PROGRAMAÇÃO MISTA COM VARIÁVEIS BIVALENTES:

PROGRAMA DE APLICAÇÃO DO MÉTODO DE DECOMPOSIÇÃO DE BENDERS

Octavio de Abreu Sampaio Filho

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

Aprovada por:

Nelson Maculan Filho

Presidente

Fernando Chigoski

R.C. de Sá

RIO DE JANEIRO
ESTADO DO RIO DE JANEIRO - BRASIL
SETEMBRO DE 1975

Ao Professor Nelson Maculan Filho
meus agradecimentos pelo seu apoio e encorajamento.

RESUMO

Este trabalho descreve uma experiência de aplicação do Método de Decomposição de Benders a problemas de programação linear mista em variáveis bivalentes.

O problema inteiro resultante da decomposição é resolvido de acordo com o Método do Filtro de Balas, a menos do esquema enumerativo.

ABSTRACT

This paper presents an experimental work on the application of Benders' Partitioning Procedure to mixed programming problems with binary variables. The integer programming resulting from the partitioning is solved by Balas' Filter Method with a change in the enumerative scheme.

ÍNDICE

Capítulos:		Páginas:
I	INTRODUÇÃO	1
II	MÉTODO DE DECOMPOSIÇÃO	5
	2.1 Teorema de Decomposição	6
	2.2 Processo Computacional	8
III	PROBLEMA RELAXADO INTEIRO	19
	3.1 Método do Filtro |2|	24
	3.2 Esquema Enumerativo	37
	3.3 Problema Relaxado Contínuo	41
IV	SUB-PROBLEMA CONTÍNUO	46
V	DESCRIÇÃO DO PROGRAMA	51
	5.1 Descrição de Variáveis	53
	5.2 Listagem	60
	5.3 Dados de Entrada	65
VI	COMENTÁRIOS	66
	6.1 Variante	66
	6.2 Sub-rotina de Programação Linear	69
VII	REFERÊNCIAS	83

CAPÍTULO I

INTRODUÇÃO

Este trabalho descreve uma experiência de aplicação do método de decomposição de Benders |4| a problemas de programação linear mista com variáveis bivalentes. (Para descrições alternativas, vide |3|, |10|, |11|).

Como se sabe, os resultados apresentados por Benders permitem resolver um problema de programação linear mista, através de processo iterativo em que são resolvidos alternadamente um problema parcial inteiro e outro contínuo.

Propriedade importante deste método é a disponibilidade de limites superior e inferior para o valor da solução ótima, os quais aumentam com o número de iterações. O limite inferior (em problemas de maximização) é gerado por uma sequência de

soluções viáveis para o problema misto. Assim, caso o processo seja interrompido antes de atingir a solução ótima, pode-se tomar a melhor solução viável gerada, dispondo-se ainda de uma indicação sobre a diferença entre o valor desta solução e o da solução ótima.

O método tem também a propriedade de conservar a estrutura da matriz de coeficientes das variáveis contínuas. Dessa forma, a resolução do sub-problema contínuo poderá ser simplificada se esta matriz apresentar uma estrutura particular ou ainda se for necessário resolver vários problemas mistos com a mesma parte contínua.

As aplicações a problemas de programação linear mista |6|, |3|, |4|, |10| indicam que o processo converge em número muito pequeno de iterações, sendo que este usualmente não cresce em proporção ao número de alternativas lógicas a considerar ou mesmo em proporção ao número de variáveis inteiras.

Essas características permitem considerar a decomposição de Benders como um método de grande potencialidade para solução de problemas mistos. No entanto, as aplicações conhecidas indicam que a eficiência obtida depende basicamente do processo adotado para solução do problema inteiro resultante da decomposição. Têm sido usados métodos enumerativos, principalmente os derivados do algoritmo aditivo de Balas. Os resultados

obtidos limitam as aplicações a problemas com até cerca de 50 variáveis binárias.

Por outro lado, têm sido relatados grandes avanços na solução de problemas inteiros com o uso de restrições "surrogate" em métodos derivados do algoritmo aditivo. Em especial, Geoffrion resolveu problemas com cerca de 100 variáveis bivalentes em tempos inferiores a 10 segundos em um IBM-360/91 |6|.

Esses resultados sugerem que o uso de um método adequado de enumeração implícita poderá permitir a resolução de problemas de programação mista pela decomposição de Benders com eficiência superior à das aplicações mencionadas.

O presente estudo visa uma exploração preliminar dessa possibilidade, através do desenvolvimento de um programa experimental em que o problema inteiro é resolvido por método derivado do algoritmo aditivo empregando restrição "surrogate".

O programa desenvolvido resolve o problema inteiro seguindo basicamente o enfoque proposto por Balas para o emprego de restrições "surrogate" em conjunto com o algoritmo aditivo, constituindo o método do filtro |2|. Apenas uma restrição "surrogate" é gerada durante a resolução do problema, o que parece mais adequado para resolução sequencial de uma série de problemas, como requer a decomposição de Benders.

Também se considerou, ao escolher o método do filtro, o interesse em dispor de resultados computacionais para o mesmo, até agora não encontrados na literatura.

Apresentam-se ainda considerações sobre possíveis variantes para solução do problema inteiro, levando em conta o contexto da decomposição de Benders.

O processo proposto para solução do sub-problema contínuo de Benders e do correspondente contínuo do problema inteiro, através do qual é gerada a restrição "surrogate", consiste do método simplex revisado e dos procedimentos de pós-otimização aplicáveis.

CAPÍTULO II

MÉTODO DE DECOMPOSIÇÃO

Neste capítulo apresenta-se um sumário do método de decomposição de Benders |4|, contendo os resultados necessários para a presente aplicação. Serão considerados problemas do tipo:

$$\max \quad cx + fy$$

$$Ax + Fy \leqslant b \qquad (2.1)$$

$$x \geqslant 0, \quad y \varepsilon \{0,1\}^q, \quad y \text{ inteiro}$$

onde $x \varepsilon R_p$, $y \varepsilon R_q$, A e F são matrizes de dimensões (m,p) e (m,q) enquanto que b, c e f são vetores de R_m, R_p e R_q, respectivamente.

2.1 - TEOREMA DE DECOMPOSIÇÃO

O método de Benders permite decompor o problema apresentado em dois problemas parciais, sendo um de programação inteira e outro de programação linear. Consideremos o problema (2.1) na seguinte forma equivalente:

$$\begin{aligned} \max \quad & x_0 \\ & x_0 - cx - fy < 0 \\ & Ax + Fy > b \\ & x > 0, \quad y \varepsilon \{0,1\}^q, \quad y \text{ inteiro} \end{aligned} \tag{2.2}$$

Define-se os conjuntos:

$$C = \{(u_0, u) | Au - Cu_0 > 0, u > 0, u_0 > 0\} \tag{2.3}$$

$$\begin{aligned} G = \quad & \{(x_0, y) | u_0 x_0 + u Fy - u_0 fy < u b, (u_0, u) \varepsilon C \\ & y \varepsilon \{0,1\}^q, \quad y \text{ inteiro}\} \end{aligned} \tag{2.4}$$

O problema (2.2) e, portanto, o problema (2.1), pode ser decomposto em:

$$\max \{x_0 \mid (x_0, y) \varepsilon G\} \tag{2.5}$$

$$\max \{cx \mid Ax \leqslant b - Fy, \quad x \geqslant 0\} \tag{2.6}$$

no seguinte sentido:

a) problema (2.1) inviável <=> problema (2.5) inviável;

b) problema (2.1) ilimitado <=> problema (2.5)ilimitado;

c) $(\bar{x}, \bar{y})$ resolve problema (2.1), $\bar{x}_0 = c\bar{x} + f\bar{y}$ =>$(\bar{x}_0, \bar{y})$ resolve problema (2.5), $\bar{x}$ resolve problema (2.6);

d) $(\bar{x}_0, \bar{y})$ resolve problema (2.5) => problema (6) viável, $c\bar{x} = \bar{x}_0 - f\bar{y}$, $(\bar{x}, \bar{y})$ resolve problema (2.1) (sendo $\bar{x}$ a solução do problema (2.6)).

Note-se que o conjunto G pode ser interpretado como o conjunto de valores de y para os quais o problema original tem solução viável.

De fato, para um $\bar{y}$ fixado, o problema (1) reduz-se a um problema de programação linear cujo dual é:

$$\min \{u(b - F\bar{y}) \mid A^T u \geqslant C, \quad u \geqslant 0\} \tag{2.7}$$

Para que o problema (2.1) tenha solução viável, o problema (2.7) deve ter solução limitada, isto é:

$$u(b - F\bar{y}) \geqslant 0 \quad \forall u \mid A^T u \geqslant 0, \quad u \geqslant 0 \tag{2.8}$$

Por outro lado, por dualidade e por (2.2), tem-se:

$$x_0 \leqslant cx + fy \leqslant u(b - Fy) + fy \tag{2.9}$$

As condições (2.8) e (2.9) constituem as restrições que caracterizam os limites do conjunto G, respectivamente para $u_0 = 0$ e $u_0 = 1$.

2.2 - PROCESSO COMPUTACIONAL

Uma solução direta do problema (2.5) exigiria o cálculo de todas as restrições do mesmo, de forma a determinar o conjunto G. (Pode-se mostrar que o mesmo poderia ser feito determinando todas as arestas do cone convexo poliédrico C). Mas, isto é praticamente impossível, devido ao enorme esforço de cálculo necessário. No entanto, para obter uma solução ótima desse problema, basta conhecer as restrições que determinam um ponto

de ótimo. Para tanto, Benders propõe um processo iterativo que gera restrições até completar as necessárias a determinar um ponto ótimo. O processo iterativo é baseado no Teorema de Decomposição e nos seguintes lemas.

LEMA 1:

Problema (2.5) viável, S limitado => (x_0 sem limite superior em G <=> $\nexists\ u \mid A^T u \geqslant C,\ u \geqslant 0$).

LEMA 2:

Se $\mathcal{Q} \subset C$, $\mathcal{Q} \neq \phi$

$$G(\mathcal{Q}) = \{(x_0, y) \mid u_0 x_0 + u Fy - u_0 fy \leqslant u b,\ (u_0, u) \varepsilon \mathcal{Q}$$

$$y \varepsilon \{0,1\}^q,\ y \text{ inteiro}\} \qquad (2.10)$$

dado o problema

$$\max \{x_0 \mid (x_0, y) \varepsilon G(\mathcal{Q})\} \qquad (2.11)$$

tem-se: (($\bar{x}_0$ $\bar{y}$) resolve problema (2.11) => ($\bar{x}_0$ $\bar{y}$) resolve problema (2.5)).

$$\Leftrightarrow \min \{(b - F\bar{y})u \mid A^T u \geqslant C, \quad u \geqslant 0 \quad = \quad \bar{x}_0 - f\bar{y}\} \quad (2.12)$$

Quando não é satisfeita a condição de otimalidade dada pelo lema 2, isto é:

$$(b - F\bar{y})\bar{u} < \bar{x}_0 - f\bar{y} \qquad (2.13)$$

então $(1, \bar{u}) \notin \mathcal{Q}$. Como $(1, \bar{u}) \varepsilon$ C, forma-se novo subconjunto $\mathcal{Q}^* \subset$ C pela inclusão de $(1, \bar{u})$, o que corresponde a gerar uma restrição de G não satisfeita por $(\bar{x}_0, \bar{y})$.

Pode ocorrer também que o problema (2.7) tenha solução ilimitada. Neste caso, o método simplex fornece um vértice $\bar{u}$ e um raio extremo $\bar{v}$ tais que:

$$(b - F\bar{y})\bar{v} < 0 \qquad (2.14)$$

Então $(0, \bar{v}) \notin \mathcal{Q}$. Como $(0, \bar{v}) \varepsilon$ C, inclui-se este ponto em como no caso acima. Caso a condição (2.13) seja satisfeita, inclui-se também $(1, \bar{u})$ em $\mathcal{Q}$.

Em cada iteração ν, o conjunto $G(\mathcal{Q}^{\nu})$ é obtido a partir de $G(\mathcal{Q}^{\nu-1})$ pela inclusão da respectiva restrição:

$$x_0 + \bar{u}\, Fy - fy \leqslant \bar{u}\, b$$

$$\bar{v}\, Fy \leqslant \bar{v}\, b$$

Note-se que a solução do problema misto baseia-se na solução em cada iteração de dois problemas parciais de otimização, referidos por (2.7) e (2.11).

O problema (2.7) é o dual do problema de programação linear resultante de fixar as variáveis inteiras do problema misto, constitue um sub-problema contínuo (SPC), forma pela qual será denominado.

O problema (2.10) é uma versão relaxada, isto é, com menos restrições do problema inteiro (2.5), ao qual se acrescentam restrições em cada iteração. Consiste portanto de um problema relaxado inteiro (PRI), nome pelo qual será designado.

Passamos a seguir ao enunciado do algoritmo de decomposição, na forma como foi aplicado neste estudo. O algoritmo está esquematizado no fluxograma da Figura 1.

I) Inicialização:

Fazer $\nu = 0^e, x_0^0 = +\infty$ (na prática, um número sufici

entemente grande). Tomar um y^0 arbitrário tal que $y^0 \varepsilon \{0,1\}^q$, y^0 inteiro.

II) Resolver $(SPC)^\nu$ dado por

$$\min(b - Fy^\nu)u$$

$$A^T u > C$$

$$u > 0$$

Se $(SPC)^\nu$ for inviável, para o problema original (MIX) é inviável ou ilimitado (note-se que esta situação só pode ocorrer na primeira iteração).

Se $(SPC)^\nu$ tiver solução ótima u^ν e:

$$(b - Fy^\nu)u^\nu = x_0^\nu - fy^\nu \qquad (2.15)$$

então pare: (x^ν, y^ν) é solução ótima de MIX com valor x_0^ν, sendo x^ν dado pela solução ótima dual de $(SPC)^\nu$.

Se $(SPC)^\nu$ tiver solução ótima u^ν e:

$$(b - Fy^\nu)u^\nu < x_0^\nu - fy^\nu \qquad (2.16)$$

então:

- acrescentar a PRI a restrição dada por:

$$x_0 + u^\nu Fy - fy \leqslant u^\nu b \qquad (2.17)$$

- incrementar ν e desviar para PRI (III).

