APLICAÇÃO DE BUSCA DE CAMINHOS EM GRAFOS AO PLANEJAMENTO DA OPERAÇÃO DE SISTEMAS TERMOELÉTRICOS

Adevanilde Aparecida Rotter Curotto

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:

Clóvis Caesar Gonzaga
(Presidente)

Ronaldo C.Marinho Persiano

Carlos Humes Júnior

RIO DE JANEIRO, RJ - BRASIL
MARÇO DE 1983

CUROTTO, Adevanilde Aparecida Rotter

Aplicação de Busca de Caminhos em Grafos ao Planejamento da Operação de Sistemas Termoelétricos (Rio de Janeiro) 1983.

IX, 114 p., 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia de Sistemas, 1983)

Tese - Universidade Federal do Rio de Janeiro. Faculdade de Engenharia

1. Busca de Caminhos em Grafos I. COPPE/UFRJ II. Título (série).

aos meus filhos

Sandra Mara

Marcio Luiz

AGRADECIMENTOS

Ao Clovis C. Gonzaga pelo tema de pesquisa e pela orientação da tese.

Ao Ronaldo Marinho Persiano pela orientação em parte do trabalho.

Ao meu marido pelo incentivo e apoio que determinaram a conclusão desse trabalho.

Ao CNPq pelo auxílio financeiro.

RESUMO

O planejamento da operação diária de um sistema termoelétrico é separado em dois subproblemas. O primeiro, encarregado de selecionar as unidades que estarão em operação a cada instante (Unit Commitment), é resolvido por um algoritmo de Busca de Caminhos em Grafos. O segundo, encarregado da alocação ótima da demanda entre as unidades selecionadas para operar (Despacho), é resolvido por um algoritmo desenvolvido por TURGEON[10].

O horizonte de planejamento é de um dia, discretizado em horas. A demanda é determinística e conhecida a priori. Os custos de geração das unidades do sistema são dados por funções quadráticas. O custo de aquecimento de uma unidade é representado por uma função exponencial. São consideradas as seguintes restrições: reserva de demanda, tempo mínimo no estado de operação, número máximo de unidades ativadas ou desativadas simultâneamente e comportamento da demanda.

São apresentados os resultados numéricos obtidos para dois sistemas termoelétricos.

ABSTRACT

The problem of optimal scheduling of a thermal power generating system is partitioned into a two problems. The Unit Commitment problem select the units to be placed in operation at each point in time and is solved by an algorithm of Search for Minimum Cost Paths in a Graph. The Economic Dispatch problem allocate the load among the units selected for operation and is solved by an algorithm created by TURGEON[10].

The scheduling is made in a 24 hour horizon divided in hours. The demand is deterministic and known in advance. Quadratic curves discrebe the units production costs. The unit heating cost is obtained by an exponential function. The set of constraints are: spinning-reserve, unit minimum down time and up time, maximum number of units to be started up or shuted down simultaneously and demand variation.

Numerical results obtained for two thermal power generating systems are presented.

ÍNDICE

I - INTRODUÇÃO

Em um sistema de geração de energia elétrica, composto por unidades termoelétricas, constitui-se um problema de otimização estabelecer quais unidades estarão operando a cada instante, dentro de um horizonte de planejamento. Isto se deve ao fato que normalmente não é econômico permanecer com todas as unidades ativas interruptamente, devido à grande variação da demanda e aos altos custos de operação de cada unidade.

Geralmente o horizonte de planejamento é de um dia, discretizado em horas. A demanda é determinística e conhecida a priori. O planejamento da operação do sistema é feito, hora a hora, visando atender à demanda e minimizar o custo total de operação.

O problema pode ser separado em dois subproblemas. Um, conhecido na literatura como "Unit Commitment", seleciona as unidades que estarão em operação a cada hora e o outro, denominado "Despacho", encarrega-se da alocação ótima da demanda entre as unidades selecionadas para operar.

O Despacho é resolvido, neste trabalho, por um algoritmo desenvolvido por TURGEON[10], bastante eficiente e rápido computacionalmente (ver Apêndice A).

O Unit Commitment, sendo um problema de decisões sequenciais (uma unidade estará ou não em operação no período seguinte), num horizonte limitado e discretizado hora a hora, torna a técnica de Busca em Grafos a mais eficiente, embora as mais utilizadas neste tipo de problema sejam as de Programação Dinâmica e Branch and Bound.

A transição da situação em que se encontra o sistema no inicio do planejamento para a situação em que se encontrará no final do planejamento, através da sequência de decisões tomadas, associa-se à busca de caminhos no grafo modelado por essas decisões. A busca de políticas ótimas para essas decisões, associa se à busca de caminhos de custo mínimo entre os nós do grafo.

O subproblema Despacho é resolvido a cada ramo do grafo, apenas para se calcular o custo de geração do ramo.

Para a resolução do Unit Commitment existem informações que permitem a estimativa de custos futuros, tornando o algoritmo A* (formalizado por NILSSON[20]) muito eficiente.

Assim, neste trabalho, aplica-se a técnica de Busca de Caminhos em Grafos para o Unit Commitment, utilizando-se para tanto o algoritmo A* com estimativas de custos futuros (redefinidas a cada novo caminho encontrado).

No capítulo II é feita a definição da modelagem do problema, sendo formuladas as funções de custos, as restrições e a função objetivo.

No capítulo III o problema é modelado como um problema de busca em grafos, definindo-se o grafo e seus componentes.

No capítulo IV é feita a esquematização da resolução tanto para o Despacho como para o Unit Commitment, definindo-se os algoritmos utilizados e as estimativas de custos futuros.

No capítulo V é descrito o funcionamento do programa de computador (feito em ALGOL para o B-6700).

Exemplos de dois sistemas termoelétricos são apresentados no capítulo VI e finalmente as conclusões são feitas no capítulo VII.

O Unit Commitment já foi extensamente estudado por vários autores, que foram introduzindo técnicas cada vez mais aprimoradas mas nem sempre utilizáveis na prática.

Os primeiros trabalhos apresentados baseavam-se em uma lista de prioridades para a escolha das unidades a entrar em operação. Esta lista era feita baseada em critérios tais como: eficiência, custos, locação, tipos de turbina e outros.

Em 1953, CHANDLER, DANDENO, GLIMM e KIRCHMAYER[1] utilizaram es-

ta técnica, não considerando restrição alguma de operação. Para o Despacho, o custo do combustível em função da potência gerada foi considerado como uma função quadrática, com níveis mínimo e máximo de geração para cada unidade.

Também BALDWIN, DALE e DITTRICH[2] utilizaram lista de prioridades. Naquele trabalho a restrição de Reserva foi introduzida como sendo uma constante, isto é, as unidades em operação devem ter capacidade para cobrir a demanda mais uma constante como margem de garantia do suprimento do mercado. O custo de geração era também uma função quadrática, com níveis mínimo e máximo de geração. Um novo custo foi introduzido: o custo de aquecimento de uma unidade.

GARVER[3] em 1963, mostrou a possibilidade do problema ser formulado como um problema de Programação Inteira. A função custo de geração foi aproximada para uma função linear por partes e não foram consideradas restrições de operação. É um trabalho válido pela introdução de novas idéias na área.

LOWERY[4] em 1966, apresentou a Programação Dinâmica como uma técnica viável para a resolução do problema. Neste mesmo ano KERR, SCHEIDT, FONTANA e WILEY[5], apresentaram um trabalho utilizando lista de prioridades.

MUCKSTADT e WILSON[6] em 1968, formularam também como Programação Inteira, fazendo um estudo mais completo do que Garver. A função custo de geração foi aproximada para uma função linear por partes. A demanda foi considerada uma variável aleatória com uma distribuição de probabilidade conhecida. Foram consideradas algumas restrições de operação tais como: Reserva, como uma constante; Intercâmbio de sistemas, como sendo uma unidade específica. O custo de aquecimento de uma unidade foi considerado constante.

Em 1971, GUY[7] e AYOUB e PATTON[8], introduziram a consideração da confiabilidade como uma função probabilidade de satisfazer o mercado em todos os instantes do planejamento. O Unit Commitment foi tratado por uma técnica de enumeração não ótima.

MUCKSTADT e KOENIG[9], em 1977, consideraram a função custo de geração linear por partes, apenas uma restrição (Reserva) e utilizaram uma técnica misto de Lagrange, Programação Dinâmica e Branch and Bound.

TURGEON[11,12] considerou a função custo de geração quadrática e a de aquecimento exponencial. A técnica utilizada foi a de Branch and Bound com as seguintes restrições: Reserva, como a maior unidade em operação; Tempos mínimos em que uma unidade deve permanecer no estado de operação; Uma unidade não deve ser aquecida mais que uma vez ao dia; Número máximo de unidades que são ativadas ou desativadas simultaneamente. Foram acrescidos alguns critérios para diminuir o número de unidades a serem testadas para ativação ou desativação. Ainda assim, a técnica mostrou-se inviável para o planejamento da operação de um sistema composto por dez unidades, ao longo de um dia. Novas restrições foram então impostas após a análise de alguns resultados obtidos a priori.

DILLON, EDWIN, KOCHS e TAND[13], em 1978, também utilizaram a técnica de Branch and Bound. A função custo de geração foi considerada como uma função linear por partes.

YAMAYEE, ABIAD e MORIN[14], em 1979, utilizaram uma técnica mista de Programação Dinâmica e Branch and Bound. O custo de geração foi dado por funções quadráticas e o de aquecimento por funções exponenciais. As restrições consideradas foram: Reserva, como uma constante e Tempos mínimos em que uma unidade deve permanecer no estado de operação.

GONZAGA[15] mostrou a superioridade dos algoritmos de Busca em Grafos sobre as técnicas de Programação Dinâmica e Branch and Bound para problemas de decições sequenciais determinísticos a horizontes limitados e totalmente discretos.

Neste trabalho a função custo de geração é quadrática e a de aquecimento exponencial. As restrições consideradas são as mesmas de TURGEON[12] com algumas variações.

II - O MODELO

É considerado um sistema composto somente por unidades termoelétricas e um mercado determinístico conhecido a priori. O planejamento da operação do sistema é feito ao longo de um dia, discretizado em horas, a fim de minimizar o custo total de operação.

As restrições consideradas para o planejamento são separáveis em dois blocos distintos: um contendo as usadas para o Despacho, que dizem respeito as características individuais de cada unidade ou do sistema e outro contendo as usadas para o Unit Commitment, que visam facilitar a operação e manutenção do sistema.

Este trabalho tem a pretensão de não se fixando por demais em um modelo, ser facilmente adaptável a qualquer inclusão ou exclusão de restrição.

São assumidos dois custos: o custo de geração e o custo de aquecimento. O custo de geração envolve duas parcelas: uma correspondendo ao custo de manutenção de uma unidade aquecida na temperatura de operação, dado por constantes, e outra ao custo de produção de energia em função do nível de produção de cada unidade, dado por funções quadráticas. O custo de aquecimento é o custo de se levar uma unidade de uma temperatura inicial até a temperatura de operação, determinado por funções exponenciais.

II.1 - O Sistema

O sistema será formado apenas por unidades termoelétricas.

Define-se: $i = 1, N$ as unidades termoelétricas

II.2 - O Mercado

O mercado a ser satisfeito é dado como determinístico e conhe-

cido a priori. A figura II.1 mostra um exemplo de uma curva de carga utilizada para o planejamento diário da operação de um sistema.

Define-se: k = 1, P os períodos do planejamento.

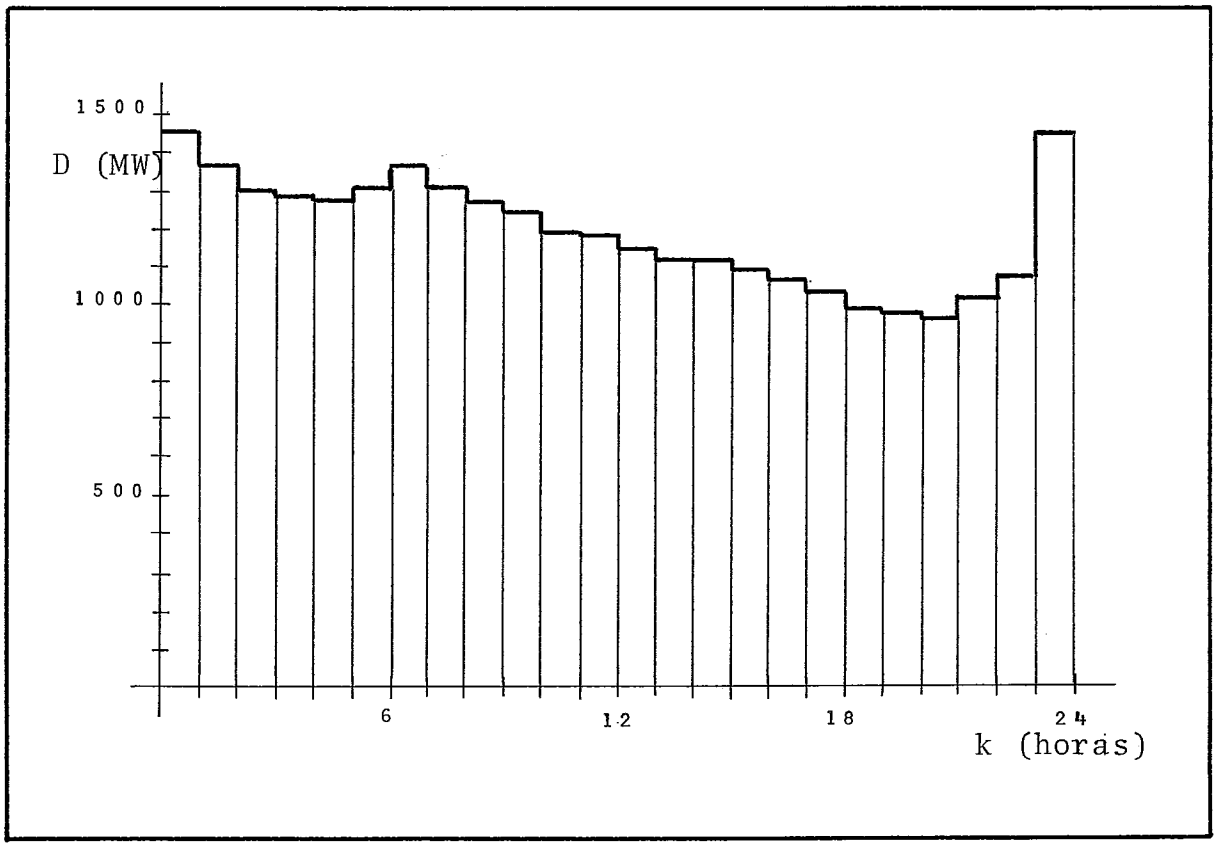

FIGURA II.1 - Exemplo de uma curva de carga.

II.3 - Restrições para o Despacho

II.3.1 - Energia Gerada

A cada período a energia gerada pelo sistema deve ser igual a demanda.

$$\Sigma_i \; G_i^k = D^k \qquad \text{II.1}$$

onde:

G_i^k = nível de geração da unidade i no período k

D^k = demanda no período k

II.3.2 - Níveis Mínimos e Máximos de Geração

Uma unidade em operação deve gerar entre seus níveis mínimo e máximo de geração, ou não gerar.

$$\underline{G}_i \leq G_i^k \leq \overline{G}_i \text{ ou } G_i^k = 0 \qquad \text{II.2}$$

onde:

$\underline{G}_i$ = nível mínimo de geração da unidade i

$\overline{G}_i$ = nível máximo de geração da unidade i

Muitos trabalhos consideram que uma unidade escolhida para operar deve sempre gerar entre seus limites de geração. Neste modelo é considerada a possibilidade de não gerar, apesar de estar aquecida para tanto, pagando por isto um custo de manutenção, desde que há possibilidade de ser mais econômico manter a unidade aquecida por alguns períodos do que reaquece-la quando sua utilização se fizer necessária.

II.3.3 - Reserva

Várias são as formas consideradas para a garantia do suprimento do mercado com uma certa margem de segurança. Alguns consideram que o sistema deve ser capaz de atender a demanda mais uma constante no período (deve ter unidades aquecidas prontas para operar em alguma eventualidade). Outros consideram que o sistema deve ser capaz de atender a demanda mais uma percentagem. Ainda outros, introduzem a confiabilidade como uma função probabilidade de satisfazer o mercado em todos os períodos.

Aqui a reserva é considerada igual a capacidade da maior unidade em operação no período. Assim o sistema terá capacidade para atender a demanda mesmo ocorrendo falha na maior unidade em operação a cada período.

$$\Sigma_i \overline{G}_i - \max(\overline{G}_i) \geq D^k \qquad \text{II.3}$$

sendo:

$i \in O^k$ (conjunto das unidades em operação no período k)

II.4 - Restrições para o Unit Commitment

II.4.1 - Tempo Mínimo no Estado de Operação

Não é conveniente que uma unidade seja ativada e desativada seguidamente. Assim, num determinado período, para que uma unidade possa ser escolhida para operar, ela deve ter permanecido alguns períodos desativada e para que uma unidade possa ser escolhida para ser desativada ela deve ter permanecido alguns períodos em operação.

$h_i^{k-1} \geq f_i$ (para possibilitar a ativação da unidade i no período k) II.4

$h_i^{k-1} \geq q_i$ (para possibilitar a desativação da unidade i no período k) II.5

sendo:

h_i^{k-1} = número de períodos em que a unidade i está no estado de operação (ativada ou desativada) no período k-1

f_i = número mínimo de períodos em que a unidade i deve permanecer desativada a partir do período em que ela passou para esse estado

q_i = número mínimo de períodos em que a unidade i deve permanecer ativada a partir do período em que ela passou para esse estado

II.4.2 - Número Máximo de Unidades que são Ativadas ou Desativadas Simultaneamente

Será restringido o número de unidades que podem ser ativadas ou desativadas simultaneamente. Não mais que R unidades podem

ser ativadas simultâneamente e não mais que S podem ser desativadas simultâneamente.

II.4.3 - Comportamento da Demanda

Se a demanda aumenta do período k-1 para o período k, não se permite que unidades sejam desativadas. Quando a demanda diminuir do período k-1 para o período k, não se permite que unidades sejam ativadas.

II.5 - Custo de Geração

Uma unidade escolhida para operar deve gerar entre seus limites de geração, ou não gerar, acarretando um custo de manutenção da unidade aquecida.

O custo de geração é dado em função do nível de geração de cada unidade, dado por funções quadráticas (ver figura II.2).

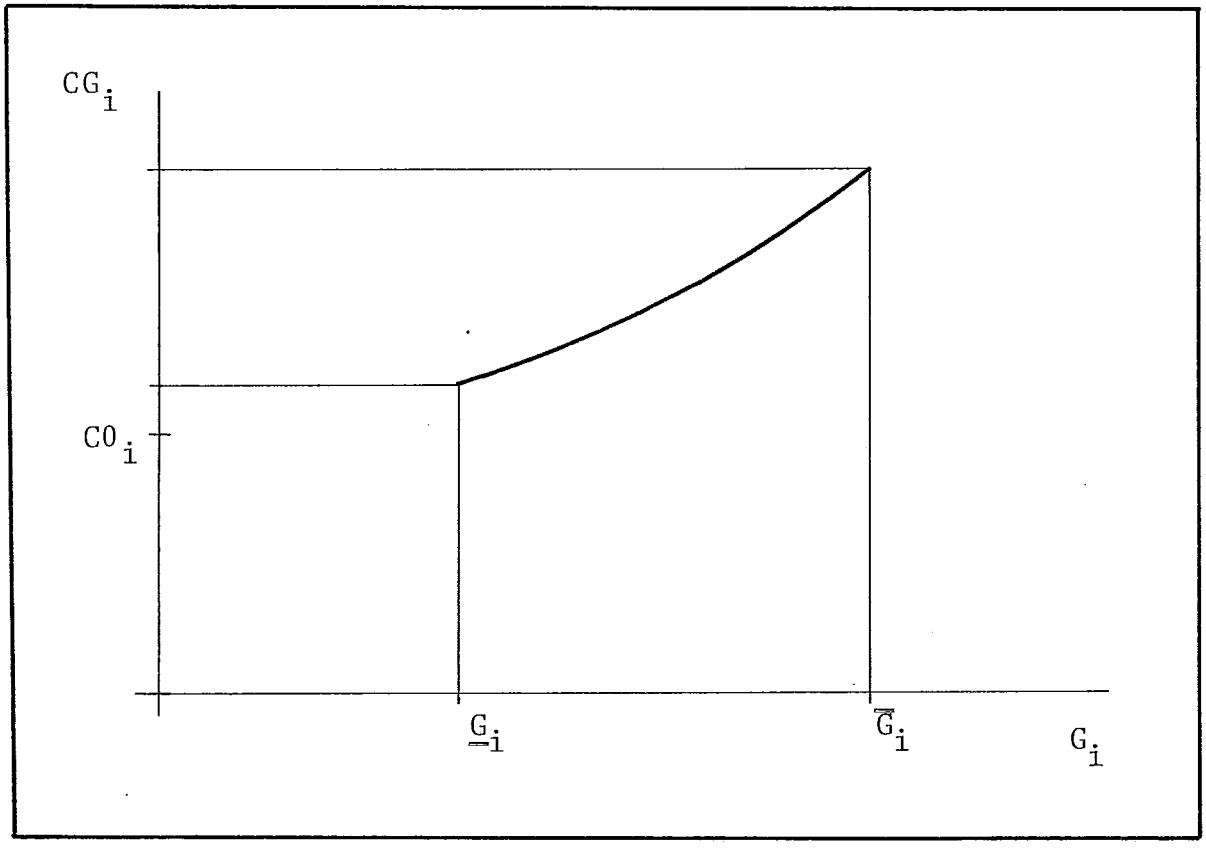

FIGURA II.2 - Exemplo de uma função custo de geração.

O custo unitário de geração num período k é dado por:

$$CG_i^k = al_i \, (G_i^k)^2 + bl_i \, G_i^k + cl_i \quad \text{se } i \in O^k \text{ e } \underline{G}_i \leq G_i^k \leq \overline{G}_i$$

$$= C0_i \quad \text{se } i \in O^k \text{ e } G_i^k = 0$$

$$= 0 \quad \text{senão} \qquad \text{II.6}$$

onde:

al_i, bl_i, cl_i = parâmetros da função custo de geração para a unidade i

$C0_i$ = custo de manutenção da unidade i aquecida

O^k = conjunto das unidades em operação no período k

O custo de geração total do período é obtido alocando-se de maneira ótima a demanda entre as unidades escolhidas para operar a cada período.

$$CG^k = \Sigma_i \, CG_i^k \qquad \text{II.7}$$

A alocação ótima da demanda entre as unidades é função do subproblema Despacho (resolvido através de um algoritmo não linear exposto no Apêndice A).

II.6 - Custo de Aquecimento

Uma unidade desativada é escolhida para operar. A esta unidade é atribuido um custo de aquecimento em função do número de períodos que permaneceu nesse estado. A figura II.3 mostra um exemplo de uma curva de custo de aquecimento.

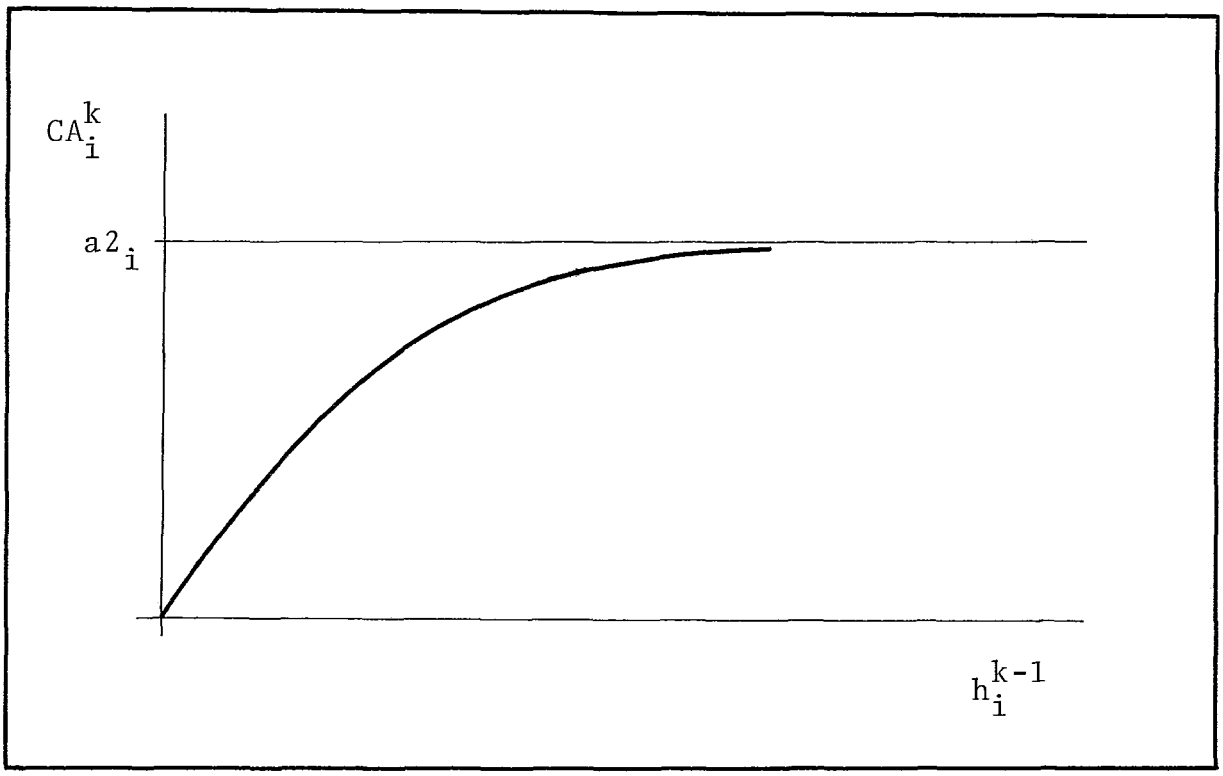

FIGURA II.3 - Exemplo de uma função custo de aquecimento.

O custo de aquecimento para uma unidade i num período k será dado por:

$$CA_i^k = a2_i \ (1 - b2_i \ e^{-(c2_i \ h_i^{k-1})}) \quad \text{se } i \in A^k$$

$$= 0 \quad \text{senão} \qquad \text{II.8}$$

sendo:

A^k = conjunto das unidades aquecidas no período k

h_i^{k-1} = número de períodos em que a unidade i permaneceu desativada

$a2_i$, $b2_i$, $c2_i$ = coeficientes da função exponencial que representa o custo de aquecimento da unidade i

O custo de aquecimento do período é dado por:

$$CA^k = \Sigma_i \ CA_i^k \qquad \text{II.9}$$

II.7 - Função Objetivo

O custo total de operação é dado pela soma dos custos dos períodos durante o horizonte de planejamento.

$$CT = \Sigma_k \, CG^k + CA^k \qquad \text{II.10}$$

O objetivo do problema é encontrar o estado operacional do sistema, a cada período, que minimize o custo total de operação , satisfazendo a todas as restrições.

O conjunto dos estados operacionais do sistema que satisfazem às restrições a cada período, formam um grafo em camadas. Encontrar o estado operacional do sistema que minimize o custo total de operação é encontrar o caminho ótimo deste grafo, cuja origem é o estado inicial do sistema e cujos alvos são os estados operacionais do sistema no último período de planejamento.

III - O GRAFO

O grafo formado pelas decisões que vão sendo tomadas sobre o sistema, ao longo do horizonte de planejamento, é um grafo em camadas, onde cada camada corresponde a um período de planejamento. Neste capítulo o Unit Commitment é formulado como um problema de busca em grafos, cujas componentes são definidas.

O grafo $G = (N^0, F)$ é caracterizado pela raiz N^0 e pelo operador sucessor F. A raiz é o estado operacional inicial do sistema. O operador sucessor aplicado ao estado operacional inicial gera todos os estados operacionais do sistema que satisfazem às restrições para o período um do planejamento. Aplicado a cada um destes, gera todos os estados operacionais do sistema que satisfazem às restrições para o período dois. E assim sucessivamente, tomando decisões sequenciais até o final do planejamento é formado um grafo em camadas, cujos alvos são os estados operacionais do período final do planejamento.

III.1 - Estágio

O estágio é equivalente ao período. Corresponde à discretização do tempo do planejamento em horas.

$$k = 1, P \text{ (total de horas do planejamento)} \qquad \text{III.1}$$

III.2 - Nós

Para que um nó caracterize perfeitamente a situação do sistema é necessário que seja dado por dois vetores: e^k e h^k. O primeiro representando o estado operacional do sistema e o segundo informando o número de estágios em que se encontra neste estado operacional. Assim um nó é definido por:

$$n^k = (e_i^k, h_i^k) \qquad \text{III.2}$$

onde:

e_i^k = estado operacional da unidade i no estágio k, sendo que:

e_i^k = 0 indica que a unidade i esta desativada no estágio k

e_i^k = 1 indica que a unidade i esta ativada no estágio k

h_i^k = número de estágios em que a unidade i esta no estado e_i^k

III.3 - Operador Sucessor

O operador sucessor é uma função que associa a cada nó em k-1 um conjunto de nós sucessores em k. A cada uma destas transições no estado operacional do sistema associa-se um ramo do grafo.

Dado um nó no estágio k-1, duas decisões são possíveis para cada da unidade: ou a unidade estará ativada ou desativada no estágio k, isto é:

u_i^{k-1} = 0 decide que a unidade i estará desativada no estágio k

u_i^{k-1} = 1 decide que a unidade i estará ativada no estágio k

III.3

III.3.1 - Decisão Admissível

A decisão tomada será admissível se as restrições de tempo mínimo no estado de operação e de comportamento da demanda forem satisfeitas, ou seja:

u_i^{k-1} = 1 será admissível se:

$$e_i^{k-1} = 1 \text{ ou}$$

$$e_i^{k-1} = 0, \quad h_i^{k-1} \geq f_i \quad \text{e} \quad D^{k-1} \leq D^k \qquad \text{III.4}$$

$u_i^{k-1} = 0$ será admissível se:

$$e_i^{k-1} = 0 \text{ ou}$$

$$e_i^{k-1} = 1, \quad h_i^{k-1} \geq q_i \quad \text{e} \quad D^{k-1} \geq D^k \qquad \text{III.5}$$

III.3.2 - Nó Sucessor

Um nó $n^k = (e_i^k, h_i^k)$ será sucessor de $n^{k-1} = (e_i^{k-1}, h_i^{k-1})$ se:

- e_i^k for gerado por decisões admissíveis

- h_i^k for calculado da seguinte maneira:

$$h_i^k = h_i^{k-1} + 1 \quad \text{se} \quad e_i^{k-1} = e_i^k$$

$$= 1 \quad \text{se} \quad e_i^{k-1} \neq e_i^k \qquad \text{III.6}$$

- satisfazer as restrições da reserva e do número máximo de unidades que são ativadas ou desativadas simultâneamente, isto é se:

$$\Sigma_i \, \overline{G}_i - \max (\overline{G}_i) \geq D^k$$

$$\Sigma_i \, (e_i^k - e_i^{k-1}) \leq R$$

$$\Sigma_j \, (e_j^{k-1} - e_j^k) \leq S \qquad \text{III.7}$$

onde:

$i \in O^k$ (conjunto das unidades em operação (ativadas) no estágio k)

$j \notin O^k$

III.4 - Alvo

Um nó $n^k = (e_i^k, h_i^k)$ será um nó do alvo se k for o estágio final do planejamento (k = P).

III.5 - Custo de um Ramo

A cada ramo está associado uma transição do estado operacional do sistema com o custo respectivo, dado por duas parcelas: custo de geração mais custo de aquecimento, ou seja:

$$c_j^k = CG_\ell^k + CA_\ell^k \qquad \text{III.8}$$

onde:

c_j^k = custo total do ramo dado por (n_j^{k-1}, n_ℓ^k)

CG_ℓ^k = custo de geração do ramo

CA_ℓ^k = custo de aquecimento do ramo

III.6 - Custo de um Caminho

Um caminho é dado por uma sequencia de decisões que definem a política de operação do sistema, do início ao fim do planejamento. A um caminho está associado um custo dado pela soma dos custos dos ramos pertencentes a esse caminho. Ao caminho de menor custo associa-se a política ótima de operação do sistema.

IV - OS ALGORITMOS

A resolução do problema da busca de caminho no grafo modelado anteriormente será feita através dos algoritmos descritos neste capítulo.

É dada inicialmente a nomenclatura utilizada pelos algoritmos e o algoritmo geral de grafos.

A resolução do subproblema Despacho é apresentada através da tabela de custos de geração.

As subestimativas de custos futuros são descritas. A subestimativa do custo de geração é fixa por estágio, calculada apenas no início do planejamento. A subestimativa do custo de aquecimento é calculada para todo nó do grafo.

O algoritmo A* apresentado utiliza apenas subestimativas, enquanto o algoritmo A* adaptativo além das subestimativas, utiliza também estimativas fortes para que soluções rápidas possam ser encontradas.

IV.1 - Nomenclatura

IV.1.1 - Listagem de Caminhos

Basicamente um algoritmo de grafos, na busca do caminho ótimo, deve listar todos os caminhos possíveis do nó inicial ao alvo e escolher o de menor custo.

Os caminhos partindo do nó inicial são representados de forma recursiva por uma tripla ordenada. Os algoritmos constroem listas dessas triplas, que são da forma:

$$E_j = (n_j^k, c_j, i)$$

onde:

n_j^k = nó terminal do caminho associado ao elemento E_j

c_j = custo deste caminho

i = índice do elemento antecessor ao elemento E_j (elemento ao qual o elemento E_j associou-se para a extensão do caminho)

j = índice do elemento E

IV.1.2 - Elemento Listado

Um elemento listado pode estar aberto ou fechado. Fechado indica que já foi expandido, isto é, gerou diferentes prolongamentos (associados a diferentes decisões) do caminho por ele representado. Aberto indica que ainda não foi expandido.

IV.1.3 - Expansão de um Elemento

O operador sucessor aplicado a um elemento $E_j = (n_j^k, c_j, i)$, gera sucessores da forma $E_\ell = (n_\ell^k, c_\ell, j)$ que serão abertos após a execução de uma rotina de eliminações que evita a presença de caminhos listados terminados em um mesmo nó. Obter os suces-

sores de um elemento significa expandi-lo.

IV.1.4 - Índice de Prioridade.

A cada elemento aberto E_j, estará associado um índice de prioridade $\hat{f}(E) \in R$ (conjunto dos números reais), que deverá ser tanto menor quanto "melhor" for o elemento. Um elemento será melhor que outro se tiver mais possibilidade de pertencer ao caminho ótimo.

IV.2 - Algoritmo Geral de Busca de Caminhos em Grafos

Um algoritmo geral de busca de caminhos em grafos pode ser descrito da seguinte forma:

Passo 0: Abra o elemento $E_0 = (N^0, 0, 0)$ (elemento inicial).

Passo 1: Se não há nenhum elemento aberto, pare.
Senão, escolha um aberto $E_j = (n_j^{k-1}, c_j, i)$ tal que $\hat{f}(E_j)$ seja mínimo entre os abertos.
Feche o elemento E_j.

Passo 2: Aplique o operador sucessor sobre o nó n_j^{k-1}, obtendo os nós sucessores n_ℓ^k, $\ell = 1, q$ (número de sucessores obtidos. Construa os sucessores:

$$E_\ell = (n_\ell^k, c_\ell, j)$$

onde:

$$c_\ell = c_j + c(n_j^{k-1}, n_\ell^k)$$

$\ell = 1, q$

$c(n_j^{k-1}, n_\ell^k)$ = custo da transição do nó n_j^{k-1} para n_ℓ^k

Passo 3: Para cada sucessor E_ℓ execute a rotina de eliminações.
Volte ao passo 1.

Rotina de Eliminações:

Passo 1: Compare E_ℓ com cada elemento listado E_m:
Se $n_\ell^k = n_m^k$ e $c_m \leq c_\ell$, abandone E_ℓ e retorne.
Se $n_\ell^k = n_m^k$ e $c_m > c_\ell$, retire E_m da lista.

Passo 2: Abra E_ℓ e retorne.

IV.3 - Tabela de Custos de Geração

Na expansão de um elemento, o cálculo do custo de geração dos ramos gerados pela aplicação do operador sucessor é determinado pelo subproblema Despacho (ver Apêndice A). Este cálculo depende apenas do estado operacional do nó do elemento em expansão. Para cada estágio k tem-se um conjunto T^k de configurações do sistema que satisfazem à restrição de Reserva. Dado um elemento em expansão pertencente ao estágio k-1 seus possíveis sucessores estarão contidos no conjunto T^k.

Assim é criada uma tabela que contem todas as configurações que satisfazem à restrição de Reserva em cada estágio. Deste modo é obtida uma grande economia de processamento, já que o custo de geração de uma configuração é calculado apenas uma vez e armazenado para ser acessado cada vez que necessário.

A partir do estado operacional das configurações de menor custo a cada estágio, obtem-se uma informação valiosa para eliminações na tabela e consequentemente no número de caminhos a serem pesquisados. Se uma unidade permanece em operação em todas as configurações de menor custo, obviamente pode-se eliminar da tabela configurações cujo estado operacional dessa unidade seja o contrário. E vice-versa, já que tais configurações não poderão fazer parte de nenhum caminho ótimo. A critério de ilustração, no exemplo 1, de um total inicial de 883 configurações foram eliminadas 53,22%, passando para apenas 403 configurações. Quando se opta por soluções sub-ótimas, pode-se obter maiores eliminações na tabela limitando o número de configurações possíveis a cada estágio.

Na expansão de um elemento $E_j = (n_j^{k-1}, c_j, i)$, localizada a configuração em T^k que tem o mesmo estado operacional de n_j^{k-1}, a lei de formação das configurações e sua consequente posição na tabela permitem que se faça a busca de seus possíveis sucessores apenas nas configurações de índices inferiores ao índice dessa configuração se a demanda aumenta do estágio k-1 para o estágio k e apenas nas configurações de índices superiores se a demanda diminui. Caso se permita ativar ou desativar unidades independentemente do comportamento da demanda ou não exista tal configuração, a busca é feita em todas as posições de T^k.

IV.4 - Subestimativas de Custos Futuros

A obtenção de informações que possam ser utilizadas pelos algoritmos é importante pois permite a simulação do comportamento inteligente de olhar a frente.

No Unit Commitment é possível obter informações a respeito de custos futuros a partir de um nó. As subestimativas de custos futuros obtidas a partir dessas informações servem tanto para auxiliar na escolha do elemento a ser expandido como para facilitar eliminações de elementos abertos que certamente não fariam parte de um caminho ótimo.

IV.4.1 - Cálculo da Subestimativa do Custo de Geração

A subestimativa do custo de geração futuro a partir de um dado estágio é obtida a partir da tabela descrita no ítem IV.3. Identificada a configuração de menor custo de cada estágio, para um determinado estágio k, é feita a somatória nos P-k estágios a frente, dos custos dessas configurações.

$$\hat{h}1^k = \Sigma_i \min CG_j^i \qquad \text{IV.1}$$

onde:

$i = k+1, P$

$j \in T^k$

IV.4.2 - Cálculo da Subestimativa do Custo de Aquecimento

A subestimativa do custo de aquecimento futuro para um dado nó $n_j = (e^k, h^k)$ é dependente do estado operacional do sistema, do número de estágios em que está nesse estado e do estágio de planejamento.

O procedimento usado para o seu cálculo consiste em verificar a necessidade de se ativar outras unidades até o final do planejamento e em que estágio isso será necessário. Haverá necessidade se o nó não satisfizer a restrição de Reserva para todos os estágios futuros. Assim é calculado para cada unidade fora de operação o seu custo de aquecimento por unidade de capacidade de geração (CA_i / $\overline{G}_i$). São escolhidas n unidades com menores CA_i / $\overline{G}_i$ para complementar a capacidade máxima de geração do nó de maneira a satisfazer a maior demanda futura. A subestimativa do custo de aquecimento é dada pela somatória dos custos de aquecimento das unidades escolhidas.

O custo de aquecimento de uma unidade é dependente do número de estágios em que a mesma esteve desativada. Essa variável poderá ser calculada de duas maneiras: uma considerando que a unidade será ativada no estágio seguinte e outra considerando que a unidade será ativada apenas no primeiro estágio para o qual a restrição de Reserva não é satisfeita.

O algoritmo descrito a seguir permite que seja obtida a subestimativa das duas formas mencionadas, sendo o parâmetro lógico "liga quando necessário" a variável que permite a opção.

A subestimativa do custo de aquecimento até o final do planejamento a partir de um estágio k, para um nó n_j^k será dada por $\hat{h}2_j^k$ e a média dessa subestimativa será dada por $\hat{h}3_j^k$.

Passo 0: Faça $\hat{h}2_j^k = 0$

$$\hat{h}3_j^k = 0$$

Se $Gmax_j^k \geq Dmax^k$ pare.

Passo 1: $C = O^{jk}$ (conjunto das unidades em operação do nó n_j^k)

Passo 2: Se liga quando necessário

então faça $t = \ell - k - 1$

senão faça $t = 0$

Passo 3: Para $i = 1, N$ tal que $i \notin O^{jk}$

faça $CA_i = a2_i \ (1 - b2_i \ e^{-c2_i \ (h_i^k + t)})$

$CAI_i = CA_i \ / \ \overline{G}_i$

Passo 4: Faça:

Escolha de m tal que $CAI_m = \min (CAI_i)$
para $i = 1, N$ tal que $i \notin C$

$C = C \ U \ \{m\}$

$Gmax_j^k = Gmax_j^k + \overline{G}_m$

$\hat{h}2_j^k = \hat{h}2_j^k + CA_m$

até que $Gmax_j^k \geq Dmax^k$

Passo 5: Faça $\hat{h}2_j^k = \hat{h}2_j^k - (Gmax_j^k - Dmax^k) \ CAI_m$

$\hat{h}3_j^k = \hat{h}2_j^k \ / \ (P - k + 1)$

Pare.

sendo:

$Gmax_j^k$ = capacidade máxima de geração do nó n_j^k

$Dmax^k$ = máxima demanda futura a partir do estágio k

ℓ = primeiro estágio em que haverá necessidade de se ativar unidades

IV.5 - Algoritmo A*

O algoritmo A*, exposto por Nilsson[21] utiliza uma estimativa $\hat{h}_j^k$ para definir um elemento "melhor" entre os abertos em cada iteração.

A um elemento listado $E_j = (n_j^k, c_j, i)$ associa-se um número real $\hat{f}1(E_j)$ que estima o menor custo possível de um caminho do nó inicial ao alvo, cujo trecho inicial é associado a E_j. O critério de prioridade $\hat{f}1(E_j)$ é interpretado como: o que se pode esperar expandindo E_j.

IV.5.1 - Definição do Índice de Prioridade

O índice de prioridade para o A* será da forma:

$$\hat{f}1(E_j) = c_j + \hat{h}_j^k \qquad \text{IV.2}$$

onde $\hat{h}_j^k$ é uma subestimativa do custo futuro da operação do sistema a partir do nó n_j^k e será composta por duas parcelas:

$$\hat{h}_j^k = \hat{h}1^k + \hat{h}3_j^k \qquad \text{IV.3}$$

onde:

- $\hat{h}1^k$ = subestimativa do custo de geração futuro (Será fixa por estágio, calculada apenas no início do algoritmo. Permite que haja equiparação de nós de camadas diferentes)
- $\hat{h}3_j^k$ = média da subestimativa do custo de aquecimento a partir do nó n_j^k (É dependente do nó n_j^k e portanto calculada para cada sucessor de um elemento em expansão. Permite diferenciar nós de uma mesma camada)

IV.5.2 - Regra de Parada

A forma como é definida $\hat{h}_j^k$ garante sua admissibilidade, isto é, $\hat{h}_j^k$ realmente subestima o caminho de menor custo de n_j^k a um nó

do alvo. Assim quando o elemento escolhido para ser expandido for um elemento cujo nó associado é um nó do alvo, a solução ótima terá sido encontrada, como demonstra GONZAGA[15].

O algoritmo também pára quando a lista de elementos for vazia ou quando o tempo de processamento atingir um limite predefinido.

IV.5.3 - Descrição do Algoritmo A*

Passo 0 : Defina o processamento limite.
Abra o elemento inicial $E_0 = (N^0, 0, 0)$.
Faça $\hat{f}1\ (E^0) = 0$.

Passo 1 : Se não existir elemento aberto, pare.
Escolha um aberto $E_j = (n_j^{k-1}, c_j, i)$ tal que $\hat{f}1\ (E_j)$ seja mínimo entre os abertos.
Se n_j^{k-1} for um nó do alvo, vá para o passo 5.
Se processamento $\geq$ processamento limite, pare.
Feche o elemento E_j.

Passo 2: A cada configuração em T^k que possua todas as decisões admissíveis e satisfaça à restrição de número máximo de unidades que são ativadas ou desativadas simultaneamente, associe um nó sucessor n_ℓ^k (ℓ = contador de sucessores) e faça:

$$c_\ell = c_j + cA_\ell^k + CG_\ell^k$$

$$E_\ell = (n_\ell^k, c_\ell, j)$$

$$\hat{f}1 = (E_\ell) = c_\ell + \hat{h}1^k + \hat{h}3_\ell^k$$

Passo 3 : Para cada elemento sucessor (E_ℓ), percorra a lista de elementos listados (E_m), verificando:

Se $n_\ell^k = n_m^k$ e

Se $c_\ell < c_m$, elimine E_m da lista e vá para o passo 4

Senão, abondone E_{ℓ}.

Passo 4 : Abra os Q elementos sucessores que tenham os menores $\hat{f}1\ (E_{\ell})$ e vá para o passo 1.

Passo 5 : Recupere e imprima o caminho ótimo associado a E_j e Pare.

IV.6 - Algoritmo A* adaptativo

Para grandes sistemas, o algoritmo A* pode ser muito oneroso computacionalmente, já que ao encontrar a solução ótima, terá pesquisado todos os trechos dos caminhos que tinham possibilidades de serem ótimos. Para sistemas onde os nós sucessores pouco diferem em termos de custos, será grande o esforço para a obtenção da solução ótima. O A* adaptativo é uma extensão do A*, onde soluções sub-ótimas são encontradas rápida e iterativamente até que a solução ótima tenha sido detectada. Inicialmente é obtida uma solução chamada incremental que é geralmente uma boa política de operação do sistema, tendo uma função importante que é a de limitante da busca de novas políticas, permitindo que se faça eliminações na lista aberta de elementos que certamente não levarão a uma política melhor que a já encontrada, diminuindo sensivelmente o número de expansões do algoritmo e consequentemente o tempo de processamento necessário.

Em princípio o critério de prioridade tem a mesma interpretação para os dois algoritmos: o que se pode esperar ao expandir E_j. A diferença está em que enquanto o A* utiliza a subestimativa do custo de geração $\hat{h}1^k$ (o mínimo que será gasto do estágio K+1 até o final do planejamento), o A* adaptativo utiliza uma superestimativa do custo de geração futuro $\overset{*}{h}1^k$ (o máximo que será gasto do estágio k+1 até o final do planejamento), que vai sendo redefinida a medida que novas soluções vão sendo encontradas.

IV.6.1 - Definição do Índice de Prioridade

O índice de prioridade para o A* adaptativo será inicialmente da forma:

$$\hat{f}2\ (E_j) = c_j + \hat{h}_j^{*k} \qquad \text{IV.4}$$

onde $\hat{h}_j^{*k}$ é uma superestimativa do custo futuro de operação do sistema a partir do nó n_j^k e será composta por duas parcelas:

$$\hat{h}_j^{*k} = \hat{h}1^{*k} + \hat{h}3_j^k \qquad \text{IV.5}$$

onde:

$\hat{h}1^{*k}$ = superestimativa do custo de geração futuro (Será fixa por estágio, calculada a cada nova solução encontrada. Permite que haja equiparação de nós de camadas diferentes)

IV.6.2 - Solução Incremental

Para a obtenção de uma solução inicial $\hat{h}1^{*k}$ é definido como:

$$\hat{h}1^{*k} = (P - k)\Delta \qquad \text{IV.6}$$

onde:

Δ = número muito grande

$k = 1, P$

Assim definido $\hat{h}1^{*k}$, os elementos sucessores de um elemento em expansão terão sempre os menores índices de prioridade da lista e consequentemente um deles será escolhido para a expansão seguinte. Dessa maneira, somente um elemento em cada camada é expandido, permitindo que se conheça rapidamente uma política de operação a ser usada como limitante da busca.

IV.6.3 - Redefinição do Índice de Prioridade

A cada novo caminho encontrado, a parcela $\hat{h}1^{*k}$ é redefinida, tornando-se mais fraca já que o limite máximo vai diminuindo a medida que converge para a solução ótima, ou seja:

$$\overset{*}{h1}^{k} = (\Sigma_i \; CG_m^i)\alpha$$

onde:

i = k+1, P

m = índices dos elementos pertencentes ao caminho associado ao elemento do alvo E_j

α = constante ≤ 1

IV.6.4 - Eliminações pelo Bound

O conhecimento de uma solução permite que se faça eliminações de elementos que certamente não fariam parte do caminho ótimo. Seja Bound o custo da melhor solução encontrada e

$$\hat{f}3 = c_j + \hat{h}1^k + \hat{h}2_j^k \qquad \text{IV.8}$$

então se $\hat{f}3 \geq$ Bound para um elemento E_j este pode ser eliminado da lista de elementos.

IV.6.5 - Políticas ε Ótimas

Pode ocorrer um esforço muito grande do algoritmo para retornar uma nova solução que em pouco melhora a já encontrada. Assim é definida a constante ε como sendo a mínimo diferença entre um custo já encontrado e o custo de uma nova solução. Seja c_j o custo da última solução, faz-se:

$$\text{Bound} = c_j - \varepsilon \qquad \text{IV.9}$$

IV.6.6. - Regra de Parada

O algoritmo pára quando a lista aberta for vazia ou quando o tempo de processamento atingir um limite predefinido.

IV.6.7 - Descrição do Algoritmo A* adaptativo

Passo 0 : Defina o processamento limite.
Abra o elemento inicial $E_o = (N^0, 0, 0)$.
Faça: Bound = $+\infty$
$\hat{f}1(E_0) = \hat{f}2(E_0) = \hat{f}3(E_0) = 0$.

Passo 1: Se não existir elemento aberto, pare.
Escolha um aberto $E_j = (n_j^{k-1}, c_j, i)$ tal que $\hat{f}2(E_j)$ seja mínimo entre os abertos.
Se n_j^{k-1} for um nó do alvo, vá para o passo 5.
Se processamento $\geq$ processamento limite, pare.
Feche o elemento.

Passo 2 : A cada configuração em T^k que possua todas as decisões admissíveis e satisfaça a restrição de número máximo de unidades que são ativadas ou desativadas simultaneamente, associe um nó sucessor n_ℓ^k (ℓ = contador de sucessores) e faça:

$$c = c_j + CA_\ell^k + CG_\ell^k$$

$$E_\ell = (n_\ell^k, c_\ell, j)$$

$$\hat{f}1(E_\ell) = c_\ell + \hat{h}1^k + \hat{h}3_j^k$$

$$\hat{f}2(E_\ell) = c_\ell + \overset{*}{h}1^k + \hat{h}3_j^k$$

$$\hat{f}3(E_\ell) = c_\ell + \hat{h}1^k + \hat{h}2_j^k$$

Passo 3 : Para cada elemento sucessor (E_ℓ):
Se $\hat{f}3(E_\ell) \geq$ Bound, abandone E_ℓ.
Compare para cada elemento listado E_m:
Se $n_\ell^k = n_m^k$ e
Se $c_\ell < c_m$, elimine E_m da lista e vá para o passo 4.
Senão, abandone E_ℓ.

Passo 4 : Abra os Q elementos sucessores que tenham os menores $\hat{f}1(E_\ell)$ e vá para o passo 1.

Passo 5 : Feche o elemento E_j.
Recupere e imprima o caminho associado a E_j.
Faça Bound = $c_j - \varepsilon$.
Elimine da lista todos os elementos E_m tal que $\hat{f}3\ (E_m) \geq$ Bound.
Redefina $\overset{*}{h}1^k$ para todo estágio k.
Atualize $\hat{f}2\ (E_m)$ para todo E_m aberto e vá para o passo 1.

V - O PROGRAMA

O programa foi escrito na linguagem ALGOL para o computador B6700 do NCE/UFRJ, que conta com 2,4 Mbytes de memória central e trabalha com um sistema de memória virtual e tempo partilhado.

V.1 - Programa Principal

O programa principal coordena o fluxo do processamento e comanda a análise pelo algoritmo escolhido. O seu fluxograma pode ser visto nas figuras V.1 a V.3.

V.2 - Rotinas de Entrada de Dados

LEREIMPRIMIRDADOSDOSISTEMA: le e imprime os dados referentes ao sistema termoelétrico.

LEREIMPRIMIRDADOSGERAIS: le e imprime os dados referentes ao modelo e ao algoritmo utilizado.

V.3 - Rotinas de Análise

V.3.1 - Rotinas do Despacho

CRIARTABELA: cria uma tabela (que é gravada em disco) contendo informações sobre todos os estados operacionais viáveis do sistema para cada estágio (geração ótima de cada estado operacional e respectivos custos, capacidade máxima de geração e maior unidade em operação). Também gera a subestimativa do custo total de geração.

MONTARPRECOS: monta o vetor que contém os preços de transição de cada unidade assim como os preços mínimos e máximos.

ORDENARPRECOS: ordena o vetor de preços em ordem crescente.

EXPANDIR: cria recursivamente, para um determinado estágio, todos os estados operacionais viáveis do sistema.

CALCULARCUSTODODESPACHO: para um determinado estágio e um determinado estado operacional, obtem a geração ótima do sistema, assim como o custo ótimo.

PROCURARPRECOOTIMO: cria recursivamente a árvore de soluções para o Despacho.

CALCULARPRECOOTIMO: calcula o preço ótimo, os níveis de geração de cada unidade para esse preço, bem como o custo ótimo.

FORCARGERARMINIMO: força determinada unidade a gerar refazendo o vetor de geração.

FORCARGERARZERO: força determinada unidade a não gerar refazendo o vetor de geração.

V.3.2 - Rotinas Auxiliares

CALCULARSUPESTIMATIVADOESTAGIO: calcula a superestimativa usada pelo algoritmo A* adaptativo na busca incremental.

CALCULARDEMANDAMAXIMAFUTURA: fornece a cada estágio a demanda máxima até o final do planejamento.

VERIFICARVARIACAODADEMANDA: associa a cada estágio um código indicando a variação da demanda.

INICIALIZARLISTADEELEMENTOS: faz as inicializações nos vetores utilizados na execução dos algoritmos.

FECHARELEMENTO: busca na tabela os possíveis sucessores de um elemento em expansão e testa para cada um sua viabilidade em relação as restrições.

CALCULARCUSTOTOTAL: calcula o custo de ligação e os índices de prioridade de um elemento sucessor.

CALCULARSUBESTIMATIVADOESTADO: calcula a subestimativa de um elemento dependente do estado.

NAOELIMINARNOVOELEMENTO: compara o novo elemento com os outros existentes na lista de elementos fazendo as eliminações necessárias.

ALOCARNOVOINDICE: aloca um indice para um elemento a ser colocado na lista de elementos.

LIBERARINDICE: libera um indice de um elemento eliminado da lista de elementos.

SELECIONARSUCESSORES: seleciona sucessores de um elemento.

ORDENARSUCESSORES: ordena os sucessores de um elemento em ordem crescente de custo.

DESCARREGARNOVOELEMENTO: coloca um novo elemento na lista de elementos.

RECUPERARCAMINHO: recupera o caminho associado a um elemento do alvo.

ELIMINARELEMENTOSPELOBOUND: elimina da lista de elementos os que tiverem custo superior ao Bound.

REDEFINIRESTIMATIVASUPERIOR: refaz a superestimativa dependente do estágio.

V.3.3 - Rotinas dos Algoritmos

ESCOLHERELEMENTOPORFMINIMO: escolhe da lista de elementos abertos um elemento para ser expandido de acordo com o algoritmo A*.

ESCOLHERELEMENTOPORFMAXIMO: escolhe da lista de elementos abertos um elemento para ser expandido de acordo com o algoritmo A* adaptativo.

V.4 - Rotinas de Saida de Dados

IMPRIMIRMENSAGEM: imprime mensagens de ocorrências durante o processamento.

IMPRIMIRSUBESTIMATIVADOESTAGIO: imprime as subestimativas dos estágios.

IMPRIMIRESTIMATIVASUPERIOR: imprime as superestimativas dos estágios.

IMPRIMIRVARIACAODADEMANDA: imprime os códigos de variação da demanda para os estágios.

IMPRIMIRCAMINHO: imprime os dados para o planejamento de um caminho ótimo.

IMPRIMIRELEMENTO: imprime os dados de um elemento.

IMPRIMIRTABELA: imprime a tabela que contem as informações sobre todos os estados operacionais viáveis do sistema para cada estágio.

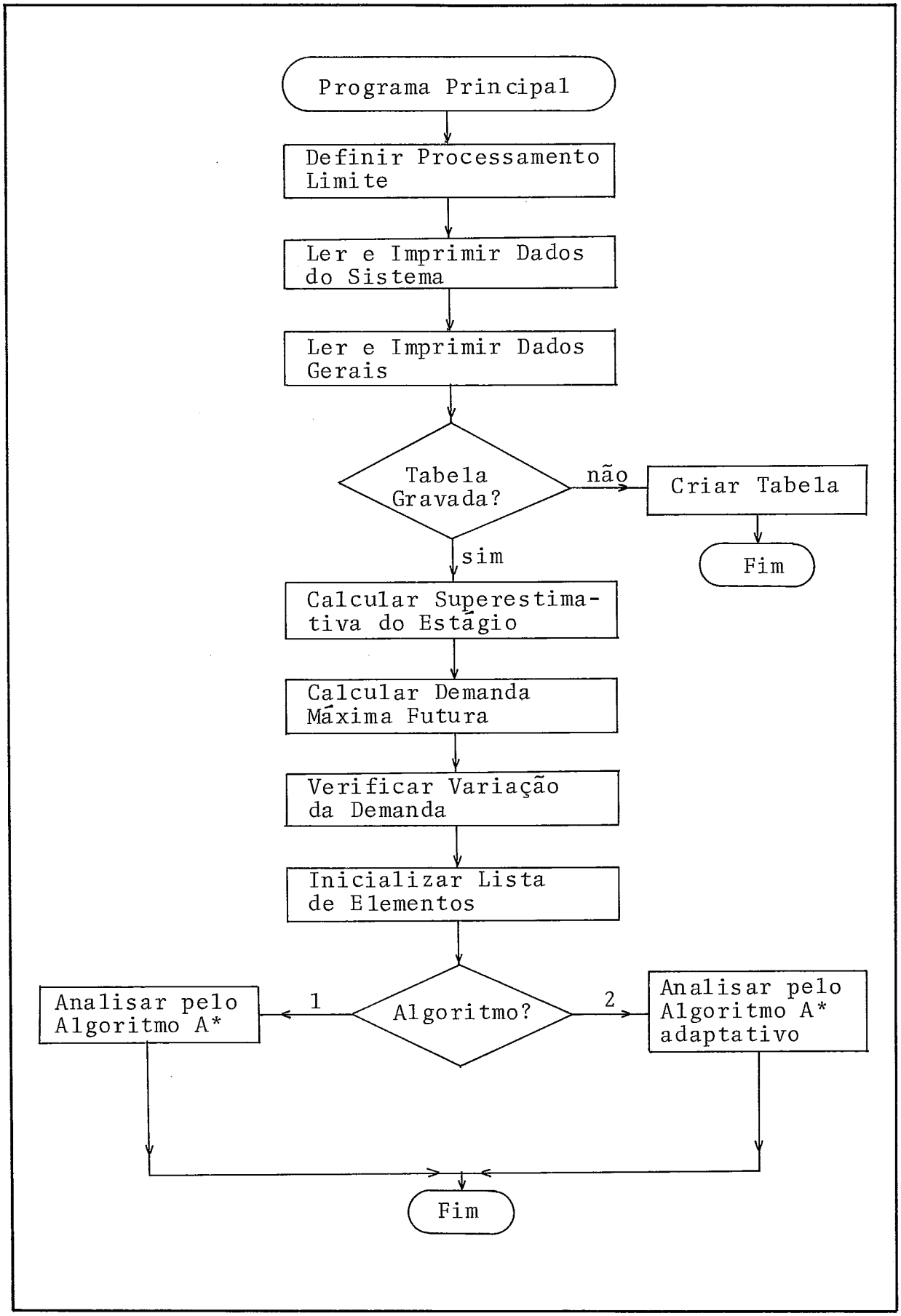

FIGURA V.1 - Fluxograma do Programa Principal.

