UMA RESOLUÇÃO PARA PROGRAMAS MISTOS

PELO MÉTODO DE DRIEBEEK

Humberto do Nascimento Carreira

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 CIÊNCIAS (M.Sc.)

Aprovada por:

Prof. Nelson Maculan Filho
Presidente

Prof. João Lizardo R.H. de Araújo

Prof. Paulo O. Boaventura Neto

RIO DE JANEIRO, BRASIL

ABRIL DE 1977

CARREIRA, HUMBERTO DO NASCIMENTO

UMA RESOLUÇÃO PARA PROGRAMAS MISTOS PELO MÉTODO DE DRIEBEEK |Rio de Janeiro| 1977.

IX, 117p. 29,7cm (COPPE-UFRJ, M.Sc., Engenharia de Sistemas e Computação, 1977).

Tese - Univ. Fed. Rio de Janeiro. Fac. Engenharia

1. Programação Inteira I. COPPE/UFRJ II. UMA RESOLUÇÃO PARA PROGRAMAS MISTOS PELO MÉTODO DE DRIEBEEK.

A memória de minha mãe

Erzila N. Carreira

(1927-1976)

A Mauriceia,

A Isabela.

AGRADECIMENTOS

As provas de amizade que recebi foram inúmeras. A todos que cooperaram para que este trabalho fosse concretizado, o meu mais sincero agradecimento.

Ao Prof. Nelson Maculan Filho, que sempre me orientou com dedicação inigualável. Ao Maculan que está sempre disposto a ajudar e que foi de uma capacidade de compreensão incalculável durante momentos extremamente difíceis pelos quais passei. Qualquer forma de agradecimento é insignificante; a minha eterna gratidão.

Aos Professores João Lizardo R.H. de Araújo e Paulo Oswaldo Boaventura Neto pelas sugestões e críticas feitas ao trabalho.

A Profa. Dulcinéia de Lourdes Varela Ferreira, minha mestra no IMUFRJ, que muito me ajudou e incentivou, o meu reconhecimento.

A Celia Regina de Carvalho, bibliotecária da Biblioteca do Centro de Tecnologia da UFRJ pelo apoio e irrestrita amizade.

Aos Professores Tercio de Albuquerque Maranhão (FERP) e Antonio Braga Coscarelli (USU) pelo apoio e incentivo que me deram.

A Mauriceia, Yeda, Marilda e Getulio.

A Suely Klajman que datilografou os originais o meu mais sincero agradecimento.

RESUMO

Foi estudado neste trabalho um método de resolução para programas mistos, que contem poucas variáveis inteiras em níveis pequenos, idealizado por Norman J. Driebeek. As publicações referentes ao método são poucas e procurou-se tornar o método mais compreensível sob o ponto de vista didático. O trabalho está dividido em três partes contendo ainda um apêndice que reune dois programas para os exemplos escolhidos. Na primeira parte é feito um estudo sobre os métodos em programação inteira-mista; a segunda compõe-se da construção do Método de Driebeek, que tem a particularidade de limitar o espaço solução viável a um simples ponto inteiro, seguido de dois exemplos de aplicação; a terceira trata de estudar o método de Driebeek segundo Stanley Zionts com uma comparação sobre o artigo original de Driebeek.

ABSTRACT

This work presents a method due to Norman J. Driebeek for solving mixed programming problems containing few integer variables on small levels. References to this method are scarce and we tried to make the presentation from a didactical viewpoint. This work is devided into three parts, plus an appendice containing two programs for the given examples. The first part contains a study about mixed and integer-programming methods; the second part presents the Driebeek method, which limits the set of possible solutions to a single (integer) point, two examples are given; the third part studies the Driebeek method according to Stanley Ziohts making a comparison with the original paper of Driebeek.

ÍNDICE

Páginas

CAPÍTULO I - INTRODUÇÃO 1

1.1. Métodos de Corte 1

1.1.1. - Algoritmo Fracionário para Programação Inteira 1

1.1.2. - Algoritmo (Dual) totalmente Inteiro 2

1.1.3. - Algoritmo (Primal) totalmente Inteiro 3

1.2. Métodos de Enumeração Implícita 4

1.2.1. - Método de Enumeração Implícita de Balas 4

1.2.2. - Método de Driebeek 6

1.3. Método de Branch and Bound 7

1.4. Situação do algoritmo de Driebeek 8

CAPÍTULO II - Construção do algoritmo de Driebeek 12

2.1. Generalidades sobre o método simplex 12

2.2. Aplicação na resolução do sistema $\hat{A}\hat{\tilde{x}}=\hat{\tilde{b}}$ 15

2.3. Desenvolvimento do algoritmo de Driebeek 20

2.4. Exploração de Estratégias 30

2.5. Algoritmo de Driebeek 31

2.6. Exemplos de Aplicação do algoritmo de Driebeek 35

	Páginas
CAPÍTULO III	73
3.1. O método de Driebeek segundo Stanley Zionts	73
3.2. Exemplo de aplicação	75
3.3. Comparação entre o método de Driebeek segundo o artigo original e segundo Stanley Zionts	90
3.4. Algoritmo de Driebeek segundo Stanley Zionts	90
BIBLIOGRAFIA	94
APÊNDICE	96

CAPÍTULO I

INTRODUÇÃO

Os problemas envolvendo variáveis do tipo 0-1 e variáveis inteiras têm sido amplamente examinados por vários métodos em programação linear inteira; as idéias básicas de alguns desses métodos serão descritas a seguir.

Os métodos de programação inteira são classificados em:

- Métodos de Corte
- Métodos de Enumeração Implícita
- Método de Branch and Bound

1.1.- Metodos de Corte

1.1.1. - Algoritmo fracionário para programação inteira

Êste algoritmo é baseado no artigo de Gomory, |4| "An algorithm for integer solutions to linear Programming" - Princeton - IBM - November 1958. O problema linear é resolvido seja pelo primal, pelo dual ou pelo primal-dual. Se o problema tem solução finita, a coluna zero do quadro ótimo nos fornece $\bar{a}_{io} \geq 0$. Se todos os $\bar{a}_{io}$ forem inteiros, então as soluções do problema linear e do problema linear inteiro coincidem. Se ocorre que um certo $\bar{a}_{io} \notin Z^+ \cup \{0\}$ (Z^+ é o conjunto dos inteiros positivos), uma restrição é adicionada (Corte de Gomory) provocando uma inviabilidade no primal do quadro ótimo. Esta enviabilidade é removida usando o método do dual do simplex e este procedimento é repetido até que tenhamos todos os $\bar{a}_{io} \geq 0$ inteiros. Cada restrição adicional (corte) transforma o quadro primal viável em primal não viável, e significa que a solução vigente representa um vértice do poliedro de soluções não satisfazendo a restrição adicional Se a restrição não suprime nenhuma solução inteira viável, é de se esperar que após um certo número de cortes, se tenha uma solução inteira ótima. A prova de convergência é feita supondo-se um limite inferior para a_{oo} no caso de maximização e que após um certo número finito de iterações, todos os componentes da coluna zero do quadro, tornam-se inteiros não negativos, ou que não há solução viável.

1.1.2.- Algoritmo (Dual) Totalmente Inteiro

Este algoritmo é baseado no artigo de Gomory, |5| All - Integer Integer Programming Algorithm" - IBM - Research Center Report - RC - 189 - January 1960.

O método simplex dual lexicográfico é usado e a partir de qualquer linha primal inviável é gerado um corte, cuja restrição é adicionada no final do quadro. Após adicio nar esta restrição, e mediante certos critérios de escolha, teremos obtido um pivô igual a -1, e este se encontra na menor coluna no sentido lexicográfico. A escolha desse pivô faz com que o quadro seja sempre dual viável e ao mesmo tempo dá o maior decréscimo na coluna de índice zero do quadro.

A prova da convergencia deste algoritmo é feita por absurdo, partindo das hipóteses de que a_{oo} é limitado inferiormente e de que o algoritmo não é finito.

1.1.3.- <u>Algoritmo (Primal) Totalmente Inteiro</u>

Êste algoritmo é baseado no artigo de R. D. Young, |12| "A simplified Primal (All-Integer) Integer Programming Algorithm" - J.ORSA, 16(4) (July - August 1968). Sua característica é em cada etapa obter uma solução não pior que a anterior e tem uma grande vantagem sobre outros métodos que é a possibilidade de interromper os cálculos ao final de qualquer etapa, antes de atingir a solução ótima, conseguindo assim uma solução "boa" viável para o problema, a qual será tanto melhor quanto maior for o número de etapas, desde que não haja degenerescência.

A alteração essencial consiste na geração de um corte de Gomory a partir da linha natural do pivô, provocando com isto um pivô efetivo cujo valor é igual a 1 e que garante a primal viabilidade após o pivoteamento.

Tendo em vista o fato de poder ocorrer degenerescência, i.e, o fenômeno de ciclagem, a prova de convergência deste algoritmo é feita mediante certas modificações necessárias, e estas fazem com que o fenômeno de ciclagem persista por um certo número finito de iterações. Estas modificações, propostas por GLOVER, podem ser encontradas em:

1. INTEGER PROGRAMMING AND NETWORK FLOWS
   T.C.HU - PG.292/310
2. LINEAR AND INTEGER PROGRAMMING
   S.ZIONTS - PG.353

1.2.- Métodos de Enumeração Implícita

1.2.1.- Método de Enumeração Implícita de Balas

O princípio de enumeração implícita é o desenvolvimento em forma de arborescência tal que para um PPL com n variáveis do tipo 0-1 se garanta a enumeração de todas as 2^n soluções do PPL de maneira implícita. A idéia do algoritmo de Balas é partir de uma solução δ_o viável, introduzindo uma regra de parada e uma regra de retorno à enumeração, desenvolvendo descendentes de δ_o. Para cada solução examinada estará associado um valor $\bar{Z}_{sol}$ da função objetivo associado à melhor solução viável encontrada anteriormente à solução examinada. Se até o momento de examinar uma certa solução, não tiver aparecido uma solução viável então $\overline{Z}_{sol} = +\infty$, caso de minimização. (se maximização $\overline{Z}_{sol} = -\infty$)

Parada

Parar-se-á em uma solução δ_p, i.e, não serão examinados seus descendentes se:

a) δ_p é uma solução viável

b) $N_p = \emptyset$, onde N_p é a união do conjunto de índices que não pertencem a J_p (conj. dos índices das variáveis $x_j=1$) e J_p

c) Quando $N_p \neq \emptyset$, porém, não existe nenhuma combinação dos x_j, $j \in N_p$, que dê uma solução descendente viável.

O retorno a enumeração é feito da seguinte forma:

Atingida uma solução do problema, consideram-se todas as soluções que pertencem ao caminho, desde a solução de partida até à solução atingida, classificando-as em ordem decrescente. Encontrada uma outra solução no caminho, aplicam-se as regras da solução descendente, que será gerada mediante certos critérios de geração e as regras de parada dadas anteriormente.

Quando todas as soluções não puderem mais gerar descendentes, então o processo para. Se $\overline{Z}_{sol} = +\infty$ então o problema não tem solução viável; se $\overline{Z}_{sol}$ for finito a solução associada a $\overline{Z}_{sol}$ será a solução ótima.

1.2.2.- Método de DRIEBEEK |2|

Êste método foi apresentado por Normam J. Driebeek na revista Management Science - Vol. 12 - Nº 7 - March 1966 e foi desenvolvido essencialmente para problemas de programação que continham um grande número de variáveis contínuas bem como algumas variáveis com valores discretos. Primeiramente é resolvido o problema linear não inteiro, subsequentemente uma pesquisa para a solução ótima inteira é feita tomando por base mudanças no valor da função objetivo, pela ativação de restrições inteiras do problema.

O método se baseia na penalização das variáveis que assumem valores discretos, examinando-se uma possível solução de acôrdo com o exame de uma tabela de penalidades a ser formada mediante certos critérios. Tanto o método simplex, quanto o dual do simplex são usados e a essência está na modificação da coluna zero do quadro ótimo pelo uso da matriz inversa do simplex, fazendo com que o quadro ótimo que é primal viável, passe à condição de primal não viável. Esta inviabilidade é removida pelo dual do simplex.

O algoritmo termina quando todas as soluções prováveis tiverem sido examinadas e a solução inteira mista é a melhor solução encontrada antes da parada. Mais adiante será mostrado detalhadamente o método, que é o objetivo deste trabalho.

1.3.- Método de "Branch-and-Bound"

O princípio do "Branch-and-Bound" ganhou impulso em 1960 por A. H. LAND e A. G. DOIG em seu artigo "An Automatic Method of Solving Discrete Programming Problems" - Econometrica, 28, pgs.497-520, 1960.

O método requer o uso do simplex e tanto pode resolver um problema inteiro-misto como um problema totalmente inteiro. Achada uma solução x_k não inteira, um corte é realizado da seguinte forma:

$$x_k \leq [\, y_k \,]$$
$$x_k \geq [\, y_k \,] + 1$$

Cada desigualdade acima dá origem a uma nova restrição e um novo problema pode ser resolvido tomando as restrições do problema original e a restrição originária do corte realizado. Este procedimento pode ser repetido até que tenhamos a solução inteira desejada. A lista de soluções obtidas pode ser esquematizada por meio de uma árvore binária, sendo a raiz da árvore a solução não inteira, observe-se que para cada corte realizado, dois novos problemas devem ser re solvidos.

A fim de alcançar a solução desejada, é possível aplicar conhecimentos de programação heurística a este método.

1.4.- SITUAÇÃO DO ALGORITMO DE DRIEBEEK

"Um algoritmo para a solução de problemas de programação inteira-mista".

Na construção de modelos de programação linear, aparece frequentemente a necessidade de se restringir algumas variáveis a valores inteiros.

Tais restrições podem ser motivadas pela natureza do problema: por exemplo, o número de aviões voando entre dois aeroportos ou a construção de uma estrutura com um número discreto de lajes. Por outro lado, as restrições inteiras são valiosas quando usadas em combinações com um grupo de aproximações lineares para descrever curvas não convexas. Este último aspecto foi o movimento inicial para pesquisa em direção aos métodos para resolução de problemas de programação inteira-mista.

G. B. Dantzig, |1| em seu artigo "On the Significance of Solving Linear Programming Problems with some Integer Variables" - Econometrica, 28, Nº 1, January 1960, pp.30-40, descreveu a necessidade de resolver problemas de programação linear com algumas variáveis inteiras. Dantzig aponta que històricamente, problemas não lineares não convexos tem sido difícies de serem resolvidos. Quando variáveis inteiras assumem valores elevados em uma solução ótima contínua viável, o estudo da vizinhança dessas variáveis nos permite aproximar os resultados, algumas vêzes de maneira aceitável, embora nada garanta a obtenção de uma solução ótima inteira neste local.

Quando as variáveis inteiras tem limites superiores pequenos (digamos 5 ou menos); estudar a vizinhança da solução para o mais próximo valor inteiro não produzirá, comumente uma solução viável aceitável. A determinação de uma solução ótima global inteira-mista é difícil e esta dificuldade aumenta com o número de variáveis inteiras contidas no problema, tanto mais se são requeridas muitas variáveis inteiras.

O algoritmo de Normam J. Driebeek é formulado especificamente para resolver grandes problemas de programação linear que contenham poucas variáveis inteiras em níveis baixos.

As principais virtudes deste algoritmo são:

a) Convergência garantida para a solução ótima global inteira-mista desde que exista.
b) Insensibilidade à degeneração e a erros de arredondamento, porque soluções diversas são todas encontradas a partir de uma solução ótima contínua viável, não restringida

a valores inteiros.

No algoritmo, as variáveis inteiras são inicialmente descritas como uma soma de variáveis contínuas, cada qual com cota superior igual a 1. Adicionam-se variáveis de folga para complementar estas variáveis contínuas 0-1 e obtém-se uma solução para um programa linear comum. Faz-se a seguir uma análise de sensibilidade sobre cada variável inteira restringida. Os resultados da análise de sensibilidade são usados para construir uma tabela de penalidades, as quais são indicativas de uma redução no valor do funcional linear, quando o espaço solução viável é reduzido para vários níveis inteiros.

A matriz original é construída de uma maneira específica, de modo que as soluções inteiras podem ser definidas através da introdução de um vetor $\underset{\sim}{\Delta b}$ no quadro da solução ótima contínua. (coluna $\bar{Z}$ do quadro ótimo). Soluções inteiras válidas são obtidas pelo dual do simplex aqui aplicado algumas vêzes.

Uma escolha para a solução ótima inteira-mista é feita tomando-se por base a tabela de penalidades. Após uma solução ótima ter sido obtida, a escolha para a solução ótima inteira-mista é restringida para somente aquelas soluções inteiras cujas penalidades mínimas combinadas são menores que a diferença entre o valor da função objetivo no quadro ótimo contínuo e a mais recente solução inteira-mista válida.

A busca para a solução ótima inteira-mista é realizada efetuando-se cortes no quadro ótimo contínuo do simplex, que obrigam ao problema admitir uma solução inteira-mista. Desta forma não são constituidos erros de arredondamento durante a pesquisa para uma solução inteira-mista. Por outro lado, o algoritmo completa a pesquisa usando os custos das variáveis básicas na solução ótima contínua.

Durante uma solução-teste, os níveis efetivos que as variáveis inteiras assumem não são considerados. Por causa disso, o algoritmo é altamente insensível a alguns erros de arredondamento que porventura possam ocorrer. A pesquisa para a solução ótima inteira-mista é completada pelo método dual do simplex. Uma variável que assume o nível 0-0 não será removida para a base pelo dual do simplex. Uma solução que contenha uma ou mais variáveis no nível 0-0 é uma solução degenerada. O uso do algoritmo dual do simplex para problemas de programação inteira-mista (Gomory) efetivamente causa dificuldades de convergência, pois êste método só garante a convergência quando a solução ótima da função objeto for inteira.

CAPÍTULO II

CONSTRUÇÃO DO ALGORITMO DE DRIEBEEK

Como foi visto anteriormente, este algoritmo tem a particularidade de envolver poucas variáveis inteiras, cada uma delas sendo expressa como soma de um conjunto de variáveis definidas em $[0,1]$.

