36	3	70,96	194		
SUBPROBLEMAS														
			cortados				particionados							
PR	T	N	nct	nc0	nc5	nc6	nc7	npt	np1	np2	np3	np4	n.res.	n.ng0
10	B	2	41	0	3	0	38	47	1	13	2	27	84	58
10	B	3	41	0	3	0	38	47	1	108	2	27	179	58
10	B	4	41	0	3	0	38	47	1	167	2	27	238	58
10	B	5	32	0	3	0	29	40	1	314	2	21	370	50
			QUALIDADE DAS SOLUCOES				NUMERO DE SUBPROBLEMAS							
PR	T	N	1.subp./sol.ot.				1.viav/sol.ot.				1.subp.	1.viav.	otimo	total
10	B	2	13,5513				1,0000				1	1	1	80
10	B	3	13,5513				1,0000				1	1	1	80
10	B	4	13,5513				1,0000				1	1	1	80
10	B	5	13,5513				1,0000				1	1	1	64

PROBL.		DADOS DE ENTRADA				custo		tempo(seg.) de exec.ate:						
PR	T	N	maior	menor	napo	nasp	real	int.	i.subp.	i.viav.	otimo	final		
12	B	2	5	1	4	1	[0,100]		<1	1	24	79		
12	B	3	100	1	4	1	[0,100]		<1	1	263	297		
12	B	4	60	10	5	2	[0,100]		<1	1	279	316		
12	B	5	100	20	6	3	[0,100]		<1	1	351	384		
LISTA														
		n.subp.			n.part.D			n.part.E			tamanho da lista			
PR	T	N	medio	max.	min.	medio	max.	min.	medio	max.	medio	max.		
12	B	2	6,64	17	1	2,71	5	2	3,43	5	1075,15	3092		
12	B	3	7,06	21	1	2,75	5	1	3,11	5	1112,36	3853		
12	B	4	6,22	15	1	2,76	5	1	2,72	5	943,18	2551		
12	B	5	6,52	16	1	2,76	5	1	2,85	5	944,43	2826		
SUBPROBLEMAS														
		cortados					particionados							
PR	T	N	nct	nc0	nc5	nc6	nc7	npt	np1	np2	np3	np4	n.res.	n.ng0
12	B	2	96	0	2	5	89	46	3	1	43	0	143	14
12	B	3	329	0	149	6	174	167	14	25	77	144	589	79
12	B	4	320	0	150	3	167	183	17	64	67	169	637	103
12	B	5	348	0	164	3	181	191	13	177	90	176	804	105
QUALIDADE DAS SOLUCOES													NUMERO DE SUBPROBLEMAS	
PR	T	N	i.subp./sol.ot.			i.viav/sol.ot.		i.subp.		i.viav.		otimo	total	
12	B	2	223,1524			70,1375		1		1		31	139	
12	B	3	223,1524			70,1375		1		1		418	490	
12	B	4	223,1524			70,1375		1		1		419	498	
12	B	5	223,1524			70,1375		1		1		462	534	

As execuções do tipo B não apresentam vantagens se o número de iterações dos subproblemas gerados a partir do primeiro é pequeno (por exemplo: menor = 1,2,3 ou 5)

Todavia, para valores maiores (menor = 10,15 ou 20) uma grande redução no tempo de execução é observada, se comparada com a execução tipo A e com os mesmos dados de entrada.

No caso das execuções do tipo B, também permanece válida a observação feita no sentido da redução do número de subproblemas examinados.

REFERÊNCIAS BIBLIOGRÁFICAS

- [Ag1] AGMON, S.; "The Relaxation Method for Linear Inequalities", *Canadian Journal of Mathematics*, 6, 382-392, (1954)

- [BQ1] BALINSKI, M.L. and QUANDT, R.E.; "On an Integer Program for Delivery Problem", *Operations Research*, 12(2), 300-304, (1964)

- [BS1] BAZARAA, M. and SHERALI, H.D.; "On the Choice of Step Size in Subgradient Optimization", *European Journal of Operational Research*, 7(4), 380-388, (1981)

- [Ed1] EDMONDS, J.; "Maximum Matching and a Polyedron with 0-1 Vertices", *Journal of Research of the National Bureau of Standards*, 69B(1 e 2), 125-130, (1965)

- [Et1] ETCHEBERRY, J.; "The Set Representation Problem", *PhD. Thesis, University of Michigan*, 175 pgs., (1974)