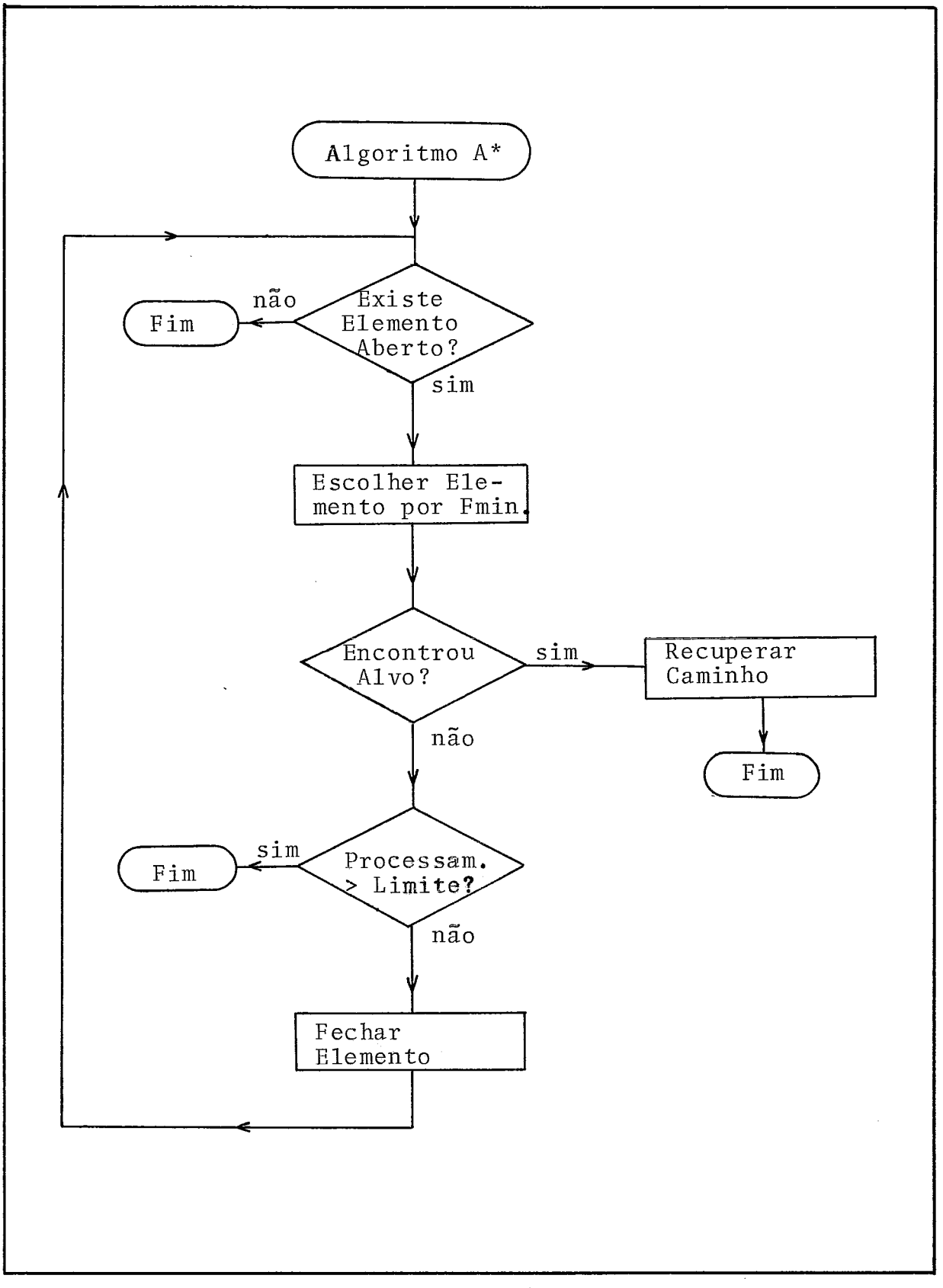

FIGURA V.2 - Algoritmo A*.

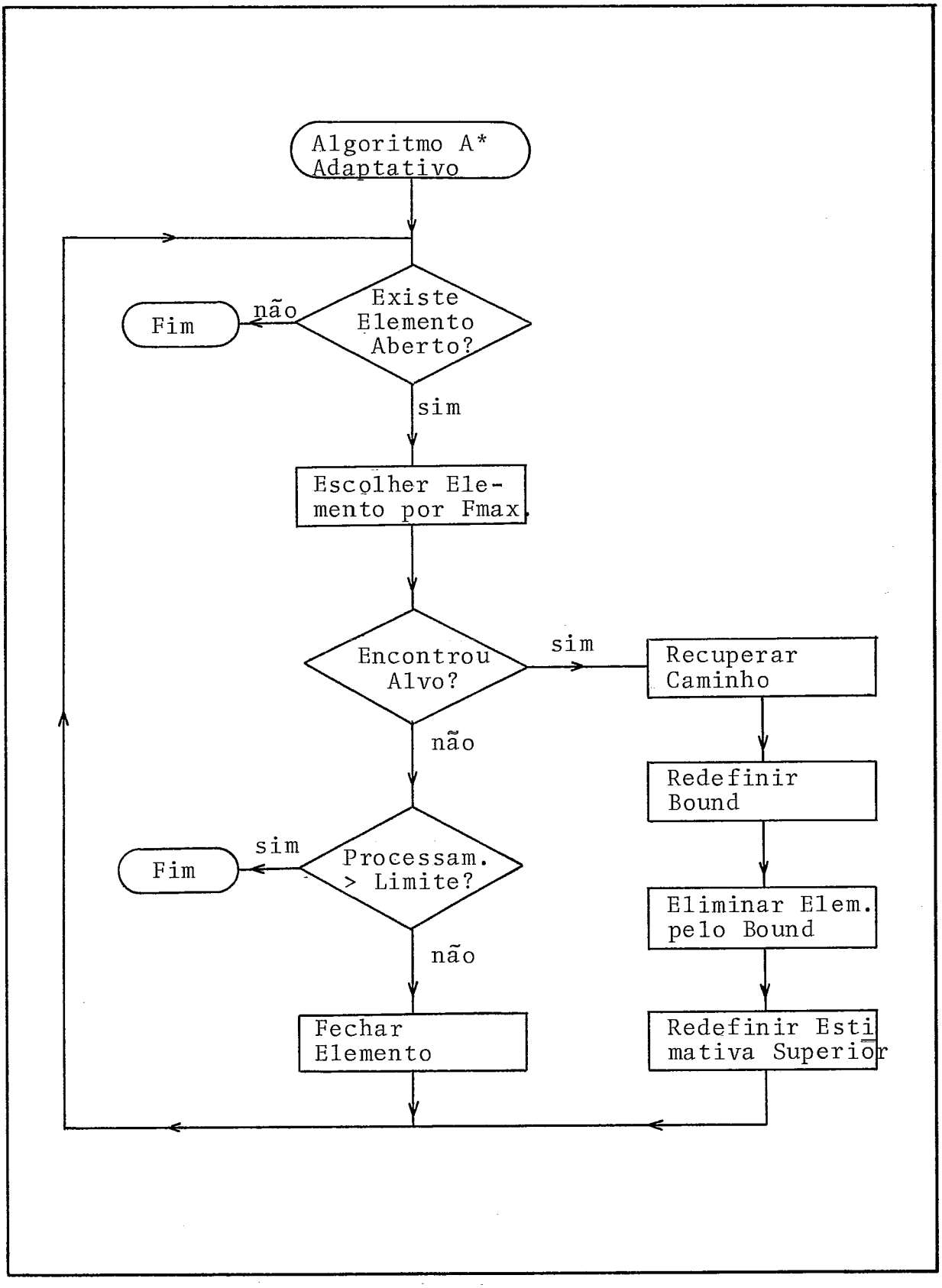

FIGURA V.3 - Algoritmo A* adaptativo.

VI - RESULTADOS

São apresentados os resultados obtidos pela aplicação dos algoritmos de Busca de Caminhos em Grafos ao planejamento da operação de dois sistemas termoelétricos. Algumas considerações são feitas à respeito do desempenho dos algoritmos e das dificuldade encontradas.

VI.1 - Exemplo 1

O planejamento da operação de um sistema composto por 4 unidades termoelétricas é feito ao longo de 10 períodos de uma hora de duração. A demanda horária é dada pela tabela VI.1, os parâmetros das unidades estão na tabela VI.2 e os dados utilizados pelos algoritmos nos diversos casos estudados são apresentados pela tabela VI.3.

As soluções encontradas são mostradas pela tabela VI.4, onde é possível observar a superioridade do algoritmo A* adaptativo sobre o algoritmo A* com relação ao número de expansões de elementos necessários para encontrar a solução ótima. Para o caso 1 houve necessidade de menos expansões que para o caso 2 devido a consideração da restrição Comportamento da Demanda, que limita o número de sucessores de um nó. Para o caso 3 a subestimativa do custo de aquecimento de um nó é calculada sob a hipótese de que uma unidade será ativada no período seguinte, ao contrário dos outros dois casos onde a hipótese é de que uma unidade será ativada apenas no primeiro período para o qual o nó não satisfaz à restrição de Reserva. Isto faz com que a subestimativa se torne mais "fraca" e portanto menos operante, tornando necessário um maior número de expansões para encontrar a solução ótima. A tabela VI.5 fornece as gerações individuais para as soluções encontradas.

TABELA VI.1 - Demanda Horária para o Exemplo 1.

estágio	1	2	3	4	5	6	7	8	9	10
demanda	203	158	105	150	105	98	105	150	158	203

TABELA VI.2 - Parâmetros das Unidades do Exemplo 1.

Unidade	$\underline{G}$	$\overline{G}$	C0	a1 a2	b1 b2	c1 c2
1	15	45	29	0.046586 295.100000	7.1030 1.0000	43.900000 0.025822
2	20	53	32	0.023020 88.900000	6.8356 1.0000	42.016000 0.453550
3	30	105	51	0.012167 103.700000	5.4530 1.0000	68.552000 0.191774
4	35	167	61	0.005206 617.800000	5.6787 1.0000	89.010000 0.025642

TABELA VI.3 - Dados para os Casos Estudados no Exemplo 1.

Dado	Caso 1	Caso 3
e_i^0 , i = 1, 4	1	1
h_i^0 , i = 1, 4	5	5
f_i , i = 1, 4	2	2
q_i , i = 1, 4	5	5
R	3	3
S	3	3
Liga quando necessário	sim	não
Q	5	5
ε	20	20
α	.995	.995

Obs.: O caso 2 é diferente do caso 1 apenas pela não consideração da restrição Comportamento da Demanda.

Para o caso 3 a restrição Comportamento da Demanda também não é considerada.

TABELA VI.4 - Soluções para o Exemplo 1.

Caso	A*			A* adaptativo		
	Custo	Proces. (s)	Num. de Expans.	Custo	Proces. (s)	Num. de Expans.
1	10825	5.13	59	10880 10825	3.42 5.10	42
2	10825	6.44	63	10880 10825	3.78 5.90	46
3	10825	5.81	65	10880 10825	3.46 5.60	51

TABELA VI.5 - Gerações Individuais (Soluções do Exemplo 1).

k	Custo 10825				Custo 10880			
	G_1^k	G_2^k	G_3^k	G_4^k	G_1^k	G_2^k	G_3^k	G_4^k
1	0	0	67	136	0	0	67	136
2	-	0	54	104	-	0	54	104
3	-	0	38	67	-	-	38	67
4	-	0	51	99	0	-	51	99
5	-	-	38	67	0	-	38	67
6	-	-	36	62	0	-	36	62
7	-	-	38	67	0	-	38	67
8	-	0	51	99	0	-	51	99
9	-	0	54	104	0	0	54	104
10	0	0	67	136	0	0	67	136

Obs.: "-" indica unidade desativada.
"0" indica unidade ativada com geração nula.

VI.2 - Exemplo 2

É feito o planejamento da operação de um sistema composto por 10 unidades termoelétricas ao longo de um dia discretizado em horas[12]. Os parâmetros das unidades são dados na tabela VI.6 e a demanda na tabela VI.7. A tabela VI.8 apresenta os dados utilizados pelos algoritmos nos diversos casos estudados.

Para esse exemplo não foi possível obter a solução ótima através do algoritmo A*, devido ao elevado tempo de processamento requerido. A tabela VI.9 fornece as soluções obtidas através do algoritmo A* adaptativo para os casos estudados. As tabelas VI.10 a VI.12 fornecem as gerações individuais para as soluções encontradas.

Os resultados para o caso 1 são bastante satisfatórios visto ser a solução ótima detectada em apenas 11.35 s de processamento.

Para o caso 2 o número de períodos em que uma unidade deve permanecer no estado de operação é aumentado em função da solução ótima já encontrada. Ao contrário do esperado isso requereu praticamente o dobro do processamento do caso 1, o que pode ser explicado pelo fato de que a solução incremental do caso 1 não se tornou viável e outra solução incremental de custo maior foi encontrada levando à eliminações menos eficazes e consequentemente à um maior número de expansões para encontrar a solução ótima.

O caso 3 libera o algoritmo das seguintes restrições: Tempo Mínimo no Estado de Operação, Número Máximo de Unidades que são Ativadas ou Desativadas Simultaneamente e Comportamento da Demanda; o que torna o número de sucessores consideravelmente maior. Apesar disso o mau desempenho do algoritmo para esse caso se deve à mesma razão que a do caso 3 do exemplo 1.

TABELA VI.6 - Parâmetros das Unidades do Exemplo 2.

Unidade	$\underline{G}$	$\overline{G}$	C0	a1 a2	b1 b2	c1 c2
1	15	60	35.020	0.00510 85.00000	1.40000 0.58800	50.02000 0.20000
2	20	80	41.006	0.00396 101.00000	1.50000 0.59400	66.00600 0.20000
3	30	100	49.020	0.00393 114.00000	1.35000 0.57000	89.02000 0.20000
4	25	120	32.900	0.00382 94.00000	1.40000 0.65000	64.90000 0.18000
5	50	150	40.793	0.00212 113.00000	1.54000 0.63900	69.79300 0.18000
6	75	280	76.032	0.00261 176.00000	1.35000 0.56800	148.03200 0.15000
7	250	520	67.017	0.00127 267.00000	1.39540 0.74900	172.01700 0.09000
8	50	150	70.782	0.00135 282.00000	1.32850 0.74900	170.78200 0.09000
9	120	320	71.621	0.00289 187.00000	1.26430 0.61700	120.62100 0.13000
10	75	200	81.493	0.00148 227.00000	1.21360 0.64100	163.49300 0.11000

TABELA VI.7 - Demanda Horária para o Exemplo 2.

Estágio	1	2	3	4	5	6	7	8
Demanda	1459	1372	1299	1285	1271	1314	1372	1314
Estágio	9	10	11	12	13	14	15	16
Demanda	1271	1242	1197	1182	1154	1138	1124	1095
Estágio	17	18	19	20	21	22	23	24
Demanda	1066	1037	993	978	963	1022	1081	1459

TABELA VI.8 - Dados para os Casos Estudados no Exemplo 2.

Dado	Caso 1	Caso 2	Caso 3
e_i^0 , i = 1, 10	1	1	1
h_i^0 , i = 1, 10	5	9	5
f_i , i = 1, 10	2	4	1
q_i , i = 1, 10	5	10	1
R	3	3	9
S	1	1	9
Liga quando necessário	sim	sim	não
Q	7	7	7
ε	30	30	300
α	1	1	1

Obs. : Para o caso 3 a restrição Comportamento da Demanda não é considerada.

TABELA VI.9 - Soluções para o Exemplo 2 (Algoritmo A* adaptativo).

Caso	Custo	Tempo (s)	Num. de expansões
1	71135	7.25	680
	71103	11.35	
2	71280	5.96	822
	71103	22.38	
3	71558	15.00	641

TABELA VI.10 - Gerações Individuais (Solução de Custo 71103).

k	G_1^k	G_2^k	G_3^k	G_4^k	G_5^k	G_6^k	G_7^k	G_8^k	G_9^k	G_{10}^k
1	60	80	100	101	149	157	305	150	157	200
2	60	--	100	100	147	156	303	150	156	200
3	--	--	100	99	145	154	298	150	154	200
4	--	--	100	97	142	151	293	150	152	200
5	--	--	100	120	150	0	367	150	184	200
6	--	--	100	100	148	156	304	150	156	200
7	60	--	100	100	147	156	303	150	156	200
8	60	--	100	118	150	0	357	150	179	200
9	60	--	--	100	147	156	302	150	156	200
10	60	--	--	97	141	151	292	150	151	200
11	60	--	--	115	150	0	347	150	175	200
12	60	--	--	112	150	0	339	150	171	200
13	60	--	--	107	150	0	323	150	165	200
14	60	--	--	104	150	0	314	150	161	200
15	60	--	--	100	148	156	304	--	156	200
16	60	--	--	97	142	151	293	--	152	200
17	--	--	--	101	148	157	304	--	156	200
18	--	--	--	97	142	152	294	--	152	200
19	--	--	--	116	150	0	350	--	177	200
20	--	--	--	113	150	0	342	--	173	200
21	--	--	--	110	150	0	333	--	169	200
22	--	--	--	95	139	149	289	--	150	200
23	60	--	--	120	150	0	367	--	184	200
24	60	80	100	101	149	157	305	150	157	200

Obs.: "--" indica unidade desativada.
"0" indica unidade ativada com geração nula.

TABELA VI.11 - Gerações Individuais (Solução de Custo 71135).

k	G_1^k	G_2^k	G_3^k	G_4^k	G_5^k	G_6^k	G_7^k	G_8^k	G_9^k	G_{10}^k
1	60	80	100	101	149	157	305	150	157	200
2	60	--	100	100	147	156	303	150	156	200
3	--	--	100	99	145	154	298	150	154	200
4	--	--	100	97	142	151	293	150	152	200
5	--	--	100	120	150	0	367	150	184	200
6	--	--	100	100	148	156	304	150	156	200
7	60	--	100	100	147	156	303	150	156	200
8	60	--	100	118	150	0	357	150	179	200
9	60	--	--	100	147	156	302	150	156	200
10	60	--	--	97	141	151	292	150	151	200
11	60	--	--	115	150	0	347	150	175	200
12	--	--	--	97	141	151	292	150	151	200
13	--	--	--	118	150	0	357	150	179	200
14	--	--	--	115	150	0	348	150	175	200
15	--	--	--	112	150	0	340	150	172	200
16	--	--	--	107	150	0	323	150	165	200
17	--	--	--	101	148	157	304	--	156	200
18	--	--	--	97	142	152	294	--	152	200
19	--	--	--	116	150	0	350	--	177	200
20	--	--	--	113	150	0	342	--	173	200
21	--	--	--	110	150	0	333	--	169	200
22	--	--	--	95	139	149	289	--	150	200
23	60	--	--	120	150	0	367	--	184	200
24	60	80	100	101	149	157	305	150	157	200

Obs.: "--" indica unidade desativada.
"0" indica unidade ativada com geração nula.

TABELA VI.12 - Gerações Individuais (Solução de Custo 71280).

k	G_1^k	G_2^k	G_3^k	G_4^k	G_5^k	G_6^k	G_7^k	G_8^k	G_9^k	G_{10}^k
1	60	80	100	101	149	157	305	150	157	200
2	60	--	100	100	147	156	303	150	156	200
3	--	--	100	99	145	154	298	150	154	200
4	--	--	100	97	142	151	293	150	152	200
5	--	--	100	120	150	0	367	150	184	200
6	--	--	100	100	148	156	304	150	156	200
7	60	--	100	100	147	156	303	150	156	200
8	60	--	100	118	150	0	357	150	179	200
9	60	--	--	100	147	156	302	150	156	200
10	60	--	--	97	141	151	292	150	151	200
11	60	--	--	115	150	0	347	150	175	200
12	60	--	--	112	150	0	339	150	171	200
13	60	--	--	--	143	153	296	150	153	200
14	60	--	--	--	139	150	289	150	150	200
15	60	--	--	--	136	147	284	150	147	200
16	60	--	--	--	150	0	356	150	179	200
17	60	--	--	--	150	0	336	150	170	200
18	60	--	--	--	150	0	316	150	161	200
19	60	--	--	--	140	150	291	--	151	200
20	60	--	--	--	137	148	285	--	148	200
21	60	--	--	--	150	0	368	--	185	200
22	60	--	--	110	150	0	333	--	169	200
23	60	--	--	120	150	0	367	--	184	200
24	60	80	100	101	149	157	305	150	157	200

Obs.: "--" indica unidade desativada.
"0" indica unidade ativada com geração nula.

VI.3 - Considerações

É possível estabelecer uma relação entre os resultados obtidos para o exemplo 2 e os obtidos por Turgeon[12], que utilizando Branch and Bound com alguns critérios de poda obteve uma solução ótima de custo 71126. As restrições utilizadas por Turgeon são as mesmas do caso 1 com uma restrição adicional: uma unidade só pode ser ativada uma vez ao dia; explicando assim a solução ótima de custo maior.

Ao longo da pesquisa pode ser observado que alterações nas subestimativas de custos futuros afetam consideravelmente o desempenho do algoritmo A* adaptativo. Assim as definições das estimativas devem ser bem elaboradas. No entanto, um afã excessivo em tornar as subestimativas mais "fortes", isto é, mais próximas dos custos mínimos realmente verificados, pode levar a tempos de processamento altos sem a devida correspondência no desempenho do algoritmo.

Na definição das subestimativas outro cuidado a ser tomado é de que essa definição seja coerente com a função da subestimativa no índice de prioridade. Por exemplo, a função da parcela $\hat{h}3_j^k$ no índice de prioridade é diferenciar elementos de uma mesma camada, por isso ao invés de ser utilizada a subestimativa do custo de aquecimento total até o final do planejamento é utilizada a média dessa subestimativa.

VII - CONCLUSÕES

O desempenho do algoritmo A* adaptativo para os casos estudados permite concluir que a técnica de Busca de Caminhos em Grafos é extremamente conveniente para estudar o planejamento diário da operação de sistemas termoelétricos. Com essa técnica as restrições do problema são facilmente manipuladas, sendo assim muito cômodo verificar a influência de determinados parâmetros do modelo.

Para o Brasil é necessário que se faça um estudo no sentido de se incorporar unidades hidroelétricas ao sistema para se verificar a aplicabilidade do método.

BIBLIOGRAFIA

1 - CHANDLER, W. G., DANDENO, P. L., GLIMM, A. F. and KIRCHMAYER, L. K. - Short-Range Economic Operation of a Combined Thermal and Hidroeletric Power System, AIEE Trans. Power App. Syst., Paper 53 - 326 (1953).

2 - BALDWIN, C. J., DALE, K. M. and DITTRICH, R. F. - A Study of the Economic Shutdown of Generating Units in Daily Dispatch, IEEE Trans. Power App. Syst., Vol. PAS-78, pp.1272-1284 (1959).

3 - GARVER, L. L. - Power Generation Scheduling by Integer Programming - Development of Theory, IEEE Trans. Power App. Syst., vol. PAS-82, pp.730-735 (1963).

4 - LOWERY, P. G. - Generating Unit Commitment by Dynamic Programming, IEEE Trans. Power App. Syst., Vol. PAS-85, pp. 422-426 (1966).

5 - KERR, R. H., SCHEIDT, J. L., FONTANA, A. J. and WILEY, J. K. - Unit Commitment , IEEE Trans. Power App. Syst., vol.PAS-85, pp.417-425 (1966).

6 - MUCKSTADT, J.A. and Wilson R. C. - An aplication of Mixed Integer Programming Duality to the Scheduling Thermal Generating Systems, IEEE Trans. Power App. Syst., vol. PAS-87, pp.1968-1978 (1968).

7 - GUY, J.D. - Security Constrained Unit Commitment, IEEE Trans. Power App. Syst., vol. PAS-90, pp.1385-1390 (1971).

8 - AYOUB; A. K. and PATTON, A. D. - Optimal Thermal Generating Unit Commitment, IEEE Trans. Power App. Syst., vol.PAS-90, pp.1752-1756 (1971).

9 - MUCKSTADT, J.A. and KOENIG, S. A. - An Aplication of Lagragian Relaxation to Scheduling in Power Generation Systems, Operations Research, vol.25,pp.387-403 (1977).

10 - TURGEON, A. - Optimal Power Dispatch - Trabalho não publicado, QUEBEC, Canadá (1977).

11 - TURGEON, A. - Optimal Unit Commitment, IEEE Trans. Automat. Contr. vol. AC-22, pp. 223-227 (1977).

12 - TURGEON, A. - Optimal Scheduling of Thermal Generating - Units, IEEE Trans. Automat. Contr., vol. AC-23, pp.1000-1005 (1978).

13 - DILLON, T.S., EDWIN, K. W., KOCKS, H. D. and TAUD, R. J. - Integer Programming Approach to the Problem of Optimal Unit Commitment with Probabilistic Reserve Determination, IEEE Trans. Power App. Syst., vol. PAS-97, (1978).

14 - YAMAYEE, A., EL-ABIAD, H. and MORIN L. - Optimal and Near Optimal Unit Commitment Scheduling, Operations Research, pp. 517-529 (1979).

15 - GONZAGA, C. C. - Busca de Caminhos em Grafos
Anais da Ia. Reunião de Matemática Aplicada, IBM, Rio de Janeiro (1978).

16 - HAPP, H. H. - Optimal Power Dispatch - A comprehensive Survey, IEEE Trans. Power App. Syst., vol PAS-96, pp.841-854 (1977).

17 - HAPP, H. H., JOHNSON, R. C. and WRIGHT, W. J. - Large Scale Hydro-Thermal Unit Commitment - Method and Results, IEEE Trans. Power App. Syst., vol. PAS-90, pp.1373-1384 (1971).

18 - GRUHL, J., SCHWEPPE, F. and RUANE, M. - Unit Commitment Scheduling of Eletric Power Systems, MIT Energy Lab., Cambridge, Mass. (1976).

19 - YAMASHIRO, S. - Scheduling of Start-Up and Shut-Down of Thermal Generating Units Considering Reliability, Electrical Engineering in Japan, vol. 94, pp. 87-94 (1974).

20 - GEOFRION, A. M. - Duality in Nonlinear Programming: A Simplified Applications Oriented Development, SIAM Review, vol.13, nº 1 (1971).

21 - NILSSON, N. - Problem-Solving Methods in Artificial Intelligence, McGraw-Hill (1971).

APÊNDICE A - O Despacho

A.1 - Formulação

Dadas as unidades que estarão em operação num determinado estágio e a demanda neste estágio, tem-se o problema do Despacho , que consiste em minimizar o custo de geração atendendo a demanda.

As unidades podem gerar dentro de um intervalo limitado pelas gerações mínima e máxima de cada unidade. O custo de geração é dado em função do nível de geração conforme pode ser visto na figura A.1. Não gerar acarreta um custo fixo, de manutenção da unidade aquecida.