2.1.- Generalidades sobre o método simplex

Consideremos o PPL abaixo:

$$\min x_o$$

$$\text{s.a} \begin{cases} x_o - \sum\limits_{j=1}^{n} c_j x_j = b_0 \\ \sum\limits_{j=1}^{n} a_{ij} x_j = b_i \ , \ i = \overline{1,m} \\ x_j \geq 0 \ , \ j = \overline{1,n} \end{cases}$$

que também pode ser escrito como:

$$\min x_o$$

$$\text{s.a} \quad \begin{cases} x_o - \underset{\sim}{c}\underset{\sim}{x} = b_o \\ \underset{\sim}{\theta} + A\underset{\sim}{x} = \underset{\sim}{b} \\ \underset{\sim}{x} \geq 0 \end{cases}$$

Com $\underset{\sim}{c}$, $\underset{\sim}{x} \in R^n$, $\underset{\sim}{\theta} \in R^m$ e $\underset{\sim}{b} \in R^m$

Por outro lado, podemos escrever o sistema acima como:

$$\hat{A}\underset{\sim}{\hat{x}} = \underset{\sim}{\hat{b}}$$

onde: $A = (a_{ij})_{mxn}$, $\underset{\sim}{\hat{x}} = (x_o, \underset{\sim}{x})'$ $\underset{\sim}{\hat{b}} = (b_0, \underset{\sim}{b})'$

$$\hat{A} = \begin{bmatrix} 1 & -\underset{\sim}{c} \\ \underset{\sim}{\theta} & A \end{bmatrix}$$

Considerando $A\underset{\sim}{x} = \underset{\sim}{b}$, seja B uma matriz quadrada associada à solução básica de $A\underset{\sim}{x} = \underset{\sim}{b}$ formada por $\underline{m}$ colunas de A. Seja $\underset{\sim}{x}_B$ o vetor das variáveis básicas e $\underset{\sim}{c}_B$ o vetor custo associado a $\underset{\sim}{x}_B$. Escolhamos $(x_o, \underset{\sim}{x}_B)'$ para representar as variáveis básicas, então a matriz $\hat{B}$ formada por (m+1) colunas de $\hat{A}$ será:

$$\hat{B} = \begin{bmatrix} 1 & -\underset{\sim}{c}_B \\ \underset{\sim}{\theta} & B \end{bmatrix}$$

Como Det B $\neq$ 0 então Det $\hat{B} \neq 0$ e então temos:

$$\hat{B}^{-1} = \begin{bmatrix} 1 & \underset{\sim}{C}_B B^{-1} \\ \underset{\sim}{\theta} & B^{-1} \end{bmatrix}$$

e para resolver $\hat{A}\hat{\underset{\sim}{x}} = \hat{\underset{\sim}{b}}$ fazemos:

$$\hat{B}^{-1}\, \hat{A}\hat{\underset{\sim}{x}} = \hat{B}^{-1}\, \hat{\underset{\sim}{b}}$$

$$\hat{B}^{-1}\, \hat{A} = \begin{bmatrix} 1 & \underset{\sim}{C}^B\, B^{-1} \\ \underset{\sim}{\theta} & B^{-1} \end{bmatrix} \begin{bmatrix} 1 & -\underset{\sim}{C} \\ \underset{\sim}{\theta} & A \end{bmatrix}$$

$$\hat{B}^{-1}\, \hat{A} = \begin{bmatrix} 1 & \underset{\sim}{C}^B B^{-1} A - \underset{\sim}{C} \\ \underset{\sim}{\theta} & B^{-1} A \end{bmatrix}$$

$$B^{-1}A = (\underset{\sim}{y}_1, \underset{\sim}{y}_2, \ldots, \underset{\sim}{y}_n)$$

$$\underset{\sim}{C}^B B^{-1} A = (\underset{\sim}{C}^B \underset{\sim}{y}_1, \underset{\sim}{C}^B \underset{\sim}{y}_2, \ldots, \underset{\sim}{C}^B \underset{\sim}{y}_n)$$

$$\underset{\sim}{C}^B B^{-1} A - \underset{\sim}{C} = (z_1 - C_1, z_2 - C_2, \ldots, z_n - C_n) \text{ pois}$$

$$\underset{\sim}{C}^B \underset{\sim}{y}_j = z_j \;,\; j = \overline{1,n}$$

Assim,

$$\hat{B}^{-1}\, \hat{A} = \begin{bmatrix} 1 & z_1 - C_1 & z_2 - C_2 & \ldots\ldots\ldots & z_n - C_n \\ \underset{\sim}{\theta} & \underset{\sim}{y}_1 & \underset{\sim}{y}_2 & \ldots\ldots\ldots & \underset{\sim}{y}_n \end{bmatrix}$$

Por outro lado,

$$\hat{B}^{-1}\,\hat{\underset{\sim}{b}} = \begin{bmatrix} 1 & \underset{\sim}{C}^B B^{-1} \\ \underset{\sim}{\theta} & B^{-1} \end{bmatrix} \begin{bmatrix} b_o \\ \underset{\sim}{b} \end{bmatrix} = \begin{bmatrix} b_o + \underset{\sim}{C}^B B^{-1} \underset{\sim}{b} \\ B^{-1} \underset{\sim}{b} \end{bmatrix}$$

e definindo

$B^{-1}\underset{\sim}{b} = \bar{\underset{\sim}{x}}_B$ (valores assumidos pelas variáveis básicas)

$b_o + \underset{\sim}{C}^B B^{-1} \underset{\sim}{b} = \bar{x}_o$ (valor assumidos pela função objetivo)

o PPL pode ser escrito:

$$\begin{bmatrix} x_o + \sum_{j=1}^{n} (z_j - C_j) x_j \\ \sum_{j=1}^{n} \underset{\sim}{y}_j x_j \end{bmatrix} = \begin{bmatrix} \bar{x}_o \\ \bar{\underset{\sim}{x}}_B \end{bmatrix}$$

A matriz $\hat{B}^{-1}$ dada acima é usada no algoritmo de Driebeek como no simplex, para colocar o quadro em dia quando modificações, tais como a introdução de uma nova restrição, são realizadas

2.2.- Vamos considerar uma aplicação na resolução de $\hat{A}\underset{\sim}{x} = \hat{\underset{\sim}{b}}$

Exemplo:

Consideremos o PPL

$$\text{Min } x_o = x_1 + x_2 + x_3$$

$$\text{s.a}\begin{cases} x_1 \quad - x_4 \quad - 2x_6 = 5 \\ \quad x_2 + 2x_4 - 3x_5 + x_6 = 3 \\ \quad\quad x_3 + 2x_4 - 5x_5 + 6x_6 = 5 \\ x_j \geq 0 \quad , \; j=\overline{1,6} \end{cases}$$

$$\begin{cases} x_0 - x_1 - x_2 - x_3 = 0 \\ \quad x_1 \quad - x_4 \quad -2x_6 = 5 \\ \quad\quad x_2 \quad +2x_4 - 3x_5 + x_6 = 3 \\ \quad\quad\quad x_3 + 2x_4 - 5x_5 + 6x_6 = 5 \end{cases}$$

$$\hat{A} = \begin{bmatrix} 1 & -1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & -2 \\ 0 & 0 & 1 & 0 & 2 & -3 & 1 \\ 0 & 0 & 0 & 1 & 2 & -5 & 6 \end{bmatrix}$$

$$\underset{\sim}{x}_B = (x_1, x_2, x_3) \qquad B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\underset{\sim}{C}_B = (1, 1, 1)$$

As variáveis básicas de $\hat{A}\hat{\underset{\sim}{x}} = \hat{\underset{\sim}{d}}$ são representadas por:

$$\hat{\underset{\sim}{x}}' = (x_o, x_1, x_2, x_3)'$$

$$\hat{B} = \begin{bmatrix} 1 & -1 & -1 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ; \quad \det \hat{B} = 1$$

$$\hat{B}^T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix}$$

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

$$\underset{\sim}{C}_B B^{-1} = (1,\ 1,\ 1) \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = (1,\ 1,\ 1)$$

$$\hat{B}^{-1} = \begin{bmatrix} 1 & \underset{\sim}{C}_B B^{-1} \\ \underset{\sim}{\theta} & B^{-1} \end{bmatrix}$$

Multiplicando ambos os membros da igualdade $\hat{A}\underset{\sim}{\hat{x}} = \underset{\sim}{\hat{b}}$ por $\hat{B}^{-1}$ vem:

$$\hat{B}^{-1}\hat{A}\underset{\sim}{\hat{x}} = \hat{B}^{-1}\underset{\sim}{\hat{b}}$$

$$\hat{B}^{-1}\hat{A} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & -2 \\ 0 & 0 & 1 & 0 & 2 & -3 & 1 \\ 0 & 0 & 0 & 1 & 2 & -5 & 6 \end{bmatrix}$$

$$\hat{B}^{-1}\hat{A} = \begin{array}{c} \begin{bmatrix} 1 & 0 & 0 & 0 & 3 & -8 & -5 \\ 0 & 1 & 0 & 0 & -1 & 0 & -2 \\ 0 & 0 & 1 & 0 & 2 & -3 & 1 \\ 0 & 0 & 0 & 1 & 2 & -5 & 6 \end{bmatrix} \\ \begin{array}{cccccccc} & \underset{\sim}{y}_1 & \underset{\sim}{y}_2 & \underset{\sim}{y}_3 & \underset{\sim}{y}_4 & \underset{\sim}{y}_5 & \underset{\sim}{y}_6 \end{array} \end{array}$$

Evidentemente $B^{-1}A = A$

$$\underset{\sim}{c}^{B}B^{-1}A = (1,1,1) \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & -2 \\ 0 & 1 & 0 & 2 & -3 & 1 \\ 0 & 0 & 1 & 2 & -5 & 6 \end{bmatrix}$$

$$\underset{\sim}{c}^{B}B^{-1}A = (1,1,1,3,-8,5)$$

$$\underset{\sim}{c}^{B}B^{-1}A - \underset{\sim}{c} = (1,1,1,3,-8,5) - (1,1,1,0,0,0)$$

$$\underset{\sim}{c}^{B}B^{-1}A - \underset{\sim}{c} = (0,0,0,3,-8,5)$$

Portanto

$$\hat{B}^{-1}A = \begin{bmatrix} 1 & z_1-c_1 & z_2-c_2 & \cdots\cdots & z_6-c_6 \\ & & & & \\ \underset{\sim}{\theta} & \underset{\sim}{y}_1 & \underset{\sim}{y}_2 & \cdots\cdots & \underset{\sim}{y}_6 \end{bmatrix}$$

$$\hat{B}^{-1}.\underset{\sim}{b} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 5 \\ 3 \\ 5 \end{bmatrix} = \begin{bmatrix} 13 \\ 5 \\ 3 \\ 5 \end{bmatrix}$$

$$B^{-1}.\underset{\sim}{b} = \underset{\sim}{b} = \begin{bmatrix} 5 \\ 3 \\ 5 \end{bmatrix}$$

$$\underset{\sim}{C}^{B} B^{-1} \underset{\sim}{b} = (1,1,1) \begin{bmatrix} 5 \\ 3 \\ 5 \end{bmatrix} = 13$$

$$b_o + \underset{\sim}{C}^{B} B^{-1} \underset{\sim}{b} = 0 + 13 = 13$$

Portanto o sistema poderá ser escrito como:

$$\begin{bmatrix} x_o + \sum_{j=1}^{6} (z_j - c_j) x_j \\ \sum_{j=1}^{6} \underset{\sim}{y}_j x_j \end{bmatrix} = \begin{bmatrix} x_o \\ \underset{\sim}{x}_B \end{bmatrix}$$

e $\underset{\sim}{x}_B$ são os valores assumidos pelas variáveis básicas ($\bar{\underset{\sim}{x}}_B' = (5,3,5)$), x_o é o valor assumido pela função objetivo ($\bar{x}_o = 13$)

2.3.- Desenvolvimento do algoritmo de Driebeek

Consideremos agora que no PPL uma variável x_j deva ser inteira, de modo que $x_j \leq \beta_j$ onde β_j é um inteiro positivo.

$$x_j = \sum_{k=1}^{\beta_j} \delta_{jk} \quad , \; j = \overline{1,n}$$

$$0 \leq \delta_{jk} \leq 1$$

$$1 \geq \delta_{j1} \geq \delta_{j2} \geq \ldots\ldots \geq \delta_{j\beta_j} \geq 0$$

A relação de desigualdade acima é equivalente as restrições:

$$\begin{cases} \delta_{j1} \leq 1 \\ -\delta_{j1} + \delta_{j2} \leq 0 \\ \ldots\ldots\ldots\ldots \\ -\delta_{j\beta_j} \leq 0 \end{cases}$$

Introduzindo variáveis de folga em (1) vem:

$$\left\{\begin{array}{l}\delta_{j1} + \alpha_{j0} = 1 \\ -\delta_{j1} + \delta_{j2} + \alpha_{j1} = 0 \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ -\delta_{j\beta_j} + \alpha_{j\beta_j} = 0\end{array}\right. \qquad j = \overline{1,n} \qquad (2)$$

Mostraremos que o sistema primitivo contém $\sum_{j=1}^{n}(\beta_j+1)$ restrições. Suponhamos $x_1, x_2, \ldots, x_n$ variáveis inteiras e $\beta_j = 5$ um limite superior

$$\left\{\begin{array}{l}x_1 = \sum_{k=1}^{5} \delta_{1k}\ , \ 0 \leq \delta_{1k} \leq 1 \\ x_2 = \sum_{k=1}^{5} \delta_{2k}\ , \ 0 \leq \delta_{2k} \leq 1 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ x_n = \sum_{k=1}^{5} \delta_{nk}\ , \ 0 \leq \delta_{nk} \leq 1\end{array}\right.$$

$$\left\{\begin{array}{l}x_1 = \delta_{11} + \delta_{12} + \delta_{13} + \delta_{14} + \delta_{15} \\ x_2 = \delta_{21} + \delta_{22} + \delta_{23} + \delta_{24} + \delta_{25} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ x_n = \delta_{n1} + \delta_{n2} + \delta_{n3} + \delta_{n4} + \delta_{n5}\end{array}\right.$$

$$\begin{cases} 1 \geq \delta_{11} \geq \delta_{12} \geq \delta_{13} \geq \delta_{14} \geq \delta_{15} \geq 0 \\ 1 \geq \delta_{21} \geq \delta_{22} \geq \delta_{23} \geq \delta_{24} \geq \delta_{25} \geq 0 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ 1 \geq \delta_{n1} \geq \delta_{n2} \geq \delta_{n3} \geq \delta_{n4} \geq \delta_{n5} \geq 0 \end{cases} \qquad (3)$$

Considerando a la. desigualdade de (3) vem:

$$\begin{cases} \delta_{11} \leq 1 \\ -\delta_{11} + \delta_{12} \leq 0 \\ \quad - \delta_{12} + \delta_{13} \leq 0 \\ \quad\quad - \delta_{13} + \delta_{14} \leq 0 \\ \quad\quad\quad - \delta_{14} + \delta_{15} \leq 0 \\ \quad\quad\quad\quad - \delta_{15} \leq 0 \end{cases}$$

Para (3) correspondente a x_1 temos $(\beta_j+1) = 6$ restrições. Como o sistema primitivo tem $\underline{n}$ variáveis, então o total de restrições é

$$\sum_{j=1}^{n} (\beta_j+1)$$

Seja $\underset{\sim}{b}$ o vetor definido pelos têrmos independentes de (2); então passando a forma de quadro vem:

$\underset{\sim}{b}$		δ_{j1}	δ_{j2}	δ_{j3}	......	$\delta_{j\beta_{j-1}}$	$\delta_{j\beta_j}$	α_{j0}	α_{j1}	....	$\alpha_{j\beta_j}$
α_{j0}	1	1	0	0	......	0	0	1	0	....	0
α_{j1}	0	-1	1	0	......	0	0	0	1	....	0
α_{j2}	0	0	-1	1	......	0	0	0	0	.1.....	0
⋮	⋮	⋮	⋮	⋮	......	⋮	⋮	⋮	⋮	....	⋮
$\alpha_{j\beta_{j-1}}$	0	0	0	0	......	-1	1	0	0	1	0
$\alpha_{j\beta_j}$	0	0	0	0	......	0	-1	0	0	....	1

$\underset{\sim}{b}$ é da forma $(\alpha_{j0}, \alpha_{j1}, \ldots, \alpha_{j\beta_j})$

As linhas de restrições associadas com as variáveis de folga α_{ij} na solução inicial são consideradas como restrições inteiras não ativadas; porém estas poderão ser ativadas após a obtenção da solução ótima contínua pela subtração do vetor $\Delta\underset{\sim}{b}$ do vetor $\underset{\sim}{b}$.

Cada linha associada com uma variável de folga inteira, pode ser usada como possível limite do espaço solução, para um dado inteiro desde 0 até n.

Este vetor $\Delta\underset{\sim}{b}$ em combinação com a matriz $\hat{B}^{-1}$ é usado por Driebeek para provocar uma mudança do quadro ótimo primal viável em primal não viável com o propósito de chegar à solução.

Quando ativamos o 2º membro de (2) por $\Delta\underset{\sim}{b}_i = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$,

$i = \overline{0, \beta_j}$ vemos que x_j toma o valor $\underline{i}$ para toda solução de (2)

De fato, se o 2º membro de (2) for escrito como

$$\underset{\sim}{b} - \Delta\underset{\sim}{b}_0 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad \text{temos:}$$

$$\alpha_{j0} + \alpha_{j1} + \alpha_{j2} + \dots + \alpha_{j\beta_j} = 0$$

$$\rightarrow \alpha_{jk} = 0 \quad , \quad 0 \leq k \leq \beta_j$$

então $$\delta_{j1} = \delta_{j2} = \ldots\ldots = \delta_{j\beta_j} = 0 \quad \rightarrow \quad x_j = 0$$

Se o 2º membro de (2) for escrito como

$$\underset{\sim}{b} - \Delta\underset{\sim}{b}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$ temos:

$$\begin{cases} \delta_{j1} + \alpha_{j0} = 1 \\ -\delta_{j1} + \delta_{j2} + \alpha_{j1} = -1 \\ \qquad -\delta_{j2} + \delta_{j3} + \alpha_{j2} = 0 \\ \qquad \ldots\ldots\ldots\ldots\ldots\ldots \\ \qquad\qquad - \delta_{j\beta_j} + \alpha_{j\beta_j} = 0 \end{cases}$$

$$\alpha_{j0} + \alpha_{j1} + \ldots\ldots + \alpha_{j\beta_j} = 0$$

e $$\alpha_{jk} = 0 \ , \ 0 \leq k \leq \beta_j$$

então $$\delta_{j1} = 1 \text{ e } \delta_{j2} = \delta_{j3} = \ldots\ldots = \delta_{j\beta_j} = 0$$

Como $x_j = \sum_{k=1}^{\beta_j} \delta_{jk} \rightarrow x_j = 1$

Observamos que quando fazemos $\underset{\sim}{b} - \Delta\underset{\sim}{b}_1$, começamos a resolver o sistema acima pela segunda equação e os outros $\delta_{jk} = 1$ se $k \leq 1$ e 0 se $k > 1$.

Concluimos então que se o 2º membro de (2) for ativado por $\underset{\sim}{\Delta b}_i$ teremos:

$$\delta_{jk} \begin{cases} = 1 & \text{se} \quad k \leq i \\ = 0 & \text{se} \quad k > i \end{cases}$$

Portanto $x_j = i$.

Quando a matriz é construída pela maneira descrita no quadro, a introdução do vetor $\underset{\sim}{\Delta b}$ reduz o valor da função objetivo para um ótimo contínuo (b_0 cont) de $\underset{\sim}{\Pi}.\underset{\sim}{\Delta b}$, onde $\underline{\Pi}$ é o dual associado à matriz $\hat{B}^{-1}$ correspondente a base de vetores da solução ótima contínua. $\underset{\sim}{\Delta b}$ é o vetor paramétrico usado para limitar o espaço solução viável e representa um decréscimo no valor da função objetivo quando as restrições adicionais são ativadas. A introdução do vetor $\underset{\sim}{\Delta b}$ pode produzir algumas inviabilidades, que podem ser eliminadas usando o critério de pivoteamento do dual do simplex.

Não se pode estar certo da introdução de inviabilidades, mas sabe-se que a variação $\underset{\sim}{\Pi}.\underset{\sim}{\Delta b}$ será incorrida, a esta variação chamamos penalidade mínima ou penalidade verdadeira.

Quando ocorre inviabilidade por $\underset{\sim}{\Pi}.\underset{\sim}{\Delta b}$, a penalidade verdadeira é incorrida e o espaço solução do problema em relação às variáveis inteiras é reduzido para um simples ponto inteiro. Neste ponto, nota-se que na multiplicação $\underset{\sim}{\Pi}.\Delta\underset{\sim}{b}$, a penalidade verdadeira (mínima) é a soma dos custos reduzidos dos α_{ij} que estão associados com as restrições inteiras ativadas.

Por exemplo:

$$\text{Se } \underset{\sim}{\Pi} = \begin{array}{cccccc} \alpha_{10} & \alpha_{11} & \alpha_{12} & \alpha_{20} & \alpha_{21} & \alpha_{22} \\ (1.; & 1.5; & 2.; & 1.6; & 2.4; & 3.2) \end{array}$$

$$\text{e} \quad \Delta\underset{\sim}{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$\underset{\sim}{\Pi}.\Delta\underset{\sim}{b} = 2. + 2.4 = 4.4$, que representa a penalidade mínima para fazer a primeira variável inteira igual a 2 e a 2a. variável inteira igual a 1. Observemos que cada variável inteira é soma de um conjunto de variáveis binárias nos níveis 0,1,2, e no caso do exemplo acima o nível escolhido para limitar o espaço solução foi o nível (2,1).

Quando uma variável de folga está na base, sua penalidade verdadeira é zero (pois o custo da variável dual é zero), em caso contrário o valor da penalidade verdadeira é o valor da variável dual como no caso de α_{12} e α_{21}.

Cabe perguntar como se poderá ativar uma restrição se uma variável de folga está na base (i.e, o custo da variável dual é zero). Isto poderá ser feito estimando-se aquela penalidade verdadeira (mínima) por uma pseudo-penalidade.

Suponhamos que a variável de folga α_{ij} esteja na base ótima, então seu nível α^*_{ij} (valor obtido no quadro da solução ótima) pode satisfazer a $0 \leq \alpha_{ij} \leq 1$. Desde que seja subtraído o vetor Δb com $\Delta b_i = 1$ e examinando a solução para este problema modificado, associado com a base ótima do nosso problema original, podemos encontrar uma nova variável α'_{ij} cujo valor é:

$$\alpha'_{ij} = \alpha^*_{ij} - 1$$

então $$-1 \leq \alpha'_{ij} \leq 0$$

e a relação acima nos mostra que podemos provocar inviabilidade, i.e, o quadro ótimo que é primal viável, passa a condição de primal não viável. Exceto no caso em que $\alpha'_{ij} = 0$, uma inviabilidade é introduzida e pode ser eliminada pelo algoritmo dual do simplex; sua eliminação reduz o valor da função objetivo no mínimo de:

$$(\alpha^*_{ij} - 1) . \text{máx} \{\frac{a_{ok}}{a_{Rk}} \mid k=\overline{1,n} \; , \; a_{Rk} < 0\} \qquad (*)$$

onde $A = (a_{jk})$ é a matriz do ótimo contínuo, a_{ok} o custo reduzido da k-ésima variável da solução ótima contínua e R o índice da mesma linha ocupada por α_{ij} na base do ótimo contínuo. A relação (*) é chamada pseudo-penalidade associada com a restrição do α_{ij} original.

Então para cada variável de folga α_{ij} original, temos associado uma penalidade verdadeira e uma pseudo-penalidade. Estas penalidades representam a redução no valor da função objetivo, quando o espaço solução é reduzido para um nível de variável inteira correspondente. Se queremos pesqui sar um ponto inteiro específico como ótimo de um problema inteiro-misto, sabemos que estamos ao menos incorrendo em:

i) soma das penalidades verdadeiras associadas com aquela solução

ii) a maior das peseudo-penalidades

Quando o primal é inviável, pela ativação de uma restrição inteira, esta viabilidade sendo removida por uma transformação da base ótima contínua pelo método dual do simplex, causa uma redução imediata no valor da função objetivo e esta redução é no mínimo igual à pseudo-penalidade associada com a variável que está sendo removida para a base. No algoritmo dual do simplex, a variável a deixar a base é escolhida por diferentes critérios; no algoritmo dual convencional, a variável a deixar a base é selecionada por

$$\min \{b_i \mid b_i < 0\}$$

Um critério diferente é a selação do pivô mostrado em (*), o qual é equivalente a escolher o decréscimo total no valor da função objetivo, quando uma variável especificada é removida para a base. A seleção da variável a deixar a base é feita de modo que se consiga a maior degradação (redução máxima) no valor da função objetivo.

2.4.- EXPLORAÇÃO DE ESTRATÉGIAS

A política de enumeração das possíveis soluções inteiras do problema é chamada estratégia. A cada possível solução está associada uma função de custo. Seja p_s o custo associado ao desenvolvimento de uma possível solução S onde S é uma combinação dos valores de x_j admissíveis. A escolha de p_s será feita de acôrdo com o seguinte critério:

$$\min \left[p_s = \text{máx} \{ \Sigma\, p_j' \;,\; \text{máx}\; p_j'' \} \right]$$

onde p_j' é a penalidade verdadeira e p_j'' é a pseudo-penalidade.

Convém observar que as penalidades verdadeiras são aditivas. No caso de termos dois níveis 0 e 1 para as variáveis binárias que compõem o problema, podemos definir o conjunto de soluções S a serem exploradas, pelas possíveis combinações dos níveis 0 e 1, i.e, S pode ser:

(0,0)

(0,1)

(1,0)

(1,1)

2.5.- ALGORITMO DE DRIEBEEK

D0 - Resolva o PPL correspondente.

D1 - Selecione um nível desejado para cada variável inteira. Usualmente selecione o nível inteiro que corresponda a mais baixa penalidade em cada conjunto.

D2 - Construa um vetor $\Delta \underset{\sim}{b}$, o que limita o espaço solução viável para os níveis inteiros desejados.

D3 - Multiplique $\hat{B}^{-1}(-\Delta \underset{\sim}{b})$ e adicione $\underset{\sim}{b}_{cont}$

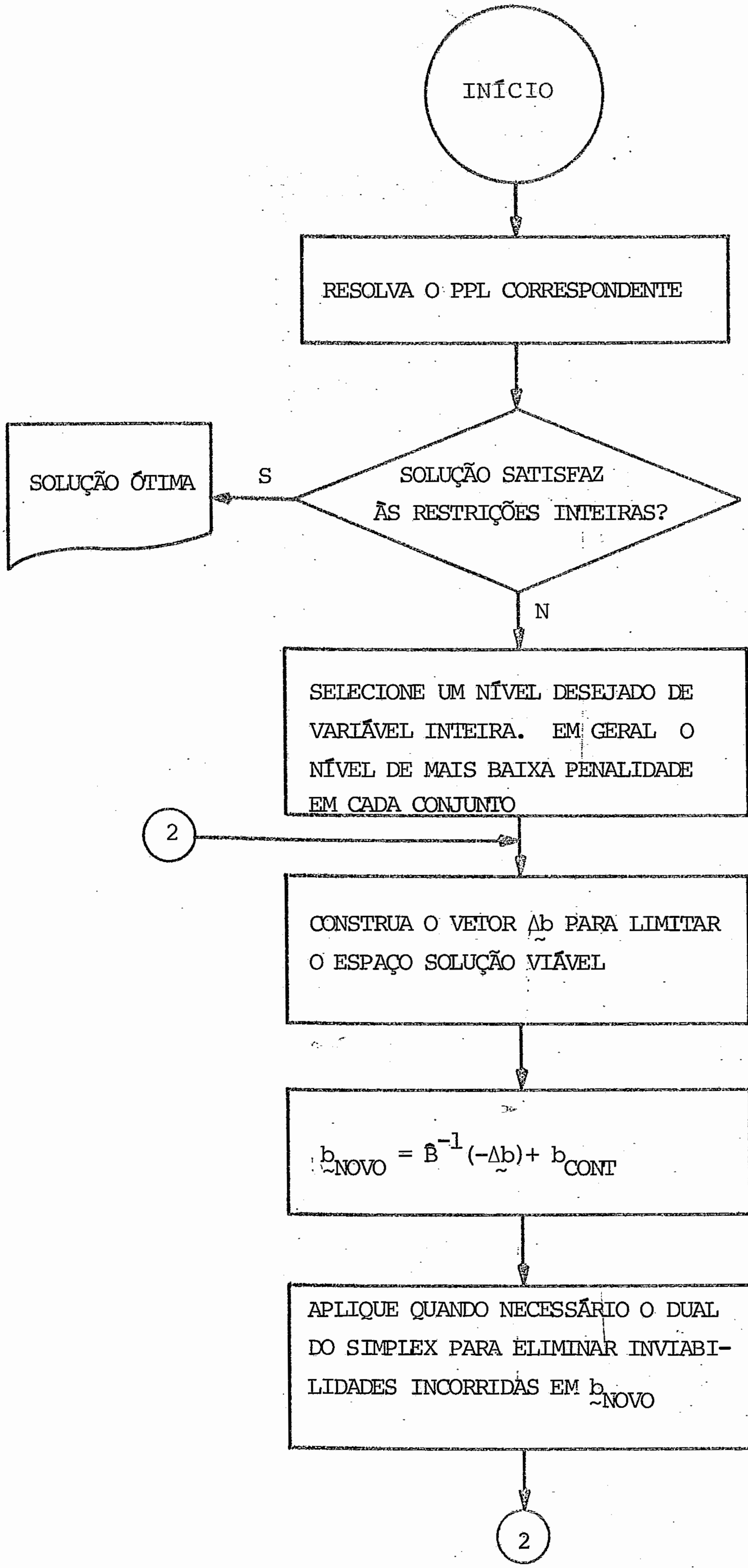
Faça $\underset{\sim}{b}_{novo} = \hat{B}^{-1}(-\Delta \underset{\sim}{b}) + \underset{\sim}{b}_{cont}$
onde $\underset{\sim}{b}_{cont}$ é o vetor solução do quadro ótimo contínuo.

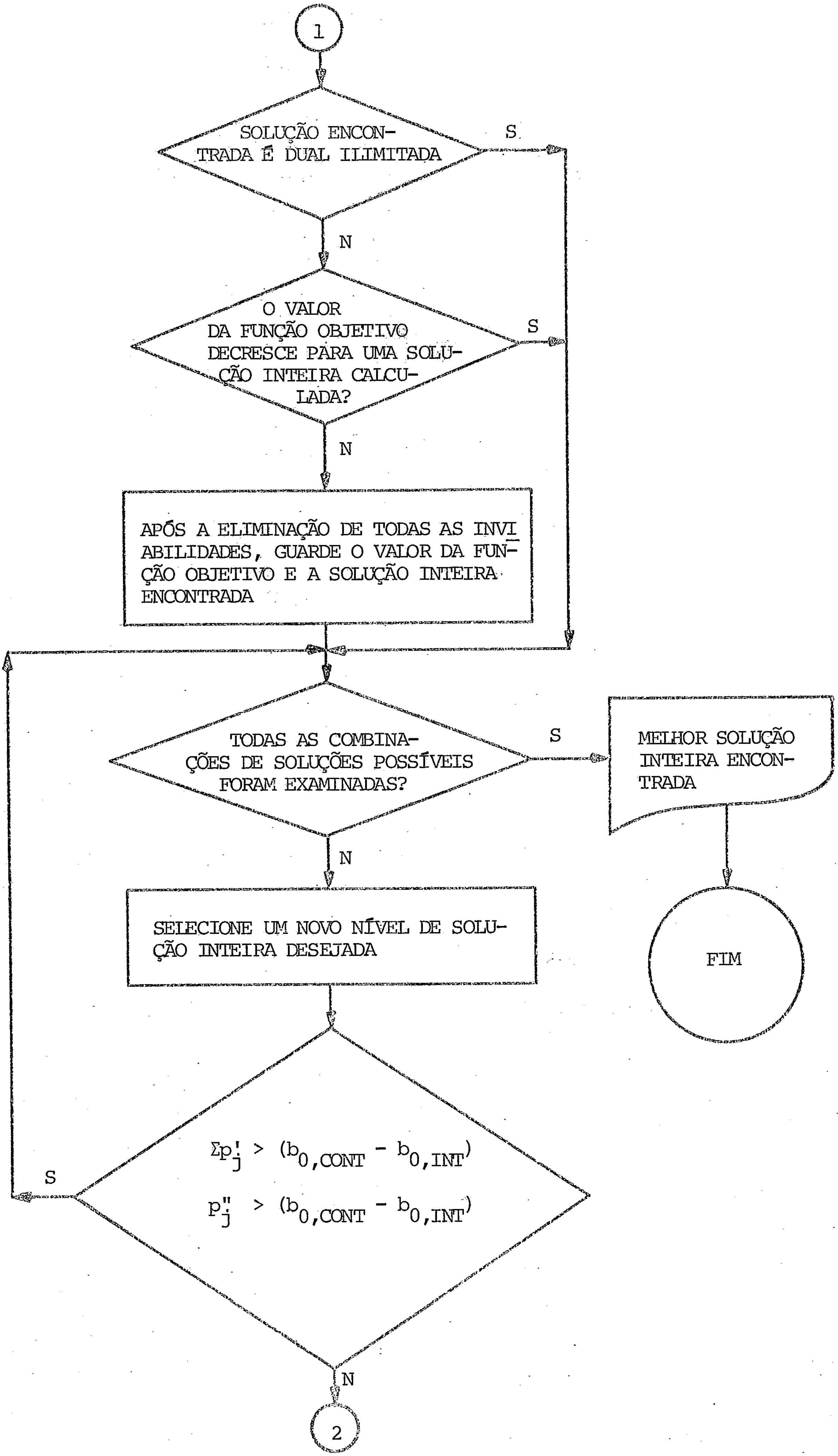
D4 - Faça as iterações do dual quando necessário para eliminar todas as inviabilidades incorridas em b_{novo}.
Quando uma solução dual ilimitada (primal inviável) é encontrada, vá para D6.
Se o valor da função objetivo decresce para uma solução inteira préviamente calculada vá para D6, senão vá para D5.

D5 - Quando todas as inviabilidades tiverem sido eliminadas, uma solução inteira válida é encontrada. Guarde a solução e o valor da função objetivo.

D6 - Selecione uma nova solução inteira desejada. Se a soma das penalidades verdadeiras ou a maior das pseudo-penalidades excede $(b_{o,cont} - b_{o,int})$ volte a D6, senão vá para D2; mas se todas as combinações possíveis para os diversos níveis tiverem sido exploradas dessa maneira, pare. A melhor solução encontrada antes de parar é a solução ótima inteira-mista.

FLUXOGRAMA DO ALGORITMO DE DRIEBEEK




1
SOLUÇÃO ENCONTRADA É DUAL ILIMITADA
S
N
O VALOR DA FUNÇÃO OBJETIVO DECRESCE PARA UMA SOLUÇÃO INTEIRA CALCULADA?
S
N
APÓS A ELIMINAÇÃO DE TODAS AS INVIABILIDADES, GUARDE O VALOR DA FUNÇÃO OBJETIVO E A SOLUÇÃO INTEIRA ENCONTRADA
TODAS AS COMBINAÇÕES DE SOLUÇÕES POSSÍVEIS FORAM EXAMINADAS?
S
MELHOR SOLUÇÃO INTEIRA ENCONTRADA
FIM
N
SELECIONE UM NOVO NÍVEL DE SOLUÇÃO INTEIRA DESEJADA
$\Sigma p'_j > (b_{0,CONT} - b_{0,INT})$
$p''_j > (b_{0,CONT} - b_{0,INT})$
S
N
2

2.6.- Exemplos de aplicação do algoritmo de Driebeek

Exemplo 1:

Consideremos o PPL

$$\max \quad Z = 10 - x_1 - x_2 - x_3 - 3x_3 - 3x_4$$

$$\text{s.a} \quad \begin{cases} -07 = -x_1 + 0.4x_2 + x_3 - x_4 \\ x_1 = 0 \text{ ou } 1 \\ x_2 = 0 \text{ ou } 1 \\ x_3,\ x_4 \geq 0 \end{cases}$$

Podemos escrever o PPL da seguinte forma:

$$\max\ z = 10 - x_1 - x_2 - 3x_3 - 3x_4 - Mx_1^a$$

$$\text{s.a} \quad \begin{cases} x_1 - 0.4x_2 - x_3 + x_4 + x_1^a = 0.7 \\ x_1 + \alpha_{10} = 1 \\ -x_1 + \alpha_{11} = 0 \\ x_2 + \alpha_{20} = 1 \\ -x_2 + \alpha_{21} = 0 \\ x_3,\ x_4 \geq 0 \end{cases}$$

Introduzimos na função objetivo uma variável artificial de grande coeficiente em valor absoluto tendo em vista a 1a. restrição do problema. (Método BIG M)

Por outro lado,

$$x_1 = \sum_{j=1}^{n} \delta_{1j}$$

$$x_2 = \sum_{j=1}^{n} \delta_{2j}$$

Como n=1 vem:

$$x_1 = \delta_{11}$$

$$x_2 = \delta_{21}$$

Sabe-se que:

$$\begin{cases} 1 \geq \delta_{11} \geq 0 \\ 1 \geq \delta_{21} \geq 0 \end{cases}$$

$$\begin{cases} 1 \geq x_1 \geq 0 \\ 1 \geq x_2 \geq 0 \end{cases}$$

então:

$$\begin{cases} x_1 \leq 1 \\ x_1 \geq 0 \end{cases} \rightarrow \begin{cases} x_1 + \alpha_{10} = 1 \\ x_1 - \alpha_{11} = 0 \end{cases} \rightarrow$$

$$\rightarrow \begin{cases} x_1 + \alpha_{10} = 1 \\ -x_1 + \alpha_{11} = 0 \end{cases}$$

$$\begin{cases} x_2 \leq 1 \\ x_2 \geq 0 \end{cases} \rightarrow \begin{cases} x_2 + \alpha_{20} = 1 \\ x_2 - \alpha_{21} = 0 \end{cases} \rightarrow \begin{cases} x_2 + \alpha_{20} = 1 \\ -x_2 + \alpha_{21} = 0 \end{cases}$$

Teremos aqui 2 níveis de variáveis possíveis, i.e, 0 e 1.

Formando o quadro do simplex vem:

quadro 1

	$\bar{Z}$	x_1	x_2	x_3	x_4	x_1^a	α_{10}	α_{11}	α_{20}	α_{21}
Z	10	1	1	3	3	M	0	0	0	0
x_1^a	0.7	1	-0.4	-1	1	1	0	0	0	0
α_{10}	1	1	0	0	0	0	1	0	0	0
α_{11}	0	-1	0	0	0	0	0	1	0	0
α_{20}	1	0	1	0	0	0	0	0	1	0
α_{21}	0	0	-1	0	0	0	0	0	0	1

Como x_1^a está na base, é preciso tornar o custo da variável dual nulo, sendo assim, multipliquemos a 2a. linha por M e subtraiamos a 1a. linha do quadro

quadro 2

	$\overline{z}$	x_1	x_2	x_3	x_4	x_1^a	α_{10}	α_{11}	α_{20}	α_{21}
z	10-0.7M	1-M	1+0.4M	3+M	3-M	0	0	0	0	0
x_1^a	0.7	1	-0.4	-1	1	1	0	0	0	0
α_{10}	1	1	0	0	0	0	1	0	0	0
α_{11}	0	-1	0	0	0	0	0	1	0	0
α_{20}	1	0	1	0	0	0	0	0	1	0
α_{21}	0	0	-1	0	0	0	0	0	0	1

Escolhendo

$$\min\ (1-M,\ 3-M) = 1-M$$

$$\min\ \left(\frac{0.7}{1},\ \frac{1}{1}\right) = 0.7$$

temos que x_1^a sai da base, entra x_1 na base e 1 é o pivô.

quadro 3

	$\bar{Z}$	x_1	x_2	x_3	x_4	x_1^a	α_{10}	α_{11}	α_{20}	α_{21}
Z	9.3	0	1.4	4	2	-1+M	0	0	0	0
x_1	0.7	1	-0.4	-1	1	1	0	0	0	0
α_{10}	0.3	0	0.4	1	-1	-1	1	0	0	0
α_{11}	0.7	0	-0.4	-1	1	1	0	1	0	0
α_{20}	1	0	1	0	0	0	0	0	1	0
α_{21}	0	0	-1	0	0	0	0	0	0	1

O quadro 3 nos dá uma solução ótima do PPL, porém x_1 não atende ainda às condições do problema e x_2 não se encontra na base.

Vamos introduzir x_2 na base, escolhendo

$$\min \left(\frac{0.3}{0.4} , \frac{1}{1} \right) = \frac{3}{4}$$

então x_2 entra na base, α_{10} sai da base e 0.4 é o pivô. Procedendo as iterações do simplex como anteriormente temos:

quadro 4

	$\bar{z}$	x_1	x_2	x_3	x_4	x_1^a	α_{10}	α_{11}	α_{20}	α_{21}
z	8.25	0	0	0.5	5.5	+2.5+M	-3.5	0	0	0
x_1	1	1	0	0	0	0	1	0	0	0
x_2	0,75	0	1	2.5	-2.5	-2.5	2.5	0	0	0
α_{11}	1	0	0	0	0	0	1	1	0	0
α_{20}	0.25	0	0	-2.5	2.5	2.5	-2.5	0	1	0
α_{21}	0.75	0	0	2.5	-2.5	-2.5	2.5	0	0	1

Na linha z do quadro 4, existe $a_{oj} < 0$ então escolhendo min $\left(\frac{1}{1} , \frac{0{,}75}{2.5} \right) = \frac{3}{10}$ temos que α_{10} entra na base, α_{21} sai da base e 2.5 é o pivô.

quadro 5

	$\bar{z}$	x_1	x_2	x_3	x_4	x_1^a	α_{10}	α_{11}	α_{20}	α_{21}
z	9.3	0	0	4	2	-1+M	0	0	0	1.4
x_1	0.7	1	0	-1	1	1	0	0	0	-0.4
x_2	0	0	1	0	0	0	0	0	0	-1
α_{11}	0.7	0	0	-1	1	1	0	1	0	-0.4
α_{20}	1	0	0	0	0	0	0	0	1	1
α_{10}	0.3	0	0	1	-1	-1	1	0	0	0.4

O quadro 5 nos dá uma solução ótima do PPL onde $x_1 = 0.7$ e $x_2 = 0$. Apenas x_2 satisfaz as condições do problema inteiro-misto; como o propósito do problema é ter também $x_1 = 0$ ou 1, recorreremos ao critério das penalidades deste algoritmo para encontrar a solução desejada.

Do quadro 5 temos que:

$$\underset{\sim}{b}_{cont} = \begin{bmatrix} 9.3 \\ 0.7 \\ 0 \\ 0.7 \\ 1 \\ 0.3 \end{bmatrix}$$

A penalidade verdadeira para cada variável binária é dada pelo custo das variáveis duais na função objetivo, assim

α_{10}	α_{11}	α_{20}	α_{21}
0	0	0	1.4

Tabela de penalidade verdadeira (mínima)

A pseudo-penalidade é dada por:

$$(\alpha^*_{ij} - 1) \,.\, \max\{\frac{a_{ok}}{a_{Rk}} \mid k=\overline{1,n} \,,\, a_{Rk} < 0\}$$

Do quadro 5 os níveis α^*_{ij} para as variáveis binárias na base são:

$$\alpha^*_{10} = 0.3 \quad ; \quad \alpha^*_{11} = 0.7 \quad \text{e} \quad \alpha^*_{20} = 1.0$$

Calculando as peseudo-penalidades vem:

a) $\alpha^*_{10} = 0.3$; $a_{54} = -1$; $a_{04} = 2$

$(0.3 - 1).\ \text{māx}\{\frac{2}{-1}\} = (-0.7).(-2) = 1.4$

b) $\alpha^*_{11} = 0.7$; $a_{33} = -1$; $a_{03} = 4$; $a_{39} = 0.4$; $a_{09} = 1.4$

$(0.7 - 1).\ \text{māx}\{\frac{4}{-1}\ ,\ \frac{1.4}{-0.4}\} = (-0.3)(-\frac{14}{4}) = 1.05$

c) $\alpha^*_{20} = 1$ e $\nexists$ $\text{māx}\{\frac{a_{ok}}{a_{Rk}}\ |\ k=\overline{1,n}\ ;\ a_{Rk} < 0\}$

$(1-1).\ _* = 0$

Lembremos que as penalidades verdadeiras são aditivas enquanto que as falsas não o são. Consideraremos a penalidade falsa de uma variável binária que não esteja na base como sendo zero para efeito de formação da tabela de penalidades e para a escolha da estratégia a ser pesquisada.

Tabela de penalidades

	Níveis	x_1	x_2
Penalidades verdadeiras (p'_j)	0	$\alpha_{10} = 0$	$\alpha_{20} = 0$
	1	$\alpha_{11} = 0$	$\alpha_{21} = 1.4$
Pseudo-penalidades (p''_j)	0	$\alpha_{10} = 1.4$	$\alpha_{20} = 0$
	1	$\alpha_{11} = 1.05$	$\alpha_{21} = 0$

Uma vez tendo os níveis 0-1, as possíveis soluções a serem pesquisadas são:

$\{(0,0),(0,1),(1,0),(1,1)\}$

Sabemos que p_s é dado por:

$p_s = \text{máx}\{\Sigma\ p'_j\ ,\ \text{máx}\ p''_j\}$

$p_{(0,0)} = \text{máx}(0;1.4) = 1.4$

$p_{(0,1)} = \text{máx}(1.4;1.4) = 1.4$

$p_{(1,0)} = \text{máx}(0;1.05) = 1.05$

$p_{(1,1)} = \text{máx}(1.4;1.05) = 1.4$

Devemos escolher uma <u>solução</u> que produza a degradação máxima da função objetivo, pois nosso problema é de maximização, caso contrário escolheríamos aquela que produz a degradação mínima da função objetivo.

Seja p_s o custo associado ao desenvolvimento de uma solução S a ser pesquisada. Temos que p_s é escolhido como:

$$\min \left[p_S = \text{máx}\{\Sigma\ p'_j,\ p''_j\} \right]$$
$$\min \left[p_{(0,0)},\ p_{(0,1)},\ p_{(1,0)},\ p_{(1,1)} \right]$$
$$\min \left[1.4;1.4;1.05;1.4 \right] = 1.05$$

Como $p_S = 1.05$ é escolhido teremos que $X_1 = 1$ e $X_2 = 0$.

Introduzamos agora o vetor $\Delta \underset{\sim}{b}$ a fim de produzir uma inviabilidade no quadro ótimo do simplex, pela substituição de $\underset{\sim}{b}_{cont}$.

Sabe-se que:

a) $\begin{cases} x_1 + \alpha_{10} = 1 \\ -x_1 + \alpha_{11} = 0 \end{cases}$

Como $x_1 = 1$ pela p_S escolhida vem:

$$\begin{cases} 1 + \alpha_{10} = 1 & \rightarrow \quad \alpha_{10} = 0 \\ -1 + \alpha_{11} = 0 & \rightarrow \quad \alpha_{11} = 1 \end{cases}$$

então reentremos $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$

b) $\begin{cases} x_2 + \alpha_{20} = 1 \\ -x_2 + \alpha_{21} = 0 \end{cases}$

Como $x_2 = 0$ pela p_S escolhida vem:

$$\begin{cases} 0 + \alpha_{20} = 1 \\ 0 + \alpha_{21} = 0 \end{cases} \rightarrow \quad \begin{matrix} \alpha_{20} = 1 \\ \alpha_{21} = 0 \end{matrix}$$

então reentremos $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$

Portanto

$$\underset{\sim}{\Delta b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \begin{matrix} \\ \\ \leftarrow \alpha_{10} \\ \leftarrow \alpha_{11} \\ \leftarrow \alpha_{20} \\ \leftarrow \alpha_{21} \end{matrix}$$

O novo vetor $\underset{\sim}{b}$ do quadro é:

$$\underset{\sim}{b}_{novo} = \hat{B}^{-1}(-\underset{\sim}{\Delta b}) + \underset{\sim}{b}_{cont}$$

onde $\hat{B}^{-1}$ é a matriz citada no início e $\underset{\sim}{b}_{cont}$ o vetor da solução ótima contínua

$$\hat{B}^{-1} = \begin{bmatrix} 1 & \underset{\sim}{C}_B \, B^{-1} \\ \underset{\sim}{\theta} & B^{-1} \end{bmatrix}$$

$$B^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & -0.4 \\ 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 1 & 0 & -0.4 \\ 0 & 0 & 0 & 1 & 1 \\ -1 & 1 & 0 & 0 & 0.4 \end{bmatrix}$$

$$\underset{\sim}{C}_B = (-1,-1,0,0,0)$$

$$\underset{\sim}{C}_B . B^{-1} = (-1,-1,0,0,0) \begin{bmatrix} 1 & 0 & 0 & 0 & -0.4 \\ 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 1 & 0 & -0.4 \\ 0 & 0 & 0 & 1 & 1 \\ -1 & 1 & 0 & 0 & 0.4 \end{bmatrix}$$

$$\underset{\sim}{C}_B . B^{-1} = (-1,0,0,0,1.4)$$

$$\hat{B}^{-1} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 1.4 \\ 0 & 1 & 0 & 0 & 0 & -0.4 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 1 & 0 & -0.4 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & -1 & 1 & 0 & 0 & 0.4 \end{bmatrix}$$

$$\underset{\sim}{b}_{novo} = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 1.4 \\ 0 & 1 & 0 & 0 & 0 & -0.4 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 1 & 0 & -0.4 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & -1 & 1 & 0 & 0 & 0.4 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \\ -1 \\ 0 \end{bmatrix} + \begin{bmatrix} 9.3 \\ 0.7 \\ 0 \\ 0.7 \\ 1 \\ 0.3 \end{bmatrix}$$

$$\underset{\sim}{b}_{novo} = \begin{bmatrix} 9.3 \\ 0.7 \\ 0 \\ -0.3 \\ 0 \\ 0.3 \end{bmatrix}$$

Substituindo $\underset{\sim}{b}_{cont}$ por $\underset{\sim}{b}_{novo}$ no quadro 5 vem:

quadro 6

	$\bar{Z}$	x_1	x_2	x_3	x_4	x_1^a	α_{10}	α_{11}	α_{20}	α_{21}
Z	9.3	0	0	4	2	-1+M	0	0	0	1.4
x_1	0.7	1	0	-1	1	1	0	0	0	-0.4
x_2	0	0	1	0	0	0	0	0	0	-1
α_{11}	-0.3	0	0	-1	1	1	0	1	0	-0.4
α_{20}	0	0	0	0	0	0	0	0	1	1
α_{10}	0.3	0	0	1	-1	-1	1	0	0	0.4

Notemos que há uma inviabilidade no primal pelo quadro 6, i.e, o nível $\alpha^*_{11} = -0.3 < 0$. A fim de remover esta inviabilidade, usemos o critério de seleção do pivô no método dual do simplex.

Escolhendo $\min\{|\frac{4}{-1}|, |\frac{1.4}{-0.4}|\} = \frac{1.4}{0.4}$ temos que α_{21} entra na base, α_{11} sai da base e -0.4 é o pivô.

quadro 7

	$\bar{z}$	x_1	x_2	x_3	x_4	x_1^a	α_{10}	α_{11}	α_{20}	α_{21}
z	8.25	0	0	0.5	5.5	2.5+M	0	3.5	0	0
X_1	1	1	0	0	0	0	0	-1	0	0
X_2	0.75	0	1	2.5	-2.5	-2.5	0	-2.5	0	0
α_{21}	0.75	0	0	2.5	-2.5	-2.5	0	-2.5	0	1
α_{20}	-0.75	0	0	-2.5	2.5	2.5	0	2.5	1	0
α_{10}	0	0	0	0	0	0	1	1	0	0

O valor da função objetivo passou de 9.3 para 8.25, a redução 1.05 (9.3-8.25 = 1.05) corresponde a pseudo penalidade associada a α_{11} como foi dito anteriormente, i.e, a eliminação de α_{11} da base reduz o valor da função objetivo no mínimo de $(\alpha^*_{ij}-1)$. $\max\{\frac{a_{ok}}{a_{Rk}} \mid k = \overline{1,n}\ ;\ a_{Rk} < 0\}$.

Convém lembrar aqui que a penalidade verdadeira (mínima) para fazer a 1a. variável inteira igual a 1(X_1=1) e a 2a. variável inteira igual a 0(X_2=0) é dada por:

$$\underset{\sim}{\Pi}.\Delta \underset{\sim}{b} = (-1,0,0,0,1.4)\begin{bmatrix}0\\0\\1\\1\\0\end{bmatrix} = 0$$

Pelo quadro 7 temos que o nível $\alpha^*_{20} < 0$ então pelo critério de seleção do pivô no método dual do simplex vem:

Escolhendo $\min\{|\frac{0.5}{-2.5}|\} = \frac{0.5}{2.5}$ vem que X_3 entra na base, α_{20} sai da base e -2.5 é o pivô.

quadro 8

	$\bar{Z}$	x_1	x_2	x_3	x_4	x_1^a	α_{10}	α_{11}	α_{20}	α_{21}
Z	8.1	0	0	0	6	3+M	0	4	0.2	0
x_1	1	1	0	0	0	0	0	-1	0	0
x_2	0	0	1	0	0	0	0	0	1	0
α_{21}	0	0	0	1	0	0	0	0	1	1
x_3	0.3	0	0	0	-1	-1	0	-1	-0.4	0
α_{10}	0	0	0	0	0	0	1	1	0	0

O quadro 8 nos mostra que uma solução inteira foi encontrada

$$\underset{\sim}{b}_{int} = \begin{bmatrix} 8.1 \\ 1 \\ 0 \\ 0 \\ 0.3 \\ 0 \end{bmatrix}$$

$x_1 = 1$ e $x_2 = 0$ atendem as condições do problema

$$b_{o,int} = 8.1$$

Suponhamos agora a seleção de um novo nível de solução desejada, considerando o ítem D6 do algoritmo. Temos ainda 3 possíveis soluções que são (0.0), (0,1) e (1,1)

a) Para $p_{(0,0)}$ temos $x_1 = x_2 = 0$

$\Sigma\, p_j' = 0$ e $p_j'' = 1.4$

Sabe-se que $b_{o,cont} = 9.3$ e $b_{o,int} = 8.1$

$b_{o,cont} - b_{o,int} = 1.2$

$$\Sigma\, p_j' < (b_{o,cont} - b_{o,int})$$
$$p_j'' > (b_{o,cont} - b_{o,int}) \qquad (1)$$

b) Para $p_{(0,1)}$ temos $x_1 = 0$ e $x_2 = 1$

$\Sigma\, p_j' = 1.4$ e $p_j'' = 1.4$

$$\left.\begin{aligned} \Sigma\, p_j' &> (b_{o,cont} - b_{o,int}) \\ p_j'' &> (b_{o,cont} - b_{o,int}) \end{aligned}\right. \qquad (2)$$

c) Para $p_{(1,1)}$ temos $x_1 = x_2 = 1$

$\Sigma\, p_j' = 1.4$ e $p_j'' = 1.05$

$$\Sigma\, p_j' \geq (b_{o,cont} - b_{o,int}) \qquad (3)$$
$$p_j'' < (b_{o,cont} - b_{o,int})$$

Observando as relações (1) em a), (2) em b) e (3) em c) vemos que nenhuma destas soluções melhorará o valor da função objetivo. Mostremos que isto é verdade considerando $p_{(0,1)}$ e voltando ao item D2 do algoritmo.

Construamos um novo vetor $\underset{\sim}{\Delta b}$ para limitar o es paço solução viável. Como $x_1=0$ e $x_2=1$ vem:

a) $\begin{cases} x_1 + \alpha_{10} = 1 \\ -x_1 + \alpha_{11} = 0 \end{cases} \rightarrow \quad \begin{matrix} \alpha_{10} = 1 \\ \alpha_{11} = 0 \end{matrix}$ então

reentremos $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$

b) $\begin{cases} x_2 + \alpha_{20} = 1 \\ -x_2 + \alpha_{21} = 0 \end{cases} \rightarrow \quad \begin{matrix} \alpha_{20} = 0 \\ \alpha_{21} = 1 \end{matrix}$ então

reentremos $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$

$$\text{Portanto } \Delta \underset{\sim}{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$B^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & -0.4 \\ 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 1 & 0 & -0.4 \\ 0 & 0 & 0 & 1 & 1 \\ -1 & 1 & 0 & 0 & 0.4 \end{bmatrix}$$

$$\underset{\sim}{C}_B = (-1,-1,0,0,0)$$

$$\underset{\sim}{C}_B B^{-1} = (-1,0,0,0,1.4)$$

$$\hat{B}^{-1} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 1.4 \\ 0 & 1 & 0 & 0 & 0 & -0.4 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 1 & 0 & -0.4 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & -1 & 1 & 0 & 0 & 0.4 \end{bmatrix}$$

$$\underset{\sim}{b}_{novo} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 1.4 \\ 0 & 1 & 0 & 0 & 0 & -0.4 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 1 & 0 & -0.4 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & -1 & 1 & 0 & 0 & 0.4 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -1 \\ 0 \\ 0 \\ -1 \end{bmatrix} + \begin{bmatrix} 9.3 \\ 0.7 \\ 0 \\ 0.7 \\ 1 \\ 0.3 \end{bmatrix}$$

$$\underset{\sim}{b}_{novo} = \begin{bmatrix} 7.9 \\ 1.1 \\ 1 \\ 1.1 \\ 0 \\ -1.1 \end{bmatrix}$$

	$\overline{Z}$	x_1	x_2	x_3	x_4	x_1^a	α_{10}	α_{11}	α_{20}	α_{21}
Z	7.9	0	0	4	2	-1+M	0	0	0	1.4
x_1	1.1	1	0	-1	1	1	0	0	0	-0.4
x_2	1	0	1	0	0	0	0	0	0	-1
α_{11}	1.1	0	0	-1	1	1	0	1	0	-0.4
α_{20}	0	0	0	0	0	0	0	0	1	1
α_{10}	-1.1	0	0	1	-1	-1	1	0	0	0.4

O quadro acima nos mostra que há uma inviabilidade.

Escolhendo $\min(|\frac{2}{-1}|) = 2$, x_4 entra na base, α_{10} sai da base e -1 é o pivô.

	$\overline{Z}$	x_1	x_2	x_3	x_4	x_1^a	α_{10}	α_{11}	α_{20}	α_{21}
Z	5.7	0	0	6	0	-3+M	2	0	0	2.2
x_1	0	1	0	0	0	0	1	0	0	0
x_2	1	0	1	0	0	0	0	0	0	-1
α_{11}	0	0	0	0	0	0	1	1	0	0
α_{20}	0	0	0	0	0	0	0	0	1	1
x_4	1.1	0	0	-1	1	1	-1	0	0	-0.4

Em relação à solução encontrada anteriormente ($x_1 = 1$ e $x_2 = 0$), pode-se notar que a solução obtida no últi mo quadro ($x_1 = 0$ e $x_2 = 1$) é pior do que aquela encontrada anteriormente (solução ótima inteira-mista).

Tomaremos agora um exemplo maior que o anterior, a fim de que possamos posteriormente fazer uma comparação, tendo em vista uma modificação proposta por Stanley Zionts |13|.

Exemplo 2

Consideremos o PPL

$$\text{Max } Z = 5x_1 + 2x_2$$

$$\text{s.a} \begin{cases} 2x_1 + 2x_2 + x_3 = 9 \\ 3x_1 + x_2 + x_4 = 11 \\ x_1 \,,\, x_2 \text{ inteiros} \\ x_j \geq 0 \,,\, j = \overline{1,4} \end{cases}$$

O exame das restrições acima nos mostra que uma cota superior para x_1 é 3 (2a. restrição) e que uma cota superior para x_2 é 4 (1a. restrição).

Tomando x_1 e x_2 como soma de variáveis binárias vem que:

$$x_1 = \sum_{k=1}^{3} \delta_{1k}$$

$$x_2 = \sum_{k=1}^{4} \delta_{2k}$$

$$\left\{\begin{array}{lcl}
x_1 = \delta_{11} + \delta_{12} + \delta_{13} & & \\
1 \geq \delta_{11} \geq \delta_{12} \geq \delta_{13} \geq 0 & & \\
\delta_{11} \leq 1 & & \delta_{11} + \delta_{10} = 1 \\
-\delta_{11} + \delta_{12} \leq 0 & \rightarrow & -\delta_{11} + \delta_{12} + \alpha_{11} = 0 \\
\quad - \delta_{12} + \delta_{13} \leq 0 & & - \delta_{12} + \delta_{13} + \alpha_{12} = 0 \\
\qquad - \delta_{13} \leq 0 & & - \delta_{13} + \alpha_{13} = 0
\end{array}\right.$$

$$\left\{\begin{array}{lcl}
x_2 = \delta_{21} + \delta_{22} + \delta_{23} + \delta_{24} & & \\
1 \geq \delta_{21} \geq \delta_{22} \geq \delta_{23} \geq \delta_{24} \geq 0 & & \\
\delta_{21} \leq 1 & & \delta_{21} + \alpha_{20} = 1 \\
-\delta_{21} + \delta_{22} \leq 0 & & -\delta_{21} + \delta_{22} + \alpha_{21} = 0 \\
\quad - \delta_{22} + \delta_{23} \leq 0 & \rightarrow & - \delta_{22} + \delta_{23} + \alpha_{22} = 0 \\
\qquad - \delta_{23} + \delta_{24} \leq 0 & & - \delta_{23} + \delta_{24} + \alpha_{23} = 0 \\
\qquad\quad - \delta_{24} \leq 0 & & - \delta_{24} + \alpha_{24} = 0
\end{array}\right.$$

Substituindo x_1 e x_2 no PPL original e acrescentando as restrições para cada nível de variável inteira vem:

$$\text{Max } Z = 5\delta_{11}+5\delta_{12}+5\delta_{13}+2\delta_{21}+2\delta_{21}+2\delta_{22}+2\delta_{23}+2\delta_{24}$$

$$\text{s.a}\left\{\begin{array}{lr}
2\delta_{11}+2\delta_{12}+2\delta_{13}+2\delta_{21}+2\delta_{22}+2\delta_{23}+2\delta_{24} \qquad +x_3 & = 9 \\
3\delta_{11}+3\delta_{12}+3\delta_{13}+\delta_{21}+\delta_{22}+\delta_{23}+\delta_{24} \qquad +x_4 & = 11 \\
\delta_{11} \qquad +\alpha_{10} & = 1 \\
-\delta_{11}+\delta_{12} \qquad +\alpha_{11} & = 0 \\
-\delta_{12}+\delta_{13} \qquad +\alpha_{12} & = 0 \\
-\delta_{13} \qquad +\alpha_{13} & = 0 \\
\delta_{21} \qquad +\alpha_{20} & = 1 \\
-\delta_{21}+\delta_{22} \qquad +\alpha_{21} & = 0 \\
-\delta_{22}+\delta_{23} \qquad +\alpha_{22} & = 0 \\
-\delta_{23}+\delta_{24} \qquad +\alpha_{23} & = 0 \\
-\delta_{24} \qquad +\alpha_{24} & = 0
\end{array}\right.$$

Quadro 1

	$\overline{Z}$	δ_{11}	δ_{12}	δ_{13}	δ_{21}	δ_{22}	δ_{23}	δ_{24}	x_3	x_4	α_{10}	α_{11}	α_{12}	α_{13}	α_{20}	α_{21}	α_{22}	α_{22}	α_{24}
Z	0	-5	-5	-5	-2	-2	-2	-2	0	0	0	0	0	0	0	0	0	0	0
x_3	9	2	2	2	2	2	2	2	1	0	0	0	0	0	0	0	0	0	0
x_4	11	3	3	3	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0
α_{10}	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
α_{11}	0	-1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
α_{12}	0	0	-1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
α_{13}	0	0	0	-1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
α_{20}	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
α_{21}	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	1	0	0	0
α_{22}	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	1	0	0
α_{23}	0	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	1	0
α_{24}	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	1

Usando o simplex e escolhendo

$\min(-5,-2) = -5$, vamos introduzir δ_{12} arbitrariamente

$\min\left(\frac{9}{2}, \frac{11}{3}, \frac{0}{1}\right) = 0$, α_{11} sai da base e 1 é o pivô

Quadro 2

	$\overline{Z}$	δ_{11}	δ_{12}	δ_{13}	δ_{21}	δ_{22}	δ_{23}	δ_{24}	x_3	x_4	α_{10}	α_{11}	α_{12}	α_{13}	α_{20}	α_{21}	α_{22}	α_{23}	α_{24}
Z	0	-10	0	-5	-2	-2	-2	-2	0	0	0	5	0	0	0	0	0	0	0
x_3	9	4	0	2	2	2	2	2	1	0	0	-2	0	0	0	0	0	0	0
x_4	11	6	0	3	1	1	1	1	0	1	0	-3	0	0	0	0	0	0	0
α_{10}	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
δ_{12}	0	-1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
α_{12}	0	-1	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
α_{13}	0	0	0	-1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
α_{20}	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
α_{21}	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	1	0	0	0
α_{22}	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	1	0	0
α_{23}	0	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	1	0
α_{24}	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	1

Escolhendo: $\min(-10,-5,-2) = -10$

$\min(\frac{9}{4}, \frac{11}{6}, \frac{1}{1}) = 1$, δ_{11} entra na base, sai α_{10} da base e 1 é o pivô.