Se $(SPC)^\nu$ tiver solução ilimitada segundo:

$$u^\nu + \lambda\, v^\nu, \quad \lambda \geqslant 0 \qquad (2.18)$$

sendo u^ν uma solução básica de $(SPC)^\nu$ e v^ν a direção do raio extremo associado à solução ilimitada, então:

- acrescentar a restrição

$$v^\nu Fy \leqslant v^\nu b \qquad (2.19)$$

- se

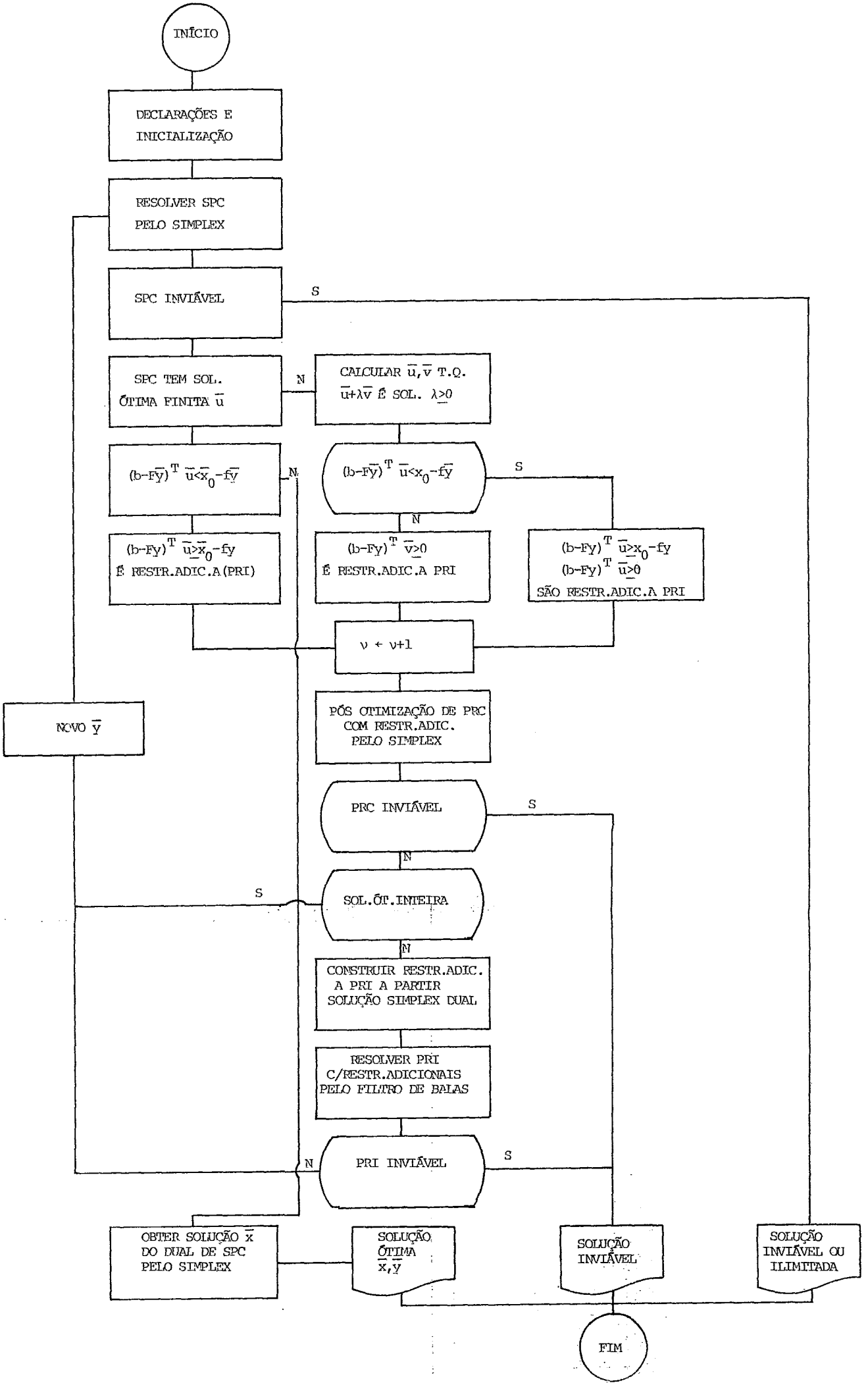
$$(b - Fy^\nu)u < x_0^\nu - fy^\nu$$

acrescentar também a restrição

$$x_0 + u^\nu Fy - fy \leqslant u^\nu b$$

- incrementar ν e passar a PRI (III).

FIG 1 - MÉTODO DE DECOMPOSIÇÃO DE BENDERS PARA MIX



III) Resolver $(PRI)^{\nu}$ dado por:

$$\max x_0$$

$$x_0 - fy + u^l Fy \overset{\downarrow}{\leq} u^l b \qquad \forall l \varepsilon L$$

$$v^k Fy \underset{\uparrow}{\leq} v^k b \qquad \forall k \varepsilon K$$

$$y \varepsilon \{0,1\}^q$$

$$y \text{ inteiro}$$

$$x_0 \text{ livre}$$

Se $(PRI)^{\nu}$ for inviável, pare: o problema original (MIX) é inviável.

Se $(PRI)^{\nu}$ tiver solução ótima (x_0^{ν}, y^{ν}), desviar para o passo II.

Observe-se que o raio extremo v pode ser obtido do último quadro de algoritmo simplex usado para resolver SPC. No caso de se verificar a condição de otimalidade dada por (2.15), a solução ótima do dual de SPC também pode ser obtida do último quadro do simplex.

Característica importante do algoritmo é possibilitar o conhecimento de limites superior e inferior do valor da solução ótima do problema original (MIX). A cada vez que se obtem uma solução ótima u^{ν} de $(SPC)^{\nu}$, obtem-se uma solução ótima x^{ν} de seu dual, dado por:

$$\max Cx$$

$$Ax \leq b - Fy^{\nu} \qquad (2.20)$$

$$x \geq 0$$

tal que

$$(b - Fy^{\nu})u^{\nu} = cx^{\nu} \qquad (2.21)$$

Como (x^{ν}, y^{ν}) satisfazem (2.20) e y^{ν} é inteiro bivalente, pois satisfaz a PRI, então (x^{ν}, y^{ν}) é viável para MIX e tem valor:

$$(b - Fy^{\nu})u^{\nu} + fy^{\nu} \leq x_0^*$$

onde x_0^* é a solução ótima de MIX.

Por outro lado, como (PRI) não contém todas as restrições ativas no ponto de ótimo, tem-se:

$$x_0^* \leq x_0^\nu$$

Assim, os limites superior e inferior são conhecidos no final de cada iteração do algoritmo.

As regras de parada empregadas são justificadas como segue:

a) se $(PRI)^\nu$ for inviável então $G(\mathcal{Q}^\nu) = \phi$; como $G \subset G(\mathcal{Q}^\nu)$ tem-se $G = \phi$, portanto o problema (2.5) é inviável e pelo Teorema de Decomposição (item a), MIX é inviável.

b) se $(SPC)^\nu$ for inviável, então:

b.1) se $G \neq \phi$ então, pelo Lema 1, o problema (2.5) é ilimitado e, portanto, pelo Teorema de Decomposição (item b), MIX é ilimitado;

b.2) se $G = \phi$, então MIX é inviável;

c) se $(SPC)^\nu$ tem solução ótima u com valor

$$(b - Fy^\nu)u^\nu = x_0^\nu - fy^\nu$$

então pelo Lema 1 (x_0^{ν}, y^{ν}) resolve o problema (2.5) e, portanto, pelo Teorema de Decomposição (item d), resolve também MIX.

A convergência do algoritmo é assegurada, pois a cada nova iteração pelo menos uma aresta do cone poliédrico C, não pertencente ao conjunto o^{ν}, é acrescentado a este para formar $\mathcal{O}^{\nu+1}$. Portanto, em um número finito de iterações ou o algoritmo termina, pela aplicação de uma das regras de parada, ou se dispõe de um conjunto completo de restrições para o problema (2.5) e, portanto, o algoritmo terminará antes da iteração seguinte, de acordo com o Teorema de Decomposição (item d).

CAPÍTULO III

PROBLEMA RELAXADO INTEIRO

Aplicando o método de decomposição de Benders, um problema de programação mista pode ser reduzido à solução de um problema inteiro e outro contínuo.

Neste capítulo apresenta-se o método de solução adotado no presente trabalho para o primeiro problema, denominado problema relaxado inteiro (PRI).

De acordo com o capítulo anterior, o mesmo tem a forma:

$$\max \{x_0 \mid (x_0, y) \varepsilon G(\alpha^{\nu})\}$$

onde:

$$G(\mathcal{Q}^{\nu}) = \bigcap\{(x_0, y) | u_0 x_0 + u Fy - u_0 fy \leq u b,$$

$$(u_0, u) \varepsilon \mathcal{Q}^{\nu} \qquad y \varepsilon \{0,1\}^q, \quad y \text{ inteiro}\}$$

Considerando que as restrições do conjunto $G(\mathcal{Q}^{\nu})$ podem apresentar duas formas, componentes a $u_0 = 1$ e $u_0 = 0$, o problema relaxado inteiro pode ser formulado alternativamente por:

$$\max x_0$$

$$x_0 - fy + u^i Fy \leq u^i b \qquad \forall i \varepsilon L$$

$$u^i Fy \leq u^i b \qquad \forall i \varepsilon K$$

$$y \varepsilon \{0,1\}^q$$

$$y \text{ inteiro}$$

$$x_0 \text{ livre}$$

onde, por definição:

$$L = \{i | (1, u^i) \varepsilon \mathcal{Q}^{\nu}\}$$

$$K = \{i | (0, 1^i) \varepsilon \mathcal{Q}^{\nu}\}$$

Com vistas a escolher um método de solução para o $(PRI)^{\nu}$, observemos inicialmente suas peculiaridades:

a) não se trata de um problema totalmente inteiro mas, de um problema misto com apenas uma variável contínua x_0;

b) x_0 tem coeficientes um ou zero, apenas;

c) todas as restrições são da forma $\geqslant$;

d) a função objetivo não depende de nenhuma das variáveis inteiras, sendo igual à variável x_0;

e) deve-se resolver uma sequência de problemas $(PRI)^{\nu}$, sendo cada um obtido do anterior pelo acréscimo de uma ou duas restrições.

Considerando as propriedades (a) a (d) acima, resolver o PRI correspondente a encontrar a solução de um problema totalmente inteiro cuja mínima folga $i \varepsilon L$ tem o máximo valor. Essa característica fica realçada por um rearranjo das variáveis do problema, como segue:

$$\max x_0$$

$$x_0 \leq u^i b - u^i F y + f y \qquad \forall i \varepsilon L$$

$$0 \leq u^i b - u\ F y \qquad \forall i \varepsilon K$$

$$y \varepsilon \{0,1\}^q$$

$$y \text{ inteiro}$$

$$x_0 \text{ livre}$$

Esse resultado será aplicado no método de solução utilizado, como se verá adiante.

Neste trabalho optou-se por resolver o problema inteiro segundo um método de enumeração implícita derivado do algoritmo aditivo de Balas. Pelas suas características, este método é especialmente apropriado para problemas mistos em variáveis bivalentes.

O algoritmo aditivo, na forma original proposta por Balas |1|, apresenta eficiência muito sensível ao número de variáveis. No entanto, o acréscimo de uma restrição derivada das iniciais leva a grande melhoria em eficiência, conforme mostrado por Geoffrion |5|, |6|, permitindo resultados satisfatórios com até cerca de 100 variáveis.

A limitação do algoritmo aditivo em sua forma original pode ser explicada por se basear em testes aplicados a cada restrição individualmente. A implicação conjunta de diversas restrições não é, portanto, devidamente considerada. A restrição adicional, denominada "surrogate", foi proposta por Glover |7| visando expressar a implicação conjunta do conjunto de restrições originais. (Para uma discussão do papel da restrição "surrogate", vide |8|, |13|).

A restrição "surrogate" é uma combinação linear das restrições originais a coeficientes não-negativos e não todos nulos com ponderação arbitrária. Geoffrion usou, como fatores de ponderação, os valores ótimos das variáveis duais do correspondente contínuo do problema inteiro a resolver para cada solução parcial. A estes somou a restrição que limita a função objetivo a valor melhor do que o da melhor solução encontrada antes da iteração atual.

Tendo em vista esses resultados, o problema relaxado inteiro será resolvido por método de enumeração implícita usando restrição "surrogate". Esta será empregada na forma proposta no método do filtro de Balas, como se descreve a seguir.

3.1 - MÉTODO DO FILTRO |2|

Balas propõe o uso de um problema relaxado para testar novas soluções parciais antes de submetê-las aos testes usuais do algoritmo aditivo, de forma a "filtrar" o conjunto de soluções a examinar. Esse problema, denominado filtro, é constituído da mesma função objetivo do problema original e de uma restrição "surrogate".

A restrição "surrogate" é uma combinação linear das restrições originais, ponderados pelos valores ótimos das variáveis duais do correspondente contínuo do problema original.

Balas demonstra que o filtro elimina pelo menos todas as soluções parciais, cujo valor funcional é melhor (superior no caso de maximização) do que o da solução ótima do problema inteiro.

A força do método reside nessa filtragem, executada a baixo custo computacional, pois testar soluções para o filtro é naturalmente mais simples do que testá-las para o problema original.

A restrição "surrogate" difere de Geoffrion por ser gerada apenas uma vez, referindo-se ao problema original.

O método pode ser facilmente adpatado para solução do problema relaxado inteiro resultante da decomposição de Benders, conforme indicado por Balas na mesma referência.

Apresenta-se a seguir um sumário do método do filtro, na forma usada neste trabalho.

Inicialmente, o problema relaxado inteiro é enunciado em formulação mais adequada:

$$\max x_0 \tag{3.1}$$

$$\alpha_i x_0 + \sum_{j=1}^{q} s_{ij} y_j \leq \beta_i \qquad i = \overline{1,t} \tag{3.2}$$

$$y_j \leq 1 \qquad j = \overline{1,q} \tag{3.3}$$

$$x_0 \text{ livre} \tag{3.4}$$

$$y_j \geq 0 \tag{3.5}$$

$$y_j \text{ inteiro } \forall_j = \overline{1,q} \tag{3.6}$$

onde:

t é o número de restrições geradas pelo método de decomposição.

$$\alpha_i = \begin{cases} 1 & \text{se } i\ L \\ 0 & \text{se } i\ K \end{cases} \qquad i = \overline{1,t}$$

$$\beta_i = u^i b \qquad i = \overline{1,t}$$

s_i é um vetor linha da dimensão q dado por:

$$s_i = \alpha_i f + u^i F$$

Pela propriedade (b) de PRI e por dualidade resulta que $\alpha_0 = 0$ apenas quando PRI é ilimitado. Esse caso não será considerado e supõe se $\alpha_0 \neq 0$.

Supõe-se também que $s_{0j} \geq 0$, $\forall_j = \overline{1,q}$, sem perda de generalidade, já que se houve j tal que $s_{0j} < 0$, pode-se fazer a substituição de $y_j = 1 - y_j$. Define-se ainda uma indexação ótima das variáveis, tal que $s_{0j_1} > s_{0j_2} \Rightarrow j_1 < j_2$.

Estas hipóteses resultam em que a solução ótima do filtro é dado por $(x^f, 0)$ com valor $x^f = \beta_0/\alpha_0$. Resulta também que os valores das soluções do problema do filtro para as descendentes de uma solução parcial dada não são superiores ao valor da solução do problema do filtro para aquela solução parcial.

Como x_0 é livre, um vetor y associado a uma solu

ção parcial de PRI sempre define uma solução viável (x_0, y) para o problema do filtro, cujo valor é dado por:

$$y_0 = (\beta_0 - \sum_{j=1}^{q} s_{0j}\, y_j)/\alpha_0 \qquad (3.7)$$

Assim, toda solução examinada pelo filtro é submetida aos testes de enumeração implícita.

Antes de apresentar os testes empregados, deve ser mencionado que neste trabalho adota-se processo diverso do proposto por Balas no que respeita à ordem na qual são examinadas as soluções parciais. Enquanto o método do filtro emprega um procedimento do tipo "branch-and-bound" com lista de soluções parciais ativas, optou-se aqui pelo uso do esquema enumerativo originalmente usado com o algoritmo aditivo (backtracking). Dessa forma, embora sejam usados os mesmos testes propostos por Balas, suas implicações poderão ser diferentes. O esquema enumerativo será tratado mais adiante.

Algumas definições devem ser estabelecidas antes de tratar dos testes de enumeração implícita, como segue:

N = conjunto das variáveis livres, isto é, não designadas para a solução parcial em exame;

$\bar{N}$ = conjunto das variáveis fixas, isto é, designadas para a solução parcial em exame;

$$\bar{\beta}_i = \beta_i - \sum_{j \in \bar{N}} s_{ij} y_j \qquad i = \overline{1,t}$$

O primeiro teste verifica se a solução parcial em exame é viável para o problema inteiro. Dado um vetor zero-um y satisfazendo (3.3), sempre se pode encontrar um escalar ξ tal que (x_0, y) é solução de PRI para todo $x_0 < \xi$. Vê-se facilmente que esse escalar é:

$$\xi = \bar{\beta}_{i0} = \min_{i \in L} \bar{\beta}_i \tag{3.8}$$

Consideremos o correspondente contínuo de PRI, obtido eliminando a restrição de integralidade.

O novo problema será determinado problema relaxado contínuo (PRC). Suponhamos que PRC e, portanto, seu dual, tenham solução ótima finita. Sejam $(\bar{x}_0, \bar{y})$ e $(\bar{w})$ as soluções de PRC primal e dual respectivamente, sendo $w_i (i = \overline{1,t})$ e $w_{t+j} (j = \overline{1,q})$ as variáveis duais associadas com a i-ésima restrição (3.2) e com a j-ésima restrição (limite superior) (3.3) respectivamente.

O problema do filtro é obtido substituindo as restrições (3.2) pela restrição "surrogate" já descrita, resultando:

$$\max x_0 \tag{3.1}$$

$$\alpha_0 x_0 + \sum_{j=1}^{q} s_{0j} y_j \leqslant \beta_0 \tag{3.2'}$$

$$\forall_j = \overline{1,q} \qquad y_j \leqslant 1 \tag{3.3}$$

$$x_0 \quad \text{livre} \tag{3.4}$$

$$\forall_j = \overline{1,q} \qquad y_j \geqslant 0 \tag{3.5}$$

$$\forall_j = \overline{1,q} \qquad y_j \quad \text{inteiro} \tag{3.6}$$

onde:

$$\alpha_0 = \sum_{i=1}^{t} \bar{w}_i \alpha_i$$

$$s_{0j} = \sum_{i=1}^{t} \bar{w}_i s_{ij}$$

Assim, calcula-se $\hat{\beta}_i$ para todo $i \varepsilon K$. Se $\hat{\beta}_i < 0$ para algum $i \varepsilon K$, a solução parcial não é viável e passamos ao teste seguinte. No entanto, se $\hat{\beta}_i \geqslant 0$ para todo $i \varepsilon K$ ou $K = \phi$, en

tão a solução parcial é viável e o melhor valor de x_0 para esta solução parcial é dado por (3.8). Nesse caso, calcula-se $\bar{\beta}_i$ para todo $i \varepsilon L$ e seu máximo fornece ξ.

O método requer que sempre se conheça o maior valor de ξ obtido até então. Assim, se ξ_*^{n-1} é o maior valor de ξ obtido antes da iteração atual, então depois de calcular ξ^n (novo valor obtido para ξ na iteração atual) escolhe-se e guarda-se o valor:

$$\xi_*^n = \max \{\xi_*^{n-1}, \xi^n\} \tag{3.9}$$

Em seguida, verifica-se se há coeficientes livres negativos para a restrição i_0 (que fornecem min $\bar{\beta}_i | i \varepsilon L$ e, portanto, ξ^n). Em caso contrário nenhuma das descendentes da solução parcial em exame pode fornecer solução melhor do que a atual, portanto essas descendentes são implicitamente enumeradas. Para reforçar esse teste, pode-se comparar a soma dos módulos dos coeficientes negativos livres com a diferença $\xi_*^n - \xi^n$. Caso aquela seja menor do que esta, não há descendente viável e aplica-se a regra de retorno. Esta variante é usada neste trabalho em substituição ao teste proposto por Balas.

Caso a solução parcial não seja viável, verifica-se se pode ter descendente viável. Caso não possa, aplica-se a regra de retorno.

Se não se puder excluir a existência de descendentes viáveis com esse teste, verifica-se se alguma das variáveis livres devem assumir necessariamente um certo valor (zero ou um), a fim de assegurar que exista descendente viável.

Se tal não ocorrer, aumenta-se a atual solução parcial pelo acréscimo da variável livre de índice mais baixo, passa-se a examinar uma de suas descendentes. Por outro lado, se houver tais variáveis condicionadas, aumenta-se a solução parcial designando-lhes os valores correspondentes (em qualquer dos dois casos inicia-se uma nova iteração). Pode ocorrer também que a mesma variável esteja condicionada a assumir valores diferentes. Neste caso a solução parcial não tem descendentes viáveis e aplicamos a regra de retorno.

Ao iniciar uma nova iteração n e, portanto, o exame de uma nova solução parcial, compara-se a solução $x_0 f$ do problema do filtro com ξ_*^{n-1}. Se $x_0^f > \xi_*^{n-1}$, pode-se prosseguir, isto é, passar ao teste de viabilidade. Em caso contrário, a atual solução parcial não tem descendente com solução melhor do que a melhor encontrada (isto é, ξ_*^{n-1}) e aplica-se a regra de retorno.

Tendo em vista a propriedade (e) de PRI, Balas propõe ainda uma modificação pela qual em algumas iterações do método de Benders, ou seja, para algumas $(PRI)^{\nu}$, não será necessário

gerar a restrição "surrogate". Sendo $\bar{x}_0^{\nu}$ o valor da solução ótima de (PRC) e $\bar{x}_0^{\nu-1}$ o valor da solução ótima de $(PRI)^{\nu-1}$ introduz-se a seguinte regra:

a) Se $\bar{x}_0^{\nu} \leq \bar{x}_0^{\nu-1}$, aplicar o método do filtro como acima;

b) Se $\bar{x}_0^{\nu} > \bar{x}_0^{\nu-1}$, usar o problema do filtro da iteração anterior.

Apresenta-se a seguir o enunciado do algoritmo do filtro na forma usada neste trabalho. O mesmo está esquematizado no fluxograma da Figura 2.

A. Resolver (PRC) por programação linear (ver adiante).
Se $\bar{x}_0^{\nu} \leq x_0^{\nu-1}$, gerar a restrição "surrogate" a partir do dual de $(PRC)^{\nu}$. Converte os coeficientes desta restrição para positivos caso não o sejam.
Introduzir indexação ótima.
Se $\bar{x}_0^{\nu} > \bar{x}_0^{\nu-1}$ usar restrição "surrogate" da iteração anterior.

B. Inicializar:

ξ_*^0 valor da solução de $(PRC)^{\nu-1}$ ($-\infty$ para $\nu=0$)

y em zero (vetor nulo), n em zero.

Para cada iteração n, procede como segue:

I) Fazer $n \leftarrow n+1$;

Se $x_0^f \leq \xi_*^{n-1}$, não há descendente melhor-desviar para a regra de retorno. Senão:

II) Calcular $\hat{\beta}_i = \beta_i - \sum_{j \in \bar{N}} s_{ij}\, y_j \qquad i = \overline{1,t}$

Se $\hat{\beta}_i \geqslant 0 \quad \forall i \in K$, a solução parcial é viável-desviar para cálculo de f.O. (VI)

Senão:

III) Calcular $\bar{\beta}_i = \hat{\beta}_i - \sum_{j \in N^{i-}} s_{ij} \qquad \forall i \in I^+$

onde

$$N^{i-} = \{j \in N \mid s_{ij} < 0\}$$

$$I^+ = \{i \in I \mid \hat{\beta}_i < 0\}$$

$$I = K \cup L$$

Se $\exists\, \bar{\beta}_i < 0$, não pode haver descendente viável-desviar para a regra de retorno.

Senão:

IV) Formar os conjuntos:

$$FO = \bigcup_{i \varepsilon I^+} \{j \varepsilon N^{i+} | s_{ij} > \bar{\beta}_i\}$$

$$F1 = \bigcup_{i \varepsilon I^-} \{j \varepsilon N^{i-} | s_{ij} > \bar{\beta}_i\}$$

onde:

$$N^i = \{j \varepsilon N | s_{ij} > 0\}$$

aplicar sequencialmente os seguintes testes.

Se $F1 \cap FO \neq \phi$ há contradição entre os valores a que estão condicionadas as variáveis - desviar para regra de retorno.

Se $F1 \cup FO \neq \phi$ há variáveis condicionadas mas sem contradição - desviar para salto para nova solução (VIII).

Caso a solução passe por estes testes, então $F1 \cup FO = \phi$ - prosseguir.

V) Caso a solução parcial atual não seja completa, aumentá-la fixando a variável livre de mais baixo índice e voltar ao passo I. Senão desviar para a regra de retorno.

VI) Cálculo de f.o., e atualização:

Calcular

$$\hat{\beta}_i = \beta_i - \sum_{j \in \bar{N}} s_{ij} \, y_j \qquad \forall i \in L$$

$$\xi^n = \beta_{i_0} = \max_{i \in L} \beta_i$$

Se $\xi^n \leq \xi_*^{n-1}$ então fazer $\xi_*^n = \xi_*^{n-1}$

Senão fazer $\xi_*^n = \xi^n$ e mandar y associado à atual solução parcial como a melhor solução até agora.

VII) Teste de aumento de f.o.:

Calcular

$$d_{i_0} = \sum_{j \in N^{i-}_0} s_{ij_0}$$

Se $d_{i_0} < \xi_*^n - \xi^n$ não pode haver descendente viável em valor superior a ξ_*^n - desviar para regra de retorno.

Senão desviar para aumento da solução (V).

VIII) Salto para nova solução:

Construir a solução parcial tal que

$$N = \bar{N} \cup F0 \cup F1 \text{ e } =$$

$$y_j = 0 \quad \forall i \in F0$$

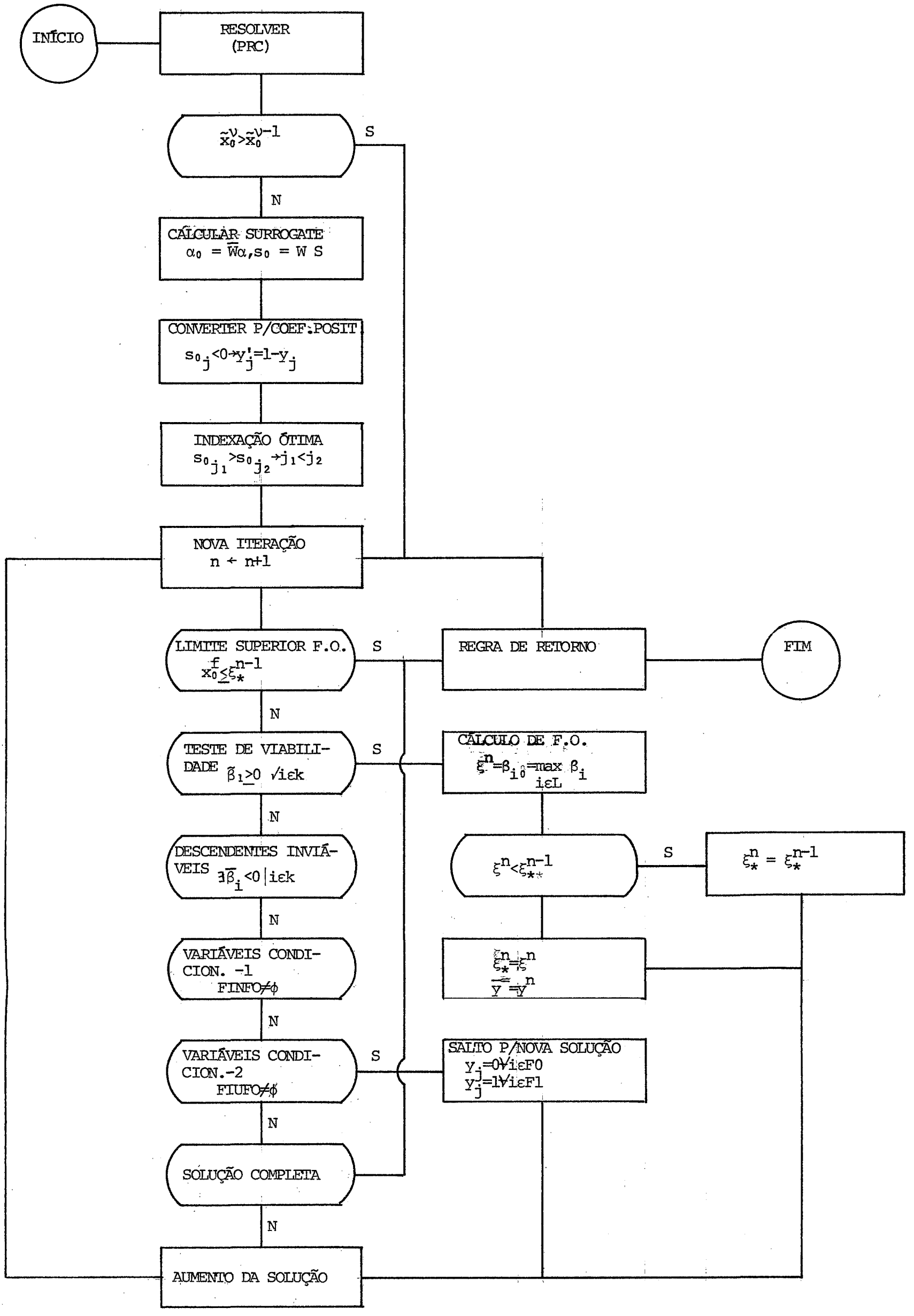
$$y_j = 1 \quad \forall i \in F1$$

y_j mantém o mesmo valor $\forall j \in \bar{N}$

Desviar para o passo I.

A regra de retorno será apresentada na próxima seção.

FIG 2 - FLUXOGRAMA PARA SOLUÇÃO DE PRI



3.2 - ESQUEMA ENUMERATIVO

O método adotado neste trabalho para resolver o problema relaxado inteiro, apresenta variação significativa em relação ao método do filtro de Balas no que se refere à ordem em que são examinadas as soluções parciais.

Balas propõe um procedimento do tipo "branch-and-bound" particularizado, visando garantir:

a) a sequência de valores das soluções do problema do filtro é não-crescente (para maximização);

b) quando um certo valor $\bar{x}_0^f$ da solução do filtro é atingido, todas as soluções viáveis do filtro tais que $x_0^f > \bar{x}_0^f$ já foram implícita ou explicitamente enumeradas.

Desta forma, o primeiro termo da sequência de soluções para o problema do filtro que for solução viável para o problema inteiro será também solução ótima deste.

Este procedimento exige que se armazenem todas as soluções parciais ativas, isto é, aquelas que ainda não geraram descendentes nem foram submetidas a testes. Esta contingência re

presenta uma dificuldade computacional significativa e a eficiência resultante ainda está por ser estabelecida.

Neste estudo optou-se pelo uso da regra de retorno ("Backtracking") na forma já usada com bons resultados em programas derivados do algoritmo aditivo de Balas (vide, por exemplo, Woiler |12|). A seguir, resumimos esse esquema enumerativo.

A solução parcial a ser aumentada será chamada S_q. Seja K o número de variáveis fixas, isto é, com valores designados na solução parcial S_q.

O esquema enumerativo que será usado consiste de gerar uma sequência de soluções parciais determinando a melhor descendente viável das mesmas. Caso esta descendente seja a melhor encontrada até então, ela é armazenada; em caso contrário, guarda-se aquela anteriormente armazenada e termina-se S_q. Esta também será determinada se não tiver descendente viável.

São introduzidas algumas definições a fim de descrever o algoritmo usado.

Seja u_k, $k = 1,K$ uma variável tal que $u_k = -j$ se x_j é uma variável marcada da solução parcial S_q e $u_k = j$ em caso contrário.

Consideremos a solução parcial em que são designadas, e com o mesmo valor presente, todas as variáveis fixas da presente solução parcial exceto uma, que é livre. Caso se possa concluir que em todas as descendentes ainda não enumeradas daquela solução esta última variável terá apenas o seu valor atual, esta é marcada. Como o aumento das soluções parciais é feito pela inclusão de variáveis livres com o valor inicial um, x_j assume os valores um ou zero para variáveis marcadas mas, apenas o valor um, para variáveis fixas não-marcadas. Para as variáveis livres, $u_k = 0$. Será designado por k^* o valor k tal que $u_k^* > 0$ e $u_{k+1}^*, u_{k+2}^*, u_K < 0$.

Quando, em decorrência dos testes apresentados na seção anterior, conclui-se que podem ser implicitamente enumeradas as descendentes da solução parcial corrente, uma nova solução parcial é escolhida determinando k^* e alterando a solução parcial atual como segue. Para $k > k^*$ fazemos x_j livre, para $k = k^*$ x_j tem seu valor alterado de um para zero e passa a ser marcada, isto é, $u_k^* = -j$. O número de variáveis fixas passa a $K - k^*$.

Em seguida recomeça-se uma nova iteração, recalculando os valores de x_0, $\bar{\beta}_i$ e d_i associados à nova solução parcial.

O esquema enumerativo adotado está esquematizado no

fluxograma da Figura 3, e corresponde ao bloco designado por regra de retorno na seção anterior.

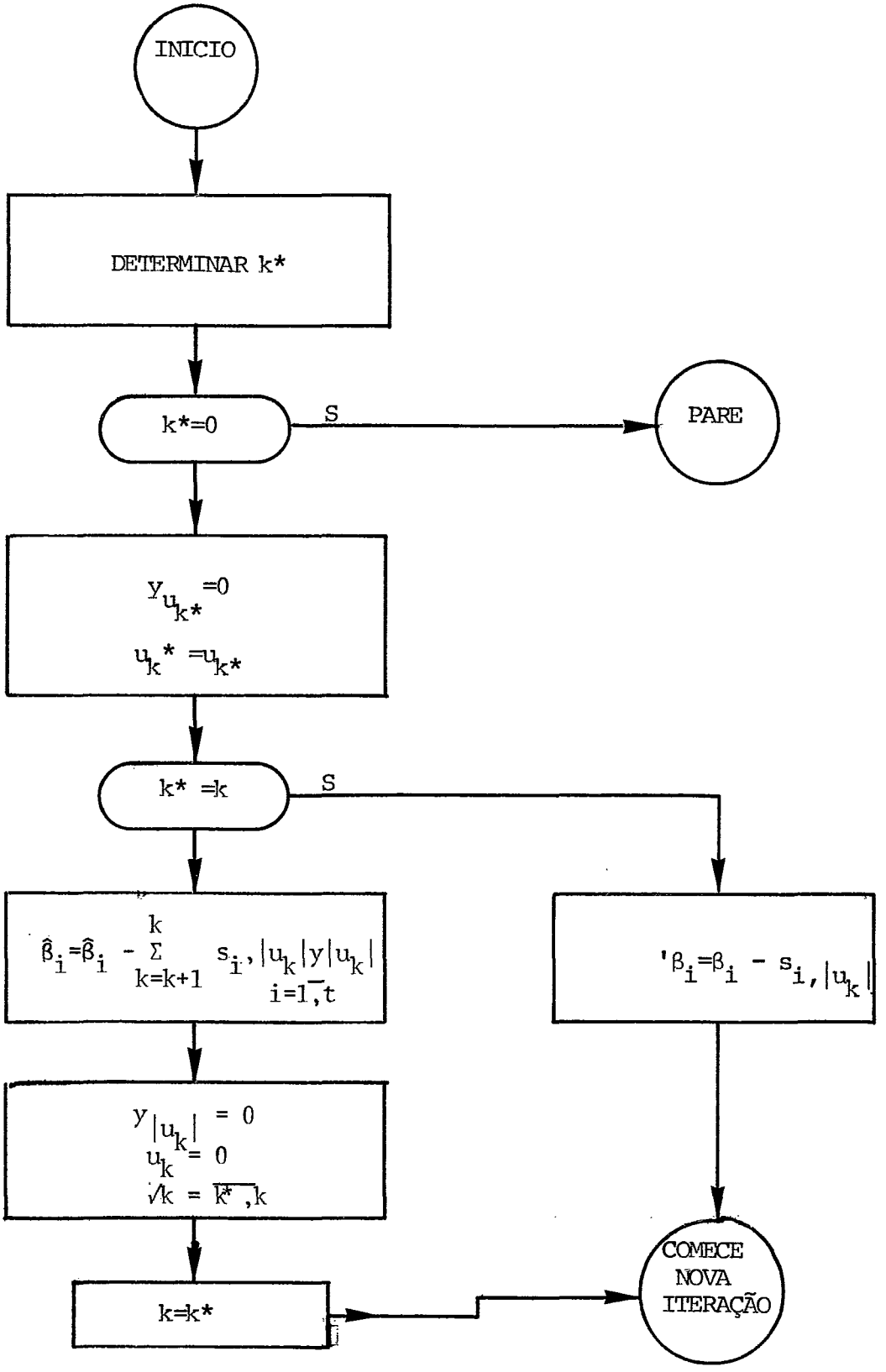


Fig. 3 - ESQUEMA ENUMERATIVO

3.3 - PROBLEMA RELAXADO CONTÍNUO

A geração de restrição "surrogate" ou do filtro requer o conhecimento da solução dual do correspondente contínuo de $(PRI)^{\nu}$, que denominamos problema relaxado contínuo $(PRC)^{\nu}$. De acordo com a seção 3.1, PRC é formulado como segue:

$$\max x_0 \tag{3.1}$$

$$x_0 + \sum_{j=1}^{q} s_{ij}\, y_j \leqslant \beta_i \qquad i = \overline{1,t} \tag{3.2}$$

$$y_j \leqslant 1 \qquad j = \overline{1,q} \tag{3.3}$$

$$y_j \geqslant 0 \qquad j = \overline{1,q} \tag{3.4}$$

$$x_0 \text{ livre} \tag{3.6}$$

Em uma dada iteração ν, $(PRC)^{\nu}$ difere de $(PRC)^{\nu-1}$ da iteração anterior apenas pelo acréscimo de uma ou duas restrições (3.2). Assim, após a primeira iteração, a resolução de $(PRC)^{\nu}$ reduz-se a pós-otimização da solução obtida na iteração anterior.

A análise de pós-otimização de $(PRC)^{\nu}$ consiste de considerar o efeito resultante do acréscimo de uma ou duas restrições sobre a solução do problema. No caso de se usar o método do simplex revisado, como é o caso deste estudo, sabe-se que a

matriz inversa da base deve ser aumentada de uma ou duas linhas (e portanto colunas) antes de se retomar o método simplex (vide por exemplo, Simonnard |11|).

A fim de evitar a complexidade computacional derivado de uma matriz inversa cujas dimensões variam durante a execução do programa, optou-se por resolver o problema dual.

Chamando w_k as variáveis duais e ordenando-as convenientemente, resulta a seguinte formulação para $(PRC)^{\nu}$ dual:

$$\min \sum_{j=1}^{q} w_j + \sum_{i=1}^{t} \beta_i w_{q+1}$$

$$w_j + \sum_{i=1}^{t} s_{ij} w_{q+1} \geq 0 \qquad j = \overline{1,q}$$

$$\sum_{i=1}^{t} \alpha_i w_{q+1} = 1$$

Como o efeito do aumento do número t de restrições do primal traduz-se por acréscimo de variáveis w_{q+1} no dual, a análise de pós-otimização para o dual pode ser feita sem alteração da matriz inversa da base. Basta acrescentar colunas à matriz correspondente ao quadro inicial do simplex.

Observe-se que $(PRC)^{\nu}$ dual já dispõe, a menos de u

ma variável, de uma base inicial de partida fornecida pelas variáveis duais $w_j (j = \overline{1,q})$ correspondentes à limitação superior (≤ 1) das variáveis primais y_j. (Supõe-se que o problema MIX apresente restrições com termo independente $b \geqslant 0$, a que resulta, em $\beta \geqslant 0$, pois u^ν, $v^\nu \geqslant 0$. Em caso contrário, invertendo o sinal da restrição, pode ser incluída na base a variável de folga).

Incluindo variáveis de folga e completando a base inicial com o acréscimo de uma variável inicial com o acréscimo de uma variável artificial na última restrição, tem-se o quadro inicial, esquematizado na Figura 4. As colunas estão renumeradas de forma a levar em conta a inclusão das (q+1) variáveis de folga (à esquerda) e da variável artificial (à direita da base inicial).

O problema é resolvido em cada iteração utilizando uma sub-rotina que executa o método simplex. Inicialmente, o problema é resolvido a partir do quadro inicial (caso em que a matriz inversa é obviamente I_{q+1}). Para soluções posteriores, fornece-se as colunas acrescidas ao quadro inicial e a sub-rotina o método simplex a partir do último quadro.

Observe-se que pelo menos uma variável w_{q+i}, correspondente à restrição (3.2) com $\alpha_i = 1$ deve estar presente ao se solucionar $(PRC)^\nu$ dual. Caso contrário, este será inviável,

pois o primal será ilimitado.

Por outro lado, a partir de um certo número de iterações, pode-se eventualmente retirar do quadro inicial colunas correspondentes a restrições (3.2) do primal, à medida que novas restrições do mesmo tipo são acrescentadas, a fim de limitar as dimensões do quadro inicial. No entanto, essa sistemática requer um método que permita determinar quais restrições (do primal) podem ser dispensadas. No presente estudo não foi considerada a aplicação de uma tal sistemática.

A solução de $(PRC)^{\nu}$ pode ser eventualmente inteira, especialmente em problemas a coeficientes inteiros. Nas iterações em que isso ocorrer, pode-se dispensar a solução de $(PRI)^{\nu}$ com sifnificativa economia computacional. Por esse motivo, inclui-se um teste de integralidade após a solução de $(PRC)^{\nu}$.

	i = 1	q	2q				2q+t+1
j = 0		0	1	ZM	β_1	β_2	β_t
1 … q	0	$-I_q$	I_q	0	s_1	s_2	s_t
q + 1	1	0	0	1	α_1	α_2	α_t

FIG. 4 - Quadro Inicial do Simplex de PRC

CAPÍTULO IV

SUB-PROBLEMA CONTÍNUO

Em cada iteração do método de decomposição de Benders deve ser resolvido um sub-problema contínuo $(SPC)^{\nu}$ dado, conforme o Capítulo II, por:

$$\begin{aligned} \min\ & (b - Fy^{\nu})\, u \\ & A^{T} u \geqslant C \\ & u \geq 0 \end{aligned} \qquad (4.1)$$

Conforme a solução de $(SPC)^{\nu}$, o algoritmo de decomposição pode terminar ou prosseguir, neste caso com acréscimo de uma ou duas restrições ao último problema $(PRI)^{\nu}$.

Dois casos de terminação podem ocorrer após a resolu

ção de $(SPC)^{\nu}$ em uma dada iteração. Se $(SPC)^{\nu}$ é inviável, MIX é inviável ou ilimitado. Se $(SPC)^{\nu}$ tem solução ótima finita tal que:

$$(b - Fy^{\nu})\, u^{\nu} = x_0^{\nu} - fy^{\nu} \tag{4.2}$$

então MIX tem solução ótima (x^{ν}, y^{ν}) com valor x_0^{ν}, sendo x^{ν} a solução dual de $(SPC)^{\nu}$.

Por outro lado, caso o algoritmo não termine após a resolução de $(SPC)^{\nu}$ em uma certa iteração, a solução deste problema é usada para gerar as novas restrições a $(PRI)^{\nu}$. Se $(SPC)^{\nu}$ tiver solução ótima u^{ν} e:

$$(b - Fy^{\nu})\quad u^{\nu} < x_0^{\nu} - fy^{\nu} \tag{4.3}$$

a restrição a acrescentar é dada por:

$$x_0 - fy + u^{\nu}Fy \leq u^{\nu}b \tag{4.4}$$

Se $(SPC)^{\nu}$ tiver solução ilimitada segundo

$$u^{\nu} + \lambda v^{\nu} \qquad \lambda \geq 0 \tag{4.5}$$

onde u é o vértice do cone poliédrico C correspondente à última base do simplex e v^{ν} é a direção do raio extremo de C en

contrado.

Se neste caso, também se verificar a condição (4.3), acrescenta-se ainda uma segunda restrição dada por (4.4). Deve ser enfatizado que os diversos vetores que podem ser necessários, conforme a solução de $(SPC)^{\nu}$, isto é, x^{ν}, u^{ν} e v^{ν}, são obtidos diretamente do último quadro do simplex. Em particular v^{ν} é dado por:

$$v_j^{\nu} = r_j \qquad j = \overline{1,p}$$

onde r_j são os elementos (negativos ou nulos) da coluna escolhida pelo critério de entrada.

Como $(SPC)^{\nu}$ em uma dada iteração difere do problema $(SPC)^{\nu-1}$ resolvido na iteração anterior apenas pela troca do vetor custo, a resolução do mesmo reduz-se, a menos da primeira iteração, a análise de pós-otimização.

A mudança do vetor custo altera a primeira linha da matriz inversa ampliada. Sendo $(1, C_B \cdot B^{-1})$ esta linha antes da modificação do vetor custo e ΔC_B a diferença entre o novo vetor e o anterior, a nova primeira linha será obtida somando $(0, \Delta C_B \cdot B^{-1})$ a $(1, C_B \cdot B^{-1})$. Além disso, o novo vetor custo deverá ser incluído no quadro inicial em substituição ao anterior (vide por exemplo, Hadley |9|).

Feitas as alterações acima, o simplex pode ser retomado a partir do último quadro. Neste trabalho, a atualização da matriz inversa ampliada e do quadro inicial são feitas à parte da sub-rotina simplex e fornecidos a esta como argumentos. Desta forma, a mesma sub-rotina pode ser usada para resolver SPC e PRC alternadamente.

Para resolver SPC pela primeira vez, o quadro inicial é obtido à parte da sub-rotina simplex, a partir de um "quadro preliminar" em que o problema é apresentado na forma (4.1). O programa inclui um módulo de inicialização para SPC em que se transforma o problema para o formato requerido no quadro inicial do simplex.

Nessa fase são invertidos sinais de restrições ($c_i < 0$) e incluídas variáveis de folga e artificiais, conforme necessário. O quadro inicial obtido depois dessas transformações é esquematizado na Figura 5.

O primeiro problema SPC é resolvido a partir desse quadro inicial. A sub-rotina exclui as eventuais variáveis artificiais da base pelo método "Big M" e aplica em seguida o método do simplex.

	$i = 1$	m	$m + nfp$	$m + nfp + p$
$j = 0$		$(b - Fy^{\nu})$		$\pi_{m+nfp+j} = (-1)^{1-kj}\ ZM$
1	$(-1)^{k_j} c_j$	$\left[(-1)^{k_j} a_{ji}\right]$	-1	I_p
			-1	
			-1	
			-1	
p			-1	

$$k_j = \begin{cases} 0 & \text{se} \quad c_j > 0 \\ 1 & \text{se} \quad c_j < 0 \end{cases}$$

FIG. 5 - Quadro Inicial de SPC

CAPÍTULO V

DESCRIÇÃO DO PROGRAMA

O algoritmo apresentado nos capítulos anteriores, visando a resolução de problemas de programação mista com variáveis bivalentes, foi codificado em linguagem ALGOL para o B6700 do NCE/UFRJ.

O problema a ser resolvido deve ser formulado como segue:

$$\max\ cx + fy$$

$$Ax + Fy \leq b$$

$$x_j \geq 0 \qquad j = 1,\ldots, p$$

$$y_j \in \{0,1\} \qquad j = 1,\ldots, q$$

onde A e F são matrizes de dimensões (m x p) e (m x q), enquanto que b, c, f são vetores de dimensões m, p, q, respectivamente - todos os elementos de A, F, b, c, f são números reais quaisquer.

Obviamente problemas de minimização também podem ser colocados na formulação acima invertendo o sinal da função objetivo.

De acordo com o método adotado, o programa deve executar basicamente três funções:

- resolução do problema relaxado inteiro;
- resolução de problemas de programação linear: sub-problema contínuo e problema relaxado contínuo;
- coordenação das funções acima, incluindo testes de parada, geração de restrições adicionais e "surrogate" a PRI, etc.

Para execução dessas funções, adotou-se uma estrutura básica de três elementos. Consiste de um bloco que soluciona PRI, e de uma sub-rotina que resolve SPC e PRC pelo método simplex revisado, contidos dentro de um bloco incluindo também as instruções de coordenação citadas.

5.1 - DESCRIÇÃO DE VARIÁVEIS

Antes de entrar em pormenores sobre o programa, é necessário introduzir a definição das variáveis usadas. Os símbolos correspondentes usados nos capítulos anteriores são dados entre parênteses após cada definição, quando aplicáveis.

- Variáveis do problema misto:

M	número de restrições (m)
P	número de variáveis contínuas (p)
Q	número de variáveis binárias (q)
A|I,J|	I = 0,...,M e J = 1,...,P é a matriz dos coeficientes das variáveis contínuas, incluindo o vetor custo correspondente na linha zero (C,A)
F|I,J|	I = 0,...,M e J = 1,...,Q é a matriz dos coeficientes das variáveis binárias, incluindo o vetor custo correspondente na linha zero (f,F)
B|I|	I = 1,...,M é o vetor do termo independente de MIX (b)
X|J|	J = 1,...,P são as variáveis contínuas (x)
Y|J|	J = 1,...,Q são as variáveis binárias (y)

- Variáveis de Decomposição:

01, 00	variáveis booleanas que indicam o tipo de restrição e acrescentar a PRI na iteração corrente.
FOY	produto dos vetores F|0,J| * Y|J| (f.y)
XI, XO	limites inferior e superior do valor da solução de MIX.

- Argumentos da Sub-rotina Simplex:

(Em cada definição são relacionados respectivamente os símbolos do argumento formal, do argumento usado ao resolver SPC e do argumento usado para resolver PRC).

N,P,NF	número de restrições
NN,NA,TQ	número de variáveis (incluindo de folga e artificiais)
MM,MA,MF	matriz dos coeficientes das variáveis, incluindo vetor custo na linha zero (linhas J= 0, ...,N ou P ou NF e colunas I = 1,...,NN ou NA ou TQ)
FF,FA,FFF	vetor que modifica o vetor custo de forma a excluir as variáveis artificiais da base (J = = 0,...,N ou P ou NF)
BB,BA,BF	vetor do termo independente (linhas J = 0,...,N ou P ou NF)

MP,MPA,MPF	matriz produto ou inversa da base (ampliada) (linhas e colunas J = 0,...,N ou P ou NF)
IB,IBA,IBF	vetor da base - indica a variável básica associada a uma linha J (J = 1,...,N ou P ou NF)
VART, VARTA, VARTF	vetor que indica a variável artificial associada a uma linha J (J = 1,...,N ou P ou NF) nulo quando não há variável artificial na linha.
Z,ZA,ZF	vetor solução ótima (I = 1,...,NN ou NA ou TO)
DD,DA,DF	linha do vetor custo no quadro atualizado (I = 1,...,NN, NA ou NF)
NI,NU	número de iterações do método de decomposição
ESTADO, STATUS	indica se a solução obtida é ótima (= 0), ilimitada (= 1) ou iviável (= -1).

- Outras variáveis usadas na Sub-rotina Simplex:

ZMAN	vetor correspondente à linha zero da matriz inversa ampliada, após modificação do vetor custo para excluir as variáveis artificiais da base (J = 0,...,N ou P ou NF)
ZM	valor do "Big M" ao excluir as variáveis artificiais da base
EPS	zero computacional
DMAX	valor máximo de DD |J|
IE	índice da variável escolhida para entrar na base.

PI|J|	coluna do quadro atualizado correspondente a IE
RR	valor corrente de BB|J|/PI|J|
RMIN	valor mínimo de RR
JS	linha associada à variável escolhida para deixar a base
IS	variável que deixa a base.

- Variáveis usadas para compor o quadro do simplex para SPC:

NVF,NFP,NFN	número de variáveis de folga, de folgas positivas e de folgas negativas
SENS|J|	indica o sentido da desigualdade para a restrição da linha J (J = 1,...,P)
DC|J|	vetor que indica a diferença do coeficiente de custo da variável básica associada à linha J (J = 1,...,P) - usado para pós-otimização de SPC
DCB|J|	vetor a ser somado à linha zero da matriz inversa ao alterar os coeficientes de custo (J= = 1,...,P).

- Variáveis que definem restrições adicionais a PRI:

ALFA|I|	I = 1,...,T, coeficiente de x_0 na restrição $I(\alpha_i)$

UB|I|	I = 1,...,T termo independente da restrição $i(\beta_i)$
S|I,J|	I = 1,...,T e J = 1,...,Q coeficientes de Y|J| na restrição $I(s_{ij})$
U|I|	I = 1,...,M vetor que fornece a solução básica do último quadro de SPC anterior.
V|I|	I = 1,...,M vetor que fornece um raio extremo do cone C quando SPC é ilimitado.
T	número de restrições adicionais.

- Variáveis que definem a restrição "surrogate" ou filtro para PRI:

k 1|I|	I = 1,...,T vetor solução de PRC dual.
ALFAF	coeficiente de x_0 (α_0)
YF	termo independente (β_0)
SF|J|	coeficientes das variáveis (s_{0j})
XOF	solução corrente do problema do filtro
MC|J|	indica se y_j foi convertido a $y'_j = 1 - y_j$
JF|J|	define a indexação ótima para o filtro.

- Variáveis para exame de cada solução parcial de PRI:

YS|J|	J = 1,..., $\aleph$ vetor solução parcial em exame
XφM	valor da melhor solução encontrada até a iteração corrente.

YV|I|	I = 1,...,T termo independente da restrição I considerando variáveis fixas como constantes ($\tilde{\beta}_i$)
D|I|	I = 1,...,T soma dos coeficientes negativos das variáveis livres da restrição I (valor absoluto) (d_i)
F1|J|, F0|J|	J = 1,...,Q conjuntos de variáveis booleanas que indicam se a variável (livre) Y|J| está condicionada a assumir o valor um ou zero (F1, F0)
YM	valor mínimo de YV|I| para ALFA I = 1 linha associada a YM.

- Variáveis usadas no esquema enumerativo:

K	número de variáveis designadas na solução parcial corrente (K)
US|J|	J = 1,..., K indica se a variável Y|J| tem valor designado e, em caso positivo, se é marcada (u_k)
KSTAR	indica variável não-marcada de índice j mais alto (k^*)

- Contadores: Os contadores destinam-se a fornecer dados para análise da eficiência do método adotado para resolução de PRI e da importância relativa dos testes empregados.

NU	número de iterações do método de decomposição
ITER	número de iterações do método simplex
C1	número de iterações do método do filtro
C2	número de desvios no passo (I) do método filtro
C3	número de desvios no passo (II) do método filtro
C4	número de desvios no passo (III) do método filtro
C5	número de desvios no passo (IV-1) do método filtro
C6	número de desvios no passo (IV-2) do método filtro
C7	número de desvios no passo (VII) do método filtro
C8	número de soluções inteiras de $(PRC)^{\nu}$, $\nu = 1,2,...$

- Marcadores de tempo (segundos):

WMIX1	tempo gasto na última iteração de MIX
WSPC1	tempo gasto pela sub-rotina simplex no último SPC

WPRC1	tempo gasto pela sub-rotina simplex no último PRC
WMIX2	tempo acumulado para resolução de MIX
WSPC2	tempo acumulado pela sub-rotina simplex para SPC
WPRC2	tempo acumulado pela sub-rotina simplex para PRC
W1	tempo decorrido a cada iteração do método do filtro.

5.2 - LISTAGEM

A listagem do programa consta das figuras a seguir. Foram incluídos comentários a cada etapa a fim de facilitar a compreensão do mesmo.

Apresenta-se aqui alguns comentários complementares (apresentados na sequência da listagem).

- Bloco Externo:

  As dimensões do problema (m,p,q) são lidas antes de entrar no bloco em que o mesmo é resolvido a fim de dimensionar matrizes e vetores adequadamente.

- Declarações de Variáveis:

São aqui declaradas todas as variáveis do programa, com exceção das da sub-rotina simplex e de conjuntos de dimensões dinâmicas usados para PRI (ver adiante).

- Declarações de Formato:

Muito numerosas dado o caráter experimental do programa, a fim de permitir acompanhamento de passagens intermediárias. Para aplicações normais, deverão ser muito reduzidas. O mesmo vale para as instruções de impressão.

- Sub-rotina Simplex (revisado):

Resolve problemas de programação linear (minimização) a partir de um quadro inicial dado (vide Figura 5). Usa o método "Big M" para exclusão de variáveis artificiais.

- Atualização da linha zero do quadro (simplex):

Nesta etapa, o vetor DD|I| da primeira linha do quadro é obtido como produto matricial FF * MP * MM onde MP * MM representa o quadro atual (B_1^{-1} A na notação tradicional, isto é, denotando por B_1^{-1} a matriz inversa estendida e por A a matriz dos coeficientes, inclusive do vetor custo, referente ao quadro inicial).

O produto matricial de FF pelo quadro atual equivale a subtrair da linha zero as linhas associadas a variáveis artificiais multiplicadas por ZM, considerando que FF|0|= = -1, FF|J|= ZM ∀J/VART|J| ≠ 0 FF|J| = 0 ∀J/VART|J|= = 0. Quando uma variável artificial VART|J| é excluída da base, faz-se FF|J| = 0, o que equivale a eliminá-la do quadro.

- Critério de Entrada (simplex):

Note-se que as variáveis básicas e artificiais não são consideradas como candidatas para entrar na base.

- Solução ótima ou Inviável (simplex):

A existência de variáveis artificiais não nulas na base terminal, implicará em solução inviável. A solução ótima é obtida em SFIM.

- Critério de saída (simplex):

PI||J|| representa a coluna da variável que entra na base; é obtida multiplicando a inversa da base pela coluna correspondente do quadro inicial.

- Pivoteamente (simplex):

Emprega-se a forma do produto da inversa.

- Quadro Inicial de SPC:

  Na primeira iteração do método de decomposição, constrói-se o quadro esquematizado na Figura 5.

- Solução de SPC:

  Antes de chamar a sub-rotina simplex, atualiza-se o vetor custo (quadro inicial) e a primeira linha da matriz inversa.

- SPC tem Solução Ótima (ou ilimitada):

  Se não for satisfeita a condição de otimalidade, determina-se o tipo de restrições a acrescentar (U01 e U00). Se for satisfeita o algoritmo termina.

- Declaração de Conjuntos Dinâmicos:

  Como o número de restrições de PRI aumenta a cada iteração, é conveniente que as dimensões dos vetores e matrizes correspondentes façam o mesmo, a fim de minimizar o uso de memória. Esse dimensionamento dinâmico é feito declarando essas variáveis em um bloco interno. No entanto, algumas variáveis devem ter seus valores guardados (declaração OWN) de uma iteração do método de decomposição para a seguinte. Para estas (ALFA, MF, DF), é necessário adotar um dimensionamento fixo em todo o pro

grama.

- Quadro Inicial de PRC (dual):

  Se $T = 1$ PRC tem solução imediata, dispensando o método do simplex. Na primeira iteração em que $T \neq 1$, constrói-se o quadro esquematizado na Figura 4.

- Teste de Integralidade:

  Se as variáveis duais ótimas de PRC (dual) são inteiras a menos de um número muito pequeno (EPS), toma-se a solução inteira correspondente como ótima para PRI. O valor do x_0 é calculado e partir dos $Y|J|$.

- Construção do Filtro e Solução de PRI (Inicialização):

  $W|K|$, $K = 1,\ldots,T$ são os valores ótimos das variáveis de PRC dual associadas às restrições geradas pelo método de decomposição (isto é, as restrições de PRI).