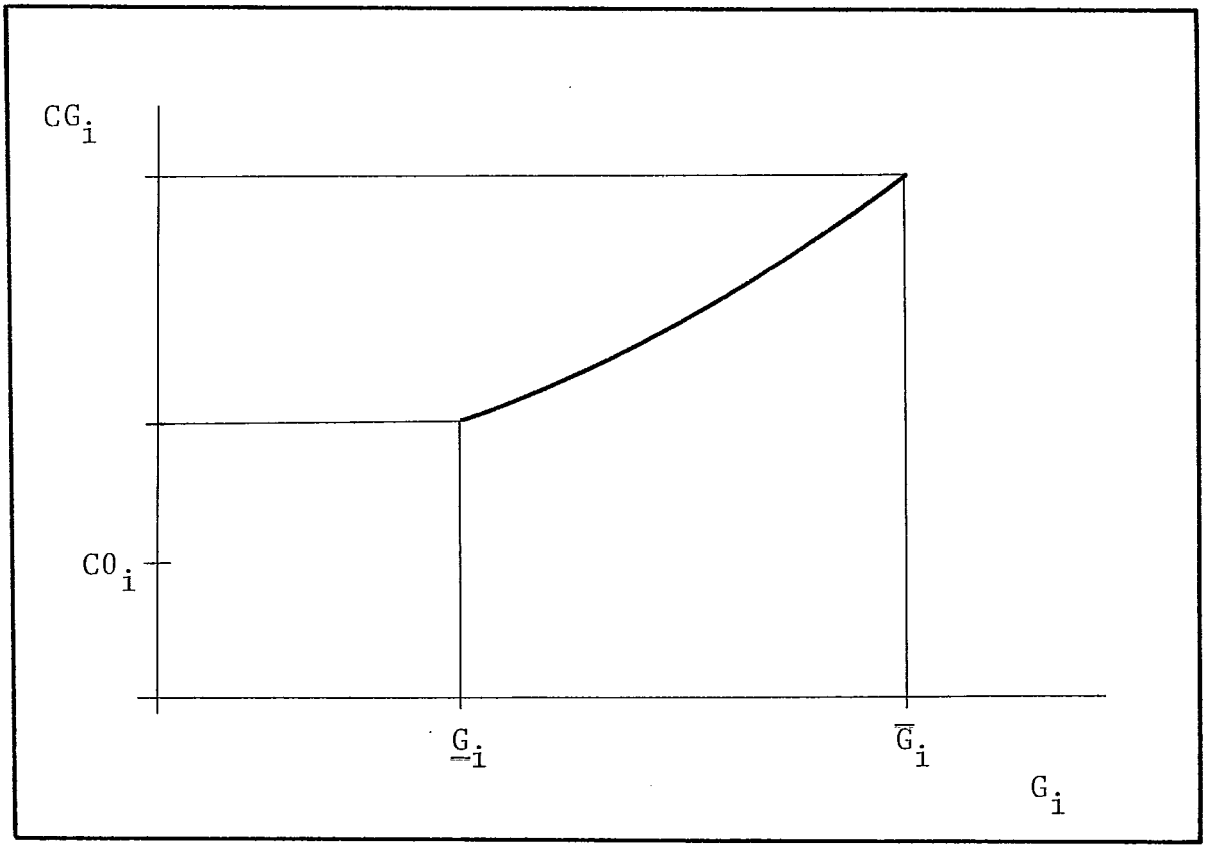

FIGURA A.1 - Exemplo de função custo de geração.

A função custo de geração é uma função quadrática, sendo dada por:

$$CG_i = a_i (G_i)^2 + b_i G_i + c_i \quad \text{se } \underline{G}_i \leq G_i \leq \overline{G}_i$$

$$= CO_i \quad \text{se } G_i = 0 \qquad \text{A.1}$$

onde:

CG_i = custo de geração da unidade i

G_i = nível de geração da unidade i

$\underline{G}_i$ = nível mínimo de geração da unidade i

$\overline{G}_i$ = nível máximo de geração da unidade i

CO_i = custo de manutenção da unidade i aquecida

a_i, b_i, c_i = parâmetros da função custo de geração da unidade i

$i \in O$ = conjunto das unidades em operação

O problema pode ser representado pelo seguinte modelo matemático:

Minimizar $\Sigma_i CG_i$

sujeito a:

$$\Sigma_i G_i = D \qquad \text{A.2}$$

onde:

$G_i \in S_i$

$S_i = \{0\} \cup [\underline{G}_i, \overline{G}_i]$

D = demanda no estágio considerado

$i \in O$ (conjunto das unidades em operação)

E o seu dual será dado por:

$$\text{Maximizar } \{\min (\Sigma_i \, CG_i) + \lambda \, (\Sigma_i \, G_i - D)\} \qquad A.3$$

sendo:

$\lambda \geq 0$ (variável dual que pode ser interpretada como o preço da energia)

A.2 - Resolução

TURGEON[10], desenvolveu um algoritmo simples para a determinação da alocação ótima da demanda entre as unidades em operação, o qual será exposto a seguir.

Existirá para cada unidade um preço de transição a partir do qual se torna vantajoso gerar energia. Esse preço pode ser obtido através da equação de prejuízo:

$$CG_i - \lambda_i \, G_i \leq CO_i \qquad A.4$$

sendo:

λ_i = preço da energia para a unidade i

que pode ser interpretada da seguinte forma: o que for gasto menos o que se receber deve ser menor ou igual ao prejuízo já existente. Ou seja, será vantajoso gerar quando o prejuízo for igual ou inferior ao prejuízo de não gerar (sendo o custo de manutenção da unidade aquecida assumido como prejuízo).

Considerando que o preço da energia pode ser dado pela derivada da função custo de geração em relação ao nível de geração, ou seja:

$$\lambda_i = 2 \, a_i \, G_i + b_i \qquad A.5$$

Pela substituição de A.5 em A.4, obtem-se:

$$(G_i)^2 \geq (c_i - CO_i) \, / \, a_i \qquad A.6$$

Para a igualdade da inequação A.6 tem-se a geração de transição que corresponde ao nível de geração a partir do qual é vantajoso gerar:

$$GT_i = \sqrt{((c_i - C0_i) / a_i)} \quad \text{se } c_i \geq C0_i$$

$$= \underline{G}_i \qquad \text{senão} \qquad \text{A.7}$$

O preço de transição pode agora ser obtido a partir da geração de transição da unidade, ou seja:

$$\lambda t_i = 2\, a_i\, GT_i + b_i \qquad \text{A.8}$$

Geometricamente o preço de transição pode ser interpretado como o coeficiente angular da reta que passa por $C0_i$ e tangencia a curva de custo de geração (ver figura A.2).

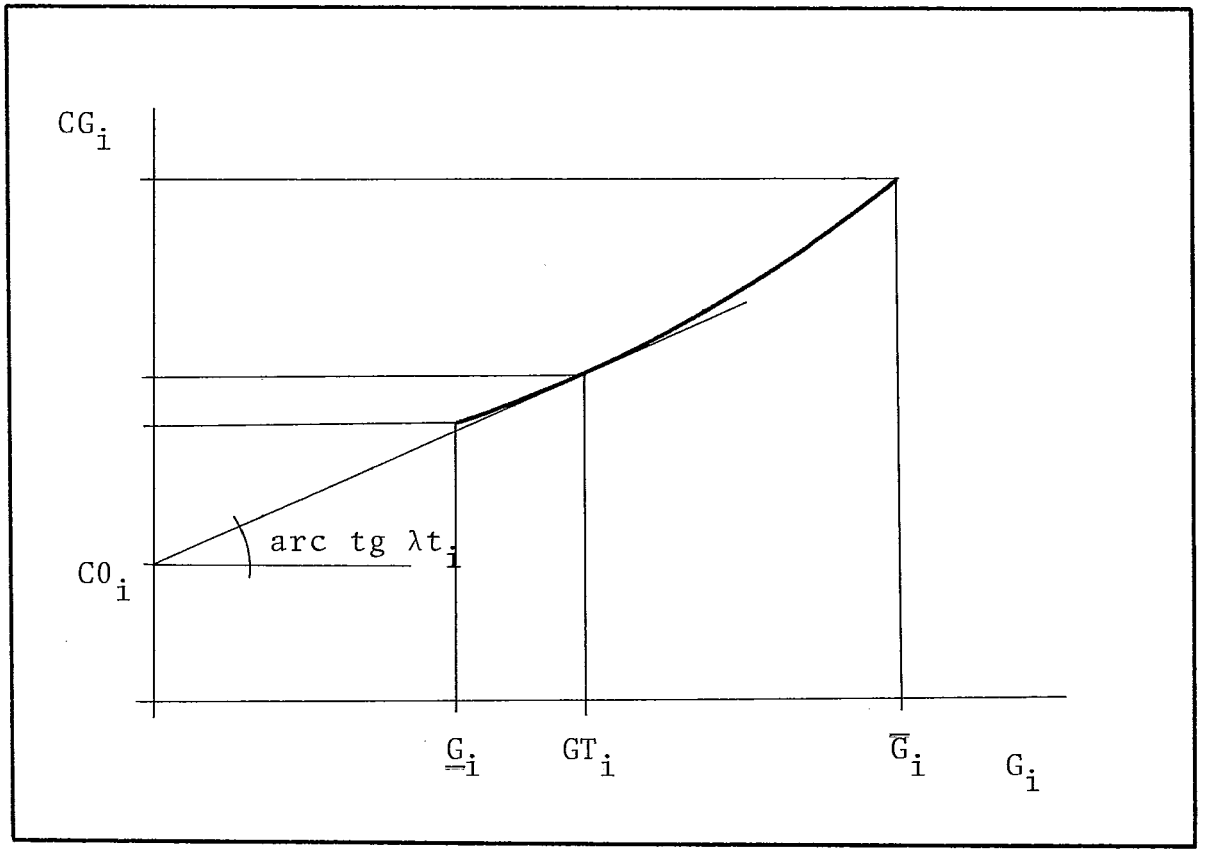

FIGURA A.2 - Representação geométrica do preço de transição.

Assim cada unidade terá um preço mínimo (preço de transição) a partir do qual será vantajoso gerar energia, o que se dará dentro de um intervalo: desde a geração de transição até a geração máxima dessa unidade. A geração ótima de cada unidade se situará dentro desse intervalo.

Como o nível de geração de uma unidade pode ser expresso em função do preço da energia através da seguinte fórmula:

$$G_i = (\lambda_i - b_i) / (2\ a_i) \qquad A.9$$

então para cada unidade pode ser construida uma curva geração em função do preço da energia (conforme figura A.3).

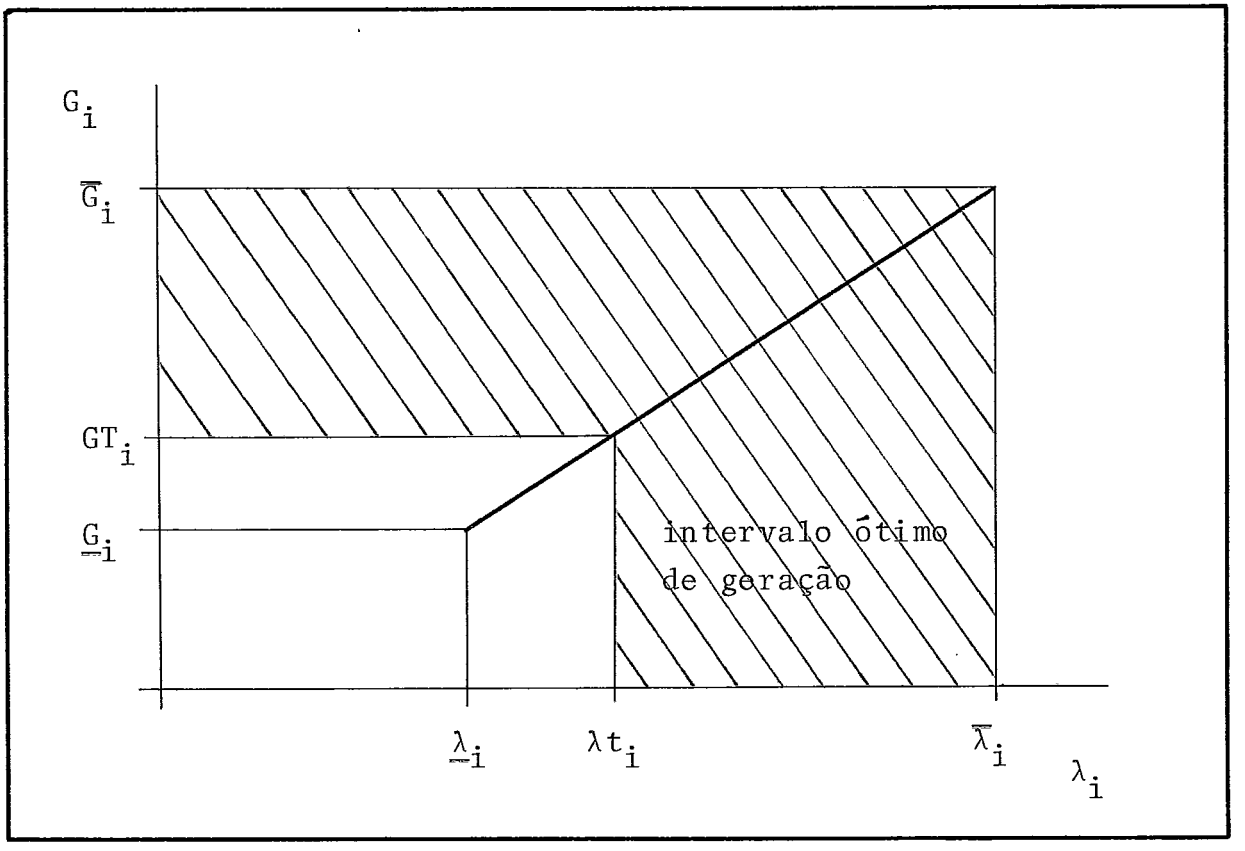

FIGURA A.3 - Curva típica de geração em função do preço.

A partir das curvas geração x preço das unidades em operação pode ser montada uma curva geração total em função do preço. Esta curva é composta por partes crescentes, por descontinuidades e por partes constantes (ver figura A.4). As partes crescentes in

dicam que há unidades com possibilidade de aumentar a geração devido a um aumento no preço. As abcissas das descontinuidades são os preços de transição para um conjunto de unidades. As partes constantes indicam que todas as unidades naquele trecho estão gerando na sua capacidade máxima. A figura A.4 representa uma curva típica de geração total em função do preço, onde λ^1, λ^3 e λ^4 são preços de transição e λ^2, λ^5 e λ^6 são preços máximos.

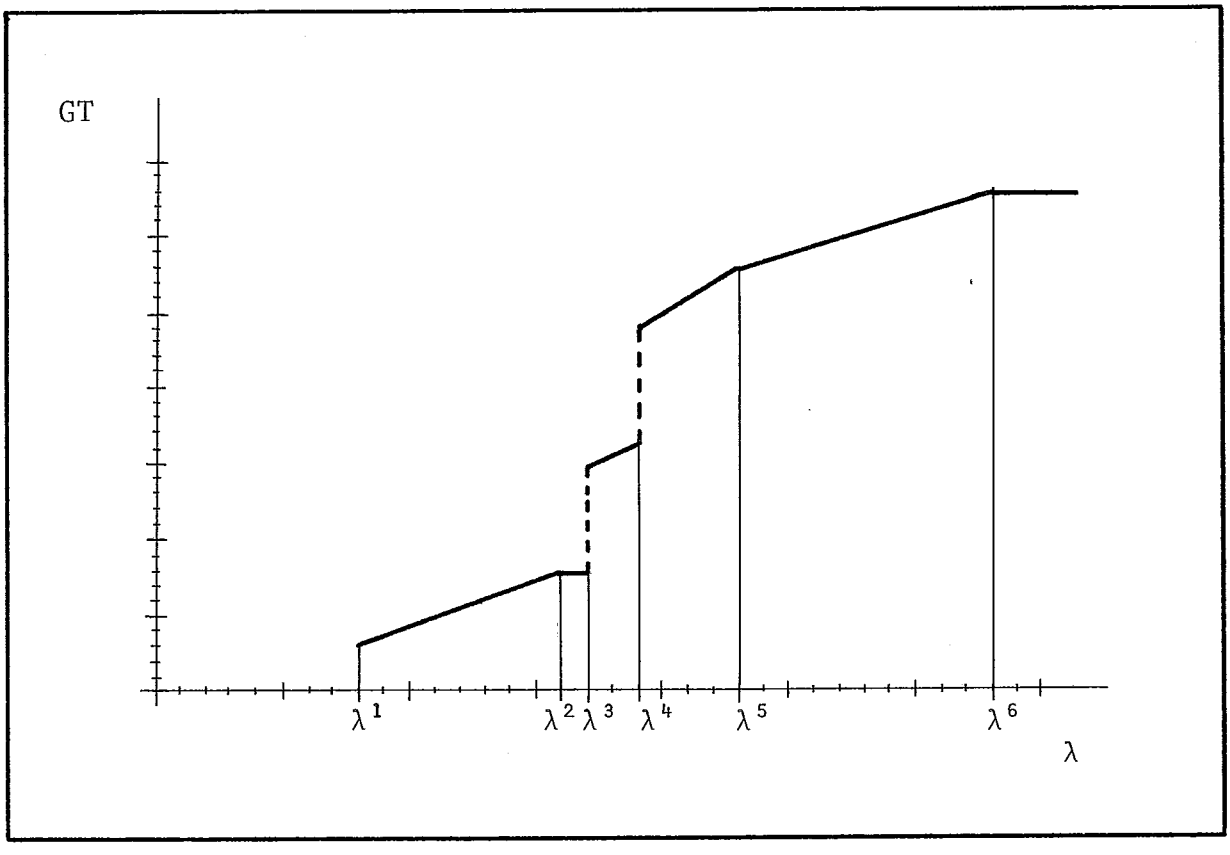

FIGURA A.4 - Curva típica de geração total em função do preço.

A solução ótima do Despacho será obtida através desta curva.

Dada uma determinada demanda, se a ela corresponder um ponto em uma região contínua da curva, a solução ótima é imediata a partir do preço ótimo (λ^*) correspondente.

Se à demanda estiver associado um ponto em uma descontinuidade da curva, são criados dois sub-problemas: um forçando uma das unidades que iniciam a geração naquele ponto a não gerar e outro forçando essa unidade a gerar por qualquer preço (ver figura A.5). A cada sub-problema, a curva é refeita de acordo com

as modificações realizadas. Se nestes sub-problemas a descontinuidade permanecer, são criados a partir deles novos sub-problemas da mesma forma anterior. Assim é gerada uma árvore cujos nós finais são sub-problemas com soluções viáveis ou inviáveis. A solução ótima esta associada ao nó final que corresponder ao menor custo.

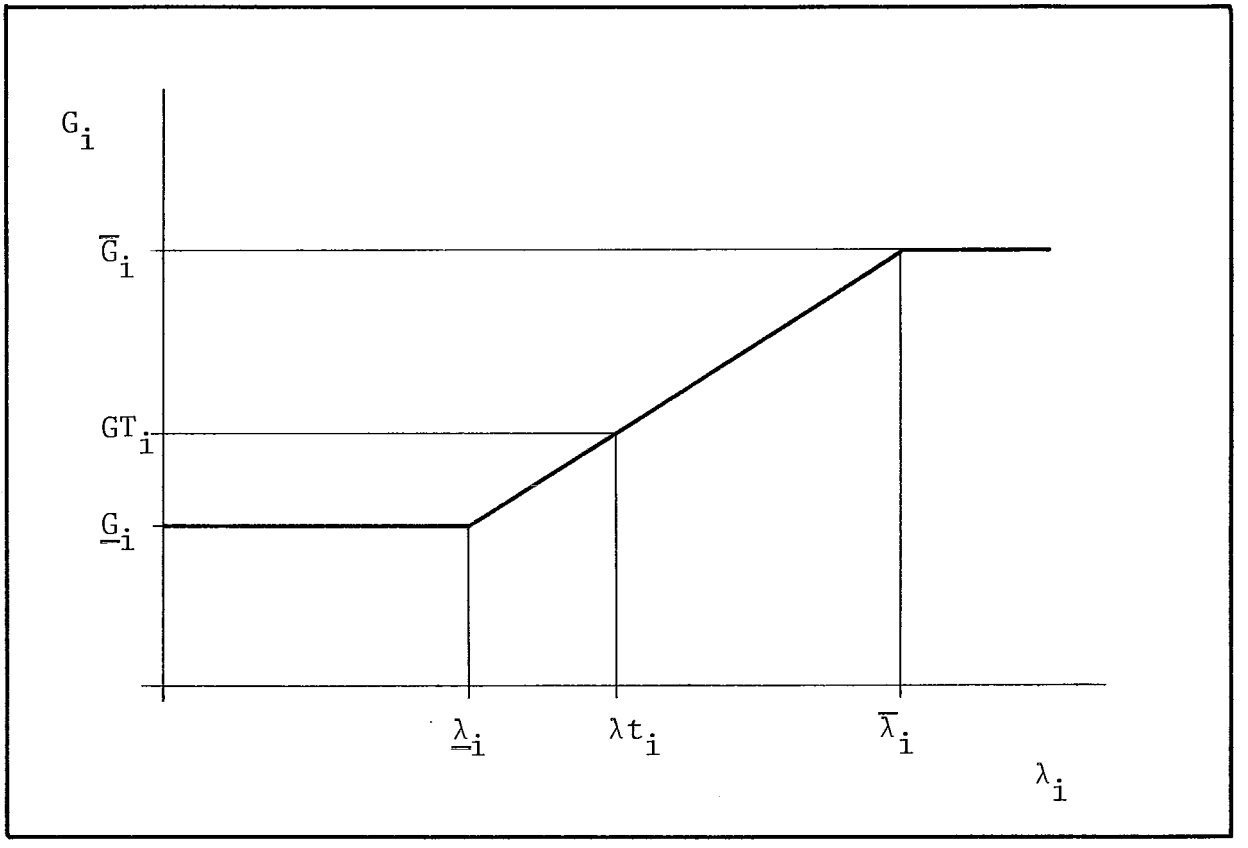

FIGURA A.5 - Curva típica de geração em função do preço para uma unidade força a gerar por qualquer preço.

Se uma unidade foi forçada a não gerar, então:

$$G_j^* = 0$$

$$CG_j^* = C0_j \qquad \text{A.10}$$

onde:

j ε conjunto das unidades forçadas a não gerar

Se uma unidade foi forçada a gerar a qualquer preço, então:

$$G_j^* = \underline{G}_j \qquad \text{se } \lambda^* \leq \underline{\lambda}_j$$

$$= (\lambda^* - b_j) / (2\, a_j) \qquad \text{se } \underline{\lambda}_j < \lambda^* < \overline{\lambda}_j$$

$$= \overline{G}_j \qquad \text{se } \lambda^* \geq \overline{\lambda}_j$$

$$CG^*_j = a_j\, (G_j^*)^2 + b_j\, G_j^* + c_j \qquad \text{A.11}$$

onde:

j ε conjunto das unidades forçadas a gerar

E se uma unidade gerar dentro do seu intervalo ótimo:

$$G_j^* = 0 \qquad \text{se } \lambda^* < \lambda t_j$$

$$= (\lambda^* - b_j) / (2\, a_j) \qquad \text{se } \lambda t_j \leq \lambda^* < \overline{\lambda}_j$$

$$= \overline{G}_j \qquad \text{se } \lambda^* \geq \overline{\lambda}_j$$

$$CG_j^* = C0_j \qquad \text{se } G_j^* = 0$$

$$= a_j\, (G_j^*)^2 + b_j\, G_j^* + c_j \qquad \text{se } G_j^* > 0 \qquad \text{A.12}$$

onde:

j ε conjunto das unidades que geram no seu intervalo ótimo

O custo total ótimo de geração será dado por:

$$CG^* = \Sigma_i\, CG_i^* \qquad \text{A.13}$$

onde:

i ε conjunto das unidades em operação

A.3 - Exemplo

Seja o problema de se encontrar a geração ótima de um sistema composto por quatro unidades em operação num determinado estágio para satisfazer uma demanda D = 17. Os parâmetros das unidades do sistema são dados na tabela A.1.

TABELA A.1 - Parâmetros das unidades do sistema exemplo.

unidade	$\underline{G}$	GT	$\overline{G}$	$\underline{\lambda}$	λt	$\overline{\lambda}$	a	b	c	CO
1	0.2	3	7	2.4	8	16	1	2	9	0
2	2.0	8	10	7.0	19	23	1	3	65	1
3	3.0	4	8	13.0	17	33	2	1	36	4
4	3.0	4	8	13.0	17	33	2	1	36	4

Inicialmente é montada a curva 1 (figura A.6) com os dados da tabela A.1. Para a demanda dada corresponde um ponto de descontinuidade na curva, causada pelo inicio do intervalo ótimo de geração da unidade 2. São criados então dois subproblemas: um onde a unidade 2 é forçada a não gerar e outro onde a unidade 2 é forçada a gerar por qualquer preço, resultando nas curvas 1.1 e 1.2 respectivamente.

Para a curva 1.1 a solução é imediata, pois para a demanda dada o ponto correspondente esta num trecho contínuo da curva com um preço ótimo $\lambda^* = 21$ e com um custo total ótimo calculado $CG^* = 254$.

Já na curva 1.2 o ponto correspondente a demanda dada pertence a uma descontinuidade da curva onde iniciam os intervalos ótimos de geração das unidades 3 e 4. Escolhendo arbitrariamente a unidade 3 para a separação do problema, são criadas as curvas 1.2.1 (forçando a unidade 3 a não gerar) e 1.2.2 (forçando a unidade 3 a gerar por qualquer preço) correspondentes aos dois novos subproblemas.

A partir da curva 1.2.1 são gerados outros dois subproblemas (curvas 1.2.1.1 e 1.2.1.2) pois tem-se um novo caso de descontí

nuidade.

A curva 1.2.2 apresenta solução imediata com um custo total ótimo CG* = 263 e as curvas 1.2.1.1 e 1.2.1.2 também apresentam solução imediata com custos de 267 e 263 respectivamente.

A árvore formada esta representada na figura A.13, sendo a solução ótima dada pela curva 1.1.

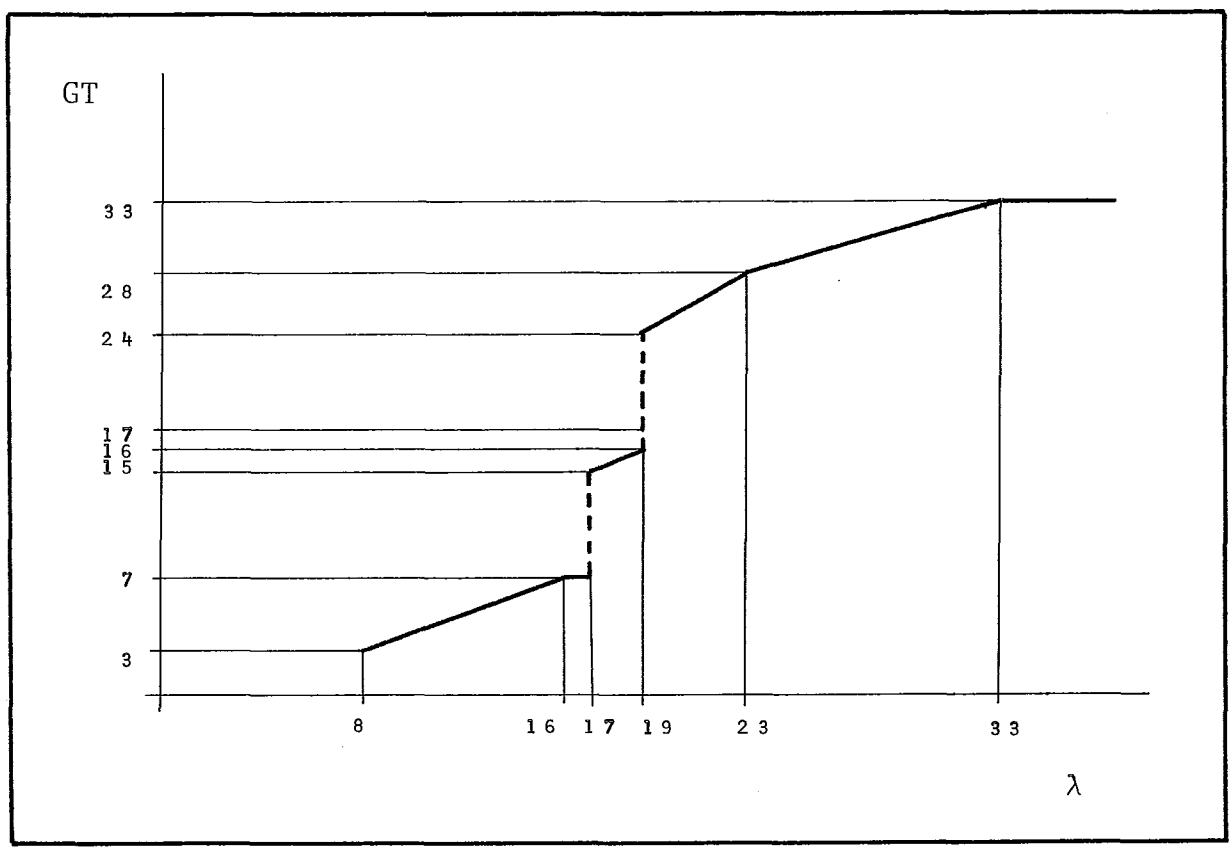

FIGURA A.6 - Curva 1 (todas unidades gerando no seu intervalo ótimo).

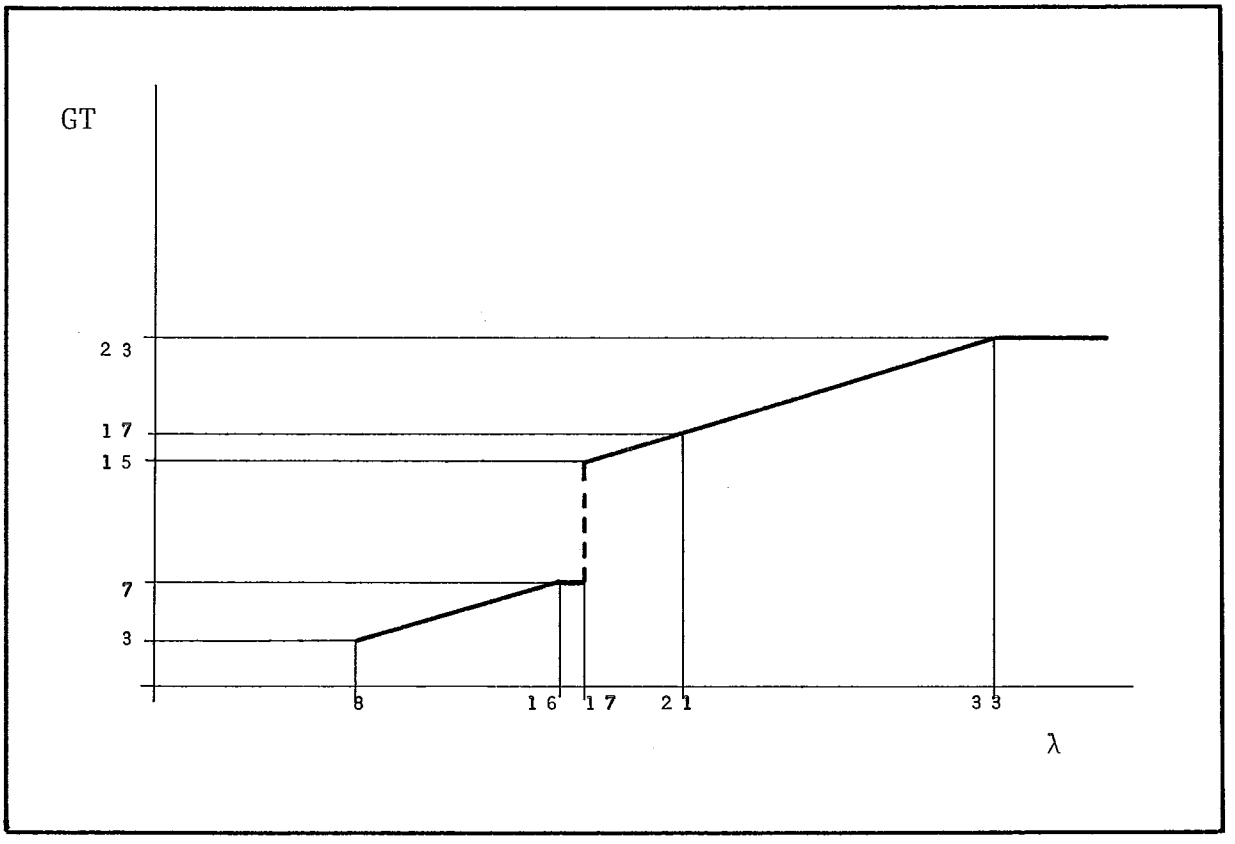

FIGURA A.7 - Curva 1.1 (unidade 2 forçada a não gerar).

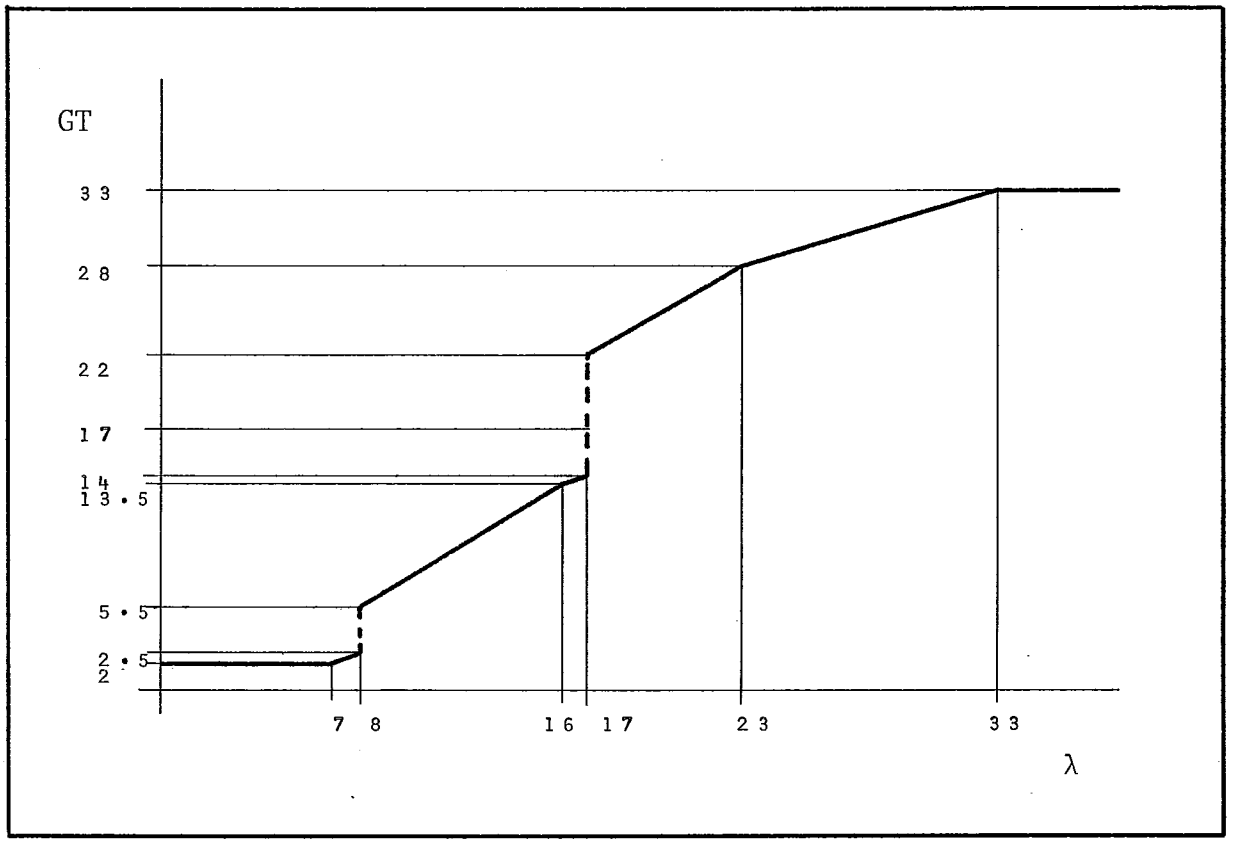

FIGURA A.8 - Curva 1.2 (unidade 2 forçada a gerar).

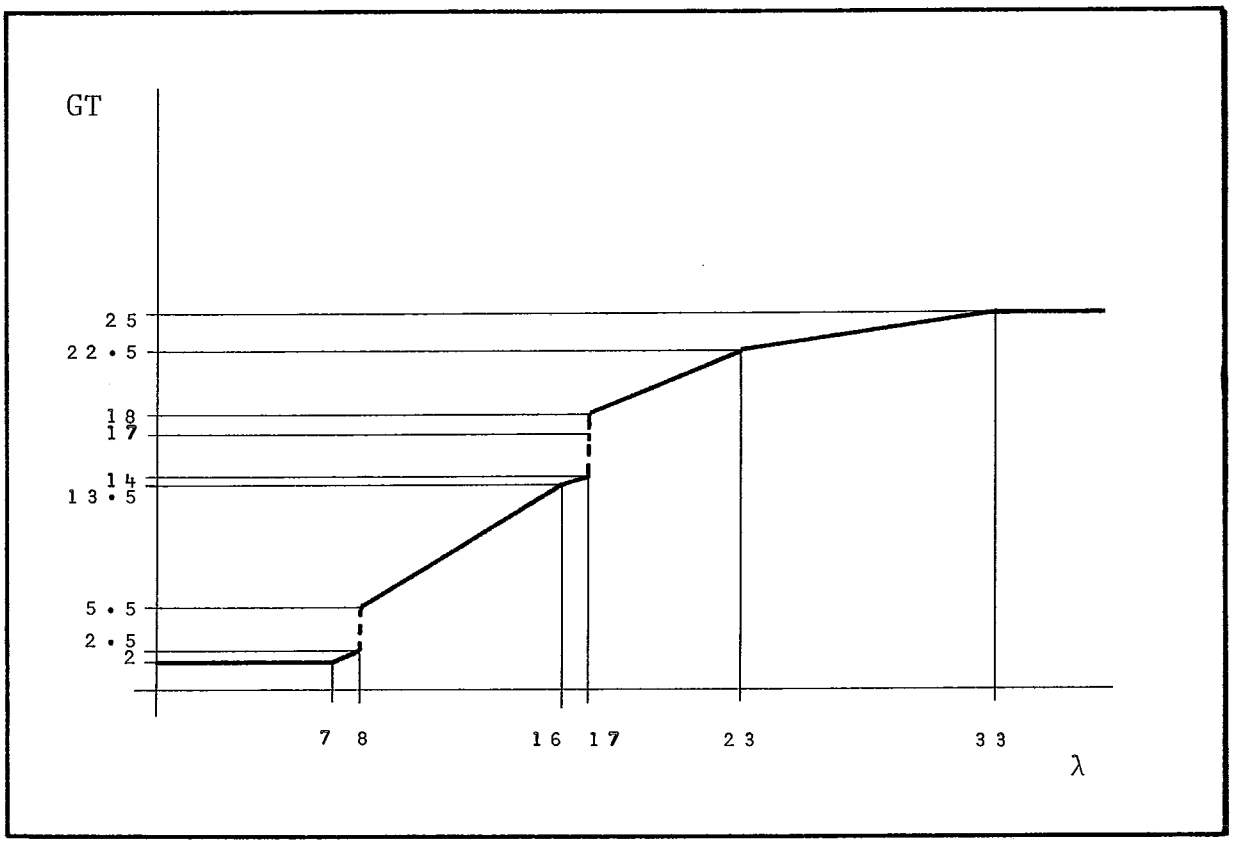

FIGURA A.9 - Curva 1.2.1 (unidade 2 forçada a gerar e unidade 3 forçada a não gerar).

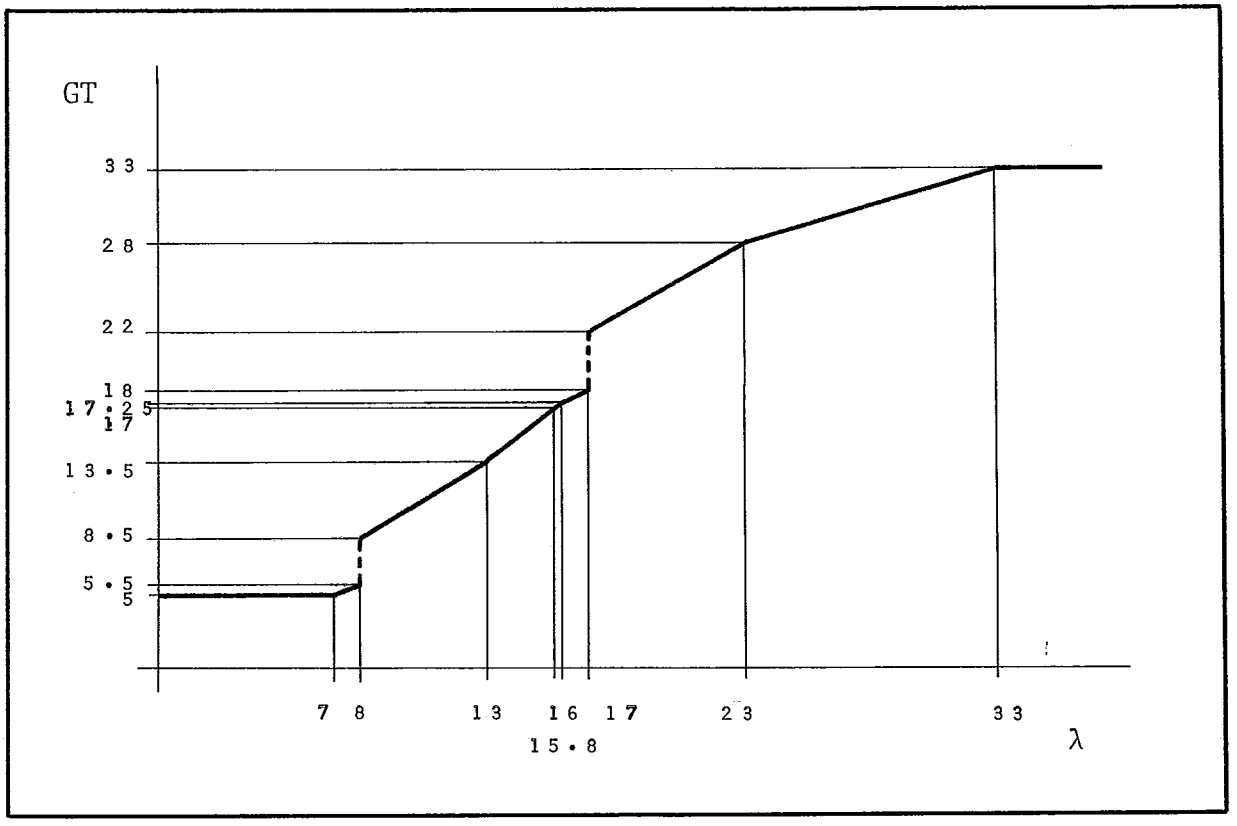

FIGURA A.10 - Curva 1.2.2 (unidades 2 e 3 forçadas a gerar).

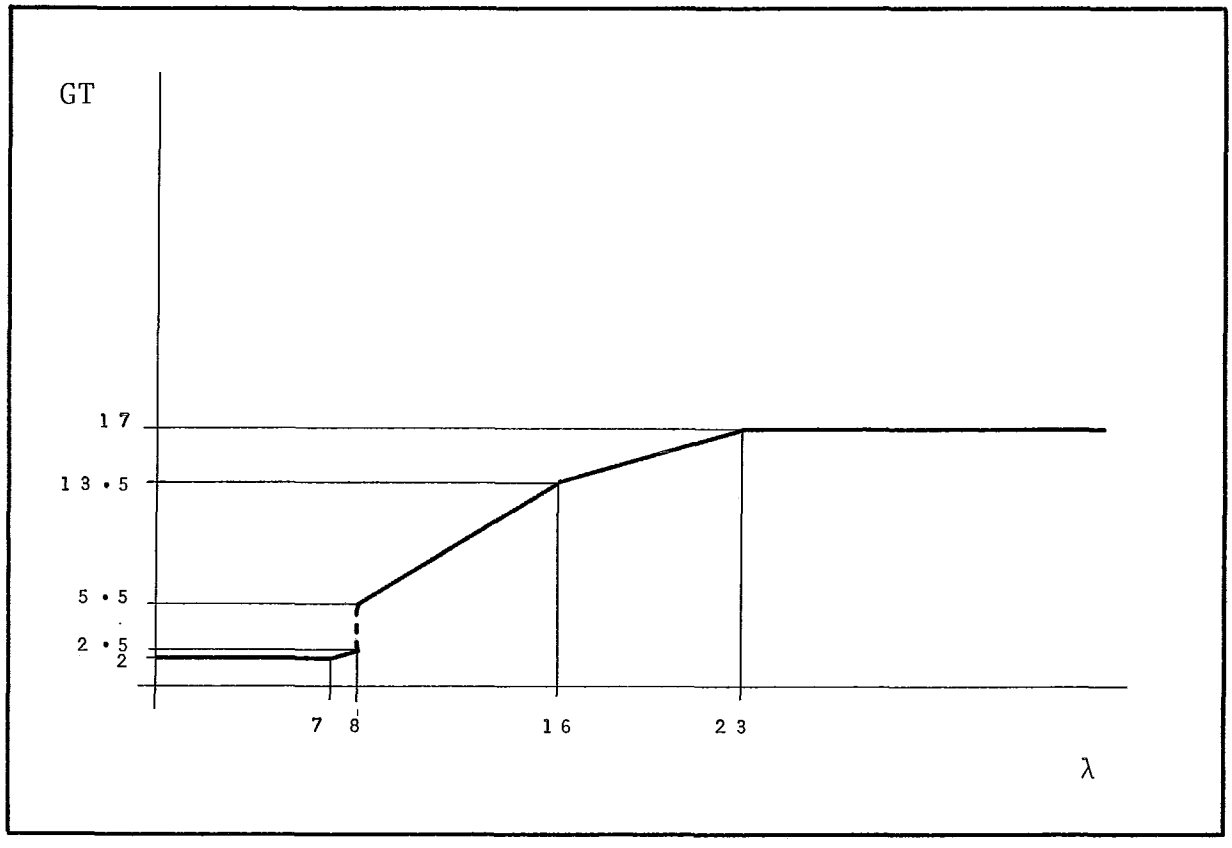

FIGURA A.11 - Curva 1.2.1.1 (unidade 2 forçada a gerar e unidades 3 e 4 forçadas a não gerar).

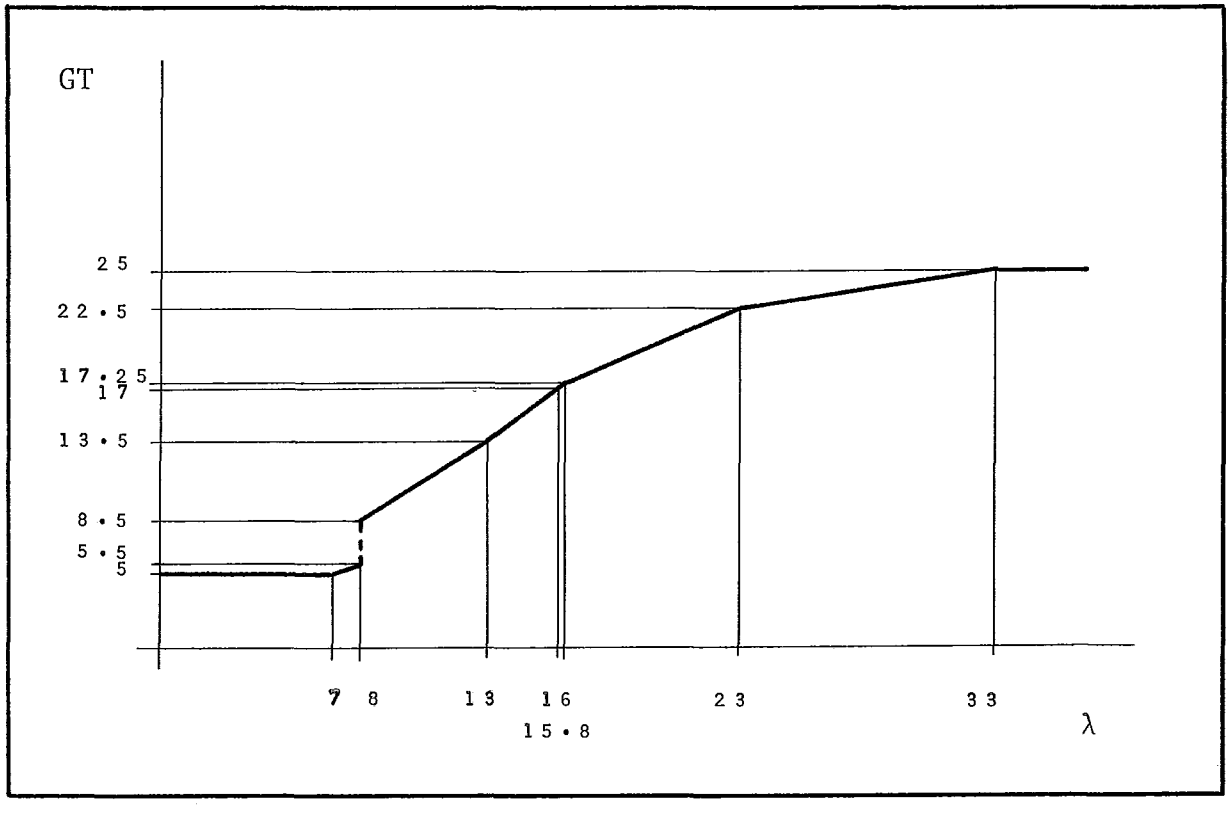

FIGURA A.12 - Curva 1.2.1.2 (unidades 2 e 4 forçadas a gerar e unidade 3 forçada a não gerar).

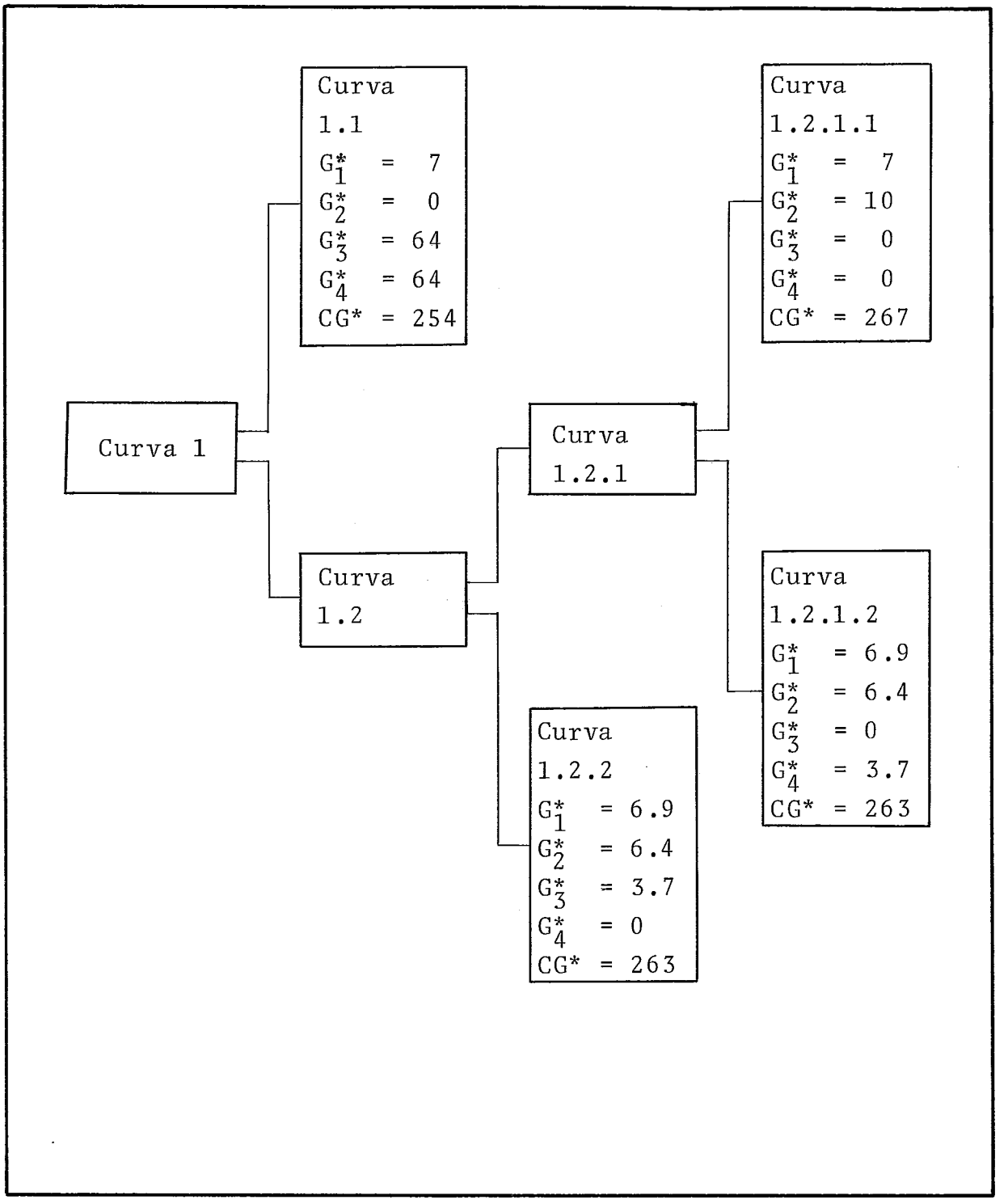

FIGURA A.13 - Árvore das soluções do Despacho

APÊNDICE B - Listagem do Programa

```
 $ SET WRITEAFTER LINEINFO                                          00100100
BEGIN                                                               00100200
   BOOLEAN                                                          00100300
      CRIATABELA,                                                   00100400
      IMPELEMENTOABERTO,                                            00100500
      IMPELEMENTOELIMINADO,                                         00100600
      IMPELEMENTOFECHADO,                                           00100700
      IMPVARIACAODADEMANDA;                                         00100800
   FILE                                                             00100900
      DADSIST                                                       00101000
        (KIND       = PACK,                                         00101100
         FILETYPE   =  7),                                          00101200
      DADGERAL                                                      00101300
        (KIND       = PACK,                                         00101400
         FILETYPE   =  7),                                          00101500
      IMPRESS                                                       00101600
        (KIND       = PRINTER),                                     00101700
      TABELA                                                        00101800
        (KIND       = PACK,                                         00101900
         FAMILYNAME = "PACK.",                                      00102000
         FILETYPE   = 6);                                           00102100
   INTEGER                                                          00102200
      HORPLAN,                                                      00102300
      NUMELEMABERTO,                                                00102400
      NUMELEMFECHADO,                                               00102500
      NUMELEMELIMABERTOS,                                           00102600
      NUMELEMELIMFECHADOS,                                          00102700
      NUMMAXELEMENTOS,                                              00102800
      NUMMAXGERACAO,                                                00102900
      NUMMAXPRECO,                                                  00103000
      NUMUNID,                                                      00103100
      NUMUNIDBAS,                                                   00103200
      NUMUNIDPONTA;                                                 00103300
   LABEL                                                            00103400
      FINAL;                                                        00103500
```

```
PROCEDURE MSG (VET);                                                    00103600
                                                                        00103700
                                                                        00103800
   ARRAY VET [*];                                                       00103900
   DISPLAY (VET);                                                       00104000
                                                                        00104100
                                                                        00104200
    %%     LEITURA DO NUMERO DE UNIDADES DO SISTEMA E     %%            00104300
    %%             DO NUMERO DE UNIDADES BASICAS.         %%           00104400
                                                                        00104500
                                                                        00104600
READ (DADSIST, //, NUMUNID, NUMUNIDBAS);                                00104700
NUMUNIDPONTA  := NUMUNID * NUMUNIDBAS;                                  00104800
NUMMAXPREC)   := NUMUNID * 3;                                           00104900
NUMMAXGERACAO := NUMUNID * 6;                                           00105000
                                                                        00105100
                                                                        00105200
    %%     LEITURA DO HORIZONTE DE PLANEJAMENTO E          %%           00105300
    %%             DO NUMERO MAXIMO DE ELEMENTOS DA LISTA  %%           00105400
                                                                        00105500
                                                                        00105600
READ (DADGERAL, //. IMPELEMENTOABERTO,                                  00105700
                    IMPELEMENTOELIMINADO,                               00105800
                    IMPELEMENTOFECHADO,                                 00105900
                    CRIATABELA,                                         00106000
                    IMPVARIACAODADEMANDA);                              00106100
READ (DADGERAL, //. HORPLAN);                                           00106200
READ (DADGERAL, //. NUMMAXELEMENTOS);                                   00106300
                                                                        00106400
                                                                        00106500
    %%     DECLARACAO DAS VARIAVEIS GLOBAIS.     %%                   00106600
                                                                        00106700
                                                                        00106800
BEGIN                                                                   00106900
   BOOLEAN                                                              00107000
      CONTINUA,                                                         00107100
      ELEMENTOEXISTE,                                                   00107200
      ELEMENTOIGUAL,                                                    00107300
```

```
    ELEMENTOVIAVEL,                                                    00107400
    ESTADOATUAL,                                                       00107500
    ESTADOFORMADO,                                                     00107600
    ESTADOINICIAL,                                                     00107700
    ESTADOMINIMO,                                                      00107800
    ESTADOPAI,                                                         00107900
    ESTADOSEMPREDESLIGADA,                                             00108000
    ESTADOSEMPRELIGADA,                                                00108100
    ESTADOSUCESSOR,                                                    00108200
    ESTADOTABATUALBOOL,                                                00108300
    ESTBOOL,                                                           00108400
    GERADEMANDA,                                                       00108500
    GERANDO,                                                           00108600
    IMPAR,                                                             00108700
    LIGAAGORA,                                                         00108800
    LIGADESLIGA,                                                       00108900
    LIGAQDONECES:                                                      00109000
DEFINE                                                                 00109100
    DEMANDAAUMENTA =                                                   00109200
       PREVISAODEMANDA [ESTAGIOPAI] = 1 #,                             00109300
    DEMANDADIMINUI =                                                   00109400
       PREVISAODEMANDA [ESTAGIOPAI] = 2 #,                             00109500
    DEMANDAIGUAL =                                                     00109600
       PREVISAODEMANDA [ESTAGIOPAI] = 0 #,                             00109700
    ENCONTROUALVO =                                                    00109800
       ESTAGIOELEMENTO [INDICE] = HORPLAN #,                           00109900
    ESTATUAL (I) =                                                     00110000
       ESTADOATUAL . [I : 1] #,                                        00110100
    ESTINICIAL (I)  =                                                  00110200
       ESTADOINICIAL . [I : 1] #,                                      00110300
    ESTFORMADO (I) =                                                   00110400
       ESTADOFORMADO . [I : 1] #,                                      00110500
    ESTMINIMO (I) =                                                    00110600
       ESTADOMINIMO . [I : 1] #,                                       00110700
    ESTPAI (I)      =                                                  00110800
       ESTADOPAI . [I : 1] #,                                          00110900
    ESTSEMPREDESLIGADA (I) =                                           00111000
       ESTADOSEMPREDESLIGADA . [I : 1] #,                              00111100
```

```
    ESTSEMPRELIGADA (I) =                                          00111200
       ESTADOSEMPRELIGADA . (I : 1) #,                             00111300
    ESTSUCESSOR (I) =                                              00111400
       ESTADOSUCESSOR . (I : 1) #,                                 00111500
    ESTTABATUALBOOL (I) =                                          00111600
       ESTADOTABATUALBOOL . (I : 1) #,                             00111700
    ESTELEMENTO (I, J) =                                           00111800
       ESTADOELEMENTO (I) . (J : 1) #,                             00111900
    INFINITOPOSITIVO =  1030 #,                                    00112000
    INFINITONEGATIVO = -1030 #,                                    00112100
    TABESTADO (I, J) =                                             00112200
       TABELADEESTADO (I) . (J : 1) #;                             00112300
INTEGER                                                            00112400
    ALGORITMO,                                                     00112500
    CONTAEXPCAMADA,                                                00112600
    CONTAESTAGIOVAZIO,                                             00112700
    ELEMENTO,                                                      00112800
    EST = ESTBOOL,                                                 00112900
    ESTADOPAIINT = ESTADOPAI,                                      00113000
    ESTADOTABATUAL = ESTADOTABATUALBOOL,                           00113100
    ESTAGIO,                                                       00113200
    ESTAGIOAEXPANDIR,                                              00113300
    ESTAGIOPAI,                                                    00113400
    ESTAGIOQUELIGA,                                                00113500
    ESTAGIOSUCESSOR,                                               00113600
    I,                                                             00113700
    IMINIMO,                                                       00113800
    IND,                                                           00113900
    INDANTERIOR,                                                   00114000
    INDAUX,                                                        00114100
    INDICE,                                                        00114200
    INDTAB,                                                        00114300
    INDTABFINAL,                                                   00114400
    INDTABINICIAL,                                                 00114500
    INDTABIGUAL,                                                   00114600
    J,                                                             00114700
    JMAX,                                                          00114800
    JMIN,                                                          00114900
```

```
      K,                                                          00115000
      L,                                                          00115100
      LIMITE,                                                     00115200
      MINTEMPOLIGA,                                               00115300
      NUMELEMENTOS,                                               00115400
      NUMELEMTAB,                                                 00115500
      NUMEXP,                                                     00115600
      NUMGERACAO,                                                 00115700
      NUMMAXSUCESSORES,                                           00115800
      NUMPRECO,                                                   00115900
      NOVOELEMENTO,                                               00116000
      NOVOINDICE,                                                 00116100
      NOVOIND,                                                    00116200
      PROXIND,                                                    00116300
      TEMPO,                                                      00116400
      UNIO;                                                       00116500
   REAL                                                           00116600
      ACRESCIMO,                                                  00116700
      AUX,                                                        00116800
      ESTIMATIVA,                                                 00116900
      BOUND,                                                      00117000
      CUSTO,                                                      00117100
      CUSTOACUM,                                                  00117200
      CUSTODESPACHO,                                              00117300
      CUSTOLIGACAOSUCESSOR,                                       00117400
      CUSTOLIGACAOPAI,                                            00117500
      CUSTOLIMITE,                                                00117600
      CUSTOMINLIGACAO,                                            00117700
      CUSTOSUBESTIMADO,                                           00117800
      CUSTOPAI,                                                   00117900
      CUSTOTOTAL,                                                 00118000
      CUSTOTOTALGERACAO,                                          00118100
      CUSTOTOTALSUCESSOR,                                         00118200
      DEMANDASUCESSOR,                                            00118300
      ESTUNID,                                                    00118400
      EPSILON,                                                    00118500
      FINFSUCESSOR,                                               00118600
      FINFSUCESSORDIV,                                            00118700
```

```
    FMINIMO,                                                    00118800
    GERMAX,                                                     00118900
    GERMAXACUM,                                                 00119000
    GERMIN,                                                     00119100
    GERUNID,                                                    00119200
    MAXB2,                                                      00119300
    MAXDEMANDA,                                                 00119400
    MAXGERMAXLIG,                                               00119500
    MINC2,                                                      00119600
    MINCUSTOINCREMENTAL,                                        00119700
    MINIMO,                                                     00119800
    MINTEMPO,                                                   00119900
    POTENCIA,                                                   00120000
    POTENCIACOMPL,                                              00120100
    PRECOMAX,                                                   00120200
    PRECOMIN,                                                   00120300
    PRECOTIMO,                                                  00120400
    PRIMEIROSUCESSOR,                                           00120500
    PROCESSAMENTOLIMITE,                                        00120600
    RESERVAACUM,                                                00120700
    RESERVASUCESSOR,                                            00120800
    SOMAGERMAX,                                                 00120900
    SOMAGERMIN,                                                 00121000
    SOMAGERMAXSUCESSOR,                                         00121100
    SUBESTADOSUCESSOR,                                          00121200
    TAXAREDUCAO;                                                00121300
BOOLEAN ARRAY                                                   00121400
    ELEMENTOABERTO,                                             00121500
    ESTADOELEMENTO,                                             00121600
    TABELADEESTADOBOOL                                          00121700
      [1 : NUMMAXELEMENTOS],                                    00121800
    ESTADOINICIALLER                                            00121900
      [1: NUMUNIDPONTA],                                        00122000
    UNIDLIGADA                                                  00122100
      [1 : NUMUNID];                                            00122200
INTEGER ARRAY                                                   00122300
    TEMPOELEMENTO                                               00122400
      [1 : NUMMAXELEMENTOS, 1 : NUMUNIDPONTA],                  00122500
```

```
      TEMPMIN                                                    00122600
         [0 : 1, 1 : NUMUNIDPONTA],                              00122700
      ANTECESSOR,                                                00122800
      ESTAGIOELEMENTO,                                           00122900
      LISTADEINDICEDEELEMENTOS,                                  00123000
      LISTADEINDICEVAGO                                          00123100
         [1 : NUMMAXELEMENTOS],                                  00123200
      TABELADEESTADO                                             00123300
         [1] = TABELADEESTADOBOOL,                               00123400
      ESTADO                                                     00123500
         [1] = UNIDLIGADA,                                       00123600
      INDUNID                                                    00123700
         [1 : NUMMAXPRECO],                                      00123800
      NUM,                                                       00123900
      NUMMAX                                                     00124000
         [0 : 1],                                                00124100
      NUMEXPCAMADA                                               00124200
         [1 : HORPLAN],                                          00124300
      CAMINHO,                                                   00124400
      INDICEDATABELA,                                            00124500
      PREVISAODEMANDA                                            00124600
         [0 : HORPLAN],                                          00124700
      TEMPOINICIAL,                                              00124800
      TEMPOPAI,                                                  00124900
      TEMPOSUCESSOR                                              00125000
         [1 : NUMUNIDPONTA];                                     00125100
   REAL ARRAY                                                    00125200
      A1,                                                        00125300
      B1,                                                        00125400
      C1,                                                        00125500
      CUSTOZERO,                                                 00125600
      GERATUAL,                                                  00125700
      GERMINIMA,                                                 00125800
      GERMAXIMA,                                                 00125900
      GEROTIMA,                                                  00126000
      GERTRANSICAO,                                              00126100
      PRECOMINIMO,                                               00126200
      PRECOMAXIMO,                                               00126300
```

```
      PRECOTRANSICAO                                                    00126400
         [1 : NUMUNID],                                                 00126500
      A2,                                                               00126600
      B2,                                                               00126700
      C2,                                                               00126800
      CUSTOLIGACAO,                                                     00126900
      CUSTODELIGACAOINCREMENTAL                                         00127000
         [1 : NUMUNIDPONTA],                                            00127100
      GERACAO                                                           00127200
         [0 : NUMMAXGERACAO],                                           00127300
      GERACAOELEMENTO,                                                  00127400
      TABELADEGERACAO                                                   00127500
         [1 : NUMMAXELEMENTOS, 1 : NUMUNID],                            00127600
      PRECO                                                             00127700
         [0 : NUMMAXPRECO],                                             00127800
      CUSTOLIGACAOELEMENTO,                                             00127900
      CUSTOELEMENTO,                                                    00128000
      FSUP,                                                             00128100
      FINF,                                                             00128200
      FINFOIV,                                                          00128300
      SUBESTADOELEMENTO,                                                00128400
      TABELADECUSTOS,                                                   00128500
      TABELADESOMAGERMAX,                                               00128600
      TABELADERESERVA                                                   00128700
         [1 : NUMMAXELEMENTOS],                                         00128800
      DEMANDA,                                                          00128900
      DEMANDAMAXFUTURA,                                                 00129000
      RESERVA,                                                          00129100
      SUBESTAGIO,                                                       00129200
      SUPESTAGIO                                                        00129300
         [1 : HORPLAN];                                                 00129400


PROCEDURE LEREIMPRIMIRDADOSDOSISTEMA;                                   00129500
                                                                        00129600
                                                                        00129700
BEGIN                                                                   00129800
```

```
DEFINE                                                                  00129900
   LISTA1 =                                                             00130000
      GERMINIMA [I], GERMAXIMA [I], CUSTOZERO [I],                      00130100
      A1 [I], B1 [I], C1 [I] #,                                         00130200
   LISTA2 =                                                             00130300
      A2 [I], B2 [I], C2 [I] #;                                         00130400
FORMAT                                                                  00130500
   CABECALHO1                                                           00130600
      (//T06, "PARAMETROS DAS UNIDADES DE PONTA"///                     00130700
         T06, "              GERACAO     GERACAO     CUSTO DE          ",  00130800
              "CONSTANTES DAS CURVAS DE CUSTO DE GERACAO"/              00130900
         T06, "UNIDADE       MINIMA      MAXIMA      GERACAO ZERO      ",  00131000
              "CONSTANTES DE LIGACAO"//),                               00131100
   CABECALHO2                                                           00131200
      (//T06, "PARAMETROS DAS UNIDADES BASICAS"///                      00131300
         T06, "              GERACAO     GERACAO     CUSTO DE"/        00131400
         T06, "UNIDADE       MINIMA      MAXIMA      GERACAO ZERO      ",  00131500
              "CONSTANTES DAS CURVAS DE CUSTO DE GERACAO"//),           00131600
   F1                                                                   00131700
      (T06, I5, X5, I5, " MW", X4, I5, " MW", X4, F12.6, X4,            00131800
            F12.6, " A1", X4, F12.6, " B1", X4, F12.6, " C1"/           00131900
       T56, F12.6, " A2", X4, F12.6, " B2", X4, F12.6, " C2"/),         00132000
   F2                                                                   00132100
      (T06, I5, X5, I5, " MW", X4, I5, " MW", X4, F12.6, X4,            00132200
            F12.6, " A1", X4, F12.6, " B1", X4, F12.6, " C1"/);         00132300
                                                                        00132400
                                                                        00132500
IF NUMUNIDPONTA > 0                                                     00132600
THEN                                                                    00132700
   BEGIN                                                                00132800
      WRITE (IMPRESS [SKIP 1]);                                         00132900
      WRITE (IMPRESS, CABECALHO1);                                      00133000
      FOR I := 1 STEP 1 UNTIL NUMUNIDPONTA DO                           00133100
      BEGIN                                                             00133200
         READ  (DADSIST, //,    LISTA1, LISTA2);                       00133300
         WRITE (IMPRESS, F1, I, LISTA1, LISTA2);                        00133400
      END;                                                              00133500
   END;                                                                 00133600
```

```
                                                                        00133700
                                                                        00133800
   IF NUMUNIDBAS > 0                                                    00133900
   THEN                                                                 00134000
      BEGIN                                                             00134100
         WRITE (IMPRESS (SKIP 1));                                      00134200
         WRITE (IMPRESS, CABECALHO2);                                   00134300
         FOR I := NUMUNIDPONTA+1 STEP 1 UNTIL NUMUNID DO                00134400
         BEGIN                                                          00134500
            READ  (DADOSIST, //,    LISTA1);                            00134600
            WRITE (IMPRESS, F2, I, LISTA1);                             00134700
         END;                                                           00134800
      END;                                                              00134900
END;                                                                    00135000


PROCEDURE LEREIMPRIMIRDADOSGERAIS;                                      00135100
                                                                        00135200
                                                                        00135300
BEGIN                                                                   00135400
   FORMAT                                                               00135500
      F1                                                                00135600
         (//T06, "DADOS DO ALGORITMO"///                                00135700
            T06, "ALGORITMO  ", C10//                                   00135800
            T06, "EPSILON          = ", F6.1//                          00135900
            T06, "TAXA DE REDUCAO = ", F8.4/),                          00136000
      F2                                                                00136100
         (//T06, "HORIZONTE DE PLANEJAMENTO  =  ", I2, "HS"//           00136200
            T06, "ESTAGIO     DEMANDA     RESERVA"//),                  00136300
      F3                                                                00136400
         (T06, X2, I2, X8, I7, X5, I7//),                               00136500
      F4                                                                00136600
         (//T06, "              TEMPO MINIMO     TEMPO MINIMO"/         00136700
            T06, "UNIDADE        DESLIGADA        LIGADA   "//),        00136800
      F5                                                                00136900
         (T06, X3, I2, X12, I2, X15, I2/),                              00137000
      F9                                                                00137100
```

```
      (//T06, "NUMERO MAXIMO DE ENTRADAS SIMULTANEAS = ", I2//          00137200
         T06, "NUMERO MAXIMO DE SAIDAS   SIMULTANEAS = ", I2//          00137300
         T06, "LIGA-DESLIGA  "  ", A3//),                              00137400
   F10                                                                  00137500
      (//T06, "ESTADO INICIAL"//),                                      00137600
   F11                                                                  00137700
      (T06, 20(X3, I2)//),                                              00137800
   F12                                                                  00137900
      (//T06, "TEMPO INICIAL"//);                                       00138000
                                                                        00138100
                                                                        00138200
READ  (DADGERAL, //, ALGORITMO, EPSILON, TAXAREDUCAO);                  00138300
READ (DADGERAL, //, LIGAQDONECES);                                      00138400
READ (DADGERAL, //, NUMMAXSUCESSORES);                                  00138500
WRITE (IMPRESS (SKIP 1));                                               00138600
WRITE (IMPRESS, F1, IF ALGORITMO = 1 THEN "AESTRELA  "                  00138700
                                     ELSE "ACHAPEU   " ,                00138800
                       EPSILON,TAXAREDUCAO);                            00138900
WRITE (IMPRESS, *//, LIGAQDONECES);                                     00139000
WRITE (IMPRESS, *//, NUMMAXSUCESSORES);                                 00139100
WRITE (IMPRESS, *//, NUMMAXELEMENTOS);                                  00139200
READ  (DADGERAL, //, DEMANDA);                                          00139300
WRITE (IMPRESS (SKIP 1));                                               00139400
WRITE (IMPRESS, F2, HORPLAN);                                           00139500
FOR I := 1 STEP 1 UNTIL HORPLAN DO                                      00139600
   WRITE (IMPRESS, F3, I, DEMANDA (I));                                 00139700
READ  (DADGERAL, //, TEMPMIN (1, *));                                   00139800
READ  (DADGERAL, //, TEMPMIN (0, *));                                   00139900
WRITE (IMPRESS (SKIP 1));                                               00140000
WRITE (IMPRESS, F4);                                                    00140100
FOR I := 1 STEP 1 UNTIL NUMUNIDPONTA DO                                 00140200
WRITE (IMPRESS, F5, I, TEMPMIN (0, I), TEMPMIN (1, I));                 00140300
READ  (DADGERAL, //, NUMMAX (0), NUMMAX (1), LIGADESLIGA);              00140400
WRITE (IMPRESS (SKIP 1));                                               00140500
WRITE (IMPRESS,  F9, NUMMAX (0), NUMMAX (1), IF LIGADESLIGA            00140600
                                        THEN "SIM" ELSE "NAO");         00140700
READ  (DADGERAL, //, FOR I := 1 STEP 1 UNTIL NUMUNIDPONTA DO            00140800
                     ESTADOINICIALLER (I));                             00140900
```

```
   FOR I := 1 STEP 1 UNTIL NUMUNIDPONTA DO                                  00141000
      ESTINICIAL (I) := ESTADOINICIALLER [I];                               00141100
   READ  (DADGERAL, //, TEMPOINICIAL);                                      00141200
   WRITE (IMPRESS [SKIP 1]);                                                00141300
   WRITE (IMPRESS, F10);                                                    00141400
   WRITE (IMPRESS, F11, FOR I := 1 STEP 1 UNTIL NUMUNIDPONTA DO             00141500
                        ESTINICIAL (I));                                    00141600
   WRITE (IMPRESS, F12);                                                    00141700
   WRITE (IMPRESS, F11, TEMPOINICIAL);                                      00141800
END;                                                                        00141900



PROCEDURE IMPRIMIRMENSAGEM                                                  00142000
   (MSG);                                                                   00142100
                                                                            00142200
                                                                            00142300
STRING                                                                      00142400
   MSG;                                                                     00142500
BEGIN                                                                       00142600
   WRITE (IMPRESS [SKIP 1]);                                                00142700
   WRITE (IMPRESS, <//, F06, C*///> , LENGTH(MSG), MSG);                    00142800
END;                                                                        00142900



PROCEDURE MONTARPRECOS;                                                     00143000
                                                                            00143100
                                                                            00143200
BEGIN                                                                       00143300
   J :=                                                                     00143400
   PRECO    [0] :=                                                          00143500
   GERACAO  [0] := 0;                                                       00143600
   FOR I := 1 STEP 1 UNTIL NUMUNID DO                                       00143700
   BEGIN                                                                    00143800
      AUX := 2 * A1 [I];                                                    00143900
      IF C1 [I] >= CUSTOZERO [I]                                            00144000
      THEN                                                                  00144100
```

```
            GERTRANSICAO [I] := SQRT ((C1 [I] - CUSTOZERO [I]) / A1 [I])   00144200
         ELSE                                                              00144300
            GERTRANSICAO [I] := 0;                                         00144400
         PRECOMINIMO [I]  := AUX * GERMINIMA [I] + B1 [I];                 00144500
         PRECOMAXIMO [I]  := AUX * GERMAXIMA [I] + B1 [I];                 00144600
         JMIN := J := J + 1;                                               00144700
         JMAX := J := J + 1;                                               00144800
         PREC] [JMIN]    := PRECOMINIMO [I];                               00144900
         PREC] [JMAX]    := PRECOMAXIMO [I];                               00145000
         INDUNID [JMIN] := -I;                                             00145100
         INDUNID [JMAX] :=  0;                                             00145200
         IF GERTRANSICAO [I] <= GERMINIMA [I]                              00145300
         THEN                                                              00145400
            BEGIN                                                          00145500
               GERTRANSICAO [I]    := GERMINIMA [I];                       00145600
               PRECOTRANSICAO [I] := PRECOMINIMO [I];                      00145700
               INDUNID [JMIN]      := I;                                   00145800
            END                                                            00145900
         ELSE                                                              00146000
            IF GERTRANSICAO [I] >= GERMAXIMA [I]                           00146100
            THEN                                                           00146200
               BEGIN                                                       00146300
                  GERTRANSICAO [I]    := GERMAXIMA [I];                    00146400
                  PRECOTRANSICAO [I] := PRECOMAXIMO [I];                   00146500
                  INDUNID [JMAX]      := I;                                00146600
               END                                                         00146700
            ELSE                                                           00146800
               BEGIN                                                       00146900
                  PREC0 [J := * + 1] :=                                    00147000
                  PRECOTRANSICAO [I] := AUX * GERTRANSICAO [I] + B1 [I];   00147100
                  INDUNID [J]         := I;                                00147200
               END;                                                        00147300
      END;                                                                 00147400
      NUMGERACAO := J * 2;                                                 00147500
      NUMPREC]    := J;                                                    00147600
      FOR I := 1 STEP 1 UNTIL NUMUNID DO                                   00147700
   END;                                                                    00147800
```

```
PROCEDURE ORDENARPRECOS;                                                00147900
                                                                        00148000
                                                                        00148100
BEGIN                                                                   00148200
   LIMITE := NUMPRECO - 1;                                              00148300
   FOR I := 1 STEP 1 UNTIL LIMITE DO                                    00148400
   BEGIN                                                                00148500
      IMINIMO := I;                                                     00148600
      MINIMO  := PRECO [I];                                             00148700
      FOR J := I+1 STEP 1 UNTIL NUMPRECO DO                             00148800
         IF PRECO [J] < MINIMO                                          00148900
         THEN                                                           00149000
            BEGIN                                                       00149100
               IMINIMO := J;                                            00149200
               MINIMO  := PRECO [J];                                    00149300
            END;                                                        00149400
         PRECO [IMINIMO]   := PRECO [I];                                00149500
         PRECO [I]         := MINIMO;                                   00149600
         INDAUX            := INDUNID [I];                            00149700
         INDUNID [I]       := INDUNID [IMINIMO];                      00149800
         INDUNID [IMINIMO] := INDAUX;                                   00149900
   END;                                                                 00150000
END;                                                                    00150100


PROCEDURE ORDENARSUCESSORES;                                            00150200
                                                                        00150300
                                                                        00150400
BEGIN                                                                   00150500
   FOR I := PRIMEIROSUCESSOR STEP 1 UNTIL NUMELEMENTOS - 1 DO           00150600
   BEGIN                                                                00150700
      INDAUX := LISTADEINDICEDEELEMENTOS [I];                           00150800
      AUX    := FINFDIV [INDAUX];                                       00150900
      FOR J := I + 1 STEP 1 UNTIL NUMELEMENTOS DO                       00151000
      BEGIN                                                             00151100
         IND := LISTADEINDICEDEELEMENTOS [J];                           00151200
         IF AUX > FINFDIV [IND]                                         00151300
```

```
            THEN                                                          00151400
               BEGIN                                                      00151500
                  LISTADEINDICEDEELEMENTOS [J] := INDAUX;                 00151600
                  INDAUX := IND;                                          00151700
                  AUX    := FINFDIV [IND];                                00151800
               END;                                                       00151900
         END;                                                             00152000
         LISTADEINDICEDEELEMENTOS [I] := INDAUX;                          00152100
      END;                                                                00152200
END DA ORDENARSUCESSORES;                                                 00152300


PROCEDURE CALCULARCUSTODODESPACHO;                                        00152400
                                                                          00152500
                                                                          00152600
BEGIN                                                                     00152700
   BOOLEAN                                                                00152800
      GERADEMANDA,                                                        00152900
      GERANDO,                                                            00153000
      IMPAR;                                                              00153100
   INTEGER                                                                00153200
      I,                                                                  00153300
      J,                                                                  00153400
      K,                                                                  00153500
      L,                                                                  00153600
      LIMITE,                                                             00153700
      UNID;                                                               00153800
   REAL                                                                   00153900
      ACRESCIMO,                                                          00154000
      CUSTO,                                                              00154100
      ESTUNID,                                                            00154200
      GERMAX,                                                             00154300
      GERUNID,                                                            00154400
      PRECOMAX;                                                           00154500
```

```
PROCEDURE PROCURARPRECOOTIMO                                          00154600
   (GERACAOVELHA, ESTADOVELHO);                                       00154700
                                                                      00154800
                                                                      00154900
INTEGER ARRAY                                                         00155000
   ESTADOVELHO [1];                                                   00155100
REAL ARRAY                                                            00155200
   GERACAOVELHA [0];                                                  00155300
BEGIN                                                                 00155400
   INTEGER                                                            00155500
      UNID;                                                           00155600
   REAL                                                               00155700
      GERMIN,                                                         00155800
      GERMAX,                                                         00155900
      PRECOMIN,                                                       00156000
      PRECOMAX;                                                       00156100
   INTEGER ARRAY                                                      00156200
      ESTADONOVO                                                      00156300
         [1 : NUMUNIO];                                               00156400
   REAL ARRAY                                                         00156500
      GERACAONOVA                                                     00156600
         [0 : NUMMAXGERACAO];                                         00156700


   PROCEDURE CALCULARPRECOOTIMO;                                      00156800
                                                                      00156900
                                                                      00157000
   BEGIN                                                              00157100
      PRECOTIMO := PRECO [J-1] + (PRECO [J] - PRECO [J-1])            00157200
                 * (DEMANDASUCESSOR      -  GERACAONOVA [I-1])        00157300
                 / (GERACAONOVA [I]  -  GERACAONOVA [I-1]);           00157400
      CUSTO := 0;                                                     00157500
      FOR UNID := 1 STEP 1 UNTIL NUMUNID DO                           00157600
      BEGIN                                                           00157700
         ESTUNID   := ESTADONOVO [UNID];                              00157800
         CASE ESTUNID OF                                              00157900
         BEGIN                                                        00158000
```

```
0:  ACRESCIMO :=                                                        00158100
    GERUNID   := 0;                                                     00158200
1:  GERANDO := FALSE;                                                   00158300
    FOR K := J STEP -1 WHILE K > 0 AND NOT GERANDO DO                   00158400
       GERANDO := INDUNID [K] = UNID;                                   00158500
    GERANDO := * AND (UNID = INDUNID [J] IMP                            00158600
                      NOT IMPAR);                                       00158700
    IF GERANDO                                                          00158800
    THEN                                                                00158900
       BEGIN                                                            00159000
          GERUNID :=                                                    00159100
             IF PRECOTIMO >= PRECOMAXIMO [UNID]                         00159200
             THEN                                                       00159300
                GERMAXIMA [UNID]                                        00159400
             ELSE                                                       00159500
                (PRECOTIMO - B1 [UNID]) /                               00159600
                          2 / A1 [UNID];                                00159700
          ACRESCIMO := C1 [UNID] + GERUNID *                            00159800
                       (GERUNID * A1 [UNID] + B1 [UNID]);               00159900
       END                                                              00160000
    ELSE                                                                00160100
       BEGIN                                                            00160200
          GERUNID   := 0;                                               00160300
          ACRESCIMO := CUSTOZERO [UNID];                                00160400
       END;                                                             00160500
2:  GERUNID :=                                                          00160600
       IF PRECOTIMO > PRECOMINIMO [UNID]                                00160700
       THEN                                                             00160800
          IF PRECOTIMO >= PRECOMAXIMO [UNID]                            00160900
          THEN                                                          00161000
             GERMAXIMA [UNID]                                           00161100
          ELSE                                                          00161200
             (PRECOTIMO - B1 [UNID]) / 2 / A1 [UNID]                    00161300
       ELSE                                                             00161400
          GERMINIMA [UNID];                                             00161500
    ACRESCIMO := C1 [UNID] + GERUNID *                                  00161600
                 (GERUNID * A1 [UNID] + B1 [UNID]);                     00161700
3:  GERUNID   := 0;                                                     00161800
```

```
            ACRESCIMO := CUSTOZERO [UNID];                          00161900
        END;                                                        00162000
        CUSTO          := * + ACRESCIMO;                            00162100
        GERATUAL [UNID] := GERUNID;                                 00162200
    END;                                                            00162300
END;                                                                00162400


PROCEDURE FORCARGERARMINIMO;                                        00162500
                                                                    00162600
                                                                    00162700
BEGIN                                                               00162800
    UNID := INDUNID [J];                                            00162900
    ESTADONOVO [UNID] := 2;                                         00163000
    GERMIN     := GERMINIMA [UNID];                                 00163100
    GERMAX     := GERMAXIMA [UNID];                                 00163200
    PRECOMIN   := PRECOMINIMO [UNID];                               00163300
    PRECOMAX   := PRECOMAXIMO [UNID];                               00163400
    LIMITE     := J - 1;                                            00163500
    FOR K := 1 STEP 1 UNTIL LIMITE DO                               00163600
    BEGIN                                                           00163700
        L := 2 * K;                                                 00163800
        ACRESCIMO :=                                                00163900
            IF PRECOMIN < PRECO [K]                                 00164000
            THEN                                                    00164100
                (PRECO [K] - B1 [UNID]) / 2 / A1 [UNID]             00164200
            ELSE                                                    00164300
                GERMIN;                                             00164400
        GERACAONOVA [L-1] := * + ACRESCIMO;                         00164500
        GERACAONOVA [L]   := * + ACRESCIMO;                         00164600
    END;                                                            00164700
    GERACAONOVA [L+1] := GERACAONOVA [L+2];                         00164800
END;                                                                00164900


PROCEDURE FORCARGERARZERO;                                          00165000
```

```
                                                                        00165100
                                                                        00165200
BEGIN                                                                   00165300
   ESTADONOVO [UNID] := 3;                                              00165400
   FOR K := 1 STEP 1 UNTIL NUMPRECO DO                                  00165500
   BEGIN                                                                00165600
      ACRESCIMO :=                                                      00165700
         IF PRECOMIN < PRECO [K]                                        00165800
         THEN                                                           00165900
            IF PRECOMAX <= PRECO [K]                                    00166000
            THEN                                                        00166100
               GERMAX                                                   00166200
            ELSE                                                        00166300
               (PRECO [K] - B1 [UNID]) /                                00166400
                2 / A1 [UNID]                                           00166500
         ELSE                                                           00166600
            GERMIN;                                                     00166700
      L := 2 * K;                                                       00166800
      GERACAONOVA [L-1] := * - ACRESCIMO;                               00166900
      GERACAONOVA [L]   := * - ACRESCIMO;                               00167000
   END;                                                                 00167100
END;                                                                    00167200
                                                                        00167300
WRITE (GERACAONOVA, NUMMAXGERACAO+1, GERACAOVELHA);                     00167400
WRITE (ESTADONOVO, NUMUNID, ESTADOVELHO);                               00167500
I := 1;                                                                 00167600
WHILE GERACAONOVA [I] < DEMANDASUCESSOR DO                              00167700
   I := * + 1;                                                          00167800
J     := (I + 1) DIV 2;                                                 00167900
IMPAR := I MOD 2 NEQ 0;                                                 00168000
GERADEMANDA := GERACAONOVA [I] <= DEMANDASUCESSOR + 1@-10 AND           00168100
               GERACAONOVA [I] >= DEMANDASUCESSOR - 1@-10;              00168200
IF I > 1 OR GERADEMANDA                                                 00168300
THEN                                                                    00168400
   IF IMPAR OR GERADEMANDA                                              00168500
   THEN                                                                 00168600
      BEGIN                                                             00168700
         CALCULARPRECOOTIMO;                                            00168800
```

```
                IF CUSTO < CUSTODESPACHO                                    00168900
                THEN                                                        00169000
                   BEGIN                                                    00169100
                      CUSTODESPACHO := CUSTO;                               00169200
                      WRITE (GEROTIMA, NUMUNID, GERATUAL);                  00169300
                   END                                                      00169400
                ELSE;                                                       00169500
             END                                                            00169600
          ELSE                                                              00169700
             BEGIN                                                          00169800
                FORCARGERARMINIMO;                                          00169900
                PROCURARPRECOOTIMO (GERACAONOVA, ESTADONOVO);               00170000
                IF GERACAONOVA [NUMGERACAO] - GERMAX >= DEMANDASUCESSOR     00170100
                THEN                                                        00170200
                   BEGIN                                                    00170300
                      FORCARGERARZERO;                                      00170400
                      PROCURARPRECOOTIMO (GERACAONOVA, ESTADONOVO);         00170500
                   END                                                      00170600
                ELSE;                                                       00170700
             END                                                            00170800
       ELSE;                                                                00170900
   END;                                                                     00171000
   FOR I := 1 STEP 1 UNTIL NUMGERACAO DO                                    00171100
      GERACAO [I] := 0;                                                     00171200
   FOR J := 1 STEP 1 UNTIL NUMPRECO DO                                      00171300
      IF UNID := INDUNID [J] > 0                                            00171400
      THEN                                                                  00171500
         IF UNIDLIGADA [UNID]                                               00171600
         THEN                                                               00171700
            BEGIN                                                           00171800
               GERMAX    := GERMAXIMA [UNID];                               00171900
               PRECOMAX := PRECOMAXIMO [UNID];                              00172000
               GERACAO [2*J] := * + GERTRANSICAO [UNID];                    00172100
               FOR K := J+1 STEP 1 UNTIL NUMPRECO DO                        00172200
               BEGIN                                                        00172300
                  ACRESCIMO :=                                              00172400
                     IF PRECO [K] < PRECOMAX                                00172500
                     THEN                                                   00172600
```

```
                        (PRECO [K] - B1 [UNID]) / 2 / A1 [UNID]          00172700
                     ELSE                                                00172800
                        GERMAX;                                          00172900
                  L := 2 * K;                                            00173000
                  GERACAO [L-1] := * + ACRESCIMO;                        00173100
                  GERACAO [L]   := * + ACRESCIMO;                        00173200
               END;                                                      00173300
            END                                                          00173400
         ELSE                                                            00173500
      ELSE;                                                              00173600
   CUSTODESPACHO := 1030;                                                00173700
   PROCURARPRECOOTIMO (GERACAO, ESTADO);                                 00173800
END;                                                                     00173900


PROCEDURE CALCULARDEMANDAMAXIMAFUTURA;                                   00174000
                                                                         00174100
                                                                         00174200
BEGIN                                                                    00174300
   LIMITE := HORPLAN - 1;                                                00174400
   FOR I := 1 STEP 1 UNTIL  LIMITE DO                                    00174500
   BEGIN                                                                 00174600
      MAXDEMANDA := INFINITONEGATIVO;                                    00174700
      FOR J := I STEP 1 UNTIL HORPLAN DO                                 00174800
         IF DEMANDA [J] > MAXDEMANDA                                     00174900
         THEN                                                            00175000
            BEGIN                                                        00175100
               MAXDEMANDA := DEMANDA [J];                                00175200
               INDICE := J;                                              00175300
            END                                                          00175400
         ELSE;                                                           00175500
      DEMANDAMAXFUTURA [I] := MAXDEMANDA;                                00175600
   END;                                                                  00175700
   DEMANDAMAXFUTURA [HORPLAN] := 0;                                      00175800
END;                                                                     00175900
```

```
PROCEDURE CALCULARSUPESTIMATIVADOESTAGIO;                          00176000
                                                                   00176100
                                                                   00176200
BEGIN                                                              00176300
   FOR I := 1 STEP 1 UNTIL HORPLAN DO                              00176400
      SUPESTAGIO [I] := (HORPLAN - I) * 128;                       00176500
END;                                                               00176600


PROCEDURE IMPRIMIRSUBESTIMATIVADOESTAGIO;                          00176700
                                                                   00176800
                                                                   00176900
BEGIN                                                              00177000
   FORMAT                                                          00177100
      CABECALHO                                                    00177200
         (//T06, "ESTAGIO        SUBESTIMATIVA"//),                00177300
      F                                                            00177400
         (T06, X3, I2, X13, I10/);                                 00177500
      WRITE (IMPRESS [SKIP 1]);                                    00177600
      WRITE (IMPRESS, CABECALHO);                                  00177700
      FOR I := 1 STEP 1 UNTIL HORPLAN DO                           00177800
         WRITE (IMPRESS, F, I, SUBESTAGIO [I]);                    00177900
END;                                                               00178000


PROCEDURE IMPRIMIRESTIMATIVASUPERIOR;                              00178100
                                                                   00178200
                                                                   00178300
BEGIN                                                              00178400
   FORMAT                                                          00178500
      CABECALHO                                                    00178600
         (//T06, "ESTAGIO        SUPESTIMATIVA"//),                00178700
      F                                                            00178800
         (T06, X3, I2, X13, I10/);                                 00178900
      WRITE (IMPRESS [SKIP 1]);                                    00179000
      WRITE (IMPRESS, CABECALHO);                                  00179100
```

```
      FOR I := 1 STEP 1 UNTIL HORPLAN DO                                00179200
         WRITE (IMPRESS, F, I, SUPESTAGIO [I]);                         00179300
END;                                                                    00179400


PROCEDURE VERIFICARVARIACAODADEMANDA;                                   00179500
                                                                        00179600
                                                                        00179700
BEGIN                                                                   00179800
   FOR I := 1 STEP 1 UNTIL HORPLAN - 1 DO                               00179900
      PREVISAODEMANDA [I] :=                                            00180000
         IF DEMANDA [I + 1] > DEMANDA [I]                               00180100
         THEN                                                           00180200
            1                                                           00180300
         ELSE                                                           00180400
            IF DEMANDA [I + 1] < DEMANDA [I]                            00180500
            THEN                                                        00180600
               2                                                        00180700
            ELSE                                                        00180800
               0;                                                       00180900
   PREVISAODEMANDA [0] := 0;                                            00181000
   PREVISAODEMANDA [HORPLAN] := 0;                                      00181100
END;                                                                    00181200


PROCEDURE IMPRIMIRVARIACAODADEMANDA;                                    00181300
                                                                        00181400
                                                                        00181500
BEGIN                                                                   00181600
   FORMAT                                                               00181700
      CABECALHO                                                         00181800
         (//T06, "ESTAGIO        VARIACAO DA DEMANDA"//),               00181900
      F                                                                 00182000
         (T06, X3, I2, X19, I1/);                                       00182100
      WRITE (IMPRESS [SKIP 1]);                                         00182200
      WRITE (IMPRESS, CABECALHO);                                       00182300
```

```
      FOR I := 1 STEP 1 UNTIL HORPLAN DO                                    00182400
         WRITE (IMPRESS, F, I, PREVISAODEMANDA [I]);                        00182500
END;                                                                        00182600


PROCEDURE INICIALIZARLISTADEELEMENTOS;                                      00182700
                                                                            00182800
                                                                            00182900
BEGIN                                                                       00183000
   ELEMENTOABERTO [1] := TRUE;                                              00183100
   NUMELEMABERTO :=                                                         00183200
   NUMELEMFECHADO := 0;                                                     00183300
   LISTADEINDICEDEELEMENTOS [1] := 1;                                       00183400
   ESTADOELEMENTO [1] := ESTADOINICIAL;                                     00183500
   WRITE (TEMPOELEMENTO  [1, *], NUMUNIDPONTA, TEMPOINICIAL );              00183600
   FOR I := 1 STEP 1 UNTIL NUMMAXELEMENTOS DO                               00183700
      LISTADEINDICEVAGO [I] := I;                                           00183800
   BOUND := INFINITOPOSITIVO;                                               00183900
   CAMINHO [0] := 1;                                                        00184000
   ELEMENTO     :=                                                          00184100
   INDICE       :=                                                          00184200
   NUMELEMENTOS := 1;                                                       00184300
   ANTECESSOR [1] :=                                                        00184400
   CUSTOELEMENTO [1] :=                                                     00184500
   ESTAGIOELEMENTO [1] :=                                                   00184600
   FINF [1] :=                                                              00184700
   FSUP [1] := 0;                                                           00184800
   FINFDIV [1] := 0;                                                        00184900
END;                                                                        00185000


BOOLEAN PROCEDURE ESCOLHERELEMENTOPORFMINIMO;                               00185100
                                                                            00185200
                                                                            00185300
BEGIN                                                                       00185400
   ELEMENTO := 0;                                                           00185500
```

```
      FMINIMO  := INFINITOPOSITIVO;                                          00185600
      FOR I := 1 STEP 1 UNTIL NUMELEMENTOS DO                                00185700
      BEGIN                                                                  00185800
         IND := LISTADEINDICEDEELEMENTOS [I];                                00185900
         IF ELEMENTOABERTO [IND] AND FINFDIV [IND] < FMINIMO                 00186000
         THEN                                                                00186100
            BEGIN                                                            00186200
               ELEMENTO  := I;                                               00186300
               INDICE    := IND;                                             00186400
               FMINIMO   := FINFDIV [IND];                                   00186500
            END                                                              00186600
         ELSE;                                                               00186700
      END;                                                                   00186800
      ESCOLHERELEMENTOPORFMINIMO := ELEMENTO > 0;                            00186900
   END;                                                                      00187000


   BOOLEAN PROCEDURE ESCOLHERELEMENTOPORFMAXIMO;                             00187100
                                                                             00187200
                                                                             00187300
   BEGIN                                                                     00187400
      ELEMENTO := 0;                                                         00187500
      FMINIMO  := INFINITOPOSITIVO;                                          00187600
      FOR I := 1 STEP 1  UNTIL NUMELEMENTOS DO                               00187700
      BEGIN                                                                  00187800
         IND := LISTADEINDICEDEELEMENTOS [I];                                00187900
         IF ELEMENTOABERTO [IND] AND FSUP [IND] < FMINIMO                    00188000
         THEN                                                                00188100
            BEGIN                                                            00188200
               ELEMENTO := I;                                                00188300
               INDICE   := IND;                                              00188400
               FMINIMO  := FSUP [IND];                                       00188500
            END                                                              00188600
         ELSE;                                                               00188700
      END;                                                                   00188800
      ESCOLHERELEMENTOPORFMAXIMO := ELEMENTO > 0;                            00188900
   END;                                                                      00189000
```

```
PROCEDURE RECUPERARCAMINHO;                                          00189100
                                                                     00189200
                                                                     00189300
BEGIN                                                                00189400
   CAMINHO [HORPLAN] :=                                              00189500
   INDANTERIOR       := INDICE;                                      00189600
   FOR I := HORPLAN - 1 STEP -1  UNTIL 1 DO                          00189700
      CAMINHO [I] :=                                                 00189800
      INDANTERIOR := ANTECESSOR [INDANTERIOR];                       00189900
END;                                                                 00190000


PROCEDURE IMPRIMIRCAMINHO;                                           00190100
                                                                     00190200
                                                                     00190300
BEGIN                                                                00190400
   FORMAT                                                            00190500
      F1                                                             00190600
         (//T06, "CUSTO DA POLITICA  =  ", Z12.6//),                 00190700
      F2                                                             00190800
         (//T06, "ESTAGIO            =", I2),                        00190900
      F3                                                             00191000
         (/ T06, "CUSTO              =", Z12.6),                     00191100
      F4                                                             00191200
         (/ T06, "CUSTO DE LIGACAO =", Z12.6),                       00191300
      F5                                                             00191400
         (/ T06, "SUBESTIMATIVA    =", Z12.6),                       00191500
      F6                                                             00191600
         (/ T06, "SUBESTIM.ESTADO  =", Z12.6),                       00191700
      F7                                                             00191800
         (/ T06, "SUPERESTIMATIVA  =", Z12.6),                       00191900
      F8                                                             00192000
         (/ T06, "ESTADO   ", 20(I5)),                               00192100
      F9                                                             00192200
         (/ T06, "TEMPO    ", 20(I5)),                               00192300
      F10                                                            00192400
         (/ T06, "GERACAO  ", 20(I5));                               00192500
```

```
                                                                        00192600
   WRITE (IMPRESS [SKIP 1]);                                            00192700
   WRITE (IMPRESS, F1, CUSTOELEMENTO [CAMINHO [HORPLAN]]);              00192800
   FOR I := 0 STEP 1 UNTIL HORPLAN DO                                   00192900
   BEGIN                                                                00193000
      J := CAMINHO [I];                                                 00193100
      WRITE (IMPRESS, F2, ESTAGIOELEMENTO [J]);                         00193200
      WRITE (IMPRESS, F3, CUSTOELEMENTO [J]);                           00193300
      WRITE (IMPRESS, F4, CUSTOLIGACAOELEMENTO [J]);                    00193400
      WRITE (IMPRESS, F5, FINF [J]);                                    00193500
      WRITE (IMPRESS, F6, SUBESTADOELEMENTO [J]);                       00193600
      WRITE (IMPRESS, F7, FSUP [J]);                                    00193700
      WRITE (IMPRESS, F8, FOR UNID := 1 STEP 1 UNTIL NUMUNIDPONTA DO    00193800
                          ESTELEMENTO (J, UNID),                        00193900
                          THRU NUMUNIDBAS DO 1);                        00194000
      WRITE (IMPRESS, F9, TEMPOELEMENTO  [J, *],                        00194100
                          THRU NUMUNIDBAS DO INFINITOPOSITIVO);         00194200
      WRITE (IMPRESS, F10, GERACAOELEMENTO[J, *]);                      00194300
   END;                                                                 00194400
   WRITE (IMPRESS, *//, TIME (12) * 243=7);                             00194500
END;                                                                    00194600


PROCEDURE REDEFINIRESTIMATIVASUPERIOR;                                  00194700
                                                                        00194800
                                                                        00194900
BEGIN                                                                   00195000
   CUSTOTOTALGERACAO := CUSTOELEMENTO [CAMINHO [HORPLAN]] -             00195100
                        CUSTOLIGACAOELEMENTO [CAMINHO [HORPLAN]];       00195200
   FOR I := 1 STEP 1 UNTIL HORPLAN DO                                   00195300
      BEGIN                                                             00195400
         IND := CAMINHO [I];                                            00195500
         SUPESTAGIO [I] := (CUSTOTOTALGERACAO - CUSTOELEMENTO [IND] +   00195600
                            CUSTOLIGACAOELEMENTO[IND]) * TAXAREDUCAO;   00195700
      END;                                                              00195800
   FOR I := 2 STEP 1 UNTIL NUMELEMENTOS DO                              00195900
   BEGIN                                                                00196000
```

```
      IND := LISTADEINDICEDEELEMENTOS [I];                                  00196100
      IF ELEMENTOABERTO [IND]                                               00196200
      THEN                                                                  00196300
         BEGIN                                                              00196400
            ESTAGIO := ESTAGIOELEMENTO [IND];                               00196500
            FSUP [IND] := SUPESTAGIO [ESTAGIO] +                            00196600
                          CUSTOELEMENTO [IND] +                             00196700
                          SUBESTADOELEMENTO [IND] /                         00196800
                         (HORPLAN - ESTAGIOELEMENTO [IND] + 1);             00196900
         END;                                                               00197000
   END;                                                                     00197100
   IMPRIMIRMENSAGEM ("***** SUPERESTIMATIVA REDEFINIDA *****");             00197200
END;                                                                        00197300


PROCEDURE IMPRIMIRELEMENTO                                                  00197400
   (INDICE, CODIGO);                                                        00197500
                                                                            00197600
                                                                            00197700
VALUE                                                                       00197800
   INDICE;                                                                  00197900
INTEGER                                                                     00198000
   INDICE;                                                                  00198100
STRING                                                                      00198200
   CODIGO;                                                                  00198300
BEGIN                                                                       00198400
   FORMAT                                                                   00198500
      F1                                                                    00198600
         (//(06, C11, X3, I4, X3, I4, X3, I2, X3, 3(X3, Z12.6)/),          00198700
      F2                                                                    00198800
         (106, 20(X2, I2)/);                                                00198900
   WRITE (IMPRESS, F1, CODIGO, INDICE, ANTECESSOR [INDICE],                 00199000
                       ESTAGIOELEMENTO [INDICE],                            00199100
                       CUSTOELEMENTO [INDICE],                              00199200
                       FSUP [INDICE], FINF [INDICE]);                       00199300
   WRITE (IMPRESS, F2, FOR I := 1 STEP 1 UNTIL NUMUNIDPONTA DO              00199400
                       ESTELEMENTO (INDICE, I),                             00199500
```

```
                        THRU NUMUNIDBAS DO 1);                                  00199600
   WRITE (IMPRESS, F2, TEMPOELEMENTOIINDICE, *],                                00199700
                        THRU NUMUNIDBAS DO INFINITOPOSITIVO);                   00199800
END;                                                                            00199900



PROCEDURE DESCARREGARNOVOELEMENTO;                                              00200000
                                                                                00200100
                                                                                00200200
BEGIN                                                                           00200300
   ELEMENTOABERTO [NOVOINDICE] := TRUE;                                         00200400
   NUMELEMABERTO := * + 1;                                                      00200500
   ANTECESSOR [NOVOINDICE] := INDICE;                                           00200600
   CUSTOELEMENTO [NOVOINDICE] := CUSTOTOTALSUCESSOR;                            00200700
   CUSTOLIGACAOELEMENTO [NOVOINDICE] := CUSTOLIGACAOSUCESSOR;                   00200800
   FOR I := 1 STEP 1 UNTIL NUMUNIDPONTA DO                                      00200900
      TEMPOELEMENTO  [NOVOINDICE, I] := TEMPOSUCESSOR  [I];                     00201000
   ESTADOELEMENTO [NOVOINDICE] := ESTADOSUCESSOR;                               00201100
   ESTAGIOELEMENTO [NOVOINDICE] := ESTAGIOSUCESSOR;                             00201200
   WRITE (GERACAOELEMENTO [NOVOINDICE, *], NUMUNID,                             00201300
                                        TABELADEGERACAO [INDTAB, *]);           00201400
   SUBESTADOELEMENTO [NOVOINDICE] := SUBESTADOSUCESSOR;                         00201500
   FINF [NOVOINDICE] := FINFSUCESSOR;                                           00201600
   FINFDIV [NOVOINDICE] := FINFSUCESSORDIV;                                     00201700
   FSUP [NOVOINDICE] := CUSTOTOTALSUCESSOR +                                    00201800
                        SUPESTAGIO [ESTAGIOSUCESSOR] +                          00201900
                        SUBESTADOSUCESSOR /                                     00202000
                        (HORPLAN * ESTAGIOSUCESSOR + 1);                        00202100
   IF IMPELEMENTOABERTO                                                         00202200
   THEN                                                                         00202300
      IMPRIMIRELEMENTO (NOVOINDICE, "ABERTO");                                  00202400
END;                                                                            00202500



PROCEDURE LIBERARINDICE                                                         00202600
   (ELEMENTO, INDICE);                                                          00202700
```

```
                                                                        00202800
                                                                        00202900
VALUE                                                                   00203000
   ELEMENTO,                                                            00203100
   INDICE;                                                              00203200
INTEGER                                                                 00203300
   ELEMENTO,                                                            00203400
   INDICE;                                                              00203500
BEGIN                                                                   00203600
   LISTADEINDICEVAGO [NUMELEMENTOS] := INDICE;                          00203700
   LISTADEINDICEDEELEMENTOS [ELEMENTO] :=                               00203800
      LISTADEINDICEDEELEMENTOS [NUMELEMENTOS];                          00203900
   NUMELEMENTOS := * - 1;                                               00204000
END;                                                                    00204100


PROCEDURE ALOCARNOVOINDICE;                                             00204200
                                                                        00204300
                                                                        00204400
IF NUMELEMENTOS := * + 1 > NUMMAXELEMENTOS                              00204500
THEN                                                                    00204600
   BEGIN                                                                00204700
      IMPRIMIRMENSAGEM ("***** LIMITE DE ELEMENTOS EXCEDIDO *****");    00204800
      GO TO FINAL;                                                      00204900
   END                                                                  00205000
ELSE                                                                    00205100
   BEGIN                                                                00205200
      NOVOELEMENTO := NUMELEMENTOS;                                     00205300
      NOVOINDICE   :=                                                   00205400
         LISTADEINDICEDEELEMENTOS [NOVOELEMENTO] :=                     00205500
         LISTADEINDICEVAGO [NOVOELEMENTO];                              00205600
   END;                                                                 00205700


PROCEDURE ELIMINARELEMENTOSPELOBOUND;                                   00205800
                                                                        00205900
```

```
                                                                        00206000
BEGIN                                                                   00206100
   IMPRIMIRMENSAGEM ("***** ELEMENTOS ELIMINADOS PELO BOUND *****");    00206200
   FOR I := 1 STEP 1 UNTIL NUMELEMENTOS DO                              00206300
      IF FINF (IND := LISTADEINDICEDEELEMENTOS [I]) > BOUND             00206400
      THEN                                                              00206500
         BEGIN                                                          00206600
            IF ELEMENTOABERTO (IND)                                     00206700
            THEN                                                        00206800
               NUMELEMELIMABERTOS := * + 1                              00206900
            ELSE                                                        00207000
               NUMELEMELIMFECHADOS := * + 1;                            00207100
            LIBERARINDICE (I, IND);                                     00207200
            IF IMPELEMENTOELIMINADO                                     00207300
            THEN                                                        00207400
               IMPRIMIRELEMENTO (IND , "ELIMINADO");                    00207500
            I := I - 1;                                                 00207600
         END                                                            00207700
      ELSE;                                                             00207800
END;                                                                    00207900
```

```
BOOLEAN PROCEDURE NAOELIMINARNOVOELEMENTO;                              00208000
                                                                        00208100
                                                                        00208200
BEGIN                                                                   00208300
   IF NAOELIMINARNOVOELEMENTO := FINFSUCESSOR < BOUND                   00208400
   THEN                                                                 00208500
      FOR I := 1 STEP 1 UNTIL NUMELEMENTOS DO                           00208600
      BEGIN                                                             00208700
         IND := LISTADEINDICEDEELEMENTOS [I];                           00208800
         ELEMENTOIGUAL := ESTAGIOELEMENTO (IND) = ESTAGIOSUCESSOR;      00208900
         FOR UNID := 1 STEP 1 WHILE ELEMENTOIGUAL AND                   00209000
                                     UNID <= NUMUNIDPONTA DO            00209100
            ELEMENTOIGUAL := ESTSUCESSOR (UNID) AND                     00209200
                             ESTELEMENTO (IND, UNID) AND                00209300
                             TEMPOSUCESSOR [UNID]  =                    00209400
```

```
                                TEMPOELEMENTO [IND, UNID];                    00209500
            IF ELEMENTOIGUAL                                                  00209600
            THEN                                                              00209700
               IF NAOELIMINARNOVOELEMENTO := FINFSUCESSOR < FINF [IND]        00209800
               THEN                                                           00209900
                  BEGIN                                                       00210000
                     LIBERARINDICE (I, IND);                                  00210100
                     IF IMPELEMENTOELIMINADO                                  00210200
                     THEN                                                     00210300
                        IMPRIMIRELEMENTO (IND, "ELIMINADO");                  00210400
                     I := I + 1;                                              00210500
                  END                                                         00210600
               ELSE                                                           00210700
                  I := NUMELEMENTOS                                           00210800
            ELSE;                                                             00210900
         END                                                                  00211000
      ELSE;                                                                   00211100
END;                                                                          00211200


PROCEDURE CALCULARSUBESTIMATIVADOESTADO;                                      00211300
                                                                              00211400
                                                                              00211500
BEGIN                                                                         00211600
   POTENCIA := SOMAGERMAXSUCESSOR - RESERVASUCESSOR;                          00211700
   IF DEMANDAMAXFUTURA [ESTAGIOSUCESSOR] * POTENCIA > 0                       00211800
   THEN                                                                       00211900
      BEGIN                                                                   00212000
         ESTAGIOQUELIGA := 0;                                                 00212100
         MINCUSTOINCREMENTAL :=                                               00212200
         MINC2 :=                                                             00212300
         MINTEMPO :=  INFINITOPOSITIVO;                                       00212400
         MAXB2    :=  INFINITONEGATIVO;                                       00212500
         SUBESTADOSUCESSOR := 0;                                              00212600
         CONTINUA := TRUE;                                                    00212700
         IF LIGAOOUNECES                                                      00212800
         THEN                                                                 00212900
```

```
      BEGIN                                                              00213000
         FOR I := ESTAGIOSUCESSOR + 1 STEP 1 UNTIL HORPLAN DO            00213100
            IF POTENCIA * DEMANDA [I] < 0                                00213200
            THEN                                                         00213300
               BEGIN                                                     00213400
                  ESTAGIOQUELIGA := I;                                   00213500
                  I := HORPLAN + 1;                                      00213600
               END                                                       00213700
            ELSE;                                                        00213800
         MINTEMPOLIGA := ESTAGIOQUELIGA - 1 - ESTAGIOSUCESSOR;           00213900
      END                                                                00214000
   ELSE                                                                  00214100
      MINTEMPOLIGA := 0;                                                 00214200
   FOR UNIO :=  1 STEP 1 UNTIL NUMUNIOPONTA DO                           00214300
      IF NOT ESTSUCESSOR (UNIO)                                          00214400
      THEN                                                               00214500
         BEGIN                                                           00214600
            CUSTOLIGACAO [UNIO] :=                                       00214700
               A2 [UNIO] * (1 - B2 [UNIO] *                              00214800
               EXP(- (TEMPOSUCESSOR [UNIO] + MINTEMPOLIGA) *             00214900
               C2 [UNIO]));                                              00215000
            CUSTOOELIGACAOINCREMENTAL [UNIO] :=                          00215100
               CUSTOLIGACAO [UNIO] / GERMAXIMA [UNIO];                   00215200
         END                                                             00215300
      ELSE;                                                              00215400
   FOR I := 1 STEP 1 UNTIL NUMUNIOPONTA DO                               00215500
      ESTATUAL (I) := ESTSUCESSOR (I);                                   00215600
   WHILE CONTINUA DO                                                     00215700
      BEGIN                                                              00215800
         FOR UNIO := 1 STEP 1 UNTIL NUMUNIOPONTA DO                      00215900
            BEGIN                                                        00216000
               IF NOT ESTATUAL (UNIO)                                    00216100
               THEN                                                      00216200
                  BEGIN                                                  00216300
                     IF MINCUSTOINCREMENTAL >=                           00216400
                        CUSTOOELIGACAOINCREMENTAL [UNIO]                 00216500
                     THEN                                                00216600
                        BEGIN                                            00216700
```

```
                                    MINCUSTOINCREMENTAL :=                    00216800
                                       CUSTODELIGACAOINCREMENTAL [UNID];      00216900
                                    IMINIMO := UNID;                          00217000
                                 END                                          00217100
                              ELSE;                                           00217200
                           END                                                00217300
                        ELSE;                                                 00217400
                     END;                                                     00217500
                  IF DEMANDAMAXFUTURA [ESTAGIOSUCESSOR] *                     00217600
                     (POTENCIA + GERMAXIMA [IMINIMO]) > 0                     00217700
                  THEN                                                        00217800
                     BEGIN                                                    00217900
                        POTENCIA := * + GERMAXIMA [IMINIMO];                  00218000
                        SUBESTADOSUCESSOR := * +  CUSTOLIGACAO [IMINIMO];     00218100
                     END                                                      00218200
                  ELSE                                                        00218300
                     BEGIN                                                    00218400
                        POTENCIACOMPL :=                                      00218500
                           DEMANDAMAXFUTURA [ESTAGIOSUCESSOR] *               00218600
                           POTENCIA;                                          00218700
                        SUBESTADOSUCESSOR := * + POTENCIACOMPL *              00218800
                           MINCUSTOINCREMENTAL;                               00218900
                        CONTINUA := FALSE;                                    00219000
                     END;                                                     00219100
                  ESTATUAL [IMINIMO] := TRUE;                                 00219200
                  MINCUSTOINCREMENTAL := INFINITOPOSITIVO;                    00219300
               END;                                                           00219400
         END                                                                  00219500
      ELSE                                                                    00219600
         SUBESTADOSUCESSOR := 0;                                              00219700
   END;                                                                       00219800


   PROCEDURE IMPRIMIRTABELA;                                                  00219900
                                                                              00220000
                                                                              00220100
   BEGIN                                                                      00220200
```

```
   FORMAT                                                                  00220300
      CABECALHO                                                            00220400
         (//T06, "TABELA DE ELEMENTOS VIAVEIS PARA CADA ESTAGIO"           00220500
          //T06, "CUSTO DE GERACAO       POTENCIA       RESERVA"          00220600
          //T06, "ESTADO"                                                 00220700
          //T06, "GERACAO"),                                              00220800
      F1                                                                   00220900
         (//T06, "ESTAGIO = ", I2, X3, "DEMANDA = ", I5),                  00221000
      F2                                                                   00221100
         (/T06, I4, 3(X3, Z12.6), X3, I2),                                 00221200
      F3                                                                   00221300
         (/T06, 20(I5));                                                   00221400
   I := 0;                                                                 00221500
   WRITE (IMPRESS, CABECALHO);                                             00221600
   FOR ESTAGIO := 1 STEP 1 UNTIL HORPLAN DO                                00221700
      BEGIN                                                                00221800
         WRITE (IMPRESS, F1, ESTAGIO, DEMANDA [ESTAGIO]);                  00221900
         INDTABINICIAL := INDICEDATABELA [ESTAGIO];                        00222000
         INDTABFINAL   := INDICEDATABELA [ESTAGIO + 1] - 1;                00222100
         FOR INDTAB := INDTABINICIAL STEP 1 UNTIL INDTABFINAL DO           00222200
            BEGIN                                                          00222300
               WRITE (IMPRESS, F2, I := I + 1,                             00222400
                                   TABELADECUSTOS [INDTAB],                00222500
                                   TABELADESOMAGERMAX [INDTAB],            00222600
                                   TABELADERESERVA [INDTAB]);              00222700
               WRITE (IMPRESS, F3, FOR UNID := 1 STEP 1 UNTIL              00222800
                                   NUMUNIDPONTA DO                         00222900
                                   TABESTADO (INDTAB, UNID),               00223000
                                   THRU NUMUNIDBAS DO 1);                  00223100
               WRITE (IMPRESS, F3, TABELADEGERACAO [INDTAB, *]);           00223200
            END;                                                           00223300
      END;                                                                 00223400
END;                                                                       00223500


PROCEDURE CALCULARCUSTOTOTAL;                                              00223600
                                                                           00223700
```

```
BEGIN                                                                   00223800
   CUSTOLIGACAOSUCESSOR := 0;                                           00223900
   FOR UNID := 1 STEP 1 UNTIL NUMUNIDPONTA DO                           00224000
      IF ESTSUCESSOR (UNID) AND NOT ESTELEMENTO (INDICE, UNID)          00224100
      THEN                                                              00224200
         CUSTOLIGACAOSUCESSOR := * + A2 [UNID] *                        00224300
                                 (1 - B2 [UNID] * EXP (-C2 [UNID] *     00224400
                                 TEMPOELEMENTO [INDICE, UNID]))         00224500
      ELSE;                                                             00224600
   CUSTOTOTALSUCESSOR := CUSTOLIGACAOSUCESSOR + TABELADECUSTOS [INDTAB] 00224700
                                              + CUSTOPAI;               00224800
   CUSTOLIGACAOSUCESSOR := * + CUSTOLIGACAOPAI;                         00224900
   IF ESTAGIOSUCESSOR = HORPLAN                                         00225000
   THEN                                                                 00225100
      SUBESTADOSUCESSOR := 0                                            00225200
   ELSE                                                                 00225300
      CALCULARSUBESTIMATIVADOESTADO;                                    00225400
      FINFSUCESSOR := CUSTOTOTALSUCESSOR +                              00225500
                      SUBESTAGIO [ESTAGIOSUCESSOR] +                    00225600
                      SUBESTADOSUCESSOR;                                00225700
      FINFSUCESSORDIV := CUSTOTOTALSUCESSOR +                           00225800
                         SUBESTAGIO [ESTAGIOSUCESSOR] +                 00225900
                         SUBESTADOSUCESSOR /                            00226000
                         (HORPLAN - ESTAGIOSUCESSOR + 1);               00226100
   IF NAOELIMINARNOVOELEMENTO                                           00226200
   THEN                                                                 00226300
      BEGIN                                                             00226400
         ALOCARNOVOINDICE;                                              00226500
         DESCARREGARNOVOELEMENTO;                                       00226600
      END;                                                              00226700
END;                                                                    00226800
                                                                        00226900


PROCEDURE SELECIONARSUCESSORES;                                         00227000
                                                                        00227100
                                                                        00227200
```

```
BEGIN                                                                   00227300
   ORDENARSUCESSORES;                                                   00227400
   FOR I := PRIMEIROSUCESSOR + NUMMAXSUCESSORES STEP 1                  00227500
                                        UNTIL NUMELEMENTOS DO           00227600
      LIBERARINDICE (I, LISTADEINDICEDEELEMENTOS [I]);                  00227700
END DA SELECIONARSUCESSORES;                                            00227800


PROCEDURE FECHARELEMENTO;                                               00227900
                                                                        00228000
                                                                        00228100
BEGIN                                                                   00228200
                                                                        00228300
                                                                        00228400
   %%   FECHA O ELEMENTO A SER EXPANDIDO E TRANSFERE   %%               00228500
   %%   SEUS PARAMETROS PARA "PAI".                    %%               00228600
                                                                        00228700
                                                                        00228800
   ELEMENTOABERTO [INDICE] := FALSE;                                    00228900
   NUMELEMFECHADO := * + 1;                                             00229000
   IF IMPELEMENTOFECHADO                                                00229100
   THEN                                                                 00229200
      IMPRIMIRELEMENTO (INDICE, "FECHADO");                             00229300
   CUSTOPAI := CUSTOELEMENTO [INDICE];                                  00229400
   CUSTOLIGACAOPAI := CUSTOLIGACAOELEMENTO [INDICE];                    00229500
   ESTAGIOPAI := ESTAGIOELEMENTO [INDICE];                              00229600
   ESTAGIOSUCESSOR := ESTAGIOPAI + 1;                                   00229700
   ESTADOPAI := ESTADOELEMENTO [INDICE];                                00229800
   WRITE (TEMPOPAI, NUMUNIDPONTA, TEMPOELEMENTO [INDICE, *]);           00229900
                                                                        00230000
                                                                        00230100
   %%   DEFINE OS INDICES DA TABELA ONDE ESTAO OS      %%               00230200
   %%   POSSIVEIS SUCESSORES.                          %%               00230300
                                                                        00230400
                                                                        00230500
   INDTABINICIAL := INDICEDATABELA [ESTAGIOSUCESSOR];                   00230600
   INDTABFINAL   := INDICEDATABELA [ESTAGIOPAI] + 1 ;                   00230700
```

```
CUSTOLIMITE   := INFINITOPOSITIVO;                                    00230800
INDTABIGUAL   := 0;                                                   00230900
                                                                      00231000
                                                                      00231100
%%   PROCURA NA TABELA O ELEMENTO QUE TEM O MESMO   %%                00231200
%%   ESTADO DO "PAI" E GUARDA SEU INDICE E CUSTO.   %%                00231300
                                                                      00231400
                                                                      00231500
FOR INDTAB := INDTABINICIAL STEP 1 UNTIL INDTABFINAL DO               00231600
   IF TABELADEESTADO [INDTAB] = ESTADOPAIINT                          00231700
   THEN                                                               00231800
      BEGIN                                                           00231900
         CUSTOLIMITE := TABELADECUSTOS [INDTAB];                      00232000
         INDTABIGUAL := INDTAB;                                       00232100
         INDTAB      := INDTABFINAL;                                  00232200
      END                                                             00232300
   ELSE;                                                              00232400
                                                                      00232500
                                                                      00232600
%%   SE A DEMANDA CAI OS ELEMENTOS VIAVEIS ESTAO ABAIXO   %%          00232700
%%   DO ELEMENTO IGUAL AO "PAI", SE SOBE , ESTAO ACIMA.   %%          00232800
                                                                      00232900
                                                                      00233000
IF INDTABIGUAL > 0                                                    00233100
   THEN                                                               00233200
      IF DEMANDADIMINUI                                               00233300
      THEN                                                            00233400
         INDTABINICIAL := INDTABIGUAL                                 00233500
      ELSE                                                            00233600
         IF DEMANDAAUMENTA                                            00233700
         THEN                                                         00233800
            INDTABFINAL := INDTABIGUAL                                00233900
         ELSE                                                         00234000
   ELSE;                                                              00234100
                                                                      00234200
                                                                      00234300
%%   BUSCA CADA ELEMENTO DO ESTAGIO SUCESSOR NA TABELA   %%           00234400
%%   E TESTA SUA VIABILIDADE.                            %%           00234500
```

```
                                                                        00234600
                                                                        00234700
PRIMEIROSUCESSOR := NUMELEMENTOS + 1;                                   00234800
FOR INDTAB := INDTABINICIAL STEP 1 UNTIL INDTABFINAL DO                 00234900
   BEGIN                                                                00235000
      NUM [0] :=                                                        00235100
      NUM [1] := 0;                                                     00235200
      ELEMENTOVIAVEL := TRUE;                                           00235300
      ESTADOSUCESSOR := TABELADEESTADOBOOL [INDTAB];                    00235400
      IF LIGADESLIGA OR DEMANDAIGUAL                                    00235500
      THEN                                                              00235600
                                                                        00235700
        %%   PODE LIGAR OU DESLIGAR UNIDADES    %%                     00235800
                                                                        00235900
        FOR UNID := 1 STEP 1 UNTIL NUMUNIDPONTA DO                      00236000
           BEGIN                                                        00236100
              ESTBOOL := ESTPAI (UNID);                                 00236200
              IF ESTSUCESSOR (UNID) EQV ESTBOOL                         00236300
              THEN                                                      00236400
                                                                        00236500
                 %%  A UNIDADE PERMANECEU NO ESTADO ANTERIOR,   %%      00236600
                 %%   INCREMENTA O TEMPO NO ESTADO.             %%      00236700
                                                                        00236800
                 TEMPOSUCESSOR [UNID] := TEMPOPAI [UNID] + 1            00236900
              ELSE                                                      00237000
                                                                        00237100
                 %%  VERIFICA SE A UNIDADE PODE MUDAR DE ESTADO. %%     00237200
                 %%  DEPENDENDO DO TEMPO EM QUE ESTA NO  MESMO . %%     00237300
                 %%  VERIFICA SE A COMBINACAO NAO LIGA OU DESLI- %%     00237400
                 %%  GA MAIS UNIDADES DO QUE O PERMITIDO.        %%     00237500
                                                                        00237600
                 IF TEMPOPAI [UNID] >= TEMPMIN [EST, UNID] AND          00237700
                    NUM [EST] := * + 1 <= NUMMAX [EST]                  00237800
                 THEN                                                   00237900
                    TEMPOSUCESSOR [UNID] := 1                           00238000
                 ELSE                                                   00238100
                    BEGIN                                               00238200
                       ELEMENTOVIAVEL := FALSE;                         00238300
```

```
                    UNID := NUMUNIDPONTA;                                  00238400
                 END                                                       00238500
           END                                                             00238600
     ELSE                                                                  00238700
                                                                           00238800
        %%   PODE LIGAR OU DESLIGAR UNIDADES DEPENDENDO    %%             00238900
        %%   DA DEMANDA.                                   %%             00239000
                                                                           00239100
        FOR UNID := 1 STEP 1 UNTIL NUMUNIDPONTA DO                         00239200
           BEGIN                                                           00239300
              ESTBOOL := ESTPAI (UNID);                                    00239400
              IF ESTSUCESSOR (UNID) EQV ESTBOOL                            00239500
              THEN                                                         00239600
                                                                           00239700
                 %%   A UNIDADE PERMANECEU NO MESMO ESTADO    %%         00239800
                 %%   ANTERIOR. INCREMENTA O TEMPO.           %%         00239900
                                                                           00240000
                 TEMPOSUCESSOR [UNID] := TEMPOPAI [UNID] + 1               00240100
              ELSE                                                         00240200
                                                                           00240300
                 %%  VERIFICA SE A UNIDADE PODE MUDAR  %%                  00240400
                 %%  DE ESTADO, DEPENDENDO DO TEMPO    %%                  00240500
                 %%  EM QUE ESTA NO MESMO.   VERIFICA  %%                  00240600
                 %%  SE A COMBINACAO NAO LIGA OU DES-  %%                  00240700
                 %%  LIGA MAIS UNIDADES DO QUE O PER-  %%                  00240800
                 %%  MITIDO.                           %%                  00240900
                                                                           00241000
                 IF ( ESTSUCESSOR (UNID) EQV DEMANDAAUMENTA )              00241100
                    AND TEMPOPAI [UNID] >= TEMPMIN [EST,UNID]              00241200
                    AND NUM [EST] := * + 1 <= NUMMAX [EST]                 00241300
                 THEN                                                      00241400
                    TEMPOSUCESSOR [UNID] := 1                              00241500
                 ELSE                                                      00241600
                    BEGIN                                                  00241700
                       ELEMENTOVIAVEL := FALSE;                            00241800
                       UNID := NUMUNIDPONTA;                               00241900
                    END;                                                   00242000
           END;                                                            00242100
```

```
                                                                        00242200
                                                                        00242300
          %%   VERIFICA SE O SUCESSOR E' VIAVEL. SE HAVIA    %%       00242400
          %%   UM SUCESSOR VIAVEL COM O MESMO ESTADO   OU   %%       00242500
          %%   "PAI" , ENTAO SO LIGA OU DESLIGA SE O CUS-   %%       00242600
          %%   TO DE GERACAO FOR MENOR.                     %%       00242700
                                                                        00242800
                                                                        00242900
          IF ELEMENTOVIAVEL AND                                         00243000
             ((( DEMANDAAUMENTA OR DEMANDADIMINUI) AND                  00243100
               TABELADECUSTOS [INDTAB] <= CUSTOLIMITE) OR               00243200
              DEMANDAIGUAL)                                             00243300
          THEN                                                          00243400
             BEGIN                                                      00243500
                RESERVASUCESSOR     := TABELADERESERVA [INDTAB];        00243600
                SOMAGERMAXSUCESSOR := TABELADESOMAGERMAX [INDTAB];      00243700
                CALCULARCUSTOTOTAL;                                     00243800
             END                                                        00243900
          ELSE;                                                         00244000
       END DA BUSCA NA TABELA;                                          00244100
    SELECIONARSUCESSORES;                                               00244200
END DA FECHARELEMENTO;                                                  00244300



PROCEDURE CRIARTABELA;                                                  00244400
                                                                        00244500
                                                                        00244600
BEGIN                                                                   00244700



   PROCEDURE EXPANDIR                                                   00244800
      (UNIDA, NUMDESLANTERIOR);                                         00244900
   VALUE                                                                00245000
      UNIDA,                                                            00245100
      NUMDESLANTERIOR;                                                  00245200
   INTEGER                                                              00245300
```

```
      UNIDA,                                                              00245400
      NUMDESLANTERIOR;                                                    00245500
   BEGIN                                                                  00245600
                                                                          00245700
                                                                          00245800
    %%    PARA UM DETERMINADO ESTAGIO, CALCULA O CUSTO DE     %%         00245900
    %%    GERACAO PARA CADA COMBINACAO QUE SATISFAZ     AS    %%         00246000
    %%    RESTRICOES DE DEMANDA , RESERVA E CONDICAO    DO    %%         00246100
    %%    ESTADO.                                             %%         00246200
                                                                          00246300
                                                                          00246400
      IF UNIDA > NUMUNIDPONTA                                             00246500
      THEN                                                                00246600
         IF TRUE                                                          00246700
         THEN                                                             00246800
            BEGIN                                                         00246900
               MAXGERMAXLIG :=                                            00247000
               SOMAGERMIN   :=                                            00247100
               SOMAGERMAX   := 0;                                         00247200
               FOR J := 1 STEP 1 UNTIL NUMUNID DO                         00247300
                  IF UNIDLIGADA [J]                                       00247400
                  THEN                                                    00247500
                     BEGIN                                                00247600
                        GERMAX := GERMAXIMA [J];                          00247700
                        IF GERMAX > MAXGERMAXLIG                          00247800
                        THEN                                              00247900
                           MAXGERMAXLIG := GERMAX                         00248000
                        ELSE;                                             00248100
                        SOMAGERMIN   := * + GERMINIMA [J];                00248200
                        SOMAGERMAX   := * + GERMAX;                       00248300
                     END                                                  00248400
                  ELSE;                                                   00248500
               RESERVASUCESSOR := MAXGERMAXLIG;                           00248600
               IF  SOMAGERMAX >= DEMANDASUCESSOR + RESERVASUCESSOR        00248700
               THEN                                                       00248800
                  BEGIN                                                   00248900
                     CALCULARCUSTODODESPACHO;                             00249000
                     IF CUSTODESPACHO < INFINITOPOSITIVO                  00249100
```

```
                    THEN                                                    00249200
                       BEGIN                                                00249300
                          IF CUSTODESPACHO < CUSTOSUBESTIMADO                00249400
                          THEN                                               00249500
                             BEGIN                                           00249600
                                CUSTOSUBESTIMADO := CUSTODESPACHO;           00249700
                                FOR J := 1 STEP 1 UNTIL NUMUNIDPONTA DO00249800
                                   ESTMINIMO (J) := UNIDLIGADA (J);          00249900
                             END                                             00250000
                          ELSE;                                              00250100
                          NUMELEMTAB := * + 1;                               00250200
                                                                             00250300
                          %%  DESCARREGA A COMBINACAO VIAVEL  %%             00250400
                          %%  NAS TABELAS.                    %%             00250500
                                                                             00250600
                          FOR J := 1 STEP 1 UNTIL NUMUNIDPONTA DO            00250700
                             TABESTADO (NUMELEMTAB, J) :=                    00250800
                                                  ESTADO (J);                00250900
                          TABELADECUSTOS (NUMELEMTAB) := CUSTODESPACHO;00251000
                          TABELADESOMAGERMAX (NUMELEMTAB) :=                 00251100
                                                   SOMAGERMAX ;00251200
                          TABELARESERVA (NUMELEMTAB) :=                      00251300
                                                RESERVASUCESSOR;00251400
                          WRITE (TABELADEGERACAO (NUMELEMTAB, *),            00251500
                                                NUMUNID, GERDTIMA);00251600
                       END                                                   00251700
                    ELSE;                                                    00251800
                 END                                                         00251900
              ELSE;                                                          00252000
           END                                                               00252100
        ELSE                                                                 00252200
     ELSE                                                                    00252300
        BEGIN                                                                00252400
           UNIDLIGADA (UNIDA) := TRUE;                                       00252500
           EXPANDIR (UNIDA + 1, NUMDESLANTERIOR);                            00252600
           IF NUMDESLANTERIOR < NUMUNIDPONTA                                 00252700
           THEN                                                              00252800
              BEGIN                                                          00252900
```

```
                NUMDESLANTERIOR := * + 1;                                  00253000
                UNIOLIGADA [UNIDA] := FALSE;                               00253100
                EXPANDIR (UNIDA + 1, NUMDESLANTERIOR);                     00253200
             END                                                           00253300
          ELSE;                                                            00253400
       END;                                                                00253500
END DA EXPANDIR;                                                           00253600
                                                                           00253700
                                                                           00253800
%%   PARA CADA ESTAGIO CHAMA A ROTINA EXPANDIR QUE CRIA    %%              00253900
%%   TODAS AS COMBINACOES VIAVEIS DE ESTADO.              %%              00254000
%%   CALCULA UMA SUBESTIMATIVA DO CUSTO DE GERACAO PARA   %%              00254100
%%   CADA ESTAGIO.                                        %%              00254200
                                                                           00254300
                                                                           00254400
ESTIMATIVA := 0;                                                           00254500
FOR J := 1 STEP 1 UNTIL NUMUNIDPONTA DO                                    00254600
   BEGIN                                                                   00254700
      ESTSEMPRELIGADA (J)    := TRUE;                                      00254800
      ESTSEMPREDESLIGADA (J) := TRUE;                                      00254900
   END;                                                                    00255000
FOR ESTAGIO := HORPLAN STEP -1 UNTIL 1 DO                                  00255100
   BEGIN                                                                   00255200
      DEMANDASUCESSOR := DEMANDA [ESTAGIO];                                00255300
      RESERVASUCESSOR := RESERVA [ESTAGIO];                                00255400
      CUSTOSUBESTIMADO := INFINITOPOSITIVO;                                00255500
      SUBESTAGIO [ESTAGIO] := ESTIMATIVA;                                  00255600
      FOR J := 1 STEP 1 UNTIL NUMUNIO DO                                   00255700
         UNIOLIGADA [J] := TRUE;                                           00255800
      EXPANDIR (1, 0);                                                     00255900
      FOR J := 1 STEP 1 UNTIL NUMUNIDPONTA DO                              00256000
         BEGIN                                                             00256100
            ESTSEMPRELIGADA (J)    :=                                      00256200
               ESTSEMPRELIGADA (J) AND ESTMINIMO (J);                      00256300
            ESTSEMPREDESLIGADA (J) :=                                      00256400
               NOT (ESTSEMPREDESLIGADA (J) IMP ESTMINIMO (J));             00256500
         END;                                                              00256600
      ESTIMATIVA := * + CUSTOSUBESTIMADO;                                  00256700
```

```
      INDICEDATABELA [ESTAGIO - 1] := NUMELEMTAB + 1;                    00256800
   END;                                                                  00256900
SUBESTAGIO [1] := ESTIMATIVA - CUSTOSUBESTIMADO;                         00257000
INDICEDATABELA [HORPLAN] := 1;                                           00257100
                                                                         00257200
                                                                         00257300
                                                                         00257400
%%  ELIMINA DA TABELA OS ELEMENTOS QUE NUNCA PODERAO  %%                 00257500
%%  FAZER PARTE DE UMA POLITICA OTIMA.                %%                 00257600
                                                                         00257700
                                                                         00257800
NOVOIND := 0;                                                            00257900
ESTAGIO := HORPLAN;                                                      00258000
PROXIND := 1;                                                            00258100
FOR INDTAB := 1 STEP 1 UNTIL NUMELEMTAB DO                               00258200
   BEGIN                                                                 00258300
      ESTADOTABATUAL := TABELADEESTADO [INDTAB];                         00258400
      FOR J := 1 STEP 1 UNTIL NUMUNIDPONTA DO                            00258500
         IF NOT ELEMENTOEXISTE :=                                        00258600
            ((ESTSEMPRELIGADA (J) IMP ESTTABATUALBOOL (J)) AND           00258700
             (NOT (ESTSEMPREDESLIGADA (J) AND ESTTABATUALBOOL (J))))     00258800
         THEN                                                            00258900
            J := NUMUNIDPONTA                                            00259000
         ELSE;                                                           00259100
      IF ELEMENTOEXISTE                                                  00259200
      THEN                                                               00259300
         BEGIN                                                           00259400
            NOVOIND := * + 1;                                            00259500
            IF NOVOIND NEQ INDTAB                                        00259600
            THEN                                                         00259700
               BEGIN                                                     00259800
                  TABELADEESTADO [NOVOIND] :=                            00259900
                                    TABELADEESTADO [INDTAB];             00260000
                  TABELADECUSTOS [NOVOIND]  :=                           00260100
                                    TABELADECUSTOS [INDTAB];             00260200
                  TABELADERESERVA [NOVOIND] :=                           00260300
                                   TABELADERESERVA [INDTAB];             00260400
                  TABELADESOMAGERMAX [NOVOIND] :=                        00260500
```

```
                                                TABELADESOMAGERMAX [INDTAB];00260600
                    WRITE(TABELADEGERACAO [NOVOIND, *], *,                  00260700
                                                TABELADEGERACAO [INDTAB, *]);00260800
                       END                                                  00260900
                    ELSE;                                                   00261000
                    IF INDTAB >= PROXIND                                    00261100
                    THEN                                                    00261200
                       BEGIN                                                00261300
                          INDICEDATABELA [ESTAGIO] := NOVOIND;              00261400
                          ESTAGIO := ESTAGIO - 1;                           00261500
                          PROXIND := INDICEDATABELA [ESTAGIO];              00261600
                       END                                                  00261700
                    ELSE;                                                   00261800
                 END                                                        00261900
              ELSE;                                                         00262000
           END;                                                             00262100
NUMELEMTAB := NOVOIND;                                                      00262200
INDICEDATABELA [0] := NOVOIND + 1;                                          00262300
WRITE (IMPRESS, *//, INDICEDATABELA);                                       00262400
                                                                            00262500
                                                                            00262600
%%   GRAVA AS TABELAS GERADAS EM DISCO.   %%                                00262700
                                                                            00262800
                                                                            00262900
WRITE (TABELA, *, NUMELEMTAB,                                               00263000
                  INDICEDATABELA [*],                                       00263100
                  SUBESTAGIO [*],                                           00263200
                  FOR I := 1 STEP 1 UNTIL NUMELEMTAB DO                     00263300
                  TABELADEESTADO [I],                                       00263400
                  FOR I := 1 STEP 1 UNTIL NUMELEMTAB DO                     00263500
                  TABELADECUSTOS [I],                                       00263600
                  FOR I := 1 STEP 1 UNTIL NUMELEMTAB DO                     00263700
                  TABELADERESERVA [I],                                      00263800
                  FOR I := 1 STEP 1 UNTIL NUMELEMTAB DO                     00263900
                  TABELADESOMAGERMAX [I]);                                  00264000
 FOR I := 1 STEP 1 UNTIL NONUNID DO                                         00264100
    WRITE (TABELA, *, FOR J := 1 STEP 1 UNTIL NUMELEMTAB DO                 00264200
                         TABELADEGERACAO [J, I]);                           00264300
```

```
   LOCK (TABELA, CRUNCH);                                                00264400
   DISPLAY ("TABELA GRAVADA.");                                          00264500
END DA CRIARTABELA;                                                      00264600


PROCEDURE LERTABELA;                                                     00264700
                                                                         00264800
                                                                         00264900
BEGIN                                                                    00265000
   READ  (TABELA, *, NUMELEMTAB,                                         00265100
                     INDICEDATABELA (*),                                 00265200
                     SUBESTAGIO (*),                                     00265300
                     FOR I := 1 STEP 1 UNTIL NUMELEMTAB DO               00265400
                     TABELADEESTADO (I),                                 00265500
                     FOR I := 1 STEP 1 UNTIL NUMELEMTAB DO               00265600
                     TABELADECUSTOS (I),                                 00265700
                     FOR I := 1 STEP 1 UNTIL NUMELEMTAB DO               00265800
                     TABELADERESERVA (I),                                00265900
                     FOR I := 1 STEP 1 UNTIL NUMELEMTAB DO               00266000
                     TABELADESOMAGERMAX (I));                            00266100
    FOR I := 1 STEP 1 UNTIL NUMUNID DO                                   00266200
      READ (TABELA, *, FOR J := 1 STEP 1 UNTIL NUMELEMTAB DO             00266300
                       TABELADEGERACAO (J, I));                          00266400
   CLOSE (TABELA);                                                       00266500
END DA LERTABELA;                                                        00266600
                                                                         00266700
                                                                         00266800
%          P R O G R A M A    P R I N C I P A L                          00266900
%          - - - - - - - -    - - - - - - - - -                          00267000
                                                                         00267100
                                                                         00267200
   PROCESSAMENTOLIMITE := (MYSELF.MAXPROCTIME / 24@=7) * .9;             00267300
   LEREIMPRIMIRDADOSDOSISTEMA;                                           00267400
   LEREIMPRIMIRDADOSGERAIS;                                              00267500
   IF CRIATABELA                                                         00267600
   THEN                                                                  00267700
      BEGIN                                                              00267800
```

```
      MONTARPRECOS;                                                    00267900
      ORDENARPRECOS;                                                   00268000
      CRIARTABELA;                                                     00268100
      IMPRIMIRTABELA;                                                  00268200
      GO TO FINAL;                                                     00268300
   END                                                                 00268400
ELSE                                                                   00268500
   LERTABELA;                                                          00268600
CALCULARSUPESTIMATIVADOESTAGIO;                                        00268700
CALCULARDEMANDAMAXIMAFUTURA;                                           00268800
VERIFICARVARIACAODADEMANDA;                                            00268900
IF IMPVARIACAODADEMANDA                                                00269000
THEN                                                                   00269100
   IMPRIMIRVARIACAODADEMANDA;                                          00269200
IMPRIMIRSUBESTIMATIVADOESTAGIO;                                        00269300
IMPRIMIRMENSAGEM ("***** INICIO DO ALGORITMO *****");                  00269400
INICIALIZARLISTADEELEMENTOS;                                           00269500
CASE ALGORITMO OF                                                      00269600
BEGIN                                                                  00269700
   1: WHILE ESCOLHERELEMENTOPORFMINIMO DO                              00269800
      IF ENCONTROUALVO                                                 00269900
      THEN                                                             00270000
         BEGIN                                                         00270100
            ELEMENTOABERTO [INDICE] := FALSE;                          00270200
            RECUPERARCAMINHO;                                          00270300
            IMPRIMIRCAMINHO;                                           00270400
            IMPRIMIRMENSAGEM("*****SOLUCAO OTIMA*****");               00270500
            GO TO FINAL;                                               00270600
         END                                                           00270700
      ELSE                                                             00270800
         IF MYSELF.PROCESSTIME > PROCESSAMENTOLIMITE                   00270900
         THEN                                                          00271000
            GO TO FINAL                                                00271100
         ELSE                                                          00271200
            FECHARELEMENTO;                                            00271300
      IMPRIMIRMENSAGEM("LISTA ABERTA VAZIA");                          00271400
   2: WHILE ESCOLHERELEMENTOPORFMAXIMO DO                              00271500
      IF ENCONTROUALVO                                                 00271600
```

```
            THEN                                                                00271700
               BEGIN                                                            00271800
                  ELEMENTOABERTO [INDICE] := FALSE;                             00271900
                  RECUPERARCAMINHO;                                             00272000
                  IMPRIMIRCAMINHO;                                              00272100
                  BOUND := CUSTOELEMENTO [INDICE] - EPSILON;                    00272200
                  ELIMINARELEMENTOSPELOBOUND;                                   00272300
                  REDEFINIRESTIMATIVASUPERIOR;                                  00272400
                  IMPRIMIRESTIMATIVASUPERIOR;                                   00272500
               END                                                              00272600
            ELSE                                                                00272700
               IF MYSELF.PROCESSTIME > PROCESSAMENTOLIMITE                      00272800
               THEN                                                             00272900
                  GO TO FINAL                                                   00273000
               ELSE                                                             00273100
                  FECHARELEMENTO;                                               00273200
            IMPRIMIRMENSAGEM("*****LISTA ABERTA VAZIA*****");                   00273300
      END;                                                                      00273400
   END;                                                                         00273500
   FINAL:                                                                       00273600
      WRITE (IMPRESS, *//, NUMELEMFECHADO, NUMELEMABERTO,                       00273700
                           NUMELEMELIMABERTOS, NUMELEMELIMFECHADOS,             00273800
                           TIME (12) * 24@-7);                                  00273900
   END.                                                                         00274000
```