- [FS1] FISHER, M.L. and SHAPIRO, J.F.; "Construtive Duality in Integer Programming", *SIAM J. Appl. Math.*, 27(1), 31-52, (1974)

- [Ga1] GABOW, H.; "An Efficient Implementation of Edmond's Matching Algorithm", *Technical Report nº31, D.S.L., Stanford University - Califórnia*, 68 pgs., (1972)

- [GN1] GARFINKEL, R. and NEMHAUSER, G.; *Integer Programming*, New York, J. Wiley, (1972)

- [GN2] GARFINKEL,R.and NEMHAUSER,G.;"Optimal Set Covering: A Survey", *in Perspectives in Optimization: A Collection of Expository Articles* (ed.Geoffrion),Mass.,Addison-Wesley,(1972)

- [Ge1] GEOFFRION,A.M.;"Lagrange Relaxation for Integer Programming",*Mathematical Programming Study 2*,North-Holland,82-114,(1974)

- [HK1] HELD,M.and KARP,R.M.;"The Travelling-Salesman Problem and Minimum Spanning Trees:Part II", *Mathematical Programming* ,1,6-25,(1971)

- [HW1] HELD,M.,WOLFE,P.and CROWDER,H.P.;"Validation of Subgradient Optimization", *Mathematical Programming*,6,62-88,(1974)

- [HK1] HOPCROFT,J.E.and KARP,R.;"An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs",*SIAM. J. Comput.*,2(4),225-231,(1973)

- [La1] LAWLER,E.L.;"Discussion of on the Set Representation and the Set Covering Problem",*Symposium on the Theory of Scheduling and its Applications* ,Springer-Verlag,Heidelberg,164-166,(1973)

- [Lo1] LOMBARDO,D.H.;"Estudo dos Problemas do Carteiro Chinês e do Caixeiro Viajante", *Tese de Mestrado,IMECC-UNJCAMP.*,131 pages.,(1986)

- [MS1] MARSTEN, E.R. and SHEPARDSON, F.; "Exact Solution of Crew Scheduling Problems Using the Set Partitioning Model: Recent Successful Applications", *Networks*, 1, 165-177 , (1981)

- [Mi1] MINIEKA, E.; *Optimization Algorithms for Networks and Graphs* , New York and Basel, Dekker, 1, (1978)

- [Mi2] MINOUX, M.; *Programmation Mathématique - Théorie et Algorithmes* , 1, Paris, Dunod, (1983)

- [Mi3] MINOUX, M.; *Programmation Mathématique - Théorie et Algorithmes* , 2, Paris, Dunod, (1983)

- [Mi4] MITTEN, L.G.; "Branch and Bounds Methods: General Formulation and Properties", *Operations Research*, 8(1), 24-34 (1970)

- [MS1] MOTZKIN, T. and SCHOENBERG, I.J.; "The Relaxation Method for Linear Inequalities", *Canadian Journal Of Mathematics*, 6, 393-404, (1954)

- [Mu1] Murty, K.G.; "On the Set Representation and the Set Covering Problems", *Symposium on the Theory of Scheduling and its Applications*, Springer-Verlag, Heidelberg, 143-163, (1973)

- [Mu2] Murty, K.G.; *Linear and Combinatorial Programming*, New York, J. Wiley, (1976)

- [MP1] MURTY,K.G.and PERIN,C.; "A 1-Matching Blossom-Type Algorithm for Edge Covering Problems, *Networks*, 12, 379-391, (1982)

- [Po1] POLJAK,B.T.; "A General Method of Solving Extremum Problems", *Soviet Mathematics Dokl.*, 8(3), 593-597, (1967)

- [Po2] POLJAK,B.T.; "Minimization of Unsmooth Functionals" , *USSR - Computational Mathematics and Mathematical Physics*, 9(3), 14-29, (1969)

- [Sa1] SANDI,C.; "Subgradient Optimization", in *Combinatorial Optimization* ,New York, Wiley, (1979)

- [Sl1] SALKIN,H.M.; *Integer Programming* ,Mass.,Addison-Wesley, (1975)

- [Sh1] SHAPIRO,J.F.; "A survey of Lagrangean Techniques for Discrete Optimization", *Annals of Discrete Mathematics*, 5, 113-138, (1979)

- [TS1] TOREGAS,C.,SWAIN,R.; REVELLE,C.and BERGMAN,L.; "The Location of Emergency Service Facilities"; *Operations Research*, 19(6), 1363-1373, (1971)