- Filtro Anterior:

  Se for usada a mesma "restrição surrogate", desvia-se para LE e LF. Caso contrário, é feita a reconversão $y_j = 1 - y_j'$ para variáveis convertidas anteriormente (dadas por $MC|J| = 1$).

5.3 - DADOS DE ENTRADA

As instruções de leitura do programa prevêm entrada de dados por cartões em formato livre (dados separados por vírgulas) na seguinte ordem:

1) M, P, Q

2) B|1|, B|2|,..., B|M|

3) Y|1|, Y|2|,..., Y|Q| (vide abaixo).

4+0) A|0,1|, A|0,2|,... A|0,P|, F|0,1|, F|0,2|,... F|0,Q|

4+M) A|M,1|, A|M,2|,... A|M,P|, F|M,1|, F|M,2|,... F|M,Q|

Caso não se conheça um vetor inteiro inicial, fornecer Y|J| = 0, J = 1,...,Q.

CAPÍTULO VI

COMENTÁRIOS

6.1 - VARIANTE

No algoritmo adotado para solução do problema relaxado inteiro, a restrição "surrogate" não é apenas uma restrição a mais. Seu papel é preponderante na busca de uma solução ótima, pois determina a solução parcial inicial e a variável de entrada em cada iteração.

De acordo com o método do filtro, toma-se como solução parcial inicial a solução ótima do problema do filtro. Para tanto, impõe-se que a restrição "surrogate" tenha apenas coeficientes positivos ou nulos (pela conversão $y'_j = 1 - y_j$ para $s_{0j} < 0$) e toma-se a solução nula ($y_j = 0$, $\forall_j = 1,\ldots,q$) como solução parcial inicial.

Por outro lado, o problema relaxado inteiro a ser resolvido em cada iteração difere do de iteração anterior apenas pelo acréscimo de uma ou duas restrições (propriedade (e) do PRI). Assim, é razoável presumir que o uso da informação representada pela última solução de PRI, permitirá uma "boa" escolha da solução parcial inicial.

A variante proposta adota como solução parcial inicial, a última solução do PRI.

A transformação $y_j' = 1 - y$; é usada quando $\bar{y}_j = 1$ na solução anterior, conforme necessário.

Nessa variante deve ser eliminado o teste de limite superior da função objetivo ($x_0^f \leqslant \xi_*^{n-1}$) e modificado o critério de aumento da solução. Essas alterações devem-se a não mais valerem todas as propriedades do filtro.

Na escolha de uma nova regra para aumento das soluções, parece interessante considerar o critério proposto por Woiler |12|, pelo qual se minimiza a inviabilidade da restrição de máxima inviabilidade.

Woiler considerou problemas totalmente inteiros. Naquele caso, ao alcançar uma solução parcial viável, pode-se implicitamente enumerar suas descendentes.

Na resolução do PRI pelo esquema enumerativo adotado, após alcançar uma solução parcial viável, ainda é necessário considerar suas descendentes. Assim, a solução parcial viável também é aumentada.

Daí resulta que a construção de uma sequência de soluções parciais do PRI, pela regra de aumento da solução, pode ser visualizada em duas fases. Inicialmente, busca-se atingir uma solução parcial viável, como no algoritmo aditivo. Após encontrar uma solução parcial viável, continua-se a aumentar a solução visando maximizar a mínima folga dentre as restrições $l \varepsilon L$.

No algoritmo adotado, a regra de aumento da solução constroi uma sequência de soluções parciais à qual está associada uma sequência não-crescente de soluções do problema do filtro.

Nesta variante, propõe-se usar o critério de Woiler, considerando o papel especial da restrição de mínima folga em $l \varepsilon L$, a qual determina o valor da função objetivo associado a cada solução parcial. Dessa forma, a partir da obtenção de uma solução parcial viável, o aumento das soluções parciais é dirigido para aumentar o valor da função objetivo do PRI.

Note-se que esta variante para o critério de entrada é especialmente importante após a obtenção de uma solução parcial viável. De fato, o algoritmo adotado, seguindo a proposta

de Balas, só dispõe de poucos testes efetivos nesta fase, pois todos os testes de viabilidade deixam de atuar então.

Além da alteração do critério de entrada, parece conveniente pesquisar testes adicionais que sejam a contrapartida, nesta segunda fase, dos testes de viabilidade.

6.2 - SUB-ROTINA DE PROGRAMAÇÃO LINEAR

O programa desenvolvido inclui uma sub-rotina para resolução de problemas de programação linear (SPC e PRC) pelo método simplex revisado. Também prevê os procedimentos de pós-otimização aplicáveis a esses dois problemas.

Optou-se pelo uso de sub-rotina adaptada a esta aplicação e incorporada dentro do programa em lugar de usar um código comercial de programação linear. Esta opção foi considerada a mais adequada para as finalidades do presente estudo.

Tendo em vista que o primeiro objetivo deste trabalho é estudar o método de solução para o problema relaxado inteiro, o programa inclui marcadores de tempo que permitem distinguir o tempo consumido pela sub-rotina simplex do tempo total.

Para tratamento de problemas com grande número de variáveis contínuas, recomenda-se considerar o uso de um código comercial para resolução e pós-otimização de SPC e PRC.

Sabe-se que em trabalhos anteriores já foi usado dessa forma o "Mathematical Programming System" (IBM) |6|.

DECLARAÇÕES (1)

D E C O M P

```
COMMENT PROGRAMA DECOMP;
COMMENT DIMENSOES;
BEGIN
FILE CARTOES (KIND=READER): FILE IMPRESS (KIND=PRINTER);

INTEGER M, P,Q,IMAX,UU,VV;LABEL L7;UU:=0;READ (CATOES,/,VV);
L7:UU:=JU+1;READ (CARTOES,/,M,P,Q,TMAX);
BEGIN
COMMENT DECLARAÇOES;
  INTEGER K,L,R,J1,J2,IM,KSTAR   ,C1,C2,C3,C4,C5,C6,C7,I,J:INTEGER C8;

  INTEGER NU,T,K1,ITER,IE,IS,JS,NVF,NFP,NFN,MA,NF,STATUS;INTEGER TQ;
  REAL 7M,XQ,EPS,DMAX,RMIN,RR,FOY,XOC,W1,WO,XOF,YF,ALFAF,XOM,XI;
  REAL EPSS;REAL X1,YM;
  REAL WMIX1X2,WSPC1,WSPC2,WPRC1,WPRC2;
  INTEGER ARRAY SENS|1:P|, VAPTA|1:P|,VARTF|1:Q+1|, Y|1:Q|,YS|1:Q|;
  INTEGER ARRAY IBF|1:Q+1|;
  REAL ARRAY A|0:M,1:P|,F|0:M,1:Q|,8|1:M|,X|1:P|,PI|0:P+C|,SF|1:Q|;
  REAL ARRAY MA|0:P,1:M+2-P|,BA|0:P|,FA|0:P|,MPA|0:P,Q:P|,ZA|1:M+2-P|;
  REAL ARRAY DA|1:M+2-P|,FFF|0:Q+1|,BF|Q:Q+1|,MPF|Q:Q+1,0:Q+1|;
  REAL ARRAY XOI|0:50|; REAL ARRAY DC|1:P|,DCB|1:P|;
  BOOLEAN UO1,UOO;
  LABEL SPC,ILIM,FIM,RESTR;
  FORMAT FMTA (//"**SOL. INV. OU ILIM.");
  FORMAT FMTB (//"**SOL. OTIMA"/"  XO=",F12,5/10(X2,F12,5));

  FORMAT FMIC (/"**",60I2);
  FORMAT FMTD (//"**SOL. INV.");
  FORMAT FMA1 (//"  INICIALIZAÇAO"/);
  FORMAT FMA2("  XOO=",F6.1,X6,"X) ="R8.1);
  FORMAT FMA3("  ALFA|I|=");
  FORMAT FMA4(X2,20I6);
  FORMAT FMA5("  UB|I|=");
  FORMAT FMA6(X2,20F6.1);
  FORMAT FMA7("  S|I,J|=");
  FORMAT FMA8(//"  FILTRO ANTERIOR"/"  YV|I|=");
  FORMAT FMAØ(BBY(J)=",6Ø12);
  FORMAT FMA10(//"  RESTRIÇAO F");
  FORMAT FMA11("  YF=",F6.1,"ALFAF=",F6.1);
  FORMAT FMA12("  SF|J|=");
  FORMAT FMA13(//"  CONVERSAO P/ COEF. POSIT."/"  MCI|J|=");

  FORMAT FMA14(//"  INDEXAÇAO OTIMA"/"  JF|J|=");
  FORMAT FMA15(//"  SOMA COEF. NEGAT. LIVRES"/"  D|I|=");
  FORMAT FMA17("  XOF=",F6.1,X6,"TEMPO=",F6.1,X6,"CONTADORES=",7I6);
  FORMAT FMA21(//"  PODE TER DESC. VIAVEL");
  FORMAT FMA22(//"  VAR. LIVRES CONDIC. A UM E ZERO");
```

```
  FORMAT FMA23(//" F1 E FQ DISJUNTOS");
  FORMAT FMA24(//" F1 E FQ VAZIOS");
  FORMAT FMA25(//"**RESULTADOS FINAIS");
  FORMAT FMA26(" XQ =",F6.1,X6,"TEMPO=",F6.1,X6,"CONTADORES=",7I6/
  " Y|J|=");
  FORMAT FMA27(//"**NAO HA DESC. MELHOR: RETORNO");
  FORMAT FMA28(//"**SOLUÇAO INVIAVEL");
  FORMAT FMA29(//"**INICIO");
FORMAT FMS0(//" NOVA BASE DO SIMPLEX"/" ITER=",I3,X6,"IE=",I3,X6,"IS="
,I3,X6,"BB(0)=",R8.1/" IB(J)=");
  FORMAT FMS3(//" LINHA ZERO DO QUADRO ATUAL"/" DD(I)=");
  FORMAT FMS4(//" COLUNA IE=",I3," DO QUADRO ATUAL"/" PI(J)=");
  FORMAT FMS5 (//" PIVOTEAMENTO"/" JS=",I3/" BB(J)=");
  FORMAT FMS6(/ " MP(Q,I)=");
FORMAT FMS7(" BB(J)=");
FORMAT FMS8(//"**FIM DO SIMPLEX"/" STATUS=",I1);
  FORMAT FMT1(//"**DADOS DO PROBLEMA MISTO"//" VETOR DE RESTRIÇOES"/
  " 8(I)=")
  FORMAT FMT2(X2,14R9.2);
  FORMAT FMT3(//" VETOR INTEIRO INICIAL"/" Y(J)=");
  FORMAT FMT4(X2,14I9);
  FORMAT FMT5(//" MATRIZ DA PARTE CONTINUA"/" A(I,J)=");
  FORMAT FMT6(//" MATRIZ DA PARTE INTEIRA "/" F(I,J)=");
  FORMAT FMT7(//"**QUADRO INICIAL DE SPC"/" VETOR DE RESTRIÇOES"/
  " BA(J)=");
  FORMAT FMT8(// " SENTIDO DAS DESIGUALDADES"/" SENS(J)=");
  FORMAT FMT9(// " QUADRO INICIAL"/" MA(J,I)=");
  FORMAT FMT10(//"**NOVO VETOR CUSTO DE SPC"/" MA(Q,I)=");
  FORMAT FMT11(//" RESTRIÇAO ADICIONAL"/" T=",I3,X6,"UB(T)=",R8.1,X6,
  " ALFA(T)=",I3/" S(T,J)=");
  FORMAT FMT12(//" SOLUÇAO DE PRC"/" -XOC=",R8.1/" -Y(J)=");
  FORMAT FMT13(//"********"/" INICIO DE ITERAÇAO"// X2,
  "LIMITES INFERIOR E SUPERIOR",X5,R8.1,X3,R8.1/X2,
  "NUMERO DE ITERAÇOES",X20,"NU=",I3/X2,
  "NEMERO DE RESTRIÇOES",X20,"T=",I3/X2,
  "VETOR INTEIRO ATUAL",X3,"Y(J)=");
  FORMAT FMT14(//" SPC TEM SOLUÇAO ILIMITADA");
  FORMAT FMT15(" TEMPO NA ITERAÇAO"/" WMIX1=",F6.1,X6,"WSPC1=",F6.1,X6,
  "WPRC1=",F6.1/" TEMPO ACUM. "/" WMIX2=",F6.1,X6,"WSPC2=",F6.1,X6,
  "WPRC2=",F6.1);
```

DECLARAÇÕES (2)

```
  FORMAT FMT16(//"**ERRO:  SOL. MIX > SOL. PRI.");
PROCEDURE SIMPLEX(N,NN,MM,FF,BB,MP,IB,VART,Z,DO,NI,ESTADO);
COMMENT DECLARAÇOES E INICIALIZAÇAO;
  INTEGER N,NN;REAL ARRAY MM|0,1|,FF|0|,BB|0|,MP|0,0|,IB|1|;
  INTEGER ESTADO;
  REAL ARRAY DO|*|,VART|1|,Z|1|;INTEGER NI;
  BEGIN REAL ARRAY ZMAN|0:N|;LABEL NBASE,LO,SFIM;
  REAL ARRAY AM|0:N,1:NN|;
  EPS:=1.0Q-8;FF|0|:=-1.0;
COMMENT INICIO DE ITERAÇAO;
  NBASE:ITER:=ITER+1;
COMMENT ATUALIZAÇAO DA LINHA ZERO DO QUADRO;
  FOR  J:=0 STEP 1 UNTIL N DO
    BEGIN ZMAN|J|:=0;FOR K:=0 STEP 1 UNTIL N DO
    ZMAN|J|:=ZMAN|J|+FF|K|+MP|K,J|;END;
  FOR I:=1 STEP 1 UNTIL NN DO
    BEGIN DO|I|:=0;FOR J:=0 STEP 1 UNTIL N DO
    DO|I|       :=DO|I|+ZMAN|J|+MM|J,I|;END;
  WRITE (IMPRESS,FMS3);
  WRITE (IMPRESS,FMT2,FOR I:=1 STEP 1 UNTIL NN   DO DO|I|);
COMMENT CRITERIO DE ENTRADA;
  DMAX:=0;E:=L:=0;
  LO:IF L<NN+N THEN BEGIN
  L:=L+1;   FOR I:=L STEP 1 UNTIL NN DO
  BEGIN IF DO|I|  >  DMAX THEN BEGIN
  FOR J:=1 STEP 1 UNTIL N DO IF I=IB|J| OR I=VART|J| THEN GO TO LO;
  IE:=I;
COMMENT SOL. OTIMA OU INVIAVEL;
  IF IE=0 THEN
    BEGIN FOR J:=1 STEP 1 UNTIL N DO IF FF|J|>EPS THEN
    BEGIN ESTADO:=-1;GO TO SFIM;END;
COMMENT CRITERIO DE SAIDA;
  FOR J:=0 STEP 1 UNTIL N DO
  BEGIN PI|J|:=0;FOR K:=0 STEP 1 UNTIL N DO
  PI|J|:=PI|J|+MP|J,K|+MM|K,IE|;END;
  WRITE (IMPRESS,FMS4,IE);
  WRITE (IMPRESS,FMT2,FOR J:=0 STEP 1 UNTIL N DO PI|J|);
  RMIN:=1.0020;IS:=0;
  FOR J:=1 STEP 1 UNTIL N DO IF PI|J| NEQ 0 THEN
    BEGIN RR:=BB|J|/PI|J|;IF RR LEQ RMIN AND RR>0 THEN
    BEGIN IS:=IB|J|;RMIN:=RR;JS:=J;END;END;
  IF IS=0 THEN BEGIN ESTADO := 1;
GO TO SFIM;END;
COMMENT PIVOTEAMENTO;
  RR:=-1.0/PI|JS|;PI|JS|:=-1.0;FOR J:=0 STEP L UNTIL N DO
    BEGIN PI|J|:=PI|J|+RR;IF J NEQ JS THEN
      BEGIN    BB|J|:=BB|J|+PI|J|+BB|JS|;
      FOR  I:=0 STEP 1 UNTIL N DO MP|J,I|:=MP|J,I|+PI|J|*MP|JS,I|;END;
```

```
END;IB|JS|:=IE;FF|JS|:=0;
BB|JS|:= PI|JS|*BB|JS|;
FOR I:=0 STEP 1 UNTIL N DO MP|JS,I|:=PI|JS|*MP|JS,I|;
WRITE (IMPRESS,FMS0,ITER,IE,IS,BB|0|);
WRITE (IMPRESS,FMT4,FOR J:=1 STEP 1 UNTIL N DO IB|J|);
WRITE (IMPRESS,FMS7);
WRITE (IMPRESS,FMT2,FOR J:=1 STEP 1 UNTIL N DO BB|J|;GO TO NBASE;
SFIM:                          FOR I:=1 STEP 1 UNTIL NN  DO Z|I|:=0;
  FOR J:=1 STEP 1 UNTIL N DO Z|IB|J||:=BB|J|;
    WRITE (IMPRESS,FMS8,ESTADO);END;
```

DECOMPOSIÇÃO

```
COMMENT FIM DA PROCEDURE;
COMMENT INICIALIZAÇAO;
  READ (CARTOES./,FOR I:=1 STEP 1 UNTIL M DO B|I|,
  FOR J:=1 STEP 1 UNTIL Q DO Y|J|);
  FOR I:=) STEP 1 UNTIL M DO
  READ (CARTOES,/,FOR J:=1 STEP 1 UNTIL P DO A|I,J|,FOR J:=1 STEP 1
  UNTIL Q DO F|I,J|);
  WRITE (IMPRESS,FMT1);
  WRITE (IMPRESS,FMT2,FOR I:=1 STEP 1 UNTIL M DO B|I|);
  WRITE (IMPRESS,FMT3);
  WRITE (IMPRESS,FMT4,FOR J:=1 STEP 1 UNTIL Q DO Y|J|);
  FOR I:=) STEP 1 UNTIL M DO
  WRITE (IMPRESS,FMT2,FOR J:=1 STEP 1 UNTIL P DO A|I,J|);
  WRITE (IMPRESS,FMT6);
  FOR I:=0 STEP 1 UNTIL M DO
  WRITE (IMPRESS,FMT2,FOR J:=1 STEP 1 UNTIL Q DO F|I,J|);
  WO:=WMIX1:=TIME(2)/60;WMIX2:=WSPC2:=WPRC2:=0;
  ZM:=XQ:=1.0Q7 ;T:=0;K1:=0;FOR J:=1 STEP 1 UNTIL P DO SENS|J|:=1;
  EPSS:=1.0Q-5;X1:=-ZM;
COMMENT QUADRO INICIAL DE SPC;
  IF NU=0 THEN
    BEGIN FOR J:=1 STEP 1 UNTIL P DO
  BEGIN BA|J|:=A|0,J|;FOR I:=1 STEP 1 UNTIL M DO MA|J,I|:=A|I,J|;END;
  FOR I:=1 STEP 1 UNTIL M DO
    BEGIN MA|0,I|:=B|I|:FOR J:=1 STEP 1 UNTIL Q DO
    MA|0,I|:=MA|0,I|-F|0,J|*Y|J|:END;
  FOR J:=1 STEP 1 UNTIL P DO IF BA|J|<0 THEN
  BEGIN BA|J|:=-BA|J|;SENS|J|;-SENS|J|;FOR I:=1 STEP 1 UNTIL M DO
  MA|J,I|:=-MA|J,I|;END;
  NVF:=NFP:=NFN:=0;
  FOR J:=1 STEP 1 UNTIL P DO
    IF       SENS|J| NEQ 0 THEN
    BEGIN NVF:=NVF+1;NFP+(SENS|J|+1)/2;NFN:=NFN-(SENS|J|-1)/2;
  VARTA|J|:=0;END;
  FOR J:=1 STEP 1 UNTIL P DO
  IF SENS|J|=1 THEN
    BEGIN MA|J,M,K1+1|:=-1.0;K1+1;FA|J|:=ZM;VARTA|J|:=M+NFP+J;
    MA|J,M+NFP+J|;=1.0;IBA|J|:=M+NFP+J;END;
    ELSE BEGIN MA|J,M+NFP+J|:=1.0;IBA|J|:=M+NFP+J;IF SENS|J|=0 THEN
    BEGIN FA|J|:=0;VARTA|J|:=M+NFP+J;END ELSE BEGIN FA|J|:=ZM;
    VARTA|J|:=0;END;END;
  NA:=M+NFP+P;
  WRITE (IMPRESS,FMT7);
  WRITE (IMPRESS,FMT2,FOR J:=1 STEP 1 UNTIL P DO BA|J|);
  WRITE (IMPRESS,FMT8);
  WRITE (IMPRESS,FMT4,FOR J:=1 STEP 1 UNTIL P DO SENS|J|);
  WRITE (IMPRESS,FMT9);
  FOR J:=1 STEP 1 UNTIL P DO
WRITE (IMPRESS,FMT2,FOR I:=1 STEP 1 UNTIL NA   DO MA|J,I|);
```

```
COMMENT MATRIZ INICIAL DA BASE;
   MPA|0,0|:=1.0;FOR J:=1 STEP 1 UNTIL P DO
    BEGIN MPA|J,J|:=1.0;IBA|J|:=M+NFP+J;END;END;
COMMENT SOLUÇAO DE SPC;
   SPC:WMIX1:=TIME(2)/60-WMIX1;WMIX2:=WMIX2+WMIX1;
   WRITE (IMPRESS,FMT13,X1,X0,NU,T);
   WRITE (IMPRESS,FMT4,FOR J:=1 STEP 1 UNTIL Q DO Y|J|);
   WRITE (IMPRESS,FMT15,WMIX1,WSPC1,WPRC1,WMIX2,WSPC2,WPRC2);
   WMIX1:=TIME(2)/60;
   FOR J:=1 STEP 1 UNTIL P DO DC|J|:=-MA|0,IBA|J||;
   FOR     I:=1 STEP 1 UNTIL M DO
     BEGIN MA|0,I|:=B|I|;FOR J:=1 STEP 1 UNTIL Q DO
     MA|0,I|:=MA|0,I|-F|I,J|-Y|J|;END;
   FOR J:=1 STEP 1 UNTIL P DO BEGIN
   DC|J|:=0C|J|+MA|0,IBA|J||; BA|0|:=RA|0|-DC|J|*BA|J|;END;
   IF NU>0 THEN FOR I:=1 STEP 1 UNTIL P DO
     BEGIN DCB|I|:=0;FOR K:=1 STEP 1 UNTIL P DO
     DCB|I|:=DCB|I|+DC|K|*MPA|K,I|;
     MPA|0,I|:=MPA|0,I|-DCB|I|;END;
   WRITE (IMPRESS,FMT10);
   WRITE (IMPRESS,FMT2,FOR I:=1 STEP 1 UNTIL NA   DO MA|0,I|);
   WSPC1:=TIME(2)/60;
   SIMPLEX(P,NA,MA,FA,RA,MPA,TRA,VARTA,ZA,DA,NU,STATUS);
   WSPC1:=TIME(2)/60-WSPC1;WSPC2:=WSPC2+WSPC1;
   IF STATUS=-1 THEN BEGIN WRITE (IMPRESS,FMTA);GO TO FIM;END;
   FOY:=0; FOR J:=1 STEP 1 UNTIL Q DO FOY+F|0,J|*Y|J|;
   IF STATUS =1 THEN GO TO ILIM;
COMMENT SPC TEM SOL. OTIMA;
   IF FOY-BA|0|<X0 THEN
     BEGIN X0I|NU|:=FOY-BA|0|; IF X0 I|NU|>X1 THEN X1:=X0I|NU|;
     U01 :=   TRUE;U00:=FALSE;GO TO RESTR;END;
   IF FOY-BA|0|>X0 THEN BEGIN WRITE (IMPRESS,FMT16);GO TO FIM;END;
   FOR J:=1 STEP 1 UNTIL P DO X|J|:=DA|M+NFP+J|;
     WRITE (IMPRESS,FMTB,X0,FOR J:=1 STEP 1 UNTIL P DO X|J|;
     WRITE (IMPRESS,FMTC,FOR J:=1 STEP 1 UNTIL Q DO Y|J|);GO TO FIM;
   ILIM:J00:=TRUE;IF FOY-BA|0|<X0 THEN 001:=TRUE ELSE U01:= FALSE;
   WRITE (IMPRESS,FMT14);
```

SOLUÇÃO DE PRI - PARTE I (1)

```
COMMENT SOLUÇAO DE PRC EPRI COM RESIR. AUTO.;
RESTR:IF U00 AND U01 THEN I:=T+2 ELSE T:=T+1;
BEGIN
COMMENT DECLARAÇAO DE CONJUNTOS DINAMICOS;
  OWN INTEGER ARRAY ALFA|1:TMAX|;
  OWN REAL ARRAY MF|0:Q+1,1:2+Q+2+TMAX|,DF|1:2+Q+2+TMAX|;
  OWN REAL ARRAY UB|1:TMAX|,S|1:TMAX,1:Q|;
  REAL ARRAY U|1:M|,V|1:M|,W|1:T|                    ,YV|1:T|,D|1:T|;
  LABEL L1,L2,L3,L4,L5,L6,L2A,LA,LB,LC,LD,LE,LF,PRC;LABEL PRI;
  REAL ARRAY ZF|1:2+Q+2+T|;
COMMENT RESTR. ADIC.;
  IF U00 AND U01 THEN T:=T-1;IF U00 THEN
    BEGIN ALFA|I|:=0;
    FOR I:=1 STEP 1 UNTIL M DO V|I|:=0;
    FOR J:=1 STEP 1 UNTIL P DO IF PI|J| 0 AND IBA|J| LEQ M THEN
    V|IBA|J||:=-PI|J|;
    UB|I|:=0;FOR I:=1 STEP 1 UNTIL M DO UB|T|:=UB|T|+V|I|*B|I|;
    FOR J:=I STEP 1 UNTIL Q DO
      BEGIN S|T,J|:=0;FOR I:=1 STEP 1 UNTIL M DO
      S|T,J|:=S|T,J|+V|I|*F|I,J|;END;
  WRITE (IMPRESS,FMT11,T,UB|I|,ALFA|T|);
  WRITE (IMPRESS,FMT2,FOR J:=1 STEP 1 UNTIL Q DO S|T,J|);END;
  IF U01 AND U00 THEN T:T+1;IF U01 THEN
    BEGIN ALFA|T|:=1;FOR I:=1 STEP 1 UNTIL M DO U|I|:=ZA|I|;
    UB|T|:=0;FOR I:=1 STEP 1 UNTIL M DO UB|T|:=UB|T|+U|I|+B|I|;
    FOR J:=1 STEP 1 UNTIL Q DO
      BEGIN S|T,J|:=-F|Q,J|;FOR I:=1 STEP 1 UNTIL M DO
      S|T,J|:=S|T,J|+U|I|*F|I,J|;END;
  WRITE (IMPRESS,FMT11,T,UB|T|,ALFA|T|);
  WRITE (IMPRESS,FMT2,FOR J:=1 STEP 1 UNTIL Q DO S|T,J|);END;
COMMENT QUADRO INICIAL DE PRO(DUAL);
  IF T=1 THEN
    BEGIN
    X0:=UB|I|;   FOR J:=1 STEP 1 UNTIL Q DO IF S|T,J|<0 THEN
    BEGIN Y|I|:=1;X0:=X0-S|T,J|;END ELSE Y|J|:=0;
    WRITE (IMPRESS,FMT12,X0);WRITE (IMPRESS,FMT4,FOR J:=1 STEP 1 UNTIL Q
    DO Y|J|);
    NU:=NU+1;GO TO SPC;END
  ELSE IF T  LEQ 3 AND NU LEQ 1 THEN
    BEGIN FOR J:=1 STEP 1 UNTIL Q DO
    BEGIN MF|J,J|:=-1.0;MF|J,Q  +J|:=MF|0,Q  +J|:=1.0;IBF|J|:=0  +J;END;
  BF|Q+1|:=1.0;MF|Q+1,2*Q+1|:=1.0                    ;FFF|Q+1|:=ZM;
  IBF|Q+1|:=2*Q+1;
    FOR K:=1 STEP 1 UNTIL T DO
      BEGIN MF|0,2*Q+1+K|:=UB|K|;MF|Q+1,2*Q+1+K|:=ALFA|K|;
      FOR J:=1 STEP 1 UNTIL Q DO MF|J,2*Q+1+K|:=S|K,J|;END;
```

```
    NF:=Q+1;VARTF|Q+1|:=2*Q+1;
COMMENT MATRIZ INICIAL DA BASE;
    MPF|0,0|:=1.0;FOR J:=1 STEP 1 UNTIL Q+1 DO
    BEGIN MPF J,J :=I.0;IBE|J|:=Q  +J;END;
    FOR J:=1 STEP 1 UNTIL Q DO
    BEGIN MPF|0,J|:=-1.0;BF|J|:=EPSS;END;END;
COMMENT SOLUÇAO DE PRO(DUAL);
  PRC:TQ:=2*Q+2+T;
  WPRC1:=TIME(2)/60;
     SIMPLEX(NF,TQ,MF,FFF,BE,MPF,IBE,VARTF,ZF,DF, 0,STATUS);
  WPRC1:=TIME(2)/60-WPRC1;WPRC2:=WPRC2+WPRC1;
  IF STATUS=1 THEN BEGIN WRITE (IMPRESS,FMT0);GO TO FIM;END;
COMMENT TESTE DE INTEGRALIDADE;
  FOR J:=1 STEP 1 UNTIL Q DO
    BEGIN IF EPS + DF|Q+J| > 0 THEN Y|J|:=0
    ELSE IF 1.0+DF|Q  +J|>EPS THEN Y|J|:=1
    ELSE BEGIN WRITE (IMPRESS,FMT12,BF|0|);
    WRITE (IMPRESS,FMT2,FOR J:=1 STEP 1 UNTIL Q DO DF|Q  +J|);
    GO TO PRI;END;END;
COMMENT PRC TEM SOLUÇAO OTIMA INTEIRA;
  C8:=C8+1;FOR I:=1 STEP 1 UNTIL T DO IF ALFA|I|=1 THEN
    BEGIN YV|I|:=UB|I|;FOR J:=1 STEP 1 UNTIL Q DO
    YV|I|:=YV|I|-S|I,J|*Y|J|;IF YV|I|<XO THEN XO:=YV|I|;END;
  WRITE (IMPRESS,FMT12,XO);
  WRITE (IMPRESS,FMT4,FOR J:=1 STEP 1 UNTIL Q DO Y|J|);
  NU:=NU+1;GO TO SPC;
COMMENT CONSTR. FILTRO E SOLUÇAO DE PRI;
  PRI:XOC:=-BF|0|;
  FOR            K:=1 STEP 1 UNTIL T DO W|K|:=ZF|2*Q+1+K|;
  FOR J:=1 STEP 1 UNTIL Q DO YS|J|:=US|J|:=Y|J|:=0;
  US 0 :=K:=C1:=C2:=C3:=C4:=C5:=C6:=C7:=0;XOM:=-99999.0;
  FOR I:=1 STEP 1 UNTIL T DO YV|I|:=UB|I|;
  WRITE (IMPRESS,FMA1);
  WRITE (IMPRESS,FMA2,XOC,XO );
  WRITE (IMPRESS,FMA3);
  WRITE (IMPRESS,FMA4,FOR I:=1 STEP 1 UNTIL T DO ALFA|I|);
  WRITE (IMPRESS,FMA5);
  WRITE (IMPRESS,FMA6,FOR I:=1 STEP 1 UNTIL T DO UB|I|);
  WRITE (IMPRESS,FMA7);
  FOR I:=1 STEP 1 UNTIL T DO
  WRITE (IMPRESS,FMA5,FOR J:=1 STEP 1 UNTIL Q DO S|I,J|);
COMMENT FILTRO ANTERIOR;
  IF XOC OEO  XOM THEN FOR J:=1 STEP 1 UNTIL Q DO IF MC|J|=1 THEN
    BEGIN FOR I:=1 STEP 1 UNTIL T DO YV|I|:=YV|I|-S|I,J|;
    S|I,J|:=-S|I,J|;GO TO LE; END;GO TO LF;
  LE:J:=J;
  WRITE (IMPRESS,FMA8);
  WRITE (IMPRESS,FMA6,FOR I:=1 STEP 1 UNTIL T DO YV|I|);
  WRITE (IMPRESS,FMA7);
  FOR I:=1 STEP 1 UNTIL T DO
  WRITE (IMPRESS,FMA6,FOR J:=1 STEP 1 UNTIL Q DO S|I,J|);GO TO LB;
COMMENT FILTRO NOVO;
```

```
COMMENT RESTRIÇAO F;
  LF:J:=J;
  FOR J:=1 STEP 1 UNTIL Q DO
    BEGIN SF|J|:=0; FOR I:=1 STEP 1 UNTIL T DO
    SF|J|:=SF|J|+W|I|+S|I,J|;END;
  YF:=A,FAF:=0;FOR I:=1 STEP 1 UNTIL T DO
    BEGIN YF:=YF+W|I|*UB|I|; ALFAF:=ALFAF+W|I|*ALFA|I|;END;
  WRITE (IMPRESS,FMA10);
  WRITE (IMPRESS,FMA11,YF,ALFAF);
  WRITE (IMPRESS,FMA12);
  WRITE (IMPRESS,FMA6,FOR J:=1 STEP 1 UNTIL Q DO SF|J|);
```

SOLUÇÃO DE PRI - PARTE I (2)

```
COMMENT CONVERSAO P/ COEF. POSIT.;
  FOR J:=1 STEP 1 UNTIL Q DO IF SF|J|<0 THEN
    BEGIN MC|J|:=1;YF:=YF-SF|J|;SF|J|:=-SF|J|;
  FOR I:=1 STEP 1 UNTIL T DO
    BEGIN YV|I|:=YV|I|-S|I,J|;S|I,J|:=-S|I,J|;END;END;
     WRITE (IMPRESS,FMA13);
     WRITE (IMPRESS,FMA4,FOR J:=1 STEP 1 UNTIL Q DO MC|J|);
     WRITE (IMPRESS,FMA11,YF,ALFAF);
     WRITE (IMPRESS,FMA12);
     WRITE (IMPRESS,FMA6,FOR J:=1 STEP 1 UNTIL Q DO SF|J|);
COMMENT INDEXAÇAO OTIMA;
  JF|I|:=1; J:=1;
  LA:L:=J+1;FOR J:=L STEP 1 UNTIL Q DO
    BEGIN JF J :=J; FOR J1:=1 STEP 1 UNTIL J-1 DO IF SF|J|<SF|J1||THEN
      BEGIN FOR J2:=0 STEP 1 UNTIL J-J1-1 DO
      JF|J-12|:=JF|J-J2-1|;JF|J1|:=J;GO TO LA;END;END;
  WRITE (IMPRESS,FMA14);
  WRITE (IMPRESS,FMA4,FOR  J:=1 STEP 1 UNTIL Q DO JF|J|);
COMMENT SOMA COEF. NEG. LIVRES;
  LB:FOR I:=1 STEP 1 UNTIL T DO D|I|:=0;
  FOR    I:=1 STEP 1 UNTIL T DO FOR J:=1 STEP 1 UNTIL Q DC
  IF S|I,J|<0 THEN D|I|:=0|I|-S|I,J|;
  WRITE (IMPRESS,FMA15);
  WRITE (IMPRESS,FMA6,FOR I:=1 STEP 1 UNTIL T DO D|I|);
```

SOLUÇÃO DE PRI - PARTE II

```
COMMENT NOVA ITERAÇAO;
  WRITE (IMPRESS,FMA20);
COMMENT TESTE LIM. SOP. F.O.;
  L1:C1:=C1+1;
            W1:=TIME(2)/60-W0; XOF:=YF/ALFAF;
  WRITE (IMPRESS,FMA16)
  WRITE (IMPRESS,FMA17,XOF,W1
                            ,C1,C2,C3,C4,C5,C6,C7);
  WRITE (IMPRESS,FMAØ,FOR J:=1 STEP 1 UNTIL Q DO YS|J|);
  WRITE (IMPRESS,/,YF,K,FOR J:=1 STEP 1 UNTIL Q DO US|J|);
  IF XIF LEQ XOM THEN BEGIN C2:=C2+1; GO TO L2;END;
COMMENT TESTE VIABILIDADE;
  FOR I:=1 STEP 1 UNTIL T DO IF ALFA|I|=0 AND YV|I|<0 THEN GO TO LC;
COMMENT CALCULO F.O. F ATUALIZAÇAO;
  C3:=C3+1;YM:=XOF;IM:=0;
  FOR I:=1 STEP 1 UNTIL T DO IF ALFA|I|=1 AND YV|I|LEQ YN THEN
  BEGIN YM:=YV|I|;IM:=I;END;
  WRITE (IMPRESS,FMA18,YM,IM);
  IF YM>XOM THEN
  BEGIN XOM:=YM;FOR J:=1 STEP 1 UNTIL Q DO Y|J|:=YS|J|;END;
  WRITE (IMPRESS,FMA19,XOM);END;
COMMENT TESTE AUMENTO F.O.;
  IF IM NEQ Q THEN BEGIN
  IF O|IM|>XOM-YM THEN GO TO LO ELSE BEGIN C7:=C7+1;GO TO L2;END;END;
  ELSE GO TO LO;
COMMENT TESTE INVAR. EST.;
  LC:FOR I:=1 STEP 1 UNTIL T DO IF ALFA|I|=0 AND YV|I|<0 THEN
  BEGIN IF YV|I|+O|I|<0 THEN BEGIN C4:=C4+1;GO TOL2;END;END;
  WRITE (IMPRESS,FMA21);
COMMENT VAR. LIVRES CONDIC.;
  FOR I:=1 STEP 1 UNTIL T DO IF ALFA|I|=0 AND YV|I|<0 THEN
  FOR J1:=K+1 STEP 1 UNTIL Q DO
    BEGIN J:=JF|J1|;
    IF YV|I|+O|I|-S|I,J|<0 THEN F1|J|:=1;
    IF YV|I|+O|I|+S|I,J|<0 THEN FO|J|:=1;END;
  WRITE (IMPRESS,FMA22);
  WRITE (IMPRESS,FMA4,FOR J:=1 STEP 1 UNTIL Q DO F1|J|);
  WRITE (IMPRESS,FMA4,FOR J:=1 STEP 1 UNTIL Q DO FO|J|);
COMMENT TESTE P/ VAR. LIVRES CONDIC.;
  FOR J1:K+1 STEP 1 UNTIL Q DO
    BEGIN J:=JF|J1|;IF F1|J|=1 AND FO|J|=1 THEN
    BEGIN C5:=C5+1;GO TO L2, END;END;
  WRITE (IMPRESS,FMA23);
  FOR J1:=K+1 STEP 1 UNTIL Q DO
    BEGIN J:=JF|J1|;IF F1|J|=1 OR FO|J|=1 THEN
    BEGIN C6:=C6+1;GO TO L3;END;END;
  WRITE (IMPRESS,FMA24);
```

```
COMMENT AUMENTO DA SOLUÇAO;
  LD:IF K OEQ Q THEN GO TO L2 ELSE
     K:=K+1;J:=JF|K|;YS|J|:=J;YF:=YF-SF|J|;
  WRITE (IMPRESS,FMA20);
  FOR I:=1 STEP 1 UNTIL T DO
  BEGIN YV|I|:=YV|I|-S|I,J|;IF S|I,J|<0 THEN D|I|:=D|I|+S|I,J|;END;
  GO TO L1;
COMMENT SALTO P/ NOVA SOLUÇAO;
  L3: FOR J1:=K+1 STEP 1 UNTIL Q DO
    BEGIN J:=JF|J|;IF F1|J|=1 OR F0|J|=1 THEN
      BEGIN K:=K+1;US|K|:=-J;IF F1|J|=1 THEN YS|J|:=1 ELSE YS|J|:=0;
      FOR I:=1 STEP 1 UNTIL T DO
      BEGIN YV|I|:=YV|I|-S|I,J|*YS|J|;IF S|I,J|<0 THEN D|I|:=D|I|+S|I,J|
     ;END;YF:=YF-SF|J|*YS|J|;END;END;
  GO TO L1;
COMMENT BACKTRACKING;
  L2:WRITE (IMPRESS,FMA27);
  FOR     R:=0 STEP 1 UNTIL K DO IF US|K-R|>0 THEN GO TO L2A;GO TO L6;
  L2A:KSTAR:=K-R;IF KSTAR=R THEN
    BEGIN FOR I:=1 STEP 1 UNTIL T DO YV|I|:=YV|I|+S|I,US|K||;
    YS|JS|K||:=0; YF:=YF+SF|US|K|:=-US|K|;GO TO L1;END;
  FOR J1:=KSTAR+1 STEP 1 UNTIL K DO
    BEGIN FOR I:=1 STEP 1 UNTIL T DO
      BEGIN YV|I|+S|I,-US|J1||*YS|-US|J1||;
      IF S|I,-US|J1||<0 THEN D|I|:=D|I|-S|I,-US|J1||;END;
    YF:=YF+SF|J1|*YS|-US|J1||;YS|-US|J1||:=0;US|J1||:=0;END;
  YS|US|KSTAR||:=0;US|KSTAR|:=-US|KSTAR|;K:KSTAR
  FOR I:=1 STEP 1 UNTIL T DO YV|I|:=YV|I|+S|I,-US|K||;YF:=YF+SF|-US|K||;
  GO TO L1;
COMMENT RESULTADOS FINAIS;
  L8:FOR I:=1 STEP 1 UNTIL T DO IF ALFA|I|=0 AND YV|I|<0 THEN
  BEGIN WRITE (IMPRESS,FMA28);STATUS:=1;GO TO L5;END;
STATUS:=0;
      FOR J:=1 STEP 1 UNTIL Q DO IF MC|J|=1 THEN
  BEGIN Y|J|:=1-Y|J|;FOR I:=1 STEP 1 UNTIL T DO S|I,J|:=-S|I,J|;END;
  X0:=X0M;
  L5:W1:=TIME(2)/60-W0;
  WRITE (IMPRESS,FMA25);
  WRITE (IMPRESS,FMA26,X0,W1,C1,C2,C3,C4,C5,C6,C7);
  IF STATUS=-1 THEN BEGIN WRITE (IMPRESS,FMT0);GO TO FIM;END;
  IF STATUS=0 THEN
    BEGIN NU:=NU+1;GO TO SPC;END;END;

  FIM;END;


  IF UU<VV THEN GO TO L7;
END.
```

REFERÊNCIAS

|1| Balas, E.
"An additive algorithm for solving linear programs with zero-one variables".
Operations Research, Vol. 13, nº 4 (July-Aug. 65), pp. 517-546.

|2| Balas, E.
"Discrete Programming by the Filter Method".
Operations Research, Vol. 15, nº 5 (Sept.-Oct. 67), pp. 915-957.

|3| Balinski, M.L.
"Integer Programming: Methods, Uses, Computation".
Management Science, Vol. 12, nº 3 (Nov. 65), pp. 253--313.

|4| Benders, J.F.
"Partitioning procedures for solving mixed-variables

programming problems".
Numerische Mathematik, Vol. 4 (1962), pp. 238-252.

|5| Geoffrion, A.M.
"An Improved Implicit Enumeration Approach for Integer Programming".
Operations Research, Vol. 17, nº 3 (May-June 69), pp. 437-454.

|6| Geoffrion, A.M e Marsten, R.E.
"Integer Programming Algorithms: a Framework and State--of-the-Art Survey".
Publicado em "Perspectives on Optimization" de A.M. Geoffrion (editor), Addison-Wesley, 1972.

|7| Glover, F.
"A Multi-phase-Dual Algorithm for the Zero-One Integer Programming Problem".
Operations Research, Vol. 13, nº 6 (Nov.-Dec. 65), pp. 879-919.

|8| Glover, F.
"Surrogate Constraints".
Operations Research, Vol. 16 (1968), pp. 741-749.

|9| Hadley, G.
"Linear Programming".
Addison-Wesley, 1962.

|10| Lasdon, L.S.
"Optimization Theory for Large Systems".
Macmillan, 1970.

|11| Simonnard, M.
"Programmation Linéaire", Vol. I e II.
Dunod, 1972 e 1973.

|12| Woiler, S.
"Implicit Enumeration Algorithms for Discrete Optimization Problems".
Technical Report nº 4 (May 67), Dept. of Industrial Engineering, Stanford University.

|13| Zionts, S.
"Linear and Integer Programming"
Prentice-Hall, 1974.