Quadro 3

	$\overline{Z}$	δ_{11}	δ_{12}	δ_{13}	δ_{21}	δ_{22}	δ_{23}	δ_{24}	x_3	x_4	α_{10}	α_{11}	α_{12}	α_{13}	α_{20}	α_{21}	α_{22}	α_{23}	α_{24}
Z	10	0	0	-5	-2	-2	-2	-2	0	0	10	5	0	0	0	0	0	0	0
x_3	5	0	0	2	2	2	2	2	1	0	-4	-2	0	0	0	0	0	0	0
x_4	5	0	0	3	1	1	1	1	0	1	-6	-3	0	0	0	0	0	0	0
δ_{11}	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
δ_{12}	1	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
α_{12}	1	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
α_{13}	0	0	0	-1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
α_{20}	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
α_{21}	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	1	0	0	0
α_{22}	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	1	0	0
α_{23}	0	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	1	0
α_{24}	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	1

Escolhendo Min(-5,-2) = -5

$\mathrm{Min}\left(\frac{5}{2}, \frac{5}{3}, \frac{1}{1}\right) = 1$, entra δ_{13} na base, sai α_{12} e 1 é o pivô.

Quadro 4

	$\bar{Z}$	δ_{11}	δ_{12}	δ_{13}	δ_{21}	δ_{22}	δ_{23}	δ_{24}	x_3	x_4	α_{10}	α_{11}	α_{12}	α_{13}	α_{20}	α_{21}	α_{22}	α_{23}	α_{24}
Z	15	0	0	0	-2	-2	-2	-2	0	0	15	10	5	0	0	0	0	0	0
x_3	3	0	0	0	2	2	2	2	1	0	-6	-4	-2	0	0	0	0	0	0
x_4	2	0	0	0	1	1	1	1	0	1	-9	-6	-3	0	0	0	0	0	0
δ_{11}	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
δ_{12}	1	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
δ_{13}	1	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
α_{13}	1	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
α_{20}	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
α_{21}	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	1	0	0	0
α_{22}	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	1	0	0
α_{23}	0	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	1	0
α_{24}	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	1

Escolhendo Min(-2)

$\text{Min}(\frac{3}{2}, \frac{2}{1}, \frac{1}{1}) = 1$, entra δ_{21} na base (arbitrário), sai α_{20} e 1 é o pivô.

Quadro 5

	$\overline{Z}$	δ_{11}	δ_{12}	δ_{13}	δ_{21}	δ_{22}	δ_{23}	δ_{24}	x_3	x_4	α_{10}	α_{11}	α_{12}	α_{13}	α_{20}	α_{21}	α_{22}	α_{23}	α_{24}
Z	17	0	0	0	0	-2	-2	-2	0	0	15	10	5	0	2	0	0	0	0
x_3	1	0	0	0	0	2	2	2	1	0	-6	-4	-2	0	-2	0	0	0	0
x_4	1	0	0	0	0	1	1	1	0	1	-9	-6	-3	0	-1	0	0	0	0
δ_{11}	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
δ_{12}	1	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
δ_{13}	1	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
α_{13}	1	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
δ_{21}	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
α_{21}	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0
α_{22}	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	1	0	0
α_{23}	0	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	1	0
α_{24}	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	1

Escolhendo Min(-2) = -2

$\mathrm{Min}(\frac{1}{2}, \frac{1}{1})$, entra δ_{22} (arbitrário), sai x_3 sendo 2 o pivô.

Quadro 6

	$\bar{Z}$	δ_{11}	δ_{12}	δ_{13}	δ_{21}	δ_{22}	δ_{23}	δ_{24}	x_3	x_4	α_{10}	α_{11}	α_{12}	α_{13}	α_{20}	α_{21}	α_{22}	α_{23}	α_{24}
Z	18	0	0	0	0	0	0	0	1	0	9	6	3	0	0	0	0	0	0
δ_{22}	1/2	0	0	0	0	1	1	1	1/2	0	-3	-2	-1	0	-1	0	0	0	0
x_4	1/2	0	0	0	0	0	0	0	-1/2	1	-6	-4	-2	0	0	0	0	0	0
δ_{11}	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
δ_{12}	1	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
δ_{13}	1	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
α_{13}	1	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
δ_{21}	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
α_{21}	1/2	0	0	0	0	0	-1	-1	-1/2	0	3	2	1	0	2	1	0	0	0
α_{22}	1/2	0	0	0	0	0	2	1	1/2	0	-3	-2	-1	0	-1	0	1	0	0
α_{23}	0	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	1	0
α_{24}	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	1

O quadro 6 nos dá a solução do problema não inteiro

$$Z^* = 18$$

$\underset{\sim}{b}'_{cont} = (18,1/2,1/2,1,1,1,1,1,1/2,1/2,0,0)'$

No quadro 6 , $\delta_{22} > 1$. Como δ_{2j} deve ser 0 ou 1 pelas condições do problema, teremos que construir a tabela de penalidades.

Critério de pseudo-penalidade:

$$(\alpha^*_{ij}-1)\ \text{máx}\{\frac{a_{ok}}{a_{Rk}} \mid K=\overline{1,n}\ \ ;\ a_{Rk} < 0\}$$

Penalidade mínima ou verdadeira (aditivas):

Custo das variáveis duais no quadro ótimo contínuo.

Para uma variável que não se encontre na base, admitiremos sua pseudo penalidade como sendo zero para efeito da tabela a ser formada.

Tabela de penalidades verdadeiras

α_{10}	α_{11}	α_{12}	α_{13}	α_{20}	α_{21}	α_{22}	α_{23}	α_{24}
9	6	3	0	0	0	0	0	0

Calculando as pseudo-penalidades vem:

$\alpha^*_{13} = 1;\ \alpha^*_{21} = 1/2;\ \alpha^*_{22} = 1/2;\ \alpha^*_{23} = 0;\ \alpha^*_{24} = 0$

a) $\alpha^*_{13} = 1;\ \text{máx}\{\frac{a_{ok}}{a_{Rk}} \mid K=\overline{1,n}\ ;\ a_{Rk} < 0\}$ $\nexists$

$(1-1).{}_* = 0$

b) $\alpha^*_{21} = 0.5;\ a_{86} = -1;\ a_{87} = -1;\ a_{88} = -1/2$

$a_{06} = 0\ ;\ a_{07} = 0\ ;\ a_{08} = 1$

$(0.5-1).\text{máx}\{\frac{0}{-1},\ \frac{0}{-1},\ \frac{1}{-1/2}\} = (-0.5).0 = 0$

c) $\alpha^*_{22} = 0.5;\ a_{9,10} = -3;\ a_{9,11} = -2;\ a_{9,12} = -1$

$a_{9,14} = -1;\ a_{0,10} = 9;\ a_{0,11} = 6;\ a_{0,12} = 3;\ a_{0,14} = 0$

$(0.5-1).\text{máx}\{\frac{9}{-3},\frac{6}{-2},\frac{3}{-1},\frac{0}{-1}\} = (0.5).0 = 0$

d) $\alpha^*_{23} = 0,\ a_{10,6} = -1;\ a_{0,6} = 0$

$(0-1).\text{máx}\{\frac{0}{-1}\} = 0$

e) $\alpha^*_{24} = 0;\ a_{11,7} = -1;\ a_{0,7} = 0$

$(0-1).\text{máx}\{\frac{0}{-1}\} = 0$

Tabela de penalidades

	Níveis	x_1	x_2
Penalidades verdadeiras p'_j	0	$\alpha_{10}=9$	$\alpha_{20}=0$
	1	$\alpha_{11}=6$	$\alpha_{21}=0$
	2	$\alpha_{12}=3$	$\alpha_{22}=0$
	3	$\alpha_{13}=0$	$\alpha_{23}=0$
	4		$\alpha_{24}=0$
Pseudo-penalidades p''_j	0	$\alpha_{10}=0$	$\alpha_{20}=0$
	1	$\alpha_{11}=0$	$\alpha_{21}=0$
	2	$\alpha_{12}=0$	$\alpha_{22}=0$
	3	$\alpha_{13}=0$	$\alpha_{23}=0$
	4		$\alpha_{24}=0$

Possíveis soluções a serem pesquisadas:

(0,0) (1,0) (2,0) (3,0)

(0,1) (1,1) (2.1) (3.1)

(0,2) (1,2) (2,2) (3,2)

(0,3) (1,3) (2,3) (3,3)

(0,4) (1,4) (2,4) (3,4)

Escolha da solução a ser pesquisada:

$$\min \left[p_s = \text{máx}\{\Sigma\, p_j' \;,\; \text{máx}\; p_j''\} \right]$$

$$p_{(0,0)} = p_{(0,1)} = p_{(0,2)} = p_{(0,3)} = p_{(0,4)} = \text{máx}(9,0) = 9$$

$$p_{(1,0)} = p_{(1,1)} = p_{(1,2)} = p_{(1,3)} = p_{(1,4)} = \text{máx}(6,0) = 6$$

$$p_{(2,0)} = p_{(2,1)} = p_{(2,2)} = p_{(2,3)} = p_{(2,4)} = \text{máx}(3,0) = 3$$

$$p_{(3,0)} = p_{(3,1)} = p_{(3,2)} = p_{(3,3)} = p_{(3,4)} = \text{máx}(0,0) = 0$$

Escolhendo min p_s temos 4 possibilidades

$p_{(3,0)}$, $p_{(3,1)}$, $p_{(3,2)}$, $p_{(3,4)}$

Vamos tomar $p_{(3,1)}$ para pesquisar a solução (3,1), então $x_1=3$ e $x_2=1$ ($\alpha_{13} = 1$, $\alpha_{21} = 1$). Foi verificado para este exemplo que o vetor $\underset{\sim}{\Delta b} = (0,0,0,0,0,0,1,0,1,0,0,0,0)$ limita o espaço solução viável.

$$\underset{\sim}{C}_B' = (2,0,5,5,5,0,2,0,0,0,0)'$$

$$\hat{B}^{-1} = \begin{bmatrix} 1 & C_B B^{-1} \\ \underset{\sim}{\theta} & B^{-1} \end{bmatrix}$$

$$B^{-1} = \begin{bmatrix} \frac{1}{2} & 0 & -3 & -2 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & -6 & -4 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 3 & 2 & 1 & 0 & 2 & 1 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & -3 & -2 & -1 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\underset{\sim}{C}_B B^{-1} = (1,0,9,6,3,0,0,0,0,0,0)$$

$$\hat{B}^{-1} = \begin{bmatrix} 1 & 1 & 0 & 9 & 6 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & -3 & -2 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{2} & 1 & -6 & -4 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{2} & 0 & 3 & 2 & 1 & 0 & 2 & 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & -3 & -2 & -1 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Como $\underset{\sim}{b}_{novo} = \hat{B}^{-1}.(-\Delta\underset{\sim}{b}) + \underset{\sim}{b}_{cont}$ tem-se

$$\underset{\sim}{b}'_{novo} = (18,1/2,1/2,1,1,1,0,1,-1/2,1/2,0,0)'$$

Substituindo $\underset{\sim}{b}_{cont}$ por $\underset{\sim}{b}_{novo}$ no quadro ótimo contínuo vem:

Quadro 7

	$\overline{Z}$	δ_{11}	δ_{12}	δ_{13}	δ_{21}	δ_{22}	δ_{23}	δ_{24}	x_3	x_4	α_{10}	α_{11}	α_{12}	α_{13}	α_{20}	α_{21}	α_{22}	α_{23}	α_{24}
Z	18	0	0	0	0	0	0	0	1	0	9	6	3	0	0	0	0	0	0
δ_{22}	1/2	0	0	0	0	1	1	1	1/2	0	-3	-2	-1	0	-1	0	0	0	0
x_4	1/2	0	0	0	0	0	0	0	-1/2	1	-6	-4	-2	0	0	0	0	0	0
δ_{11}	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
δ_{12}	1	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
δ_{13}	1	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
α_{13}	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
δ_{21}	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
α_{21}	-1/2	0	0	0	0	0	-1	-1	-1/2	0	3	2	1	0	2	1	0	0	0
α_{22}	1/2	0	0	0	0	0	2	1	1/2	0	-3	-2	-1	0	-1	0	1	0	0
α_{23}	0	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	1	0
α_{24}	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	1

Escolhendo $\min\left(\left|\frac{0}{1}\right|, \left|\frac{0}{-1}\right|, \left|\frac{1}{-1/2}\right|\right) = 0$, e colocando δ_{23} na base, sai δ_{21} sendo -1 o pivô.

Quadro 8

	$\overline{Z}$	δ_{11}	δ_{12}	δ_{13}	δ_{21}	δ_{22}	δ_{23}	δ_{24}	x_3	x_4	α_{10}	α_{11}	α_{12}	α_{13}	α_{20}	α_{21}	α_{22}	α_{23}	α_{24}
Z	18	0	0	0	0	0	0	0	1	0	9	6	3	0	0	0	0	0	0
δ_{22}	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0
x_4	1/2	0	0	0	0	0	0	0	-1/2	1	-6	-4	-2	0	0	0	0	0	0
δ_{11}	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
δ_{12}	1	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
δ_{13}	1	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
α_{13}	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
δ_{21}	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
δ_{23}	1/2	0	0	0	0	0	1	1	1/2	0	-3	-2	-1	0	-2	-1	0	0	0
α_{22}	-1/2	0	0	0	0	0	0	-1	-1/2	0	3	2	1	0	3	2	1	0	0
α_{23}	1/2	0	0	0	0	0	0	2	1/2	0	-3	-2	-1	0	-2	-1	0	1	0
α_{24}	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	1

Escolhendo Min$\left(\left|\frac{0}{-1}\right|, \left|\frac{1}{-1/2}\right|\right) = 0$, entra δ_{24} na base, sai α_{22} da base sendo -1 o pivô.

Quadro 9

	$\overline{Z}$	δ_{11}	δ_{12}	δ_{13}	δ_{21}	δ_{22}	δ_{23}	δ_{24}	x_3	x_4	α_{10}	α_{11}	α_{12}	α_{13}	α_{20}	α_{21}	α_{22}	α_{23}	α_{24}
Z	18	0	0	0	0	0	0	0	1	0	9	6	3	0	0	0	0	0	0
δ_{22}	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
x_4	1/2	0	0	0	0	0	0	0	-1/2	1	-6	-4	-2	0	0	0	0	0	0
δ_{11}	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
δ_{12}	1	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
δ_{13}	1	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
α_{13}	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
δ_{21}	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
δ_{23}	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	0	0
δ_{24}	1/2	0	0	0	0	0	0	1	1/2	0	-3	-2	-1	0	-3	-2	-1	0	0
α_{23}	-1/2	0	0	0	0	0	0	0	-1/2	0	3	2	1	0	4	3	2	1	0
α_{24}	1/2	0	0	0	0	0	0	0	1/2	0	-3	-2	-1	0	-3	-2	-1	0	1

Escolhendo $\text{Mín}\left(\left|\frac{1}{-1/2}\right|\right) = 2$, x_3 entra na base,

sai α_{23} da base sendo $-\frac{1}{2}$ o pivô.

Quadro 10

	$\overline{Z}$	δ_{11}	δ_{12}	δ_{13}	δ_{21}	δ_{22}	δ_{23}	δ_{24}	x_3	x_4	α_{10}	α_{11}	α_{12}	α_{13}	α_{20}	α_{21}	α_{22}	α_{23}	α_{24}
Z	17	0	0	0	0	0	0	0	0	0	15	10	5	0	8	6	4	2	0
δ_{22}	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0
x_4	1	0	0	0	0	0	0	0	0	1	-9	-6	-3	0	-4	-3	-2	-1	0
δ_{11}	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
δ_{12}	1	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
δ_{13}	1	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
α_{13}	1	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
δ_{21}	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
δ_{23}	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	0	0
δ_{24}	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	0
x_3	1	0	0	0	0	0	0	0	1	0	-6	-4	-2	0	-8	-6	-4	-2	0
α_{24}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

A solução inteira-mista deste PPL é $Z^* = 17$; $x_1^* = 3$;

$x_2^* = 1$; $x_3^* = x_4^* = 1$

CAPÍTULO III

3.1- O método de Driebeek segundo Stanley Zionts

S. Zionts, |13| usa apenas uma restrição para cada variável inteira, ao invés de um conjunto de restrições para cada nível aceitável de variável inteira. A modificação realizada por S. Zionts não tem sido usada em programas de computador, porque requer extensas modificações nos códigos computacionais de programação linear existentes.

Segundo S. Zionts, se o PPL requer x_i como variável inteira, então x_i é escrito da seguinte forma:

$$x_i = \sum_{j=0}^{n} j\alpha_{ij} \quad \text{onde} \quad \alpha_{ij} = 0 \text{ ou } 1 \text{ e}$$

<u>n</u> é um limite superior (cota superior) para x_i.

$$\sum_{j=0}^{n} \alpha_{ij} = 1 \qquad (*)$$

Portanto $x_i = j$ se $\alpha_{ij} = 1$ (Basta que um certo $\alpha_{ik} = 1$, $k \in \{0,1,2,\ldots,n\}$ para que $\alpha_{i0} = \alpha_{i1} = \ldots = \alpha_{i,k-1} = \alpha_{i,k+1} = \ldots\ldots = \alpha_{in} = 0$.

As variáveis inteiras escritas sob a forma acima são substituídas no PPL original, dando-nos assim um novo problema em termos das variáveis binárias inteiras e das variáveis não negativas do PPL dado, acrescido da restrição (*).

A solução ótima é pesquisada empregando o método simplex ou o dual do simplex conforme seja necessário. Encontrada a solução ótima, escolhemos as soluções a serem pesquisadas, de acordo com a tabela de penalidades proposta por Driebeek. Após a escolha destas soluções, teremos encontrado um certo $\alpha_{ij}=1$ para uma variável x_i e então o espaço solução será limitado por um corte, acrescentando no quadro do ótimo contínuo as seguintes restrições:

$$\alpha_{ij} \leq 1$$

$$\alpha_{ij} \geq 1$$

Este fato não obriga a encontramos a solução desejada para certa estratégia; se isto ocorre é necessário selecionar um novo nível de solução desejada. A matriz $\tilde{B}^{-1}$ e o vetor $\Delta \underset{\sim}{b}_j$ não são usados com o propósito de produzir inviabilidades, pois o corte feito nos dá esta condição.

3.2.- Exemplo de aplicação segundo a modificação proposta por S. Zionts

Consideraremos aqui o 2º exemplo dado no capítulo II, o qual foi desenvolvido segundo o artigo original de Norman J. Driebeek.

Seja o PPL

$$\text{Max } Z = 5x_1 + 2x_2$$

$$\text{s.a} \quad \begin{cases} 2x_1+2x_2+x_3 = 9 \\ 3x_1+ x_2+x_4 = 11 \\ x_j \geq 0 \ , \ j=\overline{1,4} \\ x_1, x_2 \text{ inteiros} \end{cases}$$

Como x_1 e x_2 tem limites superiores respectivamente iguais a 3 e 4 podemos escrever.

$$x_1 = \sum_{j=0}^{3} j\alpha_{1j}$$

$$\sum_{j=0}^{3} \alpha_{1j} = 1$$

$$x_2 = \sum_{j=0}^{4} j\alpha_{2j}$$

$$\sum_{j=0}^{4} \alpha_{2j} = 1$$

$$\left.\begin{array}{l} x_1 = \alpha_{11} + 2\alpha_{12} + 3\alpha_{13} \\ \alpha_{10} + \alpha_{11} + \alpha_{12} + \alpha_{13} = 1 \end{array}\right. \qquad (1)$$

$$\left.\begin{array}{l} x_2 = \alpha_{21} + 2\alpha_{22} + 3\alpha_{23} + 4\alpha_{24} \\ \alpha_{20} + \alpha_{21} + \alpha_{22} + \alpha_{23} + \alpha_{24} = 1 \end{array}\right. \qquad (2)$$

Substituindo (1) e (2) na função objetivo e nas restrições do PPL dado vem:

$$Z = 5(\alpha_{11}+2\alpha_{12}+3\alpha_{3}) + 2(\alpha_{21}+2\alpha_{22}+3\alpha_{23}+4\alpha_{24})$$

$$2(\alpha_{11}+2\alpha_{12}+3\alpha_{13}) + 2(\alpha_{21}+2\alpha_{22}+3\alpha_{23}+4\alpha_{24}) + x_3 = 9$$

$$3(\alpha_{11}+2\alpha_{12}+3\alpha_{13}) + \alpha_{21}+2\alpha_{22}+3\alpha_{23}+4\alpha_{24}+x_4 = 11$$

O PPL passa então a ter a seguinte disposição:

$$\text{Max } z = 5\alpha_{11}+10\alpha_{12}+15\alpha_{13}+2\alpha_{21}+4\alpha_{22}+6\alpha_{23}+8\alpha_{24}$$

$$\text{s.a} \left\{\begin{array}{llll} & 2\alpha_{11}+4\alpha_{12}+6\alpha_{13} & +2\alpha_{21}+4\alpha_{22}+6\alpha_{23}+8\alpha_{24}+x_3 & = 9 \\ & 3\alpha_{11}+6\alpha_{12}+9\alpha_{13} & +\alpha_{21}+2\alpha_{22}+3\alpha_{23}+4\alpha_{24} \quad +x_4 & = 11 \\ \alpha_{10}+ & \alpha_{11}+\alpha_{12}+\alpha_{13} & & = 1 \\ & & \alpha_{20}+\alpha_{21}+\alpha_{22}+\alpha_{23}+\alpha_{24} & = 1 \\ & x_3,\ x_4 \geq 0 & & \\ & \alpha_{ij}=0 \text{ ou } 1,\ j=\overline{0,4},\ i=1,2 & & \end{array}\right.$$

Tomando o quadro do simplex vem:

Quadro 1

	$\overline{Z}$	α_{10}	α_{11}	α_{12}	α_{13}	α_{20}	α_{21}	α_{22}	α_{23}	α_{24}	x_3	x_4
Z	0	0	-5	-10	-15	0	-2	-4	-6	-8	0	0
x_3	9	0	2	4	6	0	2	4	6	8	1	0
x_4	11	0	3	6	9	0	1	2	3	4	0	1
α_{10}	1	1	1	1	1	0	0	0	0	0	0	0
α_{20}	1	0	0	0	0	1	1	1	1	1	0	0

Escolhendo Min(-5,-10,-15,-2,-4,-6,-8) = -15

$\text{Min}(\frac{9}{6},\frac{11}{9},\frac{1}{1}) = 1$, α_{13} entra na base, sai x_4 da base e 9 é o pivô

Quadro 2

	$\overline{Z}$	α_{10}	α_{11}	α_{12}	α_{13}	α_{20}	α_{21}	α_{22}	α_{23}	α_{24}	x_3	x_4
Z	15	15	10	5	0	0	-2	-4	-6	-8	0	0
x_3	3	-6	-4	-2	0	0	2	4	6	8	1	0
x_4	2	-9	-6	-3	0	0	1	2	3	4	0	1
α_{13}	1	1	1	1	1	0	0	0	0	0	0	0
α_{20}	1	0	0	0	0	1	1	1	1	1	0	0

Escolhendo Min(-2,-4,-6,-8) = -8

$\text{Min}(\frac{3}{8},\frac{2}{4},\frac{1}{1}) = \frac{3}{8}$, α_{24} entra na base, sai x_3 da base e 8 é o pivô.

	$\bar{Z}$	α_{10}	α_{11}	α_{12}	α_{13}	α_{20}	α_{21}	α_{22}	α_{23}	α_{24}	x_3	x_4
Z	18	9	6	3	0	0	0	0	0	0	1	0
α_{24}	3/8	-3/4	-1/2	-1/4	0	0	1/4	1/2	3/4	1	1/8	0
x_4	1/2	-6	-4	-2	0	0	0	0	0	0	-1/2	1
α_{13}	1	1	1	1	1	0	0	0	0	0	0	0
α_{20}	5/8	3/4	1/2	1/4	0	1	3/4	1/2	1/4	0	-1/8	0

Quadro 3

O quadro 3 nos dá uma solução ótima do PPL ($Z^*=18$; $\alpha^*_{24} = \frac{3}{8}$; $\alpha^*_{13}=1$, $x^*_4=1/2$; $\alpha^*_{20}=5/8$), porém α_{24} e α_{20} não satisfaz as condições do problema inteiro-misto; teremos pois que construir a tabela de penalidades para as variáveis binárias.

Penalidade verdadeira: custo das variáveis duais no quadro da solução ótima contínua.

Pseudo-penalidade:

$$(\alpha^*_{ij} - 1) \,.\, \text{máx}\{\frac{a_{ok}}{a_{Rk}} \mid k=\overline{1,n} \;;\; a_{Rk} < 0\}$$

Tabela de penalidade verdadeira

α_{10}	α_{11}	α_{12}	α_{13}	α_{20}	α_{21}	α_{22}	α_{23}	α_{24}
9	6	3	0	0	0	0	0	0

Cálculo das pseudo-penalidades

a) $\alpha^*_{13} = 1$ e $a_{2k} < 0$ não existe pelo quadro 3, $(1-1) \,.\, * = 0$

b) $\alpha^*_{20} = \frac{5}{8}$; $a_{4,10} = -\frac{1}{8}$; $a_{0,10} = 1$

$$(\frac{5}{8} - 1) \,.\, \text{máx}\{\frac{1}{-1/8}\} = (-\frac{3}{8})(-8) = 3$$

c) $\alpha^*_{24} = \frac{3}{8}$; $a_{11} = -\frac{3}{4}$; $a_{12} = -\frac{1}{2}$; $a_{13} = -\frac{1}{4}$

$$a_{01} = 9;\ a_{02} = 6;\ a_{03} = 3$$

$$\left(\frac{3}{8} - 1\right) \text{máx} \left\{\frac{9}{-3/4}, \frac{6}{-1/2}, \frac{3}{-1/4}\right\}$$

$$\left(\frac{3}{8} - 1\right) \text{máx} \{-12,-12,-12\} = \left(-\frac{5}{8}\right)(-12) = \frac{15}{2}$$

Tabela de penalidades

	Níveis	x_1	x_2
Penalidade verdadeira p'_j	0	$\alpha_{10}=9$	$\alpha_{20}=0$
	1	$\alpha_{11}=6$	$\alpha_{21}=0$
	2	$\alpha_{12}=3$	$\alpha_{22}=0$
	3	$\alpha_{13}=0$	$\alpha_{23}=0$
	4		$\alpha_{24}=0$
Pseudo-penalidade p''_j	0	$\alpha_{10}=0$	$\alpha_{20}=3$
	1	$\alpha_{11}=0$	$\alpha_{21}=0$
	2	$\alpha_{12}=0$	$\alpha_{22}=0$
	3	$\alpha_{13}=0$	$\alpha_{23}=0$
	4		$\alpha_{24}=15/2$

Estratégia a ser explorada:

(0,0) (1,0) (2,0) (3,0)

(0,1) (1,1) (2,1) (3,1)

(0,2) (1,2) (2,2) (3,2)

(0,3) (1,3) (2,3) (3,3)

(0,4) (1,4) (2,4) (3,4)

$$p_{(0,0)} = \text{máx}(9,3) = 9$$

$$p_{(0,1)} = p_{(0,2)} = p_{(0,3)} = \text{máx}(9,0) = 9$$

$$p_{(0,4)} = \text{máx}(9, \frac{15}{2}) = 9$$

$$p_{(1,0)} = \text{máx}(6,3) = 6$$

$$p_{(1,1)} = p_{(1,2)} = p_{(1,3)} = \text{máx}(6,0) = 6$$

$$p_{(1,4)} = \text{máx}(6, \frac{15}{2}) = 6$$

$$p_{(2,0)} = \text{máx}(3,3) = 3$$

$$p_{(2,1)} = p_{(2,2)} = p_{(2,3)} = \text{máx}(3,0) = 3$$

$$p_{(2,4)} = \text{máx}(3, \frac{15}{2}) = \frac{15}{2}$$

$$p_{(3,0)} = \text{máx}(0,3) = 3$$

$$p_{(3,1)} = p_{(3,2)} = p_{(3,3)} = \text{máx}(0,0) = 0$$

$$p_{(3,4)} = \text{máx}(0, \frac{15}{2}) = \frac{15}{2}$$

Escolhendo min p_s temos 3 possibilidades

$p_{(3,1)}$, $p_{(3,2)}$, $p_{(3,3)}$

Tendo em vista a resolução do 2º exemplo no capítulo II, escolheremos $p_{(3,1)}$ então:

$x_1=3$ e $x_2=1$

Por conseguinte,

$\alpha_{13} = 1\ (j=3)\ ;\ \alpha_{21} = 1\ (j=1)$

Como α_{13} está na base do ótimo contínuo, acrescentemos pois uma nova restrição envolvendo α_{21}, pois esta variável pela solução escolhida deve ser igual a 1 e não se encontra na base.

Pelo fato de α_{21} ser igual a 1, então podemos acrescentar as restrições:

$$\alpha_{21} \leq 1$$

$$\alpha_{21} \geq 1$$

Colocando variáveis de folga vem:

$$\begin{cases} \alpha_{21} + f_{21}^{+} = 1 \\ \alpha_{21} - f_{21}^{-} = 1 \end{cases} \rightarrow \begin{cases} \alpha_{21} + f_{21}^{+} = 1 \\ -\alpha_{21} + f_{21}^{-} = -1 \end{cases}$$

Acrescentando agora o corte realizado ao quadro 4 vem:

Quadro 4

	$\overline{Z}$	α_{10}	α_{11}	α_{12}	α_{13}	α_{20}	α_{21}	α_{22}	α_{23}	α_{24}	x_3	x_4	f_{21}^{+}	f_{21}^{-}
Z	18	9	6	3	0	0	0	0	0	0	1	0	0	0
α_{24}	3/8	-3/4	-1/2	-1/4	0	0	1/4	1/2	3/4	1	1/8	0	0	0
x_4	1/2	-6	-4	-2	0	0	0	0	0	0	-1/2	1	0	0
α_{13}	1	1	1	1	1	0	0	0	0	0	0	0	0	0
α_{20}	5/8	3/4	1/2	1/4	0	1	3/4	1/2	1/4	0	-1/8	0	0	0
f_{21}^{+}	1	0	0	0	0	0	1	0	0	0	0	0	1	0
f_{21}^{-}	-1	0	0	0	0	0	-1	0	0	0	0	0	0	0

O quadro 5 é primal inviável; então pelo critério de seleção do pivô do método dual do simplex vem:

$\min(|\frac{0}{-1}|) = 0$, α_{21} entra na base, f_{21}^{-} sai da base e -1 é o pivô.

Quadro 5

	$\bar{Z}$	α_{10}	α_{11}	α_{12}	α_{13}	α_{20}	α_{21}	α_{22}	α_{23}	α_{24}	x_3	x_4	f_{21}^+	f_{21}^-
Z	18	9	6	3	0	0	0	0	0	0	1	0	0	0
α_{24}	1/8	-3/4	-1/2	-1/4	0	0	0	1/2	3/4	1	1/8	0	0	1/4
x_4	1/2	-6	-4	-2	0	0	0	0	0	0	-1/2	1	0	0
α_{13}	1	1	1	1	1	0	0	0	0	0	0	0	0	0
α_{20}	-1/8	3/4	1/2	1/4	0	1	0	1/2	1/4	0	-1/8	0	0	3/4
f_{21}^+	0	0	0	0	0	0	0	0	0	0	0	0	1	1
α_{21}	1	0	0	0	0	0	1	0	0	0	0	0	0	-1

O quadro 6 é primal inviável, escolhendo $\text{Min}\left(\left|\frac{1}{-1/8}\right|\right) = 8$, x_3 entra na base, α_{20} sai da base e $-\frac{1}{8}$ é o pivô.

Quadro 6

	$\overline{Z}$	α_{10}	α_{11}	α_{12}	α_{13}	α_{20}	α_{21}	α_{22}	α_{23}	α_{24}	x_3	x_4	f^+_{21}	f^-_{21}
Z	17	15	10	5	0	8	0	4	2	0	0	0	0	6
α_{24}	0	0	0	0	0	1	0	1	1	1	0	0	0	1
x_4	1	-9	-6	-3	0	-4	0	-2	-1	0	0	1	0	-3
α_{13}	1	1	1	1	1	0	0	0	0	0	0	0	0	0
x_3	1	-6	-4	-2	0	-8	0	-4	-2	0	1	0	0	-6
f^+_{21}	0	0	0	0	0	0	0	0	0	0	0	0	1	1
α_{21}	1	0	0	0	0	0	1	0	0	0	0	0	0	-1

O quadro acima nos dá a solução requerida para o problema inteiro-misto, $Z^* = 17$; $x_3^* = x_4^* = 1$; $\alpha_{13}^* = 1$; $\alpha_{21}^* = 1$; $\alpha_{13}^* = 0$; $f^+_{21} = 0$

Pelo fato de $\alpha_{13} = 1$ $(j=3)$ e $\alpha_{21} = 1$ $(j=1)$ e

$$x_1 = \sum_{j=0}^{3} j\alpha_{1j} \quad \rightarrow \quad x_1 = 3$$

$$x_2 = \sum_{j=0}^{4} j\alpha_{2j} \quad \rightarrow \quad x_2 = 1$$

Suponhamos usar o custo $p_{(3,2)}$; então a solução a ser pesquisada é $S = (3,2)$

$$x_1=3 \quad \text{e} \quad x_2=2$$

logo $\alpha_{13} = 1$ $(j=3)$; $\alpha_{22} = 1$ $(j=2)$

Considerando o quadro do ótimo contínuo, α_{13} está na base com valor 1; porém α_{22} não está na base. Como $\alpha_{22} = 1$, acrescentando as restrições

$$\alpha_{22} \leq 1$$

$$\alpha_{22} \geq 1$$

vem:

$$\begin{cases} \alpha_{22} + f_{22}^{+} = 1 \\ -\alpha_{22} + f_{22}^{-} = -1 \end{cases}$$

Quadro 4'

	$\overline{Z}$	α_{10}	α_{11}	α_{12}	α_{13}	α_{20}	α_{21}	α_{22}	α_{23}	α_{24}	x_3	x_4	f_{22}^+	f_{22}^-
Z	18	9	6	3	0	0	0	0	0	0	1	0	0	0
α_{24}	3/8	-3/4	-1/2	-1/4	0	0	1/4	1/2	3/4	1	1/8	0	0	0
x_4	1/2	-6	-4	-2	0	0	0	0	0	0	-1/2	1	0	0
α_{13}	1	1	1	1	1	0	0	0	0	0	0	0	0	0
α_{20}	5/8	3/4	1/2	1/4	0	1	3/4	1/2	1/4	0	-1/8	0	0	0
f_{22}^+	1	0	0	0	0	0	0	1	0	0	0	0	1	0
f_{22}^-	-1	0	0	0	0	0	0	-1	0	0	0	0	0	1

O quadro é primal inviável, introduzindo α_{22} na base $(\min\left|\frac{0}{-1}\right|)$ f_{22}^- sai da base e -1 é o pivô.

Quadro 5'

	$\overline{Z}$	α_{10}	α_{11}	α_{12}	α_{13}	α_{20}	α_{21}	α_{22}	α_{23}	α_{24}	x_3	x_4	f^+_{22}	f^-_{22}
Z	18	9	6	3	0	0	0	0	0	0	1	0	0	0
α_{24}	-1/8	-3/4	-1/2	-1/4	0	0	1/4	0	3/4	1	1/8	0	0	1/2
x_4	1/2	-6	-4	-2	0	0	0	0	0	0	-1/2	1	0	0
α_{13}	1	1	1	1	1	0	0	0	0	0	0	0	0	0
α_{20}	1/8	3/4	1/2	1/4	0	1	3/4	0	1/4	0	-1/8	0	0	1/2
f^+_{22}	0	0	0	0	0	0	0	0	0	0	0	0	1	1
α_{22}	1	0	0	0	0	0	0	1	0	0	0	0	0	-1

O quadro é primal inviável, excolhendo $\min\left(\left|\frac{9}{-3/4}\right|, \left|\frac{6}{-1/2}\right|, \left|\frac{3}{-1/4}\right|\right) = 12$ e introduzindo α_{10} na base teremos como pivô $-\frac{3}{4}$ e α_{24} sai da base

Quadro 6'

	$\overline{Z}$	$\alpha_{1\cdot 0}$	$\alpha_{1\cdot 1}$	$\alpha_{1\cdot 2}$	$\alpha_{1\cdot 3}$	$\alpha_{2\cdot 0}$	$\alpha_{2\cdot 1}$	$\alpha_{2\cdot 2}$	$\alpha_{2\cdot 3}$	$\alpha_{2\cdot 4}$	x_3	x_4	$f^+_{2\cdot 2}$	$f^-_{2\cdot 2}$
Z	33/2	0	0	0	0	0	9	0	9	12	3/2	0	0	6
$\alpha_{1\cdot 0}$	1/6	1	2/3	1/3	0	0	-1/3	0	-1	-4/3	-1/6	0	0	-2/3
x_4	3/2	0	0	0	0	0	-2	0	-6	-8	-3/2	1	0	-4
$\alpha_{1\cdot 3}$	5/6	0	1/3	2/3	1	0	1/3	0	1	4/3	1/6	0	0	2/3
$\alpha_{2\cdot 0}$	0	0	0	0	0	1	1	0	1	1	0	0	0	1
$f^+_{2\cdot 2}$	0	0	0	0	0	0	0	0	0	0	0	0	1	1
$\alpha_{2\,2}$	1	0	0	0	0	0	0	1	0	0	0	0	0	-1

Observa-se pelo quadro 7', que a solução encontrada não é a requerida pelo problema. Se todavia procuramos fazer o mesmo para a possível solução (3,3), teremos encontrado uma solução semelhante a esta encontrada acima.

3.3.- Comparação entre um problema resolvido segundo o artigo de Norman J. Driebeek e a resolução pela modificação proposta por Stanley Zionts

A vantagem do artigo original sobre a descrição de Zionts é o não aumento da matriz A do ótimo contínuo, porém o problema de convergência mais ou menos rápida, depende do tipo de problema que está sendo examinado.

Para o exemplo anterior, a solução segundo o artigo original exigiu 10 iterações; a formação da matriz $\hat{B}^{-1}$ assim como a introdução do vetor $\underset{\sim}{\Delta b}$, embora não sejam complicadas, fazem com que demoremos um pouco mais do que se tivéssemos introduzido mais uma restrição no PPL como Zionts propôs. Neste exemplo, considerando o modo de Zionts escrever uma variável inteira, a convergência foi muito mais rápida, porém convém ressaltar que isto foi um caso isolado.

3.4.- <u>O algoritmo de Driebeek segundo Stanley Zionts</u>

Considerando Zionts, o algoritmo de Driebeek pode ser enunciado como se segue:

(1) Resolução do PPL correspondente

(2) Seleção de um nível desejado para cada variável inteira. Usualmente seleciona-se o nível que corresponde a

mais baixa penalidade em cada conjunto.

(3) Limitação do espaço solução viável, pela introdução de restrições tendo em vista a seleção do nível de variável inteira desejado, i.e, se a seleção de nível impõe $\alpha_{ij} = 1$ então acrescentamos as restrições

$$\alpha_{ij} \geq 1$$

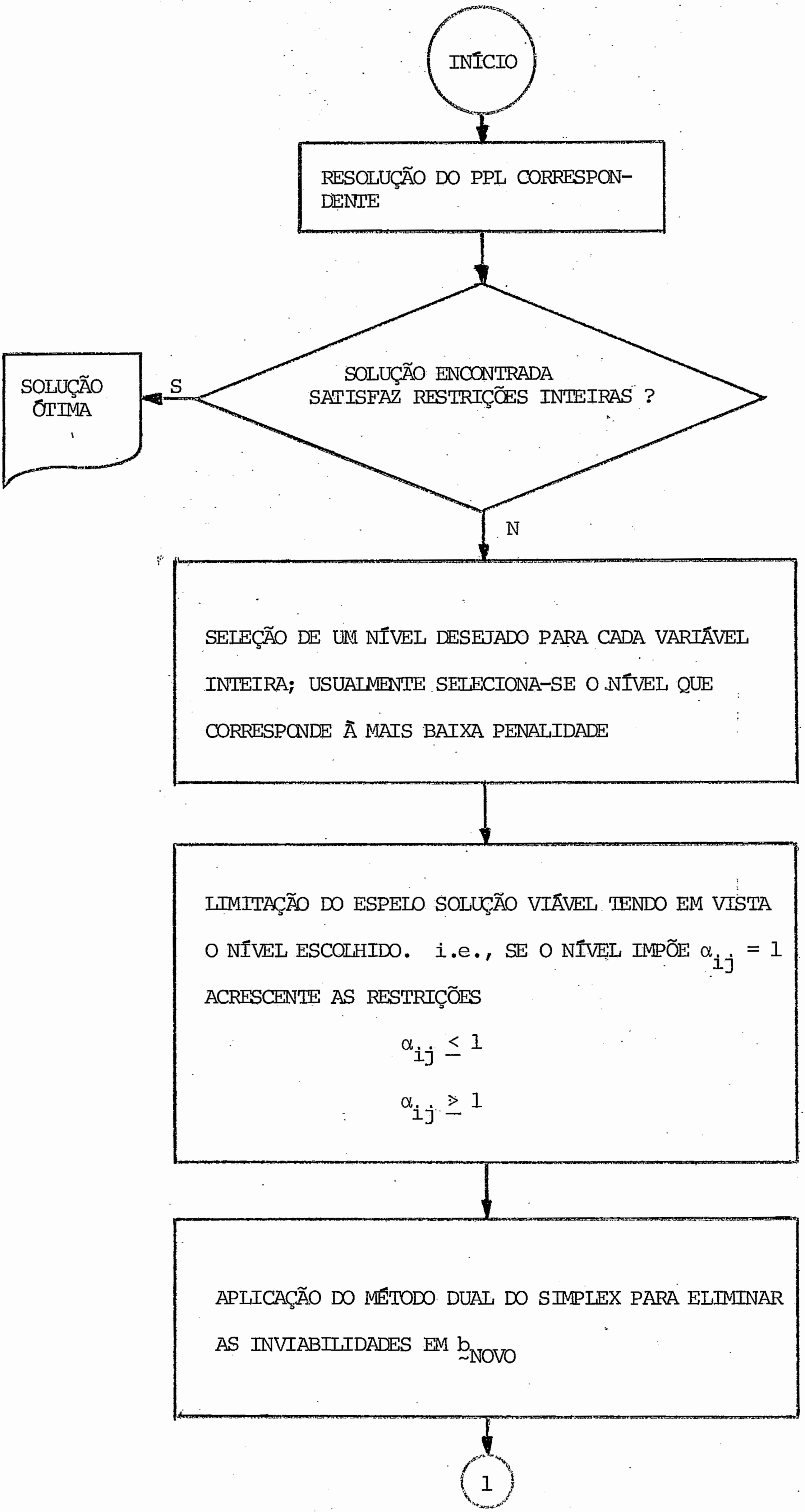
$$\alpha_{ij} \leq 1$$

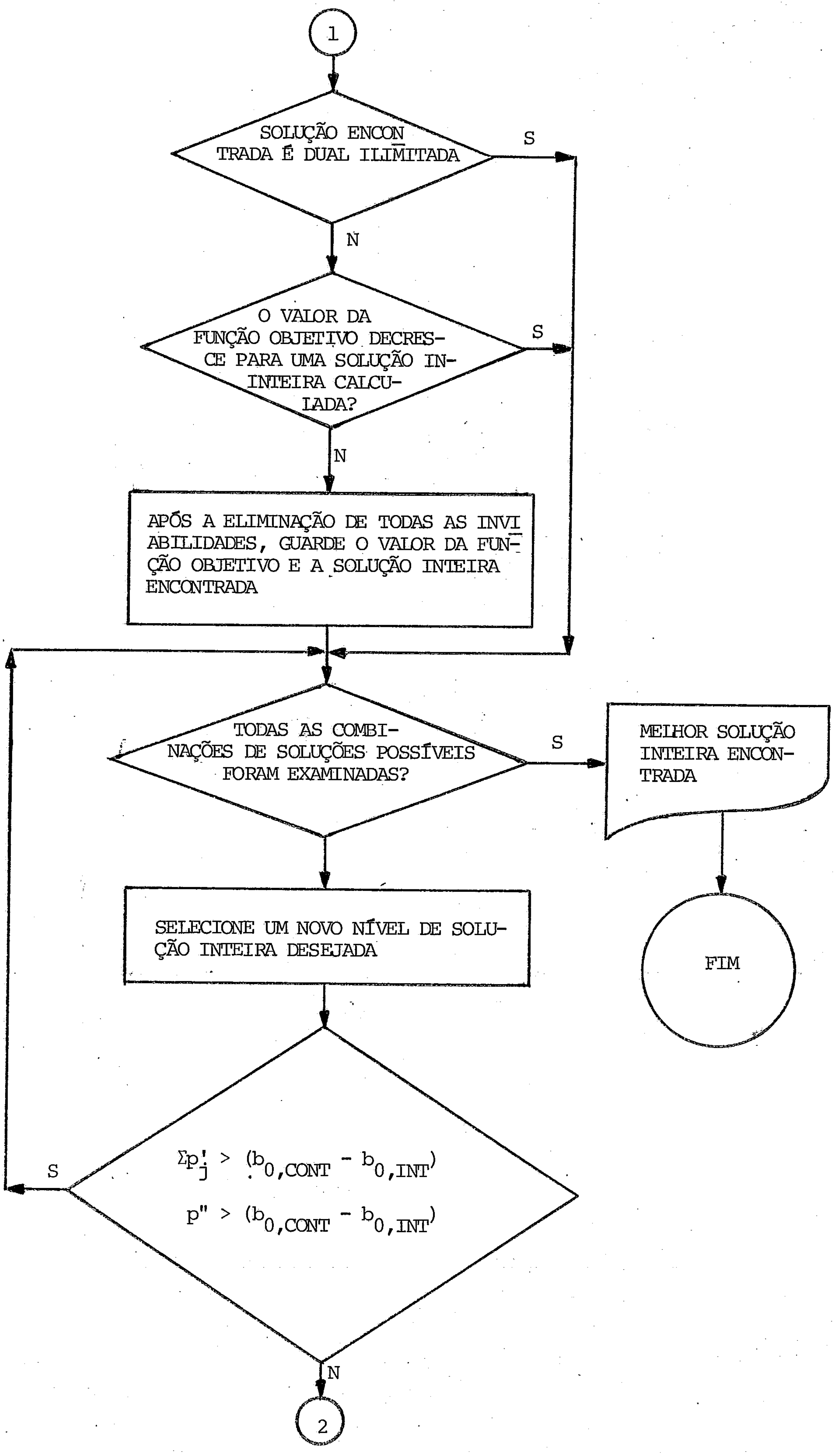
(4) Aplicação do método dual do simplex para eliminar as inviabilidades no novo $\underset{\sim}{b}_{cont}$ do PPL. Se uma solução dual ilimitada é encontrada ou se o valor da função objetivo decresce para uma solução previamente calculada vá para (6), senão vá para (5)

(5) Quando todas as inviabilidades tiverem sido eliminadas, uma solução inteira válida é encontrada; guarda-se a solução e o valor da função objetivo. $(b_{o,int})$

(6) Seleção de nova solução inteira desejada. Se $\Sigma\ p'_j$ ou máx p''_j for maior que $(b_{o,cont} - b_{o,int})$ vá para (6), senão vá para (2). Se todas as combinações possíveis tiverem sido investigadas desta maneira pare. A solução do problema misto é a melhor solução encontrada antes de parar.

FLUXOGRAMA DO ALGORITMO DE DRIEBEEK SEGUNDO ZIONTS




1
SOLUÇÃO ENCON TRADA É DUAL ILIMITADA
S
N
O VALOR DA FUNÇÃO OBJETIVO DECRES- CE PARA UMA SOLUÇÃO IN- INTEIRA CALCU- LADA?
S
N
APÓS A ELIMINAÇÃO DE TODAS AS INVI ABILIDADES, GUARDE O VALOR DA FUN- ÇÃO OBJETIVO E A SOLUÇÃO INTEIRA ENCONTRADA
TODAS AS COMBI- NAÇÕES DE SOLUÇÕES POSSÍVEIS FORAM EXAMINADAS?
S
MELHOR SOLUÇÃO INTEIRA ENCON- TRADA
FIM
SELECIONE UM NOVO NÍVEL DE SOLU- ÇÃO INTEIRA DESEJADA
$\Sigma p'_j > (b_{0,CONT} - b_{0,INT})$
$p'' > (b_{0,CONT} - b_{0,INT})$
S
N
2

BIBLIOGRAFIA

|1| Dantizig, G. B., "On the Significance of Solving Linear Programming Problems with Some Integer Variables"- Econometrica, 28, nº 1, January 1960 - pp.30-44.

|2| Driebeek, N. J., "An Algorithm For the Solution of Mixed Integer Programming" - Management Science - Vol 12, Nº 7, March 1966 - USA.

|3| Gass, S. I., Linear Programming, MCGRAW-HILL - Kōgakusha - 3rd. Edition TOKYO 1969.

|4| Gomory, R. E., "An Algorithm for Integer Solutions to Linear Programs", Princeton IBM Math. Res. Report, Nº 1, November 1958.

|5| Gomory, R. E., "All-Integer Integer Programming Algorithm", IBM Research Center Report RC-189, January 1960.

|6| Hadley, G., Linear Programming, Addison Wesley Publishing Company, Inc - 1962.

|7| Hu, T.C., Integer Programming and Network Flows, Addison-Wesley Publishing Company, Inc - 1969.

|8| Maculan Filho, N., Notas de Aula em Programação Inteira - Mista, COPPE-SISTEMAS, UFRJ, Setembro 1974.

|9| Maculan Filho, N., Notas de Aula em Programação Inteira - Mista, COPPE-SISTEMAS, UFRJ, Novembro 1976.

|10| Nascimento, P., Notas de Aula em Programação Linear, PROG. ENG. PRODUÇÃO, UFSC, 1973.

|11| Simonnard, M., Programmation Linéaire, Dunod, Paris - 2a. Edição - 1972.

|12| Young, R. D., "A Simplified Primal (All-Integer) Integer Programming Algorithm", J. ORSA, 16(4) (July-August 1968) 750-782.

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

APÊNDICE

```
          B6700/B7700   F O R T R A N   C O M P I L A T I O N   M A R K   2.
FILE    5=IMP,UNIT=PRINTER
FILE    8=LEIT,UNIT=READER
        SUBROUTINE MENLZ(A,K,M)
        DIMENSION A(12,19)
C
C       ESTA SUBROUTINE PROCURA O MENOR DA LINHA Z
C
      9 IF(A(1,K).LT.0)GOTO 7
        K=K+1
        IF(K.GT.19)GOTO 87
        GOTO 9
      7 B2=A(1,K)
        M=K
        KY=M-1
     12 K=K+1
C
C
C
        IF(K.GT.19)GOTO 11
        IF(B2.GT.A(1,K))GOTO 7
        GOTO 12
     11 WRITE(5,13)M,B2,K,KY
     13 FORMAT(10X,'O MENOR DA LINHA Z E',5X,'B=A(1,',I2,')',3X,F5.2,5X,'K
       *=',2X,I2//10X,'ENTRA X',I2/)
        K=19
     87 RETURN
        END
```

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
NÚCLEO DE COMPUTAÇÃO ELETRÔNICA

```
        SUBROUTINE PVSMP(A,N,M,J0,IX)
C
C       ESTA SUBROUTINE PROCURA O PIVOT DO SIMPLEX
C
        DIMENSION A(12,19),IX(11)
        WRITE(5,999)M
     16 IF(A(N,M).GT.0)GOTO 15
        N=N+1
        WRITE(5,999)N
    999 FORMAT(' ',I4/)
        IF(N.LE.12)GOTO 16
     17 WRITE(5,18)
     18 FORMAT(10X,'PPL TEM SOLUCAO ILIMITADA')
        RETURN
     15 F=A(N,1)/A(N,M)
        J0=N
        IF(N.EQ.12)GOTO 301
        DO 19 K0=N,12
        IF(A(K0,M).LE.0)GOTO 19
        G=A(K0,1)/A(K0,M)
        IF(F-G)19,195,21
     21 F=G
    195 J0=K0
     19 CONTINUE
C
C
        WRITE(5,9008)J0
   9008 FORMAT(/'   O VALOR DE J0  E ',10X,I4//)
        WRITE(5,22)J0,M,A(J0,M),IX(J0-1)
     22 FORMAT(10X,'A(',I2,',',I2,')=',1X,F5.2,10X,'E O PIVOT',5X,'SAI DA
       *BASE X',I2//)
    301 RETURN
        END
```

```
      SUBROUTINE PIVOTT(A,J0,M,IX,PIVOT)
      DIMENSION A(12,19),IX(11)
C
C     PIVCTEAMENTO DO SIMPLEX
C     PIVCTEAMENTO DO SIMPLEX
C
      DO 24 K2=1,12
      IF(K2.EQ.J0)GOTO 24
      DO 23 J1=1,19
      IF(J1.EQ.M)GOTO 23
      A(K2,J1)=A(K2,J1)-((A(J0,J1)/PIVOT)*A(K2,M))
   23 CONTINUE
   24 CONTINUE
      DO 1000 K2=1,12
      IF(K2.EQ.J0)GOTO 1000
      A(K2,M)=0.
 1000 CONTINUE
      DO 26 J1=1,19
   26 A(J0,J1)=A(J0,J1)/PIVOT
      L=J0-1
      IX(L)=M-1
      RETURN
      END
```

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
NÚCLEO DE COMPUTAÇÃO ELETRÔNICA

```
      SUBROUTINE PVDUAL(A,NO,MAO,J0,M)
      DIMENSION A(12,19)
C
C     ESTA SUBROUTINE PROCURA O PIVOT PELO DUAL DO SIMPLEX
C
      R=ABS(A(1,MAO)/A(NO,MAO))
      M1=MAO
      DO 400 LAN=MAO,19
      IF(A(NO,LAN).GE.0)GOTO 400
C
      S=ABS(A(1,LAN)/A(NO,LAN))
      IF(R.LE.S)GOTO 400
      R=S
      M1=LAN
  400 CONTINUE
      WRITE(5,68)R,NO,M1,A(NO,M1)
   68 FORMAT(5X,F8.3,3X,'A(',I2,',',I2,')=',3X,F8.3,2X,'E O PIVOT'//)
C
C
      J0=NO
      M=M1
      RETURN
      END
```

```
      SUBROUTINE ESTR(B1,P,P1,IAM,IAM1,KA)
      DIMENSION B1(9),P(9),P1(4,5),IAM(6),IAM1(6)
C
C     ESTA SUBROUTINE ESCOLHE AS ESTRATEGIAS MINIMAS A SEREM EXPLORADAS
C
      DO 1400 MAL=1,4
      DO 1400 LAN=1,5
      A1=B1(MAL)+B1(LAN+4)
      IF(P(MAL).GT.P(LAN+4))GOTO 871
      A2=P(LAN+4)
      GOTO 872
  871 A2=P(MAL)
  872 IF(A1.GT.A2)GOTO 873
      P1(MAL,LAN)=A2
      GOTO 1400
  873 P1(MAL,LAN)=A1
 1400 CONTINUE
      WRITE(5,874)((P1(I,J),J=1,5),I=1,4)
  874 FORMAT(//1X,'A MATRIZ DE ESTRATEGIAS E'//5(5X,F5.2)/)
C
C
C
      I=1
      AM=P1(1,1)
 1504 DO 1500 MAL=1,4
      DO 1500 LAN=1,5
      IF(KA.EQ.0)GOTO 876
      IF(AM.NE.P1(MAL,LAN))GOTO 1500
      IF(MAL.EQ.IAM(I).AND.LAN.EQ.IAM1(I))GOTO 1500
      I=I+1
      WRITE(5,1589)I
 1589 FORMAT(10X,'I=',3X,I3//)
      IAM(I)=MAL
      IAM1(I)=LAN
      GOTO 1500
  876 IF(P1(MAL,LAN).GE.AM)GOTO 1500
      AM=P1(MAL,LAN)
      IAM(I)=MAL
      IAM1(I)=LAN
 1500 CONTINUE
      KA=1
      IF(IYY1.NE.0)GOTO 1502
      IYY1=1
      GOTO 1504
 1502 WRITE(5,1590)(IAM(I),I=1,6),(IAM1(I),I=1,6)
 1590 FORMAT(//10X,'INDICES DOS MENORES NAS LINHAS',5X,6(3X,I2)//45X,6(3
     *X,I2)/)
      RETURN
      END
```

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
NÚCLEO DE COMPUTAÇÃO ELETRÔNICA

```
      SUBROUTINE PENAL(P,B1,IX,IZ,A,J)
      DIMENSION P(9),B1(9),IX(11),A(12,19)
C
C     ESTA SUBROUTINE CALCULA AS PENALIDADES PARA AS VARIAVEIS 0-1 DO PP
C
      DO 86 LJ=1,9
      B1(LJ)=0.
   86 P(LJ)=0.
      DO 85 LJ=1,9
   85 B1(LJ)=A(1,LJ+10)
      WRITE(5,851)(B1(LJ),LJ=1,9)
  851 FORMAT(//1X,'O VETOR DAS PENALIDADES VERDDR E'//1X,9(3X,F5.2))
      DO 1648 M3=1,11
      LJ=10
  850 IF(IX(M3).EQ.LJ)GOTO 852
      LJ=LJ+1
      IF(LJ.GT.18)GOTO 1648
      GOTO 850
  852 JMW=IZ+1
  854 IF(A(M3+1,JMW).LT.0)GOTO 853
      JMW=JMW+1
      IF(JMW.LT.19)GOTO 854
      P(LJ-9)=0.
      GOTO 1648
C
C
  853 G1=A(1,JMW)/A(M3+1,JMW)
      DO 1649 K20=JMW,19
      IF(A(M3+1,K20).GE.0)GOTO 1649
      G2=A(1,K20)/A(M3+1,K20)
      IF(G1.GE.G2)GOTO 1649
      G1=G2
 1649 CONTINUE
      P(LJ-9)=(A(M3+1,J)-1.)*G1
 1648 CONTINUE
      WRITE(5,856)(P(LJ),LJ=1,9)
  856 FORMAT(//1X,'O VETOR DAS PENALIDADES FALSAS E'//1X,9(3X,F5.2))
      RETURN
      END
```

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
NÚCLEO DE COMPUTAÇÃO ELETRÔNICA

```
      SUBROUTINE MENCZ(A,IX,L,IYY,IRR,NO,MAO,D,IX1)
C
C     ESTA SUBROUTINE PROCURA O MENOR DA COLUNA Z NO TABLEAU DO SIMPLEX
C
      DIMENSION A(12,19),IX(11),D(12,19),IX1(11)
   29 IF(A(L,1).LT.0)GOTO 27
      L=L+1
      IF(L.LE.12)GOTO 29
   38 WRITE(5,31)A(1,1)
      WRITE(5,907)(IX(KW),KW=1,11)
  907 FORMAT(2X,' AS VARIAVEIS NA BASE SAO ',10X,11(5X,I2)//)
   31 FORMAT(10X,'Z=',2X,F5.2,3X,'E A SOLUCAO OTIMA DO PPL')
      IF(IYY.EQ.0)GOTO 908
  909 IRR=0
      RETURN
      RETURN
  908 DO 953 I=1,12
      DO 953 J=1,19
  953 D(I,J)=A(I,J)
      DO 954 J12=1,11
  954 IX1(J12)=IX(J12)
      GOTO 909
   27 C=A(L,1)
      NO=L
   52 L=L+1
      IF(L.GT.12)GOTO 51
      IF(C.GE.A(L,1))GOTO 27
      GOTO 52
   51 WRITE(5,54)NO,C,NO
   54 FORMAT(10X,'A(',I2,',1)=',2X,F10.3,5X,'NO=',2X,I2/)
      IF(ABS(C).LE.1.0E-5)GOTO 38
      MAO=2
   58 IF(A(NO,MAO).LT.0)GOTO 569
      MAO=MAO+1
      IF(MAO.LE.19)GOTO 58
      WRITE(5,60)
   60 FORMAT(10X,' O PPL NAO POSSUE SOLUCAO '//)
      IRR=1
      RETURN
  569 IRR=-1
      RETURN
      END
```


UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
NÚCLEO DE COMPUTAÇÃO ELETRÔNICA

```
      DIMENSION A(12,19),B1(9),P(9),B(12,12),IX(11),P1(4,5),DELTA(12),CB
     *(18),IAM(6),IAM1(6),D(12,19),IX1(11),CB1(11),F(6,3)
      KA1=0
      KA=0

      MIL=0
      J0=1
      IRR=2

      IYY=0
      IYY1=0
      IAL1=0
      READ(8,1)((A(I,J),J=1,19),I=1,12)
    1 FORMAT(19F4.2)

      READ(8,2)(IX(L),L=1,11)
    2 FORMAT(11I2)
      READ(8,5001)(CB(K12),K12=1,18)
 5001 FORMAT(18F4.2)

      WRITE(5,3)
    3 FORMAT(10X,'A MATRIZ INICIAL E'///)
      WRITE(5,4)((A(I,J),J=1,19),I=1,12)
    4 FORMAT(19(2X,F5.2)/)
      WRITE(5,6)(IX(L),L=1,11)

    6 FORMAT(10X,'AS VARIAVEIS BASICAS SAO'//11(6X,I2)//)
   50 WRITE(5,4)((A(I,J),J=1,19),I=1,12)
      WRITE(5,6)(IX(L),L=1,11)
      K=2
      M=0

      CALL MENLZ(A,K,M)
      IF(K.GT.19)GOTO 80
      N=2
C
C
C     PROCURA PIVOT DO SIMPLEX
C
C
      CALL PVSMP(A,N,M,J0,IX)
      IF(N.GT.12)GOTO 1870

   30 PIVOT=A(J0,M)
C
C
C     CHAMADA DA SUBROUTINE QUE FAZ O PIVOTEAMENTO DO SIMPLEX
C
      CALL PIVOTT(A,J0,M,IX,PIVOT)
C
      GOTO 50
C
```

```
C
   80 L=2
      NO=0

      MAO=0
      CALL MENCZ(A,IX,L,IYY,IRR,NO,MAO,D,IX1)
      IF(IRR)56,7007,1870
 1870 IF(KA1.EQ.0)GOTO 187
      GOTC 2825
C
C
C     DETERMINACAO DO PIVOT PELO DUAL DO SIMPLEX

C
C
   56 CALL PVDUAL(A,NO,MAO,JO,M)
      GOTO 30
 7007 WS1=0.
      WS2=0.
      DO 1800 M3=1,11

      IF(IX(M3).GE.4)GOTO 801
      WS1=WS1+A(M3+1,1)
      GOTO 1800
  801 IF(IX(M3).GT.7)GOTO 1800
      WS2=WS2+A(M3+1,1)
 1800 CONTINUE
      SK1=WS1-((IFIX(WS1)/2.)*2.)
      SK2=WS2-((IFIX(WS2)/2.)*2.)
C

C
C
      IF(SK1.LT.1.E-3)GCTO 1621
      MIL=1
      GOTC 1620
 1621 WRITE(5,1622)WS1

 1622 FORMAT(10X,'ALGOTRITMO SATISFEITO,PRA X1',5X,'X1=',2X,F8.3/)
 1620 IF(SK2.GE.1.E-3)GCTO 1623
      WRITE(5,1624)WS2
 1624 FORMAT(10X,'ALGOTITMO SATISFEITO PARA X2',5X,'X2=',2X,F8.3/)
      GOTO 1625

 1623 MIL=1
 1625 IF(IYY.EQ.1)GOTO 2800
      IF(MIL.EQ.1)GOTO 1628

      IF(IAL1.EQ.0)GOTO 187
C
```

```
C
      GOTO 2800
 1628 IZ=1
      J=IZ
C
C     CHAMADA DA SUBROUTINE QUE  CALCULA AS PENALIDADES.
C
      CALL PENAL(P,B1,IX,IZ,A,J)
C
C     CHAMADA DA SUBROUTINE QUE  DETERMINA AS ESTRATEGIAS
C
      CALL ESTR(B1,P,P1,IAM,IAM1,KA)
C
      MY=1
  751 DO 1600 JOT=1,12
 1600 DELTA(JOT)=0.
      IF(IAM(MY).EQ.0)GOTO 756
      LW1=3+IAM(MY)
      LW2=7+IAM1(MY)
      DELTA(LW1)=1.
      DELTA(LW2)=1.
      WRITE(5,1700)(DELTA(JOT),JOT=1,12)
 1700 FORMAT(/10X,' O VETOR DELTA E '//4X,12(2X,F5.2)/)
      DO 1900 I=1,12
      DO 1900 J=1,12
 1900 B(I,J)=0.
      DO 2000 I=2,12
      DO 2006 J=2,12
 2006 B(I,J)=A(I,J+7)
 2000 CONTINUE
      WRITE(5,1701)((B(I,J),J=1,12),I=1,12)
 1701 FORMAT(10X,' A MATRIZ B MENOSUM E '///12(3X,F8.3)//)
C
C
C     NOVA MATRIZ B MENOSUM
C
C
      DO 2401 I=1,11
      LWS=IX1(I)
 2401 CB1(I)=CB(LWS)
      WRITE(5,2481)(CB1(I),I=1,11)
 2481 FORMAT(//10X,'O VETOR CUSTO E'//10X,11(3X,I2))
      B(1,1)=1.
      DO 2010 J=2,12
      S=0.
      DO 2001 I=2,12
 2001 S=S+CB1(I-1)*B(I,J)
 2010 B(1,J)=S
      WRITE(5,1702)((B(I,J),J=1,12),I=1,12)
 1702 FORMAT(//10X,' A NOVA MATRIZ B MENOSUM E '///12(3X,F8.3)//)
C
```

```
C
      DO 2003 I=1,12
      S=0.
      DO 2004 J=1,12
 2004 S=S+B(I,J)*(-DELTA(J))
      A(I,1)=S+A(I,1)
 2003 CONTINUE
      WRITE(5,1703)((A(I,J),J=1,19),I=1,12)
 1703 FORMAT(10X,' A MATRIZ APOS A INTRODUCAO DO VETOR DELTA E'///19(1X,
     *F5.2)///)
      IRR=-1
      IAL1=1
      IYY=1
      KA1=1
      GOTO 80
C
C
C
 2800 F(MY,1)=A(1,1)
      F(MY,2)=WS1
      F(MY,3)=WS2
 2825 IF(MY.GT.6)GOTO 756
      MY=MY+1
      DO 3007 I=1,12
      DO 3007 J=1,19
 3007 A(I,J)=D(I,J)
      DO 3008 J=1,11
 3008 IX(J)=IX1(J)
      WRITE(5,4)((A(I,J),J=1,19),I=1,12)
      WRITE(5,6)(IX(L),L=1,11)
      GOTO 751
CC
  756 WRITE(5,758)((F(I,J),J=1,3),I=1,6)
  758 FORMAT(10X,' A MATRIZ DE SOLUCOES E'//3(5X,F5.2)//)
      MAIOR=F(1,1)
      DO 3009 MY=1,4
      IF(MAIOR-F(MY,1))3010,3011,3009
 3010 MAIOR=F(MY,1)
      L20=MY
      GOTO 3009
 3011 WRITE(5,759)F(MY,1),F(MY,2),F(MY,3)
  759 FORMAT(10X,'Z=',F8.3,5X,'X1=',F8.3,5X,'X2=',F8.3/)
 3009 CONTINUE
      WRITE(5,760)MAIOR,F(L20,2),F(L20,3)
  760 FORMAT(//10X,'A MELHOR SOLUCAO E',10X,'ZOT=',F8.3,5X,'X1=',F8.3,5X
     *,'X2=',F8.3)
  187 STOP
      END
```

```
NO ERRORS DETECTED.  NUMBER OF CARDS = 423.

COMPILATION TIME = 44 SECONDS ELAPSED.  6.80 SECONDS PROCESSING.
D2 STACK SIZE = 14 WORDS.  FILESIZE = 140 WORDS.  ESTIMATED CORE STORAGE REQUIREMENT = 2097 WORDS.
TOTAL PROGRAM CODE = 1144 WORDS.  ARRAY STORAGE = 731 WORDS.
NUMBER OF PROGRAM SEGMENTS = 13.  NUMBER OF DISK SEGMENTS = 118.

PROGRAM CODE FILE = (COS88021)PROGRAMA ON PACK. COMPILER COMPILED ON 11/12/76(FORTRAN ON PACK)
```

```
        B6700/B7700    F O R T R A N   C O M P I L A T I O N   M A R K   2.
FILE   5=TMP,UNIT=PRINTER
FILE   8=LEIT,UNIT=READER
       SUBROUTINE MENLZ(A,K,M)
       DIMENSION A(6,10)
C      ESTA SUBROUTINE PROCURA O MENOR DA LINHA Z
     9 IF(A(1,K).LT.0)GOTO 7
       K=K+1
       IF(K.GT.10)GOTO 87
       GOTO 9
     7 B2=A(1,K)
       M=K
       KY=M-1
    12 K=K+1
C
C
C
       IF(K.GT.10)GOTO 11
       IF(B2.GE.A(1,K))GOTO 7
       GOTO 12
    11 WRITE(5,13)M,B2,K,KY
    13 FORMAT(10X,'O MENOR DA LINHA Z E',5X,'B=A(1,',I1,')',3X,F8.3,5X,'K
      *=',2X,I2//10X,'ENTRA X',I2/)
       K=10
    87 RETURN
       END
```

```
      SUBROUTINE MENCZ(A,IX,L,KA,MIL,IRR,NO,MAO)
C     ESTA SUBROUTINE PROCURA O MENOR DA COLUNA Z NO TABLEAU DO SIMPLEX
      DIMENSION A(6,10),IX(5)
   29 IF(A(L,1).LT.0)GOTO 27
      L=L+1
      IF(L.LE.6)GOTO 29
   38 WRITE(5,31)A(1,1)
      WRITE(5,908)(IX(KW),KW=1,5)
  908 FORMAT(2X,' AS VARIAVEIS NA BASE SAO ',10X,7(5X,I2)//)
   31 FORMAT(10X,'Z=',2X,F8.3,3X,'E A SOLUCAO OTIMA DO PPL')
      IRR=0
      RETURN
   27 C=A(L,1)
      NO=L
C
C
   52 L=L+1
      IF(L.GT.6)GOTO 51
      IF(C.GE.A(L,1))GOTO 27
      GOTO 52
   51 WRITE(5,54)NO,C,NO
   54 FORMAT(10X,'A(',I1,',1)=',2X,F10.3,5X,'NO=',2X,I2/)
      IF(ABS(C).LE.1.0E-5)GOTO 38
      MAO=2
   58 IF(A(NO,MAO).LT.0)GOTO 569
      MAO=MAO+1
      IF(MAO.LE.10)GOTO 58
      WRITE(5,60)
C
C
C
   60 FORMAT(10X,'O PPL NAO POSSUE SOLUCAO')
      IRR=1
      RETURN
  569 IRR=-1
      RETURN
      END
```

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
NÚCLEO DE COMPUTAÇÃO ELETRÔNICA

```
      SUBROUTINE PVSMP(A,N,M,J0,IX)
C     ESTA SUBROUTINE PROCURA O PIVOT DO SIMPLEX
      DIMENSION A(6,10),IX(5)
C
      WRITE(5,999)M
   16 IF(A(N,M).GT.0)GOTO 15
      N=N+1
      WRITE(5,999)N
  999 FORMAT(' ',I4/)
      IF(N.LE.6)GOTO 16
   17 WRITE(5,18)
   18 FORMAT(10X,'PPL TEM SOLUCAO ILIMITADA')
      RETURN
   15 F=A(N,1)/A(N,M)
      J0=N
C
C
C
      IF(N.EQ.6)GOTO 301
      DO 19 K0=N,6
      IF(A(K0,M).LE.0)GOTO 19
      G=A(K0,1)/A(K0,M)
      IF(F-G)19,195,21
   21 F=G
  195 J0=K0
   19 CONTINUE
C
C
 9008 FORMAT(/'  O VALOR DE J0 E  ',10X,I4)
C
      WRITE(5,9008)J0
      WRITE(5,22)J0,M,A(J0,M),IX(J0-1)
   22 FORMAT(10X,'A(',I1,',',I1,')=',1X,F8.3,10X,'E O PIVOT',5X,'SAI DA
     *BASE X',I2//)
  301 RETURN
      END
```

```
      SUBROUTINE PIVOTT(A,J0,M,IX,PIVOT)
      DIMENSION A(6,10),IX(5)
C     PIVOTEAMENTO DO SIMPLEX
C
C
C
      DO 24 K2=1,6
      IF(K2.EQ.J0)GOTO 24
      DO 23 J1=1,10
      IF(J1.EQ.M)GOTO 23
      A(K2,J1)=A(K2,J1)-((A(J0,J1)/PIVOT)*A(K2,M))
   23 CONTINUE
   24 CONTINUE
      DO 1000 K2=1,6
      IF(K2.EQ.J0)GOTO 1000
      A(K2,M)=0.
 1000 CONTINUE
      DO 26 J1=1,10
   26 A(J0,J1)=A(J0,J1)/PIVOT
      L=J0-1
      IX(L)=M-1
      RETURN
      END
```

```
      SUBROUTINE PVDUAL(A,NO,MAO,JO,M)
      DIMENSION A(6,10)
C     ESTA SUBROUTINE PROCURA O PIVOT PELO DUAL DO SIMPLEX
C
C
      R=ABS(A(1,MAO)/A(NO,MAO))
      M1=MAO
      DO 400 LAN=MAO,10
      IF(A(NO,LAN).GE.0)GOTO 400
C
C
      S=ABS(A(1,LAN)/A(NO,LAN))
      IF(R.LE.S)GOTO 400
      R=S
      M1=LAN
  400 CONTINUE
      WRITE(5,68)R,NO,M1,A(NO,M1)
C
C
   68 FORMAT(5X,F8.3,3X,'A(',I2,',',I2,')=',3X,F8.3,2X,'E O PIVOT'//)
      JO=NO
      M=M1
      RETURN
      END
```

```
      SUBROUTINE INVB(IX,MIL,IYY,M)
      DIMENSION IX(5)
      DO 208 M3=1,5
      IF(IX(M3).NE.1)GOTO 208
      JK=M3+1
   79 IF(JK.GT.5)GOTO 73
   72 IF(IX(JK).EQ.2)GOTO 160
      JK=JK+1
      GOTO 79
C
C
C
  160 WRITE(5,161)M3,IX(M3),JK,IX(JK)
      IYY=0
  161 FORMAT(10X,'IX(',I1,')=',I1,5X,'IX(',I2,')=',I2/)
      RETURN
   73 M=3
      MIL=1
      WRITE(5,163)M
  163 FORMAT(10X,'INTRODUZ X2',4X,'M=',1X,I1/)
      IYY=1
      RETURN
C
C
C
  208 CONTINUE
      M=2
      MIL=1
      WRITE(5,164)M
  164 FORMAT(10X,'INTRODUZ X1',4X,'M=',1X,I1/)
      IYY=1
      RETURN
      END
```

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
NÚCLEO DE COMPUTAÇÃO ELETRÔNICA

```
      SUBROUTINE PENAL (P,B1,K40,K50,IX,IZ,A,J)
      DIMENSION IX(5),A(6,10),B1(9),P(9)
      DO 201 LJ=1,9
      P(LJ)=0.
  201 B1(LJ)=0.
      DO 1200 M3=1,5
      IF(M3.EQ.K40)GOTO 1200
      IF(M3.EQ.K50)GOTO 1200
      LJ=6
  857 IF(IX(M3).EQ.LJ)GOTO 856
C
C
C
      LJ=LJ+1
      IF(LJ.GE.9)GOTO 1200
      GOTO 857
  856 B1(LJ)=A(1,LJ+1)
      JMW=IZ+1
  861 IF(A(M3+1,JMW).LT.0)GOTO 859
  858 JMW=JMW+1
      IF(JMW.EQ.6)GOTO 858
      IF(JMW.NE.11)GOTO 861
      P(LJ)=0.
      GOTO 1200
C
C
C
  859 G1=A(1,JMW)/A(M3+1,JMW)
      DO 1300 K20=JMW,10
      IF(K20.EQ.6)GOTO 1300
      IF(A(M3+1,K20).GE.0)GOTO 1300
      G2=A(1,K20)/A(M3+1,K20)
      IF(G1.GE.G2)GOTO 1300
      G1=G2
 1300 CONTINUE
      P(LJ)=(A(M3+1,J)-1.)*G1
C
C
C
 1200 CONTINUE
      DO 2030 LJ=6,9
 2030 B1(LJ)=A(1,LJ+1)
      WRITE(5,2016)(B1(LJ),LJ=1,10)
 2016 FORMAT(//1X,' O VETOR DAS PENALIDADES VERDDR E'//1X,10(3X,F5.2))
      WRITE(5,2015)(P(LJ),LJ=1,10)
C
C
 2015 FORMAT(//1X,' O VETOR DAS PENALIDADES FALSAS E'//1X,10(3X,F5.2))
      RETURN
      END
```

```
      SUBROUTINE ESTR(B1,P,P1,IAM1,IAL1)
      DIMENSION B1(9),P(9),P1(2,2)
C
C
C     ESCOLHA DAS ESTRATEGIAS
C
C
      JKL=5
      JKM=7
      DO 1400 MAL=1,2
      DO 1400 LAN=1,2
      YS11=JKL+MAL
      YS12=JKM+LAN
      A1=B1(YS11)+B1(YS12)
      IF(P(YS11).GT.P(YS12))GOTO 867
      A2=P(YS12)
      GOTO 868
  867 A2=P(YS11)
C
C
C
C
  868 IF(A1.GT.A2)GOTO 870
      P1(MAL,LAN)=A2
      GOTO 1400
  870 P1(MAL,LAN)=A1
 1400 CONTINUE
      WRITE(5,1410)((P1(I,J),J=1,2),I=1,2)
 1410 FORMAT(3(/),2(5X,F8.3))
      AM=P1(1,1)
      DO 1500 MAL=1,2
      DO 1500 LAN=1,2
      IF(p1(MAL,LAN).GT.AM)GOTO 1500
      AM=P1(MAL,LAN)
C
C
C
C
      IAM=MAL
      IAL=LAN
 1500 CONTINUE
      IAM1=IAM-1
      IAL1=IAL-1
      RETURN
      END
```

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
NÚCLEO DE COMPUTAÇÃO ELETRÔNICA

```
C
C
C
      INTEGER YS11,YS12
      DIMENSION A(6,10),P1(2,2),B1(10),P(10),B(6,6),IX(5),DELTA(6),
     *CB(5)
      KA=0
      MIL=0
      J0=1
      IRR=2
      IYY=0
      IAM1=0
      IAL1=0
      READ(8,1)((A(I,J),J=1,10),I=1,6)
    1 FORMAT(10F5.2)
      READ(8,2)(IX(L),L=1,5)
    2 FORMAT(5I2)
      READ(8,5001)(CB(K12),K12=1,5)
 5001 FORMAT(5F5.2)
      WRITE(5,3)
    3 FORMAT(10X,'A MATRIZ INICIAL E'///)
      WRITE(5,4)((A(I,J),J=1,10),I=1,6)
C
C
C
    4 FORMAT(10(3X,F8.3))
      WRITE(5,6)(IX(L),L=1,5)
    6 FORMAT(10X,'AS VARIAVEIS BASICAS SAO'//5(8X,I2))
      DO 8000 J=1,10
 8000 A(1,J)=A(1,J)-(A(2,J)*A(1,6))
      WRITE(5,8001)
 8001 FORMAT(10X,'A NOVA MATRIZ A E'///)
   50 WRITE(5,4)((A(I,J),J=1,10),I=1,6)
      WRITE(5,6)(IX(L),L=1,5)
      IF(MIL-1)801,140,801
  801 K=2
      M=0
      CALL MENLZ(A,K,M)
      IF(K.GT.10)GOTO 80
   81 N=2
C
C
C     PROCURA PIVOT DO SIMPLEX
C
      CALL PVSMP(A,N,M,J0,IX)
      IF(N.GT.6)GOTO 187
   30 PIVOT=A(J0,M)
C
C
C     CHAMADA DA SUBROUTINE QUE FAZ O PIVOTEAMENTO PELO SIMPLEX
      CALL PIVOTT(A,J0,M,IX,PIVOT)
C
      GOTO 50
C
C
```

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
NÚCLEO DE COMPUTAÇÃO ELETRÔNICA

```
C
   80 L=2
      N0=0
      MAO=0
      CALL MENCZ(A,IX,L,KA,MIL,IRR,N0,MAO)
      IF(IRR)56,7007,187
 7007 IF(KA.EQ.1)GOTO 100
      IF(MIL.EQ.1)GOTO 100
      GOTO 140
C
C
C
C     DETERMINACAO DO PIVOT PELO DUAL DO SIMPLEX
C
C
   56 CALL PVDUAL(A,N0,MAO,J0,M)
      GOTO 30
C
C
C
C     INTRODUCAO DE VARIAVEL BINARIA
C
C
C
  140 CALL INVB(IX,MIL,IYY,5)
      IF(IYY.EQ.0)GOTO 801
      GOTO 81
  100 IZ=1
      IW=IZ
      J=IW
      MILA=0
      DO 5000 M3=1,5
C
C
C
      IF(IX(M3).NE.IW)GOTO 841
      K40=M3
      GOTO 847
  841 IF(IX(M3).NE.(IW+1))GOTO 5000
      K50=M3
  847 IF(A(M3+1,J).GE.999.E-3)GOTO 845
      IF(ABS(A(M3+1,J)).GE.1.E-3)GOTO 5000
C
C
C
C
  845 WRITE(5,849)M3,IX(M3)
  849 FORMAT(///5X,'ALGORITMO SATISFEITO PARA IX(',I2,')=',I4//)
      MILA=MILA+IX(M3)
      IF(MILA.EQ.3)GOTO 187
 5000 CONTINUE
C
C
C     CONSTRUCAO DA TABELA DE PENALIDADES PELA SUBROUTINE PENAL
C
      CALL PENAL(P,B1,K40,K50,IX,IZ,A,J)
      CALL ESTR(B1,P,P1,IAM1,IAL1)
      DO 1600 JOT=1,6
 1600 DELTA(JOT)=0
      DO 1800 JOT=3,5,2
      DELTA(JOT)=1-IAM1
      DELTA(JOT+1)=IAM1
C
C
```


UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
NÚCLEO DE COMPUTAÇÃO ELETRÔNICA

```
C
      IAM1=IAL1
 1800 CONTINUE
      WRITE(5,2011)(DELTA(JOT),JOT=1,6)
 2011 FORMAT(//1X,6(5X,F5.2))
      DO 1900 I=1,6
      DO 1900 J=1,6
 1900 B(I,J)=0.
C
C
C     FORMACAO DA MATRIZ B MENOSUM
C
C
      DO 2000 I=2,6
      DO 2006 J=2,6
 2006 B(I,J)=A(I,J+4)
 2000 CONTINUE
      WRITE(5,1901)((B(I,J),J=1,6),I=1,6)
 1901 FORMAT(10X,'A MATRIZ B MENOSUM E'///6(8X,F8.3)//)
C
C     NOVA MATRIZ B MENOSUM
C
      B(1,1)=1.
      DO 2010 J=2,6
      S=0.
      DO 2001 I=2,6
C
C
C
      S=S+CB(I-1)*B(I,J)
 2001 CONTINUE
 2010 B(1,J)=S
      WRITE(5,2002)((B(I,J),J=1,6),I=1,6)
 2002 FORMAT(10X,'ANOVA MATRIZ B MENOSUM E'//6(8X,F8.3)//)
      DO 2003 I=1,6
      S=0.
      DO 2004 J=1,6
      S=S+B(I,J)*(-DELTA(J))
 2004 CONTINUE
C
C
      A(I,1)=S+A(I,1)
 2003 CONTINUE
      WRITE(5,2005)((A(I,J),J=1,10),I=1,6)
 2005 FORMAT(10X,'A MATRIZ APOS A INTRODUCAO DO VETOR DELTA E'///10(3X,F
     *8.3)///)
      KA=1
      IRR=-1
      GOTO 80
  187 STOP
      END
```