

UMA APLICAÇÃO DE PROGRAMAÇÃO INTEIRA
PRODUÇÃO DE LUBRIFICANTES NUMA REFINARIA

MICHAEL PETER MICHAILOW

TESE SUBMETIDA AO CORPO DOCENTE DA COOR-
DENAÇÃO 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:

Wilson Carneiro Filho
Roberto Luiz Gabard
Adelino
Rui Pereira

Rio de Janeiro
Estado do Rio de Janeiro - Brasil
Dezembro de 1977

A G R A D E C I M E N T O S

Ao professor Nelson Maculan Filho, pelo apoio e orientação na realização desse trabalho.

A Andrea Levy e a Paulo Fernando Hamacher pelas valiosas críticas e sugestões oferecidas.

A Ricardó N. S. do Valle pela utilização experimental de seu programa do método de Benders.

1. R E S U M O

O presente trabalho consiste de um modelo de programação linear mista, cujo equacionamento contém conjuntos de variáveis de decisão 0-1 (conjuntos ordenados especiais).

Para a sua execução foi utilizado o algoritmo branch and-bound, acrescido de controles para limitar a busca de soluções inteiras, de modo a se obter soluções a um tempo não proibitivo.

O modelo foi aplicado à determinação da programação diária do sistema de produção de óleos lubrificantes e parafinas numa refinaria de petróleo, com o objetivo de maximizar a sua produção, evitar importação pelo não fornecimento de algum tipo de óleo, e reduzir a instabilidade na operação do sistema. São satisfeitas as restrições do funcionamento dos processos, qualidade dos óleos produzidos e capacidade de armazenamento.

Foi estudado também um acoplamento adequado entre programações sucessivas, para evitar que a solução ótima para um período prejudique as soluções para os períodos seguintes.

A B S T R A C T

This work consists of a mixed linear programming model, whose equationing contains sets of 0 - 1 decision variables (called special ordered sets).

To execute it, the branch-and-bound algorithm was used with some controls to limit the search for integer solutions, so that solutions may be obtained at a non prohibitive execution time.

The model was applied to the determination of a daily programming for the production system of lubricant oils and paraffines in an oil refinerie, with the objective of maximizing its production, avoiding market losses because of non production of any type of oil, reducing the instability functioning of the processes, the quality of the oils produced and storage capacities are satisfied.

An adequate coupling between programming relative to subsequent periods was also taken into account so as to make sure that the optimal solution for a given period will turn impossible any solution to the next periods.

2. Í N D I C E

1. RESUMO

2. ÍNDICE

3. INTRODUÇÃO

4. DESCRIÇÃO DO PROBLEMA

5. SOLUÇÃO PROPOSTA

5.1. Descrição do Modelo

5.2. Terminologia

5.3. Equacionamento

5.4. Algoritmo Utilizado

6. RESULTADOS

7. AVALIAÇÃO DOS RESULTADOS

8. CONCLUSÕES

9. REFERÊNCIAS BIBLIOGRÁFICAS

Anexo : Programas

3. INTRODUÇÃO

A iniciativa desse trabalho, nasceu da necessidade de se substituir o procedimento que vinha sendo utilizado na determinação de um roteiro para o funcionamento de todos os processos industriais, no sistema de produção de óleos lubrificantes e parafinas de uma refinaria de petróleo.

Dado o grande número de unidades industriais independentes e tipos de óleos envolvidos, o procedimento tradicional consiste em se experimentar uma relação de óleos a serem processados em cada unidade num dia, e fazer trocas sempre que alguma restrição de armazenamento não seja satisfeita, prosseguindo-se assim dia após dia. Ou seja o programador da produção se satisfazia em encontrar uma solução viável para o problema por um método de tentativas.

O processo antigo consome muito tempo para os cálculos, e para não acarretar iterações inúteis exige do programador um grande conhecimento histórico do funcionamento do sistema.

Esse trabalho se propõe então a determinar uma programação diária para o funcionamento do sistema indicando quais os óleos que devem ser processados em cada unidade, para um horizonte de curto prazo (aproximadamente quinzenal), de modo a atender os pedidos do mercado.

A experiência do programador é aproveitada na determinação de preferências na seleção dos óleos a serem processados, níveis mínimos em cada tanque e custos de perda de mercado (óleos não fornecidos).

Assim o modelo procurou refletir os principais critérios de decisão usados no procedimento tradicional.

4. DESCRIÇÃO DO PROBLEMA

O sistema estudado consiste de um conjunto de processos em série que recebe como entrada uma vazão diária de óleