UNIVERSIDADE FEDERAL DE SANTA CATARINA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

UMA METODOLOGIA PARA SOLUÇÃO DE PROBLEMAS DE PROGRAMAÇÃO POR OBJETIVOS COM VARIÁVEIS INTEIRAS

DISSERTAÇÃO SUBMETIDA À UNIVERSIDADE FEDERAL DE SANTA CATARINA PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA

VÂNIA CONCEIÇÃO TAVARES

FLORIANÓPOLIS

SANTA CATARINA - BRASIL

SETEMBRO - 1980

UMA METODOLOGIA PARA SOLUÇÃO DE PROBLEMAS DE PROGRAMAÇÃO POR OBJETIVOS COM VARIÁVEIS INTEIRAS

VÂNIA CONCEIÇÃO TAVARES

ESTA DISSERTAÇÃO FOI JULGADA PARA A OBTENÇÃO DO TÍTULO DE "MESTRE EM ENGENHARIA" ESPECIALIDADE ENGENHARIA DE PRODUÇÃO E APROVADA EM SUA FORMA FINAL PELO CURSO DE PÓS-GRADUAÇÃO.

PROF. JOHN ROBERT MACKNESS
COORDENADOR

APRESENTADA PERANTE A BANCA EXAMINADORA COMPOSTA DOS PROFESSORES:

PROF. LEONARDO ENSSLIN, Ph.D.
PRESIDENTE

PROF. ROBERTO FRANCISCO KRISCHER, M.Sc.
CO-ORIENTADOR

PROF. RAUL VALENTIM DA SILVA, M.Sc.

PROF. FREDERICO AGENOR ALVAREZ, M.Sc.

A você.

"Se vi mais longe, é porque
subi nos ombros de gigantes."

Isaac Newton

A eles, meus agradecimentos.

R E S U M O

O presente trabalho tem por propósito desenvolver e analisar um procedimento que permita determinar as soluções de problemas de Programação por Objetivos nos casos particulares de variáveis inteiras, permitindo ao decisor melhor compreender as repercussões de variações tanto a nível de objetivos como de parâmetros.

Inicialmente é apresentada a técnica de Programação por Objetivos como um recurso usado para auxiliar o processo de tomada de decisão.

Posteriormente o método de "branch and bound", usado em Programação Linear Inteira, é associado ao de Programação por Objetivos, e esta metodologia é analisada sob dois aspectos: um deles sem atribuir prioridades às novas restrições geradas e o outro atribuindo-se a maior prioridade a estas restrições.

Da análise efetuada conclui-se que para resolução de Problemas de Programação por Objetivos pode ser utilizada a técnica de "branch and bound", com atribuição de maior prioridade às restrições geradas pela própria técnica.

Neste trabalho é apresentada ainda uma ilustração da metodologia desenvolvida, bem como o manual de utilização e a listagem do programa computacional utilizado na resolução do modelo matemático.

ABSTRACT

In this dissertation, a procedure is developed for solving Goal Programming problems with integer variables; this will help decision takers in their understanding of repercussions following changes in the value of the variables.

Initially the Goal Programming technique as an aid to decision taking is described; then the "branch and bound" method used in Linear Programming is combined with Goal Programming and the resulting procedure is analysed with and without the definition of priorities to the new restrictions which are created.

This analysis shows that the "branch and bound" techniques can be used together with Goal Programming with the highest priority assigned to the restrictions which the technique creates.

A pratical illustration of the methodology is presented and a manual is provided for the use of the procedure and the associated computer programme.

S U M Á R I O

Pag.

"Tive mestres eminentes.

Alegrei-me com os meus progressos
e triunfos.

Hoje, evocando o sábio que era,
comparo-me naquele tempo à água,
que toma a forma do vaso e à
fumaça que o vento dissipa."

Omar Kháyyám

C A P Í T U L O   I

1. INTRODUÇÃO

1.1. Histórico

Raciocinar, resolver problemas e tomar decisões são atividades que caracterizam o comportamento humano. A tomada de decisão tem sido alvo de atenções cada vez maiores dos pesquisadores e dos profissionais em exercício. De uma maneira cada vez mais acentuada existe a necessidade do homem usar seus recursos de uma forma adequada, visando atingir a maior parte dos seus objetivos; para tanto, é preciso adotar uma maneira racional na escolha das ações a serem tomadas.

Até o aparecimento de técnicas de decisão mais avançadas, baseadas em sólido ferramental matemático, as decisões eram tomadas com base em procedimentos padronizados por meio de normas ou metas secundárias ou, na falta destas, por experiência, hábito, intuição ou impressão.

Modernamente, entretanto, a Pesquisa Operacional, a Simulação e o Processamento de Dados, bem como Técnicas Heurísticas vieram dar uma abordagem mais científica ao processo decisório.

Uma das técnicas desenvolvidas na Pesquisa Operacional para tomada de decisão, aplicada a problemas de otimização, é a Programação por Objetivos. Esta técnica permite resolver problemas que envolvem um conjunto de metas simultâneas a serem atingidas, não apenas segundo uma ordem de prioridades pré-definida em termos do modelo matemático, mas também em função de análises com a interveniência do decisor.

1.2. O Estudo de Programação por Objetivos com Variáveis Inteiras

1.2.1. Objetivo

O presente trabalho tem por propósito desenvolver e analisar o procedimento que permita determinar as soluções de problemas de Programação por Objetivos nos casos particulares em que estas soluções admitam somente valores inteiros, visando inseri-lo em uma metodologia global de análise decisória.

1.2.2. Importância

Certos tipos de problemas de otimização apresentam uma restrição adicional, segundo a qual uma ou mais variáveis devem assumir valores inteiros. Nestes casos a solução fracionária não é aceita, seja porque não faz sentido, seja porque contraria frontalmente o significado físico da variável. Desta forma, o trabalho desenvolvido pode trazer uma contribuição para os processos decisórios que apresentam a referida característica.

1.2.3. Limitações

A metodologia desenvolvida no presente trabalho, permite determinar as soluções de problemas de Programação por Objetivos com variáveis inteiras, para os casos em que tanto a função objetivo como as restrições são lineares, e as variáveis são determinísticas. Para o caso específico de variáveis inteiras do tipo zero-um, esta metodologia pode ser utilizada, mas pode não ser a maneira mais eficaz, existindo modelos mais apropriados usando heurísticas e técnicas de enumeração.

Quanto à utilização do programa computacional desenvolvido, existem limitações decorrentes das características do equipamento disponível, em termos de dimensionamento; assim, é definido um número máximo permitido para o número de restrições, variáveis e prioridades dos problemas a serem resolvidos.

1.2.4. Metodologia

A pesquisa para o presente trabalho foi desenvolvida basicamente em duas etapas:

1. Estudo detalhado da técnica de Programação por Objetivos, bem como do programa computacional desenvolvido por Sang M. Lee[1].
2. Análise das técnicas de Programação Linear Inteira resultando na escolha do método de "branch and bound" para utilização em Programação por Objetivos com variáveis inteiras.

---

[1]LEE, Sang M. Goal Programming for Decision Analysis, 1972, p.140-157.

1.3. Estrutura do Trabalho

O capítulo 2 apresenta a técnica de Programação por Objetivos como um recurso usado para auxiliar o processo de tomada de decisão. É mostrada a formulação do modelo, bem como a resolução do mesmo através do algoritmo simplex. Uma abordagem do programa computacional referente à técnica citada faz parte do mesmo capítulo.

No capítulo 3 uma das técnicas da Programação Linear Inteira, o método de "branch and bound", é utilizado para resolver problemas de Programação por Objetivos com variáveis inteiras. Um conjunto de soluções é obtido da associação dos dois métodos atribuindo-se maior prioridade à restrição gerada ao aplicar a técnica "branch and bound".

Estas soluções representam várias alternativas que permitem ao decisor analisar as repercussões provenientes de alterações de parâmetros ou das prioridades definidas anteriormente, com a finalidade de fixá-las efetivamente ou não. Desta forma, a metodologia caracterizada neste capítulo pode ser descrita como sendo:

. resolução do problema de P.P.O. com variáveis inteiras utilizando o método "branch and bound".

. geração de soluções alternativas para o decisor.

Um problema que ilustra a metodologia desenvolvida no capítulo 3 é apresentado no capítulo 4.

O capítulo 5 contém as conclusões gerais do trabalho desenvolvido e sugestões para pesquisas futuras.

A listagem do programa computacional, bem como o manual de utilização do mesmo são apresentados em forma de anexos.

C A P Í T U L O   I I

2. RETROSPECTO

2.1. A Programação por Objetivos - P.P.O.

O conceito de Programação por Objetivos foi desenvolvido a partir de pesquisas em Programação Linear, por A. Charnes e W.W. Cooper, e estudado detalhadamente por Y. Yjiri, que as considerou como técnicas distintas de programação matemática.

A P.P.O. caracteriza-se pela existência de vários objetivos (metas) a serem atingidos, através da minimização da soma dos desvios referentes àqueles objetivos que não forem totalmente alcançados. Nesta estrutura, o administrador deve ter em mente o problema global, de forma a caracterizar qual dos objetivos deve ser alcançado em primeiro lugar, atribuindo para o desvio correspondente uma maior prioridade na minimização.

Uma vez determinada uma hierarquia para os objetivos, o algoritmo de P.P.O. procura encontrar um conjunto de soluções que satisfaça a restrição com o maior objetivo. Somente após alcançado o primeiro objetivo o algoritmo busca encontrar um sub-conjunto deste conjunto de soluções que satisfaça também a meta com segunda prioridade. E assim por diante até que todos os seus objetivos sejam alcançados, total ou parcialmente.

Assim, o método de Programação por Objetivos pode ser aplicado na resolução de problema com objetivos conflitantes.

2.1.1. Formulação do modelo de Programação por Objetivos

O modelo geral de Programação por Objetivos é da forma:

$$\text{Minimizar } Z = CD$$

$$\text{sujeito a } \quad AX + RD = B$$

$$X,\ D \geqslant 0$$

Onde:

C : vetor das prioridades e os "pesos" correspondentes

D : vetor dos desvios positivos e/ou negativos

A : matriz dos coeficientes das variáveis nas metas a serem atingidas

X : vetor das variáveis a serem determinadas

B : vetor das constantes

R : matriz de elementos unitários (positivos ou negativos) e/ou nulos.

Este modelo matemático procura minimizar tanto quanto possível os desvios relativos às metas não alcançadas. A preocupação maior é a de atingir o objetivo de maior prioridade e passar sucessivamente para o de nível imediatamente inferior, a partir de uma hierarquia pré-definida. O decisor não precisa conhecer nem dominar a técnica apresentada. Basta que o técnico em Pesquisa Operacional apresente as soluções com sua ordem de prioridades arbitrariamente definida, para que o decisor possa analisar e retificar as prioridades.

Na formulação de um problema de Programação por Objetivos devem ser considerados os seguintes passos:

1. Definição das variáveis e constantes.

Devem ser analisadas todas as variáveis relevantes do problema e as constantes do lado direito das restrições, de acordo com os recursos existentes e o efeito que estas variáveis produzem no conjunto de metas do administrador.

2. Formulação das restrições.

As restrições são formuladas em função das disponibilidades existentes, traduzindo-se como um conjunto de desigualdades e equações que relacionam as variáveis do problema e que caracterizam também as metas ou objetivos a serem atingidos.

3. Definição da função objetivo.

Esta é definida a partir da estrutura hierárquica de objetivos do administrador. Primeiro devem ser designados os níveis de prioridade relativos a estes objetivos. Se vários deles tiverem o mesmo nível de prioridade, diferentes "pesos" podem ser atribuídos para diferenciar sua importância relativa.

Caracteriza-se assim que a solução do problema está basicamente dependendo da experiência do administrador, no sentido de que, se a sua hierarquia de prioridades não está de acordo com os objetivos da organização, a solução ótima obtida pode não ser ótima no sentido global da empresa.

Este fato que pode a primeira vista parecer apresentar limitações à aplicação da técnica de P.P.O. traz, na verdade, implicitamente uma forma do decisor definir em melhores condições a ordem das prioridades e de conhecer melhor o problema como um todo. Isto porque, basta variar a ordem das prioridades para se obter uma solução que após analisada, venha de encontro aos objetivos globais procurados pelo decisor.

2.1.2. Solução do modelo de Programação por Objetivos

Uma das técnicas de solução de problemas de Programação por Objetivos é a da Inversa Generalizada, também desenvolvida por Yjiri[2], porém não muito utilizada devido à pouca literatura existente. Um método simplex, análogo ao utilizado na Programação Linear é mais usado na resolução dos problemas, tendo sido objeto de maiores estudos, visando o desenvolvimento de programas de computador para determinar a solução procurada.

O método simplex é um algoritmo iterativo, de tal forma que a solução ótima do problema é alcançada após iterações sucessivas. Teoricamente qualquer problema de programação matemática linear pode ser resolvido através deste método. Na prática, todos os problemas devem ser resolvidos com o auxílio do computador. O procedimento do método simplex para Programação por Objetivos pode ser sintetizado na figura 1.

2.2. O Programa Computacional

2.2.1. Fluxograma geral

Devido à complexidade da maioria dos problemas práticos de Programação por Objetivos, um programa computacio

---

[2] IJIRI, Y. Management Goals and Accounting for Control. Chicago, 1965.

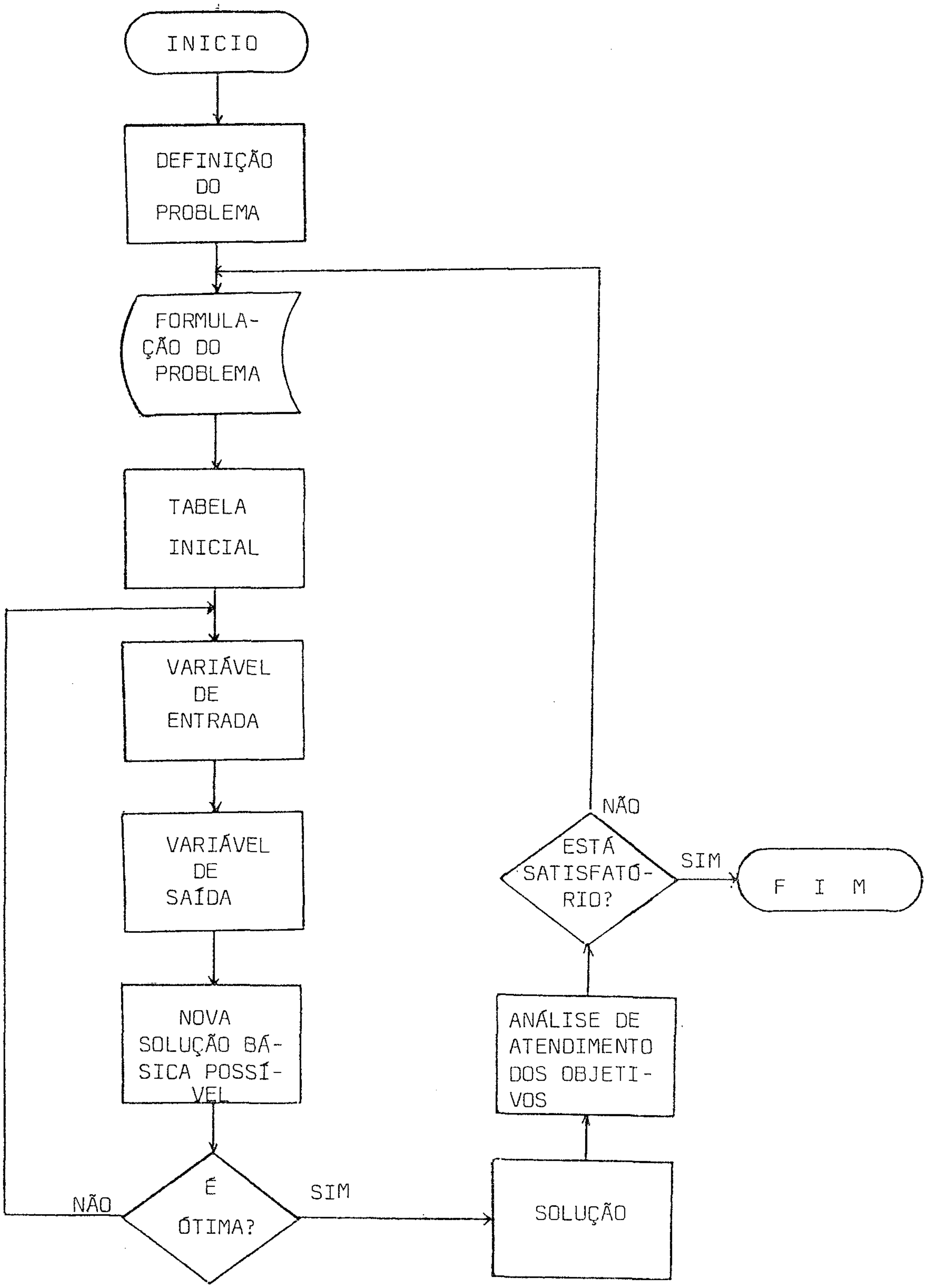

FIGURA 1 - Diagrama de Blocos para Solução de Problemas de P.P.O.

nal foi desenvolvido por Sang M. Lee[3] baseado na solução do método simplex. O diagrama de blocos referente a tal programa é sintetizado na figura 2.

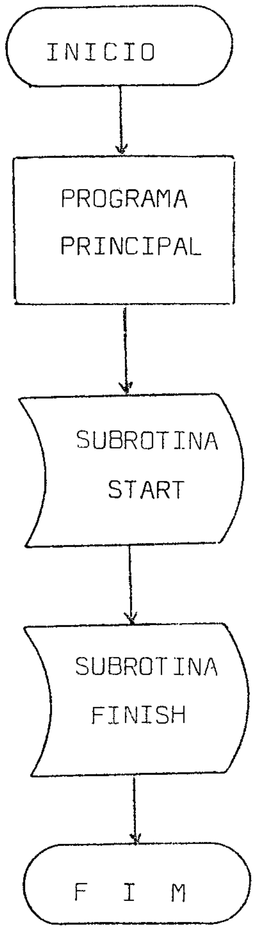

FIGURA 2 - Diagrama de Blocos Geral do Programa de P.P.O.

SUBROTINA START: Efetua a leitura dos cartões de dados e mediante uma série de testes faz a crítica destes valores, emitindo mensagens de erros quando necessário. Gera as matrizes para o procedimento simplex, imprimindo os dados iniciais através destas matrizes.

---

[3]LEE, Sang M. Goal Programming For Decision Analysis, 1972, pg. 140-157.

PROGRAMA PRINCIPAL: Executa as iterações do método simplex propriamente dito, efetuando as operações pivotais. Imprime todas as iterações ou apenas a última, que resulta na solução ótima.

SUBROTINA FINISH: Calcula os valores das variáveis de desvio para cada um dos objetivos, e indica o valor final da função objetivo em termos de cada uma das prioridades.

2.2.2. Procedimento Operacional

SUBROTINA START

Esta subrotina tem a finalidade de ler os dados do problema e prepará-los convenientemente para a utilização do método simplex, além de efetuar uma exaustiva "crítica" destes dados para evitar erros de digitação.

As metas a serem alcançadas, na formulação do problema, podem ser descritas como desigualdades matemáticas, havendo quatro possibilidades:

E : "exatamente igual" - Não existe desvio positivo ou negativo para esta meta.

L : "menor do que" - Existe somente o desvio negativo.

G : "maior do que" - Existe somente o desvio positivo.

B : "maior ou menor" - Pode existir desvio positivo ou negativo.

De acordo com os dados de entrada, a subrotina START gera as matrizes iniciais que serão utilizadas no procedimento simplex. As variáveis de desvio são criadas automaticamente, além de uma prioridade maior (gerada artificialmente) ser atribuída aos desvios das metas dos tipos E e G. Nesta subrotina são impressas todas as informações de entrada, em forma de sumário, e as matrizes geradas.

PROGRAMA PRINCIPAL

Executa as operações pivotais do método simplex propriamente dito. Algumas considerações devem ser feitas em relação ao método simplex, ao utilizá-lo na Programação por Objetivos:

1. A finalidade da função objetivo de P.P.O. é minimizar o total de objetivos não atingidos, segundo níveis de prioridades $P_1, P_2, \ldots, P_n$. Isto é obtido através da minimização da soma das variáveis de desvio associadas às prioridades e pesos designados pelo administrador.

2. Como os níveis de prioridades atribuídos a cada variável de desvio na função objetivo podem ser valores ordinais, eles não são mensuráveis. Assim sendo, os critérios de decisão do simplex, z_j ou $z_j - c_j$, não podem ser expressos através de uma única linha. Neste caso, o critério simplex é constituído de uma matriz de ordem mxn, onde m representa o número de prioridades e n o número total de variáveis do problema (variáveis reais mais variáveis de desvio).

3. Uma vez que $z_j - c_j$ é uma matriz, novo procedimento deve ser considerado para identificar a coluna pivô. As

sumindo-se que $P_j >>> P_{j+1}$, isto é, P_j tem prioridade absoluta sobre P_{j+1}, estes elementos devem ser considerados na escolha da coluna ótima. Assim, a variável que apresentar uma taxa de contribuição maior para a obtenção do objetivo de mais alta prioridade deverá ser escolhida para entrar na base. Ou seja, a coluna (k2) que apresentar o maior valor positivo na linha correspondente à maior prioridade ainda não atendida será a coluna pivô, determinando a variável X_{k2} que entra na base.

A linha pivô (k1) será aquela que apresentar o menor valor positivo para $c_j/a_{j,k2}$, determinando a variável X_{k1} que sai da base.

As operações pivotais do procedimento simplex são efetuadas em função do elemento pivô $a_{k1,k2}$.

O processo é repetido até que o objetivo de maior prioridade seja alcançado, passando-se para os demais objetivos e adotando-se a mesma técnica usada anteriormente.

A solução do problema é ótima quando o valor de $z_j - c_j$ para cada nível de prioridade for nulo. Se a solução apresentar um valor positivo para uma das prioridades, verifica-se se algum nível mais alto possui coeficiente negativo para aquela variável na matriz $z_j - c_j$. Se existir, a solução obtida é a "melhor solução possível para o problema". Caso contrário, a solução não é ótima e retoma-se o processo.

Após a obtenção da solução final do problema, esta deve ser analisada em função dos objetivos globais da organização. Esta análise pode implicar numa reformulação do modelo, através do reestudo da hierarquia dos objetivos definida pelo administrador.

SUBROTINA FINISH

De acordo com a solução obtida para o problema, a subrotina FINISH calcula os desvios para cada um dos objetivos e determina o vetor que caracteriza o valor da função objetivo em termos de cada uma das prioridades.

2.2.3. Análise do programa

Um estudo exaustivo e detalhado foi efetuado em relação ao programa descrito anteriormente no item 2.2., após ter sido verificado que apenas problemas particulares podiam ser resolvidos através de sua utilização.

Assim, incorporações foram efetuadas no programa original, tornando-o eficiente para a resolução de problemas generalizados, mediante um minucioso processo de depuração e testes. Para uma melhor visualização da solução do problema, além de possibilitar a descoberta de erros não detectáveis pelo programa, os relatórios de saída foram redefinidos facilitando a interpretação dos resultados.

C A P Í T U L O   I I I

3. METODOLOGIA DESENVOLVIDA

3.1. Analogia entre Programação Linear Inteira e Programação por Objetivos com Variáveis Inteiras

A Programação Linear é uma das técnicas desenvolvidas na Pesquisa Operacional para auxiliar na tomada de decisão, aplicável a uma grande variedade de problemas de otimização.

Sua versatilidade, aliada à possibilidade do uso de recursos computacionais, fizeram com que um grande número de pesquisas fossem desenvolvidas nesta área, resultando em novas técnicas e novas aplicações, ampliando-se ainda mais o campo da Pesquisa Operacional.

Assim sendo, os problemas de Programação Linear cuja solução necessitasse ser inteira foram pesquisados exaustivamente, resultando em grande número de técnicas específicas para resolver tais problemas.

Ainda baseado na Programação Linear, foi desenvolvido o método de Programação por Objetivos, cuja solução é obtida pelo método simplex, conforme visto no capítulo anterior. No entanto, técnicas para resolução de problemas de Programação por Objetivos com variáveis inteiras ainda não foram objeto de muitas pesquisas, havendo pouca literatura nesta área.

O procedimento adotado para resolver tais problemas, desenvolvido neste trabalho, foi o de incorporar um dos métodos da Programação Linear Inteira, o "branch and bound"[4], na Programação por Objetivos.

3.1.1. O método de "branch and bound" para Programação Linear Inteira

Dado um problema de Programação Linear Inteira, o método de "branch and bound" desenvolve uma arborescência, cuja raiz é a solução do problema inicial, retirando-lhe as condições de integralidade. Se esta solução é inteira, tem-se a solução inteira ótima. Caso contrário, resolve-se dois outros problemas de programação linear, cada um deles composto do problema original e uma restrição adicional do tipo

$$X_j \leqslant \left[X_j^*\right] \qquad e \qquad X_j \geqslant \left[X_j^*\right] + 1 \quad ,$$

onde:

X_j^* = valor de X_j na solução ótima não inteira

$\left[X_j^*\right]$ = maior inteiro menor ou igual a X_j^*.

Se as soluções encontradas para os novos problemas forem inteiras, aquela que apresentar o melhor valor para a função objetivo é considerada a solução inteira ótima procurada. Se ainda existir solução fracionária, verifica-se a conveniência de repetir o procedimento, criando-se novos problemas para serem resolvidos.

---

[4]MACULAN FILHO, Nelson. Programação Linear Inteira. Rio de Janeiro, 1978, p.2.9, 4.1-4.17.

Nem sempre, na aplicação do "branch and bound", é necessário o desenvolvimento de todas as ramificações. Se a função objetivo de uma solução fracionária apresentar um valor pior em relação às soluções inteiras já encontradas, as bifurcações originárias dela não apresentarão soluções melhores, cessando a arborescência.

3.2. Aplicação do Método de "branch and bound" em Programação por Objetivos

Um procedimento análogo ao considerado na seção anterior pode ser adotado para a resolução de problemas de Programação por Objetivos com variáveis inteiras. O problema inicialmente é resolvido sem considerar as condições de integralidade. Caso a solução seja fracionária, uma das variáveis X_j é escolhida para dar origem a dois novos problemas, incorporando a cada um deles uma das restrições.

$$X_j \geqslant \left[X_j^*\right] + 1 \qquad \text{ou} \qquad X_j \leqslant \left[X_j^*\right]$$

onde:

X_j^* = valor fracionário da variável na solução ótima

$\left[X_j^*\right]$ = maior inteiro menor ou igual a X_j^*.

A solução destes novos problemas poderá ser inteira ou não, mas a função objetivo apresentará um valor pior ou no máximo igual à solução do problema que gerou a arborescência.

Este procedimento pode ser repetido até que todas as soluções obtidas sejam inteiras; isto porém pode resultar num grande número de ramificações, dificultando a obtenção da melhor solução nos casos de problemas mais complexos, que envolvam muitas variáveis.

No caso de Programação por Objetivos, como o valor da função objetivo é dado em função das prioridades e estas não são mensuráveis, não é possível identificar "numericamente" a melhor solução. No entanto, a solução que atender o objetivo de maior prioridade pode ser considerada "melhor" do que aquela que não o satisfaz; se ambas não atenderem este objetivo, será considerada "melhor" a solução que apresentar o menor desvio correspondente àquela meta. Desta forma pode-se limitar o número de ramificações no desenvolvimento da arborescência.

Uma interferência do decisor também pode restringir a bifurcação para uma determinada solução, uma vez que ele pode analisar se uma solução é viável ou não, em termos dos objetivos globais da empresa.

Na aplicação do método de "branch and bound" em Programação por Objetivos inicialmente foram incorporadas as novas restrições ao problema inicial, mas nenhuma prioridade foi atribuída, ou seja, as variáveis de desvio correspondentes àquelas restrições não figuraram na função objetivo. Nesta alternativa, o número de ramificações do problema pode tornar-se muito grande uma vez que uma variável, após assumir valores inteiros, pode voltar a ser fracionária em outra solução, gerando nova bifurcação, até que todas as soluções sejam inteiras. Novo procedimento foi adotado para evitar tais situações, qual seja, o de atribuir prioridades às novas restrições geradas.

3.3. Incorporações

O método de resolução de um problema de Programação por Objetivos tenta encontrar uma solução que minimize tanto quanto possível a soma dos desvios relativos aos objetivos não alcançados. Se esta solução buscada deve ser uma solução inteira, utiliza-se a técnica de "branch and bound", incorporando-se novas restrições ao problema original e gerando-se dois outros novos problemas de programação por objetivos para resolver.

Estas novas restrições significam novos objetivos a serem alcançados, logo, é necessário atribuir-lhes uma prioridade.

Assim sendo, no modelo matemático de P.P.O. foi incorporado a atribuição de prioridades maiores para as restrições geradas pelo método de "branch and bound".

Inicialmente, para cada bifurcação criou-se mais um nível de prioridade, maior do que as já existentes; este procedimento mostrou-se deficiente dado que, mesmo para exemplos pequenos surgiram problemas de dimensionamento para utilização do programa computacional.

Assim, a cada nova restrição incorporada ao problema original foi atribuída a mesma prioridade, sendo esta maior do que as já existentes.

Com esta metodologia, um problema de Programação por Objetivos com variáveis inteiras pode ser solucionado, obtendo-se todas as soluções inteiras talvez com menor número de bi furcações do que pela forma inicialmente apresentada no final do item 3.2.

No exemplo analisado, todas as bifurcações foram efetuadas, até que nenhuma solução fracionária fosse obtida. Não adotou-se um critério de "poda" para limitar as ramificações.

3.4. Procedimento Operacional da Metodologia desenvolvida

Dado um problema de Programação por Objetivos com variáveis inteiras, este pode ser formulado como:

$$
\begin{aligned}
\text{Minimizar} \quad & Z = CD \\
\text{sujeito a} \quad & AX + RD = B \\
& X,\ D \geqslant 0 \\
& X,\ D \text{ inteiros}
\end{aligned}
$$

onde:

C : vetor das prioridades e os "pesos" correspondentes

D : vetor dos desvios positivos e/ou negativos

A : matriz dos coeficientes das variáveis nas metas a serem atingidas

X : vetor das variáveis a serem determinadas

B : vetor das constantes

R : matriz de elementos unitários (positivos ou negativos) e/ou nulos.

A solução deste problema é obtida através da metodologia mostrada na figura 3, utilizando o programa computacional cuja listagem e manual de utilização constam nos anexos.

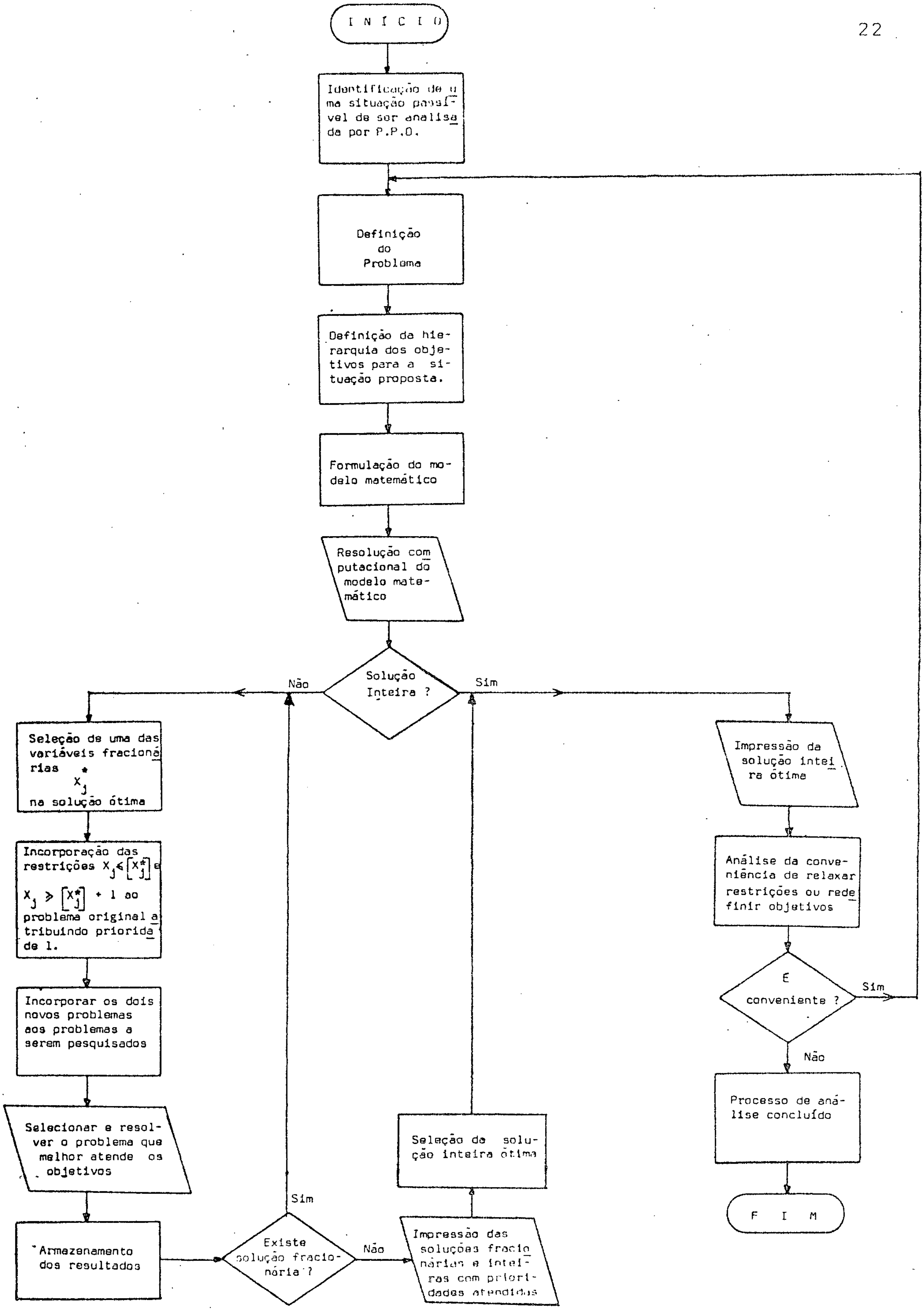

FIGURA 3 - Metodologia para análise e subsídio decisório para hierarquização de objetivos de problemas de P.P.O.

Ao incorporar as novas restrições ao problema, é necessário redefinir as prioridades, atribuindo a maior delas P_1 para estas novas restrições e "deslocando" as demais para um nível imediatamente inferior.

Após a obtenção do conjunto de soluções inteiras, o decisor tem em mãos os dados para proceder a uma análise destes resultados. Nesta análise pode ser estudada a viabilidade de alterar valores das disponibilidades (recursos b_i) para melhor atendimento dos objetivos, bem como redefinir a hierarquia das prioridades atribuídas a estes mesmos objetivos, verificando as repercussões econômicas de tal procedimento.

Este processo de realimentação permite ao decisor melhor conhecer o problema que está sendo analisado, ao mesmo tempo que lhe possibilita fundamentar sua seleção dos objetivos, que de outra forma dificilmente seria tão completa.

C A P Í T U L O I V

4. ILUSTRAÇÃO

4.1. Problema Proposto

Visando exemplificar a aplicação de Programação por Objetivos com variáveis inteiras, é estudado a produção mensal da linha de fabricação de uma indústria.

Esta fábrica apresenta três produtos diferentes (designados como 1, 2 e 3) ao mercado consumidor, e seu departamento de mercadologia informa que toda produção mensal é adquirida não existindo produtos em estoques.

A administração da fábrica definiu os seguintes objetivos para serem atingidos:

1. atender a demanda de mercado de uma certa região específica.

2. evitar ociosidade na linha de produção.

3. atender a um nível de vendas mínimo para cada tipo de produto.

4. limitar em 20 horas o número de horas extras na linha de produção.

5. minimizar o número de horas extras na linha de produção.

O atendimento destes objetivos depende das disponibilidades existentes, traduzindo-se como restrições relativas a:

- carga horária da linha de produção
- número de horas requeridas para produção de cada tipo de produto
- previsão de vendas
- atendimento da demanda específica de uma região.

O conjunto destas restrições devidamente equacionadas, pode evidenciar a solução requerida para atender os objetivos definidos. A solução obtida através do método simplex para programação por objetivos pode apresentar, no entanto, a melhor solução possível para este problema. A utilização deste método porém, requer a definição da escala decrescente de prioridades relativa aos objetivos. A administração da empresa apontou a seguinte ordenação:

1ª prioridade: evitar ociosidade na linha de produção

2ª prioridade: atender a demanda específica de uma região

3ª prioridade: não ultrapassar de 20 horas o número de horas extras da linha de produção

4ª prioridade: atender um mínimo de vendas definido para cada tipo de produto

5^{a} prioridade: minimizar o número de horas extras da linha de produção.

Assim, o problema descrito foi equacionado da seguinte maneira:

Equação

$$
\begin{array}{lll}
01) & 5X1 + 8X2 + 12X3 + d_1^- - d_1^+ & = 170 \\
02) & X1 + d_2^- - d_2^+ & = 5 \\
03) & X2 + d_3^- - d_3^+ & = 5 \\
04) & X3 + d_4^- - d_4^+ & = 8 \\
05) & X1 + d_5^- - d_5^+ & = 10 \\
06) & X2 + d_6^- - d_6^+ & = 12 \\
07) & X3 + d_7^- - d_7^+ & = 10 \\
08) & 5X1 + 8X2 + 12X3 + d_8^- - d_8^+ & = 190 \\
09) & X1, X2, X3, d_1^-, d_2^-, \ldots, d_8^-, d_1^+, d_2^+, \ldots, d_8^+ \geqslant 0 \text{ e inteiras} &
\end{array}
$$

Função objetivo:

$$
\text{Minimizar } Z = P_1 d_1^- + 20P_2 d_2^- + 18P_2 d_3^- + 21P_2 d_4^- + P_3 d_8^+ + 20P_4 d_5^- + 18P_4 d_6^- + 21P_4 d_7^- + \\ + P_5 d_1^+ .
$$

Com as seguintes interpretações:

. a equação 01 representa as horas de trabalho necessárias na linha de produção para fabricar X1, X2 e X3 unidades dos produtos 1, 2 e 3 da empresa. Sendo a capacidade de trabalho normal definida em 170 horas mensais, d_1^- representa ociosidade na linha de produção, e d_1^+ representa o número de horas extras de trabalho necessárias para aquela produção. Os coeficientes de cada variável representam o número de horas necessárias para produzir uma unidade do produto considerado.

. as equações 02, 03 e 04 são relativas ao atendimento mensal específico para uma certa região, sendo definido como no mínimo 5, 5 e 8 unidades dos produtos 1, 2 e 3 respectivamente.

. as equações 05, 06 e 07 representam as quantidades mínimas dos produtos 1, 2 e 3, estipuladas em 10, 12 e 10 unidades respectivamente, a serem produzidas mensalmente para venda.

. a equação 08 está relacionada com a equação 1, e limita em no máximo 20 horas o número de horas extras de trabalho na linha de produção. Assim, a capacidade normal de trabalho (170 horas) pode ser expandido para um valor máximo de 190 horas mensais.

. finalmente, a equação 09 indica que todas as variáveis do problema são valores não negativos, além de só poderem assumir valores inteiros. Isto significa que os produtos 1, 2 e 3 só podem ser fabricados em quantidades inteiras. Esta restrição não é considerada para a obtenção da solução inicial pelo método simplex da Programação por Objetivos.

A função objetivo corresponde à minimização dos desvios das metas definidas pela administração, de acordo com a hierarquia fixada.

Os coeficientes numéricos (pesos) relativos às prioridades $P_1, P_2, \ldots, P_5$ são proporcionais ao lucro líquido unitário para cada tipo de produto, obtidos no setor de contabilidade da empresa.

4.2. Solução do Problema Proposto

4.2.1. Solução inicial

O programa computacional listado no anexo 2 foi utilizado para fornecer a solução do problema formulado, indicando a seguinte solução:

VARIÁVEL	VALOR
X1	10
X2	5,5
X3	8

DESVIO	NEGATIVO	POSITIVO
01	0	20
02	0	5
03	0	0,5
04	0	0
05	0	0
06	6,5	0
07	2	0
08	0	0

ANÁLISE DA FUNÇÃO OBJETIVO

PRIORIDADE	ATENDIMENTO	VALOR
01	total	0
02	total	0
03	total	0
04	parcial	159
05	parcial	20

VALOR PARA A FUNÇÃO OBJETIVO EM FUNÇÃO DOS DESVIOS E PRIORIDADES:

$$Z = P_1 d_1^- + P_2(20d_2^- + 18d_3^- + 21d_4^-) + P_3 d_8^+ + P_4(20d_5^- + 18d_6^- + 21d_7^-) + P_5 d_1^+$$

$$Z = 0P_1 + 0P_2 + 0P_3 + (18 \times 6,5 + 21 \times 2)P_4 + 20P_5 = 159P_4 + 20P_5$$

Esta solução significa que, para atender os objetivos definidos pela diretoria da indústria, é necessário a produção mensal de 10 unidades do produto 1, 5,5 do produto 2 e 8 unidades do produto 3. A análise dos resultados para a função objetivo, no entanto, indica que não houve atendimento total em todas as prioridades. Assim, pode-se concluir que:

a) o atendimento total da prioridade 01 indica que nesta solução não existe ociosidade da carga horária da linha de produção.

b) o atendimento total da prioridade 02 indica que é possível atender a demanda necessária para a região específica (equações 02, 03 e 04).

c) o objetivo definido com prioridades 03 também foi atendido, ou seja, o número de horas extras da linha de produção não ultrapassa a 20 horas.

d) o atendimento parcial da prioridade 04 indica que não foi possível produzir o número esperado para venda de produtos 2 e 3 (equações 06 e 07), ocorrendo um desvio negativo de 6,5 e 2 respectivamente.

e) não foi possível evitar as horas extras na linha de produção, o que demonstra o não atendimento da prioridade 05.

Finalmente, o valor fracionário para a produção do produto 2 implica na procura de outra solução, que atenda a condição de integralidade do problema apresentado.

4.2.2. Geração de soluções inteiras

4.2.2.1. Método de "branch and bound"

De acordo com a metodologia desenvolvida, pode ser obtida uma solução inteira para o problema proposto aplicando-se o método de "branch and bound".

Dada a solução fracionária obtida para o problema: $X1 = 10$

$X2 = 5,5$

$X3 = 8$

e $Z = 0P_1 + 0P_2 + 0P_3 + 150P_4 + 20P_5$

a variável $X2 = 5,5$ foi escolhida para gerar as restrições do tipo:

$$X2 \leqslant [5,5]$$

$$\text{e} \quad X2 \geqslant [5,5] + 1$$

ou seja:

$$X2 \leqslant 5 \qquad \text{e} \qquad X2 \geqslant 6,$$

que foram incorporadas ao problema original, resultando em dois novos problemas, cada um deles agora composto das 9 equações iniciais e uma destas inequações devidamente transformada em equação.

A solução destes novos problemas, bem como o desenvolvimento de todas as ramificações das soluções fracionárias obtidas a partir desta bifurcação é apresentada na figura 4.

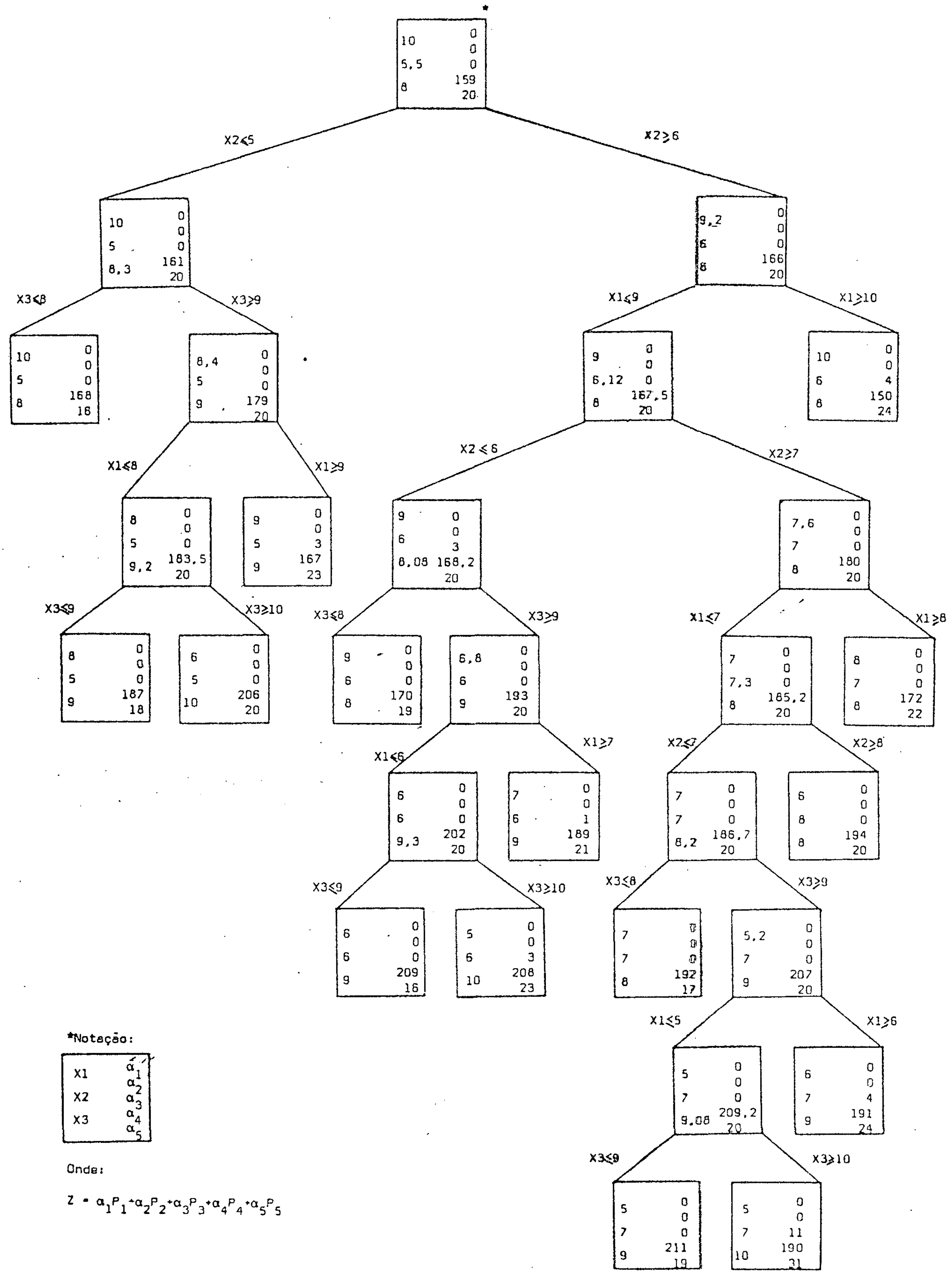

FIGURA 4 - Arborescência das soluções do problema proposto.

Para a obtenção de todas as soluções inteiras (em número de 15) foi necessária uma arborescência composta de 29 problemas.

O conjunto de todas as soluções inteiras encontradas pode ser expresso na tabela 1.

TABELA 1 - Conjunto de soluções inteiras sem atribuir nova prioridade

TABELA 1							
VARIÁVEIS			FUNÇÃO OBJETIVO				
X1	X2	X3	P_1	P_2	P_3	P_4	P_5
10	5	8	0	0	0	168	16
9	6	8	0	0	0	170	19
8	5	9	0	0	0	187	18
7	7	8	0	0	0	192	17
6	8	8	0	0	0	194	20
6	5	10	0	0	0	206	20
6	6	9	0	0	0	209	16
5	7	9	0	0	0	211	19
7	6	9	0	0	1	189	21
8	7	8	0	0	2	172	22
9	5	9	0	0	3	167	23
5	6	10	0	0	3	208	23
10	6	8	0	0	4	150	24
6	7	9	0	0	4	191	24
5	7	10	0	0	11	190	31

Uma análise da tabela 1 indica que a solução apresentada na primeira linha pode ser considerada a "melhor" de las, uma vez que, dentre as que atendem integralmente as priorida des 01, 02 e 03, apresenta o menor valor para a prioridade 04, que é atendida parcialmente, bem como para a prioridade 05.

Assim, temos para o problema formulado a seguinte solução inteira:

$$X1 = 10$$

$$X2 = 5$$

$$X3 = 8$$

$$\text{e} \quad Z = 0P_1 + 0P_2 + 0P_3 + 168P_4 + 16P_5.$$

4.2.2.2. <u>Método de "branch and bound" com atribuição de novas prioridades</u>

Após a obtenção da arborescência para determinar todas as soluções inteiras do problema abordado, o mesmo foi resolvido pelo método de "branch and bound" atribuindo-se a priori dade maior (P_1) para as novas restrições geradas. Assim, as priori dades já existentes ficaram "deslocadas de uma unidade" em relação ao que foi inicialmente definido pelo administrador. Seria análo go a atribuir a prioridade zero (P_0) às novas restrições.

Com esta metodologia as soluções estão apresen tadas na figura 5.

O campo final de soluções continua o mesmo, pe lo qual não se exclui soluções inteiras. No entanto, dadas as características de multiprioridades do problema, poderão ocorrer soluções distintas. Porém, as soluções mais recomendáveis em termos do atendimento das prioridades estarão todas contidas em ambas as arborescências.

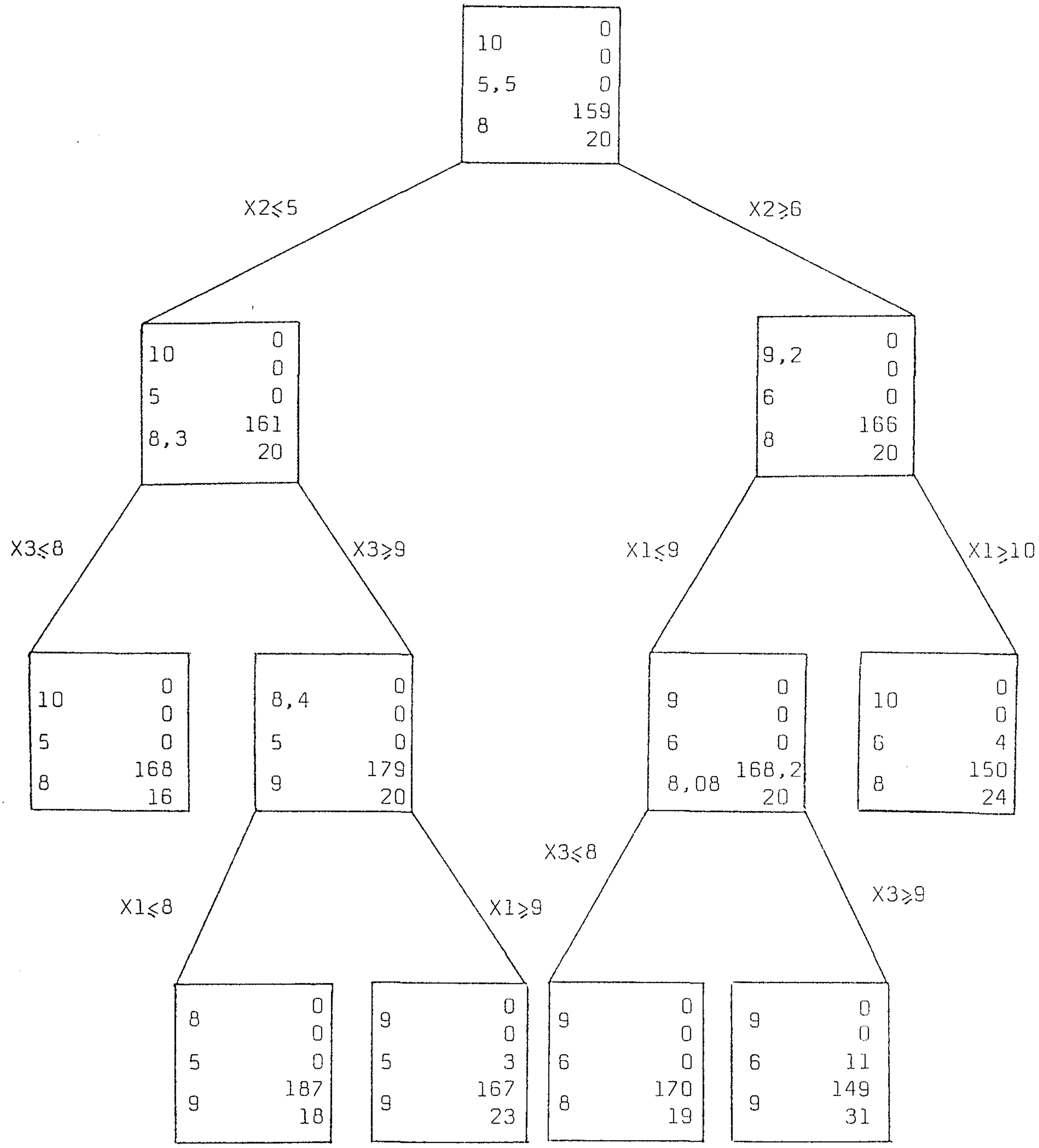

FIGURA 5 - Árvore de soluções do problema proposto atribuindo-se prioridades às novas restrições.

No que se refere à redução do número de soluções alternativas, para o exemplo proposto, isto ocorreu, mas não necessariamente isto será um fato generalizado.

As soluções inteiras obtidas estão expressas na tabela 2.

TABELA 2 - Conjunto de soluções inteiras atribuindo-se novas prioridades.

T A B E L A 2							
VARIÁVEIS			FUNÇÃO OBJETIVO				
X1	X2	X3	P_1	P_2	P_3	P_4	P_5
10	5	8	0	0	0	168	16
9	6	8	0	0	0	170	19
8	5	9	0	0	0	187	18
9	5	9	0	0	3	167	23
10	6	8	0	0	4	150	24
9	6	9	0	0	11	149	31

Em função do atendimento dos objetivos fixados, pode-se concluir que a "melhor" solução inteira para o problema dado é

X1 = 10

X2 = 5

X3 = 8

atendendo integralmente as prioridades 01, 02 e 03 e parcialmente a 04 e 05.

4.3. Análise dos Métodos

De acordo com o que foi descrito anteriormente, a solução do problema de Programação por Objetivos com variáveis inteiras, apresentado como ilustração, resultou fracionária, quando resolvido pelo algoritmo computacional.

Ao aplicar-se o método "branch and bound" sem atribuir prioridade às restrições incorporadas, obteve-se quinze soluções inteiras. Atribuindo-se a prioridade 01 às novas restrições obteve-se seis soluções inteiras. A análise das soluções obtidas, em ambos os casos, resultou numa mesma solução inteira considerada "melhor", sob o ponto de vista do atendimento das prioridades definidas pelo decisor.

Assim, a atribuição de prioridade maior às restrições incorporadas pelo método de "branch and bound" pode reduzir o total de ramificações, o que justifica em princípio sua aplicação.

Considera-se pouco conveniente adotar um critério de "poda" para evitar certas ramificações. No entanto, para problemas complexos, o próprio decisor pode evitar as bifurcações que considerar irrelevantes.

O conjunto das soluções inteiras obtidas representa as várias alternativas que permitem ao decisor escolher a "melhor solução" para o problema apresentado, em função da hierarquia de prioridades por ele definida.

Uma análise destas soluções pode também, levá-lo a efetuar variações nos parâmetros ou níveis de prioridades, avaliando as repercussões provenientes destas alterações em termos dos objetivos globais da empresa. Assim, o administrador pode ter vários conjuntos de soluções, cada um deles em função da hierarquia definida para os seus objetivos, auxiliando-o sobremaneira no processo decisório global.

Pode-se concluir que para o problema proposto como ilustração, a aplicação da metodologia desenvolvida apresentou como "melhor solução" a produção mensal de 10 unidades do produto 1, 5 unidades do produto 2 e 8 unidades do produto 3. Com estes valores, verificou-se ser possível atender integralmente os objetivos de prioridade 01, 02 e 03, e parcialmente os objetivos de prioridade 04 e 05.

Neste exemplo não foram efetuadas alterações nos parâmetros ou níveis de prioridades definidos inicialmente, considerando-se a resposta obtida satisfatória em termos dos objetivos globais da empresa. Porém o decisor pode, por exemplo, chegar à conclusão que é conveniente aumentar o limite de horas extras de trabalho na linha de produção, para melhor atender seus objetivos de venda de cada produto, ou mesmo redefinir suas prioridades, em função dos resultados obtidos.

4.4. Potencialidades Gerenciais da Utilização de Programação por Objetivos

O método de Programação por Objetivos, quando usado na resolução de problemas de otimização, mostra ser de grande valia principalmente nos casos em que os objetivos a alcançar são conflitantes.

Sua maior potencialidade, no entanto, está no fato de fornecer subsídios ao decisor na definição efetiva da melhor hierarquia para os objetivos.

Inicialmente esta hierarquia é definida, e o decisor recebe um conjunto de respostas que é função desta hierarquia. Uma análise dos resultados obtidos pode então ser efetuada, permitindo ao decisor avaliar as repercussões econômicas provenientes de variações nos parâmetros ou na hierarquia dos objetivos anteriormente definidos. Esta análise pode levá-lo então a alterar a formulação inicial, gerando um novo problema que apresentará um novo conjunto de soluções, que novamente será submetido a análise. Desta forma, a cada nova formulação, o decisor obtém um conjunto de soluções que é função da nova hierarquia definida. Ao final, o problema a ser resolvido terá como solução a que melhor se apresentar dentro do conjunto geral de soluções obtidas nas várias formulações, caracterizando também a melhor hierarquia para os objetivos definidos, auxiliando sobremaneira o decisor na efetiva fixação desta hierarquia em seu processo decisório.

CAPÍTULO V

5. CONCLUSÃO

5.1. Generalidades

A Programação por Objetivos pode ser aplicada a inúmeras áreas de decisão gerencial e administrativa, como um recurso para resolver problemas de otimização que envolvem múltiplos objetivos a serem alcançados.

Sua grande flexibilidade consiste no fato de levar em consideração interesses conflitantes existentes no problema, o que impediria a determinação da solução por técnicas conhecidas da Programação Linear, bem como a possibilidade de procurar atender objetivos "não financeiros", dentro de uma política de ação "não lucrativa".

A solução de um problema de Programação por Objetivos é obtida pelo método simplex, análogo ao da Programação Linear, tendo sido desenvolvido um algoritmo computacional para a determinação da solução ótima.

Outras pesquisas relativas à técnica de Programação por Objetivos, visando uma ampliação do conhecimento e utilização desta técnica, têm sido sugeridas por alguns autores recentemente, pouca bibliografia existindo no entanto sobre o assunto.

No presente trabalho foi apresentada uma metodologia para a determinação da solução inteira ótima de modelos de otimização utilizando-se a técnica de "branch and bound" associada à Programação por Objetivos. A atribuição de maior prioridade às restrições geradas pelo "branch and bound", para a ilustração, evidenciou sua importância em função da exigência de valores inteiros para as variáveis do problema.

A conveniência de se adotar um critério de "poda" para as ramificações da árvore de soluções do problema fica a cargo do decisor, de acordo com o grau de complexidade apresentado e as características próprias do problema em estudo.

Também a escolha da variável que vai dar origem às novas restrições não obedece a nenhum critério específico, no modelo apresentado.

Na aplicação do modelo proposto, o administrador possui recursos para determinar a melhor hierarquia dos objetos a serem alcançados, partindo de uma definição inicial e analisando os resultados obtidos correspondentes a esta definição. Desta análise, e em função dos objetivos globais da organização, o decisor pode alterar a formulação inicial do modelo, criando um novo problema que apresentará novo conjunto de soluções. Assim, ele terá vários destes conjuntos, correspondentes às várias formulações do problema existente, que o auxiliará no processo decisório para a escolha da solução ótima procurada. A formulação correspondente à solução escolhida caracteriza a melhor estrutura hierárquica para os objetivos a serem alcançados.

5.2. Importância do Trabalho

Durante o estudo desenvolvido ficou evidenciada a importância da utilização de Programação por Objetivos na resolução de problemas de otimização, não só em termos de apresentar a solução ótima do modelo formulado, mas principalmente em termos de suas potencialidades gerenciais. Na formulação inicial o decisor define as prioridades para os objetivos a atingir, porém, em função dos resultados obtidos esta definição pode ser efetivamente confirmada ou modificada. O administrador pode analisar as repercussões econômicas provenientes da relaxação de restrições ou disponibilidades da empresa.

A ênfase dada aos problemas de Programação por Objetivos com variáveis inteiras é proveniente da constatação da existência de muitas áreas que apresentam estas características, analogamente ao que ficou evidenciado nos estudos de Programação Linear Inteira.

5.3. Sugestões para Pesquisas Futuras

Em termos do algoritmo computacional apresentado para a resolução dos problemas de Programação Por Objetivos através do método simplex, recomenda-se introduzir no modelo formas de amenizar erros de arredondamento, bem como tentar obter uma melhor utilização dos recursos computacionais.

Considerando os problemas com variáveis inteiras, pesquisas podem ser desenvolvidas especificamente para as do tipo zero-um, com a incorporação de técnicas de enumeração, heurísticas ou outros métodos convenientes para estes problemas específicos.

Finalmente, dada a grande importância da aplicação da técnica de Programação por Objetivos, outras pesquisas podem ser desenvolvidas nesta área, tais como:

. Resolução de problemas de Programação por Ojetivos através do método da Inversa Generalizada.

. Análise de sensibilidade.

. Programação por Objetivos paramétrica.

. Programação por Objetivos sob condições de incerteza.

BIBLIOGRAFIA

1. ALVAREZ, Frederico Agenor. Um Algoritmo de Programação Linear Inteira Zero-Um Utilizando a Técnica Lexicográfica. Dissertação de Mestrado. UFSC. Santa Catarina, 1979.

2. CHARNES, A. & COOPER, W.W. Goal programming and multiple objective optimizations. European Journal of Operational Research. (1977) : 39-54.

3. EHRLICH, Pierre Jacques. Pesquisa Operacional - Curso Introdutório. Editora Atlas S.A., 1976.

4. GAL, Tomas. & NEDOMA, Josef. Multiparametric linear programming. Management Science, Vol. 18, nº 7, março 1972 - 406-422.

5. GARFINKEL, Robert S.; NEMHAUSER, George L.; Integer Programming, A Wiley-Intersciense Publication, 1972.

6. HADLEY, G. Nonlinear and Dynamic Programming. Addison-Wesley Publishing Company, 1964.

7. HU, T.C. Integer Programming and Network Flows. Addison-Wesley Publishing Company, 1970.

8. IGNIZIO, James P. An Approach to the Capital Budgeting Problem with Multiple Objectives. The Engineering Economist, Vol. 21, nº 4 : 259-272.

9. ISERMANN, Heinz. The enumeration of the set all efficient solutions for a linear multiple objectives program. Operational Research Quartely, Vol. 28, nº 3ii, 1977 : 711-725.

10. KRISCHER, Roberto Francisco. Desenvolvimento de uma Técnica de Otimização de Sistemas Lineares Estocásticos com Múltiplos Objetivos. Dissertação de Mestrado. UFSC. Santa Catarina, 1976.

11. KUESTER, James L.; MIZE, Joe H. Optimization Techniques with Fortran. McGraw-Hill Book Company, 1973.

12. LEÃO, Manoel Luiz; LEONEL, Neron Arruda. Curso de Programação Linear. Rio Grande do Sul, 1979.

13. LEE, Mac S. Goal programming for decisions analysis. Auerbach Publishers, U.S.A., 1972.

14. MACULAN Fº, Nelson. Programação Linear Inteira. Comissão de Publicações COPPE/RJ, 1978.

15. MORRIS, Richard L. Current Developments in Goal Programming. Clemson University Review of Industrial Management and Textile Science, Vol. 17, nº 1, 1978 : 47-53.

ANEXO 1

MANUAL DE UTILIZAÇÃO DO PROGRAMA

MANUAL DE UTILIZAÇÃO DO PROGRAMA

1. Apresentação do Problema

O problema a ser resolvido pode estar na forma:

$$\text{Min } Z = CD$$

$$\text{Sujeito a } AX \gtreqless B$$

$$X,\ D \geqslant 0$$

Onde:

C : vetor das prioridades

D : vetor dos desvios positivos e/ou negativos

A : matriz dos coeficientes das variáveis nas metas a serem atingidas

X : vetor das variáveis a serem determinadas

B : vetor das constantes

2. Limitações das Variáveis

O programa formulado está limitado quanto ao seu dimensionamento, devido às características da máquina em que foi processado. Assim, para o computador IBM 360/40 temos as seguintes dimensões:

NÚMERO MÁXIMO DE PRIORIDADES	:	10
NÚMERO MÁXIMO DE RESTRIÇÕES (LINHAS)	:	70
NÚMERO MÁXIMO DE VARIÁVEIS (COLUNAS)	:	150

Dependendo da capacidade de memória do computador a ser utilizado, este dimensionamento pode ser reformulado através da substituição dos cartões "DIMENSION".

3. Preparação dos Dados

3.1. Cartão inicial

O programa permite o processamento de até nove problemas a cada execução, com o "DECK" de cartões de entrada conforme a figura 6.

O primeiro cartão a ser introduzido é o que contém o número de problemas a serem processados (perfurado na coluna 1 do cartão) e o código de impressão (perfurado na coluna 2 do cartão). Este código será o dígito 1 (um) se todas as iterações forem impressas e deverá ter o valor zero se apenas se desejar a impressão da solução final.

3.2. Cartões de cada bloco

Cada bloco de cartões que se segue ao cartão inicial do "DECK" corresponde a um problema a ser resolvido e é constituído dos cartões indicados na figura 7.

Descrição de cada cartão:

CARTÃO 1 - Descreve os parâmetros do problema considerado.

Col. 1 - 4 : perfurar PROB

Col. 5 - 7 : número de linhas

Col. 8 - 10 : número de colunas

Col. 11 - 13 : número de prioridades

O número de linhas refere-se ao número de restrições (objetivos a alcançar) do problema.

O número de colunas indica a quantidade de variáveis reais do problema, sem incluir as variáveis de folga (desvios) e as variáveis artificiais (criadas por necessidade do simplex). Tais variáveis são geradas no próprio programa.

O número de prioridades é o total de níveis (Pi) considerados na função objetiva. O programa cria prioridades automaticamente, quando necessário.

CARTÃO 2 - Descreve o tipo de cada restrição. Existem quatro possibilidades:

E : para "exatamente igual" - não existe desvio positivo ou negativo.

L : para "menor do que" - existe somente o desvio negativo.

G : para "maior do que" - existe somente o desvio positivo.

B : quando ambas as direções são possíveis, existindo desvio positivo e negativo.

A perfuração das letras correspondentes aos tipos de restrições deve ser feita em campos consecutivos, iniciando na coluna 1 do cartão.

CARTÃO 3 - Indica cada parcela da função objetiva.

É composto de um cartão com a perfuração OBJ nas três primeiras colunas, seguido de tantos outros cartões quantos forem necessários para definir cada parcela da função objetiva. Estes são do tipo:

Col. 1 - 3 : perfurar POS ou NEG dependendo do sinal do desvio a ser minimizado.

Col. 4 - 7 : brancos.

Col. 8 - 9 : número da linha que contém o desvio considerado.

Col. 10 - 12 : brancos

Col. 13 - 14 : nível de prioridade atribuída a este desvio (valor de i correspondente a Pi).

Col. 15 - 25 : coeficiente do nível de prioridade (diferentes pesos α_i). Deverá ser perfurado mesmo quando seu valor é 1.0(um). (Formato real).

CARTÃO 4 - Especifica os coeficientes das variáveis do problema nas restrições.

É composto de um cartão com a palavra "DATA" perfurado nas quatro primeiras colunas e seguido dos cartões contendo as informações:

Col. 1 - 7 : brancos.

Col. 8 - 9 : número de linha em que o coeficiente está alocado.

Col. 10 - 12 : brancos.

Col. 13 - 14 : número da coluna na qual aparece o coeficiente.

Col. 15 - 25 : valor do coeficiente na posição indicada (formato real).

CARTÃO 5 - Indica os valores das constantes do lado direito das restrições - (formato real).

É composto de um cartão com a palavra RGHT perfurada nas quatro primeiras colunas, seguido de cartões contendo as informações:

Col. 1 - 10 : valor da constante da primeira linha.

Col. 11 - 20 : valor da constante da segunda linha.

⋮

Col. 71 - 80 : valor da constante da oitava linha.

Caso necessário, usa-se mais cartões do mesmo tipo até informar as constantes de todas as linhas (restrições).

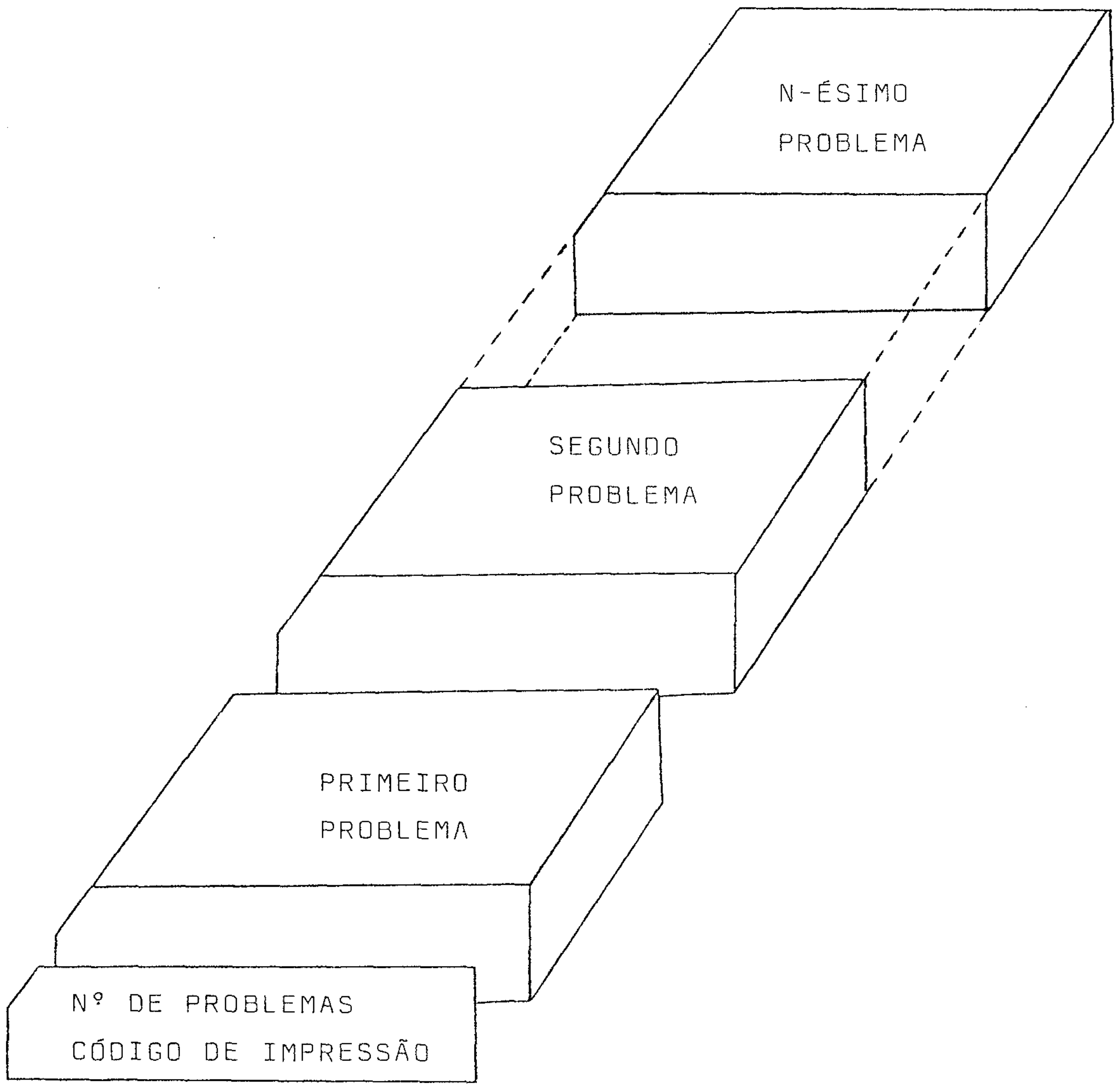

FIGURA 6 - "DECK" de cartões de entrada

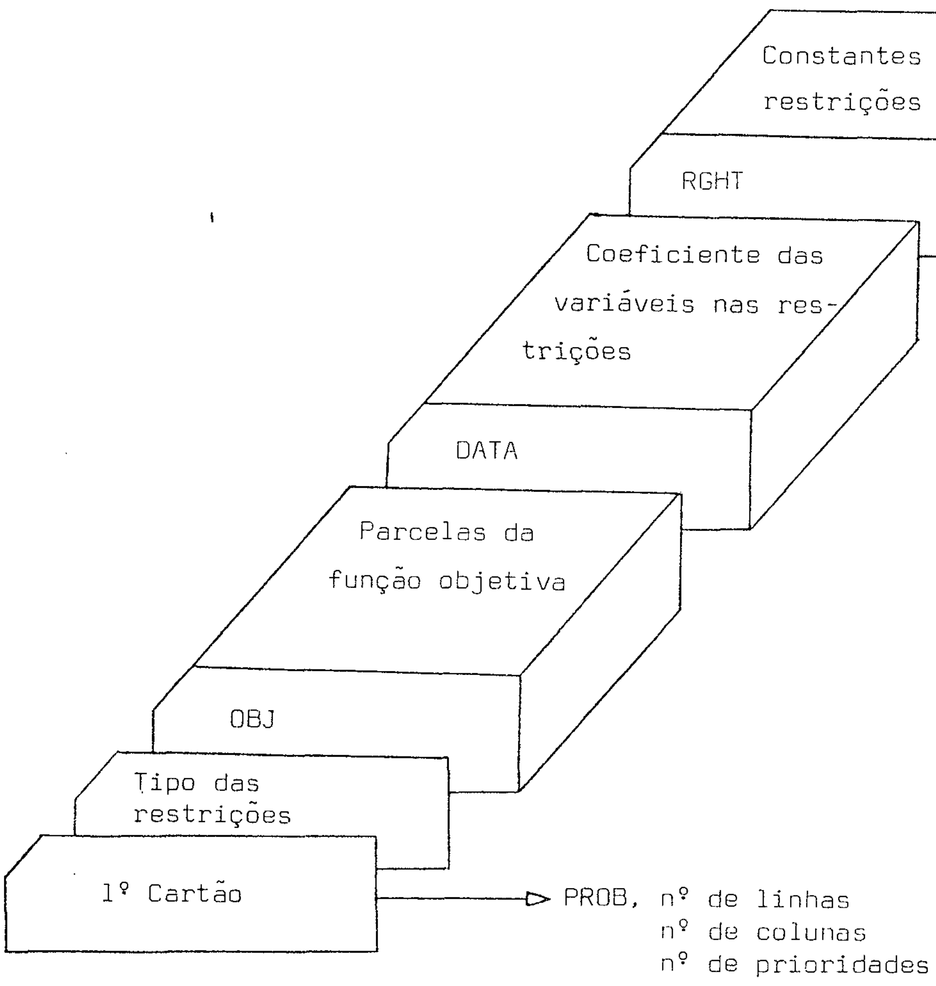

FIGURA 7 - "DECK" de cartões em cada bloco

ANEXO 2

LISTAGEM DO PROGRAMA

```
C
C
C
C
C
C                      PROGRAMA PARA RESOLVER PROBLEMAS
C                      DE PROGRAMACAO POR OBJETIVOS-PPO
C
C
C
C
C
C
C
C
C*****************************************************************
      DIMENSION KEPT(70)
      DIMENSION RHS1(70)
      DIMENSION VALY(70,10)
      DIMENSION Y(70)
      DIMENSION PRDT(70)
      DIMENSION AMT(70)
      DIMENSION ZVAL(10)
      DIMENSION C(70,150)
      DIMENSION DOD(70)
      DIMENSION DUD(150)
      DIMENSION VALX(10,150)
      DIMENSION X(150)
      DIMENSION RVLX(10,150)
      DIMENSION D(70,150)
C
C                      PROGRAMACAO POR OBJETIVOS
C
      READ(1,2)NPROB,ITAB
    2 FORMAT(2I1)
      DO 830 IREP=1,NPROB
      CALL START(N,M,L,C,VALX,VALY,PRDT,RHS1,KPCK,KEPT,TEST)
      DO 21 J=1,M
   21 X(J)=J
      DO 20 I=1,N
   20 Y(I)=I
   15 FORMAT(10X,I5,F20.2,/)
   12 FORMAT(/,5X,10F11.2)
   13 FORMAT(10X,F9.0,F9.2,/)
      DO 25 K=1,L
      DO 25 I=1,N
      VALY(I,K) = VALX(K,I)
   25 CONTINUE
C
C          CALCULO DE NOVAS VARIAVEIS
C
      ITER=0
C
C          CALCULO  DOS  VALORES  DA MATRIZ   ZJ- CJ   (RVLX(K,J))
C
   31 L1=0
   32 K3=L-L1
   33 IF(K3-1) 800,40,40
   40 DO 60 K=1,K3
      DO 60 J=1,M
      SUMP=0.
      DO 50 I=1,N
```

```
      P=VALY(I,K)*C(I,J)
      SUMP= SUMP+P
   50 CONTINUE
      RVLX(K,J)=SUMP- VALX(K,J)
   60 CONTINUE
C
      IF(ITER)62,62,54
 62   WRITE(3,3)
    3 FORMAT(///,2X,'---MATRIZ ZJ-CJ INICIAL ',/)
      DO 312 I=1,L
  312 WRITE(3,12)(RVLX(I,J),J=1,M)
      GOTO 5
 54   IF(ITAB)5,5,795
  795 WRITE(3,5003)
      DO 796 K=1,L
  796 WRITE(3,12) (RVLX(K,J),J=1,M)
C
C          DETERMINACAO  DA  COLUNA  PIVO  - K2  (VARIAVEL QUE
C          ENTRA NA BASE
C
    5 ZMAX=0
      DO 90 J=1,M
      IF(K3-L) 92,70,70
   92 K4=K3+1
      DO 91 K=K4,L
      IF(RVLX(K,J)) 90,91,91
   91 CONTINUE
   70 IF(RVLX(K3,J)-ZMAX) 90,90,80
   80 ZMAX=RVLX(K3,J)
      K2=J
   90 CONTINUE
   95 IF(ZMAX)790,790,100
C
C
C          DETERMINACAO DA VARIAVEL QUE SAI DA BASE -LINHA K1
C
  100 DO 150 I=1,N
      IF(PRDT(I)) 110,120,120
  110 WRITE(3,13)  PRDT(I)
      GO TO 830
  120 IF(C(I,K2)) 130,130,140
  130 AMT(I)=-1.
      GO TO 150
  140 AMT(I)=PRDT(I)/C(I,K2)
  150 CONTINUE
C
C          SELECAO DO MENOR  VALOR  POSITIVO   B(I)/C(I,K2)
      I=1
  160 IF(AMT(I)) 170,210,210
  170 I=I+1
      IF(I-N) 160,160,180
  180 WRITE(3,13) AMT(N)
      GO TO 830
  210 ZMIN=AMT(I)
      K1= I
  220 I=I+1
      IF(I-N) 230,230,300
  230 IF(AMT(I)) 220,240,240
  240 IF(ZMIN-AMT(I)) 220,220,210
C
C          TROCA DE VARIAVEL DA BASE
```

```
  300 Y(K1)=X(K2)
      DO 310 K=1,L
      VALY(K1,K)= VALX(K,K2)
  310 CONTINUE
      ITER =ITER + 1
      IF(ITAB)350,350,340
  340 WRITE(3,1015)
      WRITE(3,313)ITER,K1,K2,C(K1,K2)
  313 FORMAT(///,3X,'*** ITERACAO   ',I3,'  ELEMENTO PIVO -C('I3,',',I3,
     A'),= ',E16.7)
C
C
C          CALCULO DAS NOVAS CONSTANTES B(I)
C
 350  DO 400 I=1,N
      PRDT(I) = PRDT(I) - ZMIN*C(I,K2)
  400 CONTINUE
      PRDT(K1) = ZMIN
C
C          CALCULO DOS NOVOS COEFICIENTES  C(I,J)
C
      DO 500 J=1,M
      DO 500 I=1,N
      D(I,J) = C(I,J)- C(K1,J)*(C(I,K2)/C(K1,K2))
  500 CONTINUE
      DO 510 J=1,M
      D(K1,J) = C(K1,J)/C(K1,K2)
  510 CONTINUE
      DO 520 J=1,M
      DO 520 I=1,N
      C(I,J) =D(I,J)
  520 CONTINUE
C
C          IMPRESSAO DE TODAS AS ITERACOES OU  TABLEAU OTIMO
C
      IF(ITAB) 40,40,600
C
C          IMPRESSAO DE CADA  ITERACAO
C
 600  WRITE(3,5001)
      DO 610 I=1,N
      WRITE(3,13) Y(I),PRDT(I)
  610 CONTINUE
      WRITE(3,5002)
      DO 620 I=1,N
      WRITE(3,12)  (C(I,J),J=1,M)
  620 CONTINUE
      GO TO 40
C
C          VERIFICANDO      SE  TODAS AS PRIORIDADES FORAM ATINGIDAS
C
  790 L1= L1+1
      K3=L-L1
      IF(K3.GE.1.)GOTO 5
C
C          IMPRESSAO DOS RESULTADOS FINAIS
C
  800 WRITE(3,1014) ITER
      WRITE(3,1015)
 1015 FORMAT(1H1)
 1014 FORMAT(//////,10X,'TOTAL DE ITERACOES .....',I5)
```

```
      WRITE(3,5000)
 5000 FORMAT(//,20X,'****** SOLUCAO FINAL - SIMPLEX ******',////)
      WRITE(3,5001)
 5001 FORMAT(///,10X,'VALORES DAS VARIAVEIS',/)
  801 DO 810 I=1,N
      WRITE(3,13) Y(I),PROT(I)
  810 CONTINUE
      WRITE(3,5002)
 5002 FORMAT(///,10X,'MATRIZ DE COEFICIENTES C(I,J)',/)
  811 DO 812 I=1,N
      WRITE(3,12)  (C(I,J),J=1,M)
  812 CONTINUE
      WRITE(3,5003)
 5003 FORMAT(///,10X,'MATRIZ ZJ-CJ ',//)
  813 DO 814 K=1,L
      WRITE(3,12)  (RVLX(K,J),J=1,M)
  814 CONTINUE
C
C          AVALIACAO DA FUNCAO OBJETIVA
C
      DO 820 K=1,L
      ZVAL(K)=0.
      DO 820 I=1,N
      ZVAL(K)=  ZVAL(K) + PROT(I)*VALY(I,K)
  820 CONTINUE
      WRITE(3,5004)
 5004 FORMAT(//,10X,'AVALIACAO DOS OBJETIVOS',//)
      DO 821 K=1,L
      KK=L-K
      IF(TEST.EQ.1.0) GO TO 89
      KK=KK+1
   89 WRITE(3,15) KK,ZVAL(K)
  821 CONTINUE
      CALL FINISH(RHS1,PROT,VALY,L,KPCK,Y,N,KEPT,TEST)
  830 CONTINUE
      STOP
      END
```

```
      SUBROUTINE START(NROWS,NVAR,NPRT,C,VALX,VALY,RHS,RHS1,KPCK,KEPT,TE
     1ST)
C
C          A  SUBROTINA  START  TRANSFORMA OS VALORES
C          INICIAIS  LIDOS  EM UMA SERIE  DE MATRIZES
C          NECESSARIAS PARA  A RESOLUCAO DO PROBLEMA.
C
C
      DIMENSION RHS(70)
      DIMENSION VALY(70,10)
      DIMENSION C(70,150),VALX(10,150)
      DIMENSION EQUALS(70),RVLX(10,150)
      DIMENSION KEPT(70)
      DIMENSION RHS1(70)
      REAL NEG
      REAL L
      NR=70
      NV=150
    1 FORMAT(A4,3I3)
      DATA POS,NEG/'POS ','NEG '/
      DATA DATA/'DATA'/
      DATA OBJ/'OBJ '/
      DATA PROB/'PROB'/
      DATA B /'B'/
      DATA E,G,L/'E','G','L'/
      DATA RGHT/'RGHT'/
      TEST=0.
      WRITE(3,2)
    2 FORMAT(1H1,// //, 20X,'****** CARTOES DE DADOS ******',////)
C
C
C
C          LER O CARTAO  COM NUMERO DE LINHAS,VARIAVEIS E
C          PRIORIDADES
C
   10 READ(1,1) ANAME,NROWS,NVAR,NPRT
      WRITE(3,3)ANAME,NROWS,NVAR,NPRT
    3 FORMAT(27X,A4,3I3)
      LISP=NPRT +1
      IF(NVAR.LE.0) GO TO 1020
      IF(NPRT.LE.0) GO TO 1020
      IF(NROWS.LE.0) GO TO 1020
      IF(ANAME.NE.PROB) GO TO 901
C
C
C          LER O TIPO DE RESTRICOES  -SERA UMA DAS LETRAS
C
C                E -PARA  REST. DO TIPO IGUAL
C                L -PARA  REST. DO TIPO MENOR OU IGUAL
C                G -PARA  REST. DO TIPO MAIOR OU IGUAL
C                B -PARA  REST. DO TIPO MAIOR OU MENOR
C
      READ(1,11) (EQUALS(I),I=1,NROWS)
   11 FORMAT(80A1)
      WRITE(3,4)(EQUALS(I),I=1,NROWS)
    4 FORMAT(/,27X,40(A1,1X))
C
C
      NART=0
C
C          CALCULO DO NUMERO DE VAR.DEFOLGA POSITIVAS
```

```
C
      NFLDS=0
      DO 12 I=1,NROWS
      IF(EQUALS(I).EQ.B) NFLDS=NFLDS+1
   12 IF(EQUALS(I).EQ.G) NFLDS=NFLDS+1
C
C
C          TESTANDO AS DIMENSOES
C
C..............................................................
      NSIZE= NFLDS+NROWS+NVAR
      IF(NROWS.GT.NR) GO TO 911
      IF(NSIZE.GT.NV) GO TO 911
C
C          ZERANDO TODAS  AS  MATRIZES
C
C..............................................................
      KDUD=NPRT+1
      DO 16 J=1,NSIZE
      DO 16 I=1,NROWS
      KEPT(I)=0
      IF(I.GT.KDUD) GO TO 17
      K=I
      RVLX(K,J)=0.
      VALX(K,J)=0
   17 IF(I.EQ.J)  C(I,J)=1.
      VALY(I,K)=0.
      IF(I.NE.J)  C(I,J)=0.
   16 CONTINUE
      DO 109 K=1,LISP
      DO 109 J=1,NSIZE
      VALX(K,J)=0.
      RVLX(K,J)=0.
  109 VALY(K,K)=0.
  110 KPCK=0
      K=KDUD
C
C
C          AJUSTE DAS VARIAVEIS DE FOLGA E FUNCAO OBJETIVA
C          DE ACORDO COM O TIPO DE RESTRICOES
C
C
         DO 13 I=1,NROWS
      IF(EQUALS(I).EQ.E) GO TO 14
      IF(EQUALS(I).EQ.G) GO TO 15
      IF(EQUALS(I).EQ.L) GO TO 13
      IF(EQUALS(I).EQ.B) GO TO 18
      GO TO 910
  14  J=I
      VALX(K,J)=1.0
      NART=1
      TEST =1.
      GO TO 13
   15 KPCK= KPCK+1
      J=NROWS+KPCK
      C(I,J)=-1.
      C(I,I)=0.
      KEPT(I)=J
      J=I
      VALX(K,J)=1.
      TEST=1.
```

```
      NART=1
      GO TO 13
   18 KPCK=KPCK+1
      J=KPCK+NROWS
      C(I,J)=-1.
      KEPT(I)=J
   13 CONTINUE
C
C
C            LEITURA DA FUNCAO OBJETIVA
C
C.....................................................................
      READ(1,21) ANAME
      WRITE(3,5)ANAME
    5 FORMAT(/,27X,A4,2I5,F16.6)
   19 I=0
      IF(ANAME.NE.OBJ) GO TO 920
      IF(ANAME.EQ.OBJ) GO TO 20
   20 READ(1,21) ANAME,I,M,TEMP
      WRITE(3,5)ANAME,I,M,TEMP
      IF(ANAME.EQ.DATA) GO TO 30
      IF(M.LE.0) GO TO 1022
      K=LISP-M
   21 FORMAT(A4,2I5,F16.6)
      IF(J.LE.0) GO TO 1022
      IF(K.GT.NPRT) GO TO 1024
      IF(ANAME.EQ.NEG) GO TO 26
      IF(ANAME.EQ.POS) GO TO 25
      GO TO 27
   26 J=I
      VALX(K,J)=TEMP
      GO TO 20
   25 J=KEPT(I)
      IF(KEPT(I).EQ.0) GO TO 1026
      VALX(K,J)=TEMP
      GO TO 20
   27 IF(TEMP) 926,20,926
C
C
C            LEITURA  DOS  DADOS DA MATRIZ DE COEFICIENTES
C
C.....................................................................
   30 READ(1,21) ANAME,I,J,TEMP
      WRITE(3,5)ANAME,I,J,TEMP
      IF(ANAME.EQ.RGHT) GO TO 40
      IF(I.LE.0) GO TO 1090
      IF(J.EQ.0) GO TO 1090
      J= KPCK+NROWS+J
      C(I,J)= TEMP
      GO TO 30
C
C
C            LEITURA DAS  CONSTANTES  DAS RESTRICOES
C
C.....................................................................
   40 READ(1,44) (RHS(I),I=1,NROWS)
      WRITE(3,6)(RHS(I),I=1,NROWS)
    6 FORMAT(/,27X,8F12.2)
   44 FORMAT(8F10.0)
      WRITE(3,620)
  620 FORMAT(1H1)
```

```
      WRITE(3,5018)
 5018 FORMAT(/////,20X,'SUMARIO DE INFORMACOES DE ENTRADA')
      IGER=NSIZE-NVAR
      WRITE(3,2017)NROWS,NSIZE,NVAR,IGER,NPRT,NART
 2017 FORMAT(/,20X,'NUMERO DE LINHAS..........',I5,//,20X,'NUMERO DE VAR
     7IAVEIS........',I5,/,23X,'REAIS.......',I5,/,23X,'GERADAS.....',I5,
     8//,20X,'NUMERO DE PRIORIDADES.....',I5,//,20X,'PRIORIDADES ADICION
     9AIS....',I5,///)
      NVAR= NSIZE
C
C
C           IMPRESSAO  DOS  RESULTADOS
C
C.....................................................................
      WRITE(3,5015)
 5015 FORMAT(1H1,/////,20X,'****** TABELA INICIAL - ENTRADA ******',///)
      WRITE(3,7)
    7 FORMAT(///,2X,'---VALORES CONSTANTES',//)
      DO 41 I=1,NROWS
      IF(RHS(I)) 941,42,43
   42 RHS(I)= .00001
   43 RHS1(I) = RHS(I)
      WRITE(3,1111) I,RHS(I)
 1111 FORMAT(23X,I3,5X,F15.5,/)
   41 CONTINUE
      WRITE(3,5016)
 5016 FORMAT(/,2X,'---MATRIZ DE COEFICIENTES C(I,J)',//)
      DO 1112 I=1,NROWS
      WRITE(3,2519) I
 2519 FORMAT(5X,'LINHA ',I3)
 1112 WRITE(3,1113) (C(I,J),J=1,NSIZE)
 1113 FORMAT(14X,10F10.2)
      WRITE(3,5017)
 5017 FORMAT(/,2X,'---A FUNCAO OBJETIVA',//)
      DO 1114 K=1,NPRT
      M=LISP-K
      WRITE(3,2150) M
 2150 FORMAT(5X,'PRIORIDADE',I3)
 1114 WRITE(3,1113) (VALX(K,J),J=1,NSIZE)
      IF(NART.GT.0) NPRT= NPRT+1
      RETURN
  910 WRITE(3,914)
  914 FORMAT('  ERRO NO NUMERO DE LINHAS PERFURADO OU NO
     1TIPO DE RESTRICAO - B, G,E OU L ' )
      GO TO 999
 1090 WRITE(3,1091)
 1091 FORMAT('  ERRO  NA  DEFINICAO DA LINHA OU COLUNA  DO DADO ')
      GO TO 999
  920 WRITE(3,921)
  921 FORMAT('  ERRO NO CARTAO QUE INDICA O INICIO DA FUNCAO OBJETIVA')
      GO TO 999
 1020 WRITE(3,1021)
 1021 FORMAT('  NUMERO DE LINHAS,VARIAVEIS OU PRIORIDADES DEVEM SER
     2DIFERENTES DE  ZERO')
      GO TO 999
 1022 WRITE(3,1023)
 1023 FORMAT('  VALOR DA COLUNA OU PRIORIDADE IGUAL OU MENOR QUE ZERO')
      GO TO 999
  911 WRITE(3,912)
  912 FORMAT('  PROBLEMA  EXCEDE DIMENSOES  NO NUMERO DE VARIAVEIS  ,
     3VERIFIQUE  SUAS NECESSIDADES ')
```

```
      GO TO 999
 1026 WRITE(3,1027)
 1027 FORMAT('  DESVIO POSITIVO PARA MINIMIZAR NAO EXISTENTE ')
      GO TO 999
 1024 WRITE(3,1025)
 1025 FORMAT('  NUMERO DE PRIORIDADES DA FUNCAO OBJETIVA EXCEDE O NUME
     4RO DEFINIDO')
      GO TO 999
  901 WRITE(3,902)
  902 FORMAT('  PRIMEIRO CARTAO- PROB - EXTRAVIADO OU ERRADO ')
      GO TO 999
  926 WRITE(3,927)
  927 FORMAT('  FOI DEFINIDO UM VALOR NA FUNCAO OBJETIVA MAS NAO INDICA
     5SE E PARA UM DESVIO POSITIVO OU NEGATIVO ' )
  941 WRITE(3,942)
  942 FORMAT('  NAO DEVE EXISTIR VALOR NEGATIVO PARA AS CONSTANTES DAS
     6RESTRICOES ',/'   MULTIPLIQUE A RESTRICAO POR MENOS UM ' )
      GO TO 999
  999 RETURN
      END
```

```
      SUBROUTINE FINISH(RHS1,RHS,VALY,NPRT,KPCK,Y,NROWS,KEPT,TEST)
      REAL NEGSLK
      DIMENSION VALY(70,10)
      DIMENSION ZVAL(10)
      DIMENSION RHS(70)
      DIMENSION KEPT(70)
      DIMENSION Y(70),RHS1(70)
C
C
C           RHS1 E UM VETOR DE RESERVA PARA AS CONSTANTES
C           DAS RESTRICOES, DE MODO QUE A DIFERENCA DOS
C           VALORES INICIAL E FINAL  CORRESPONDEM AS VARI
C           AVEIS DE FOLGA (DESVIOS)
C
C
C
C           VARIAVEIS DE FOLGA
C
      WRITE(3,21)
   21 FORMAT(1H1,//,27X,'ANALISE DOS DESVIOS ')
      WRITE(3,1)
      WRITE(3,8)
    8 FORMAT(10X,'LINHA',8X,'VALOR',12X,'DESVIO POS.',7X,'DESVIO NEG.')
      WRITE(3,1)
      DO 19 I=1,NROWS
      NEGSLK=0.0
      POSSLK=0.0
      DO 11 J=1,NROWS
      M= Y(J)
      IF(I-M) 9,10,9
    9 IF(M-KEPT(I))  11,12,11
    1 FORMAT(////)
   11 CONTINUE
      GO TO 13
   10 NEGSLK= RHS(J)
      GO TO 13
   12 POSSLK=RHS(J)
   13 WRITE(3,14) I,RHS1(I),POSSLK,NEGSLK
   14 FORMAT(10X,I3,3F20.5,/)
   19 CONTINUE
   43 FORMAT(12X,I3,F18.5,/)
C
C
C           VALORES DAS VARIAVEIS REAIS DO PROBLEMA
C
      WRITE(3,44)
   44 FORMAT(////,27X,'ANALISE DAS VARIAVEIS')
      WRITE(3,45)
   45 FORMAT(///,10X,'VARIAVEL',10X,'VALOR',//)
      DO 41 I=1,NROWS
      NCHCK= Y(I)-KPCK-NROWS
      IF(NCHCK) 41,41,42
   42 WRITE(3,43) NCHCK,RHS(I)
   41 CONTINUE
      WRITE(3,50)
   50 FORMAT(////,27X,'ANALISE DOS OBJETIVOS',////,20X,'PRIORIDADE',9X,
     9'QUANTIDADE',/)
      DO 52 K=1,NPRT
      ZVAL(K)=0.0
      DO 51 I=1,NROWS
   51 ZVAL(K)=ZVAL(K)+ VALY(I,K)*RHS(I)
```

```
      LISP= NPRT+1
      KK= LISP-K
      IF(TEST.EQ.0) GO TO 52
      KK= NPRT-K
      IF(KK.GT.0) GO TO 52
      WRITE(3,78) ZVAL(K)
 78   FORMAT(/,20X,'ARTIFICIAL',10X,F14.5)
      GO TO 77
 52   WRITE(3,53) KK,ZVAL(K)
 53   FORMAT(/,24X,I2,14X,F14.5)
 77   CONTINUE
      RETURN
      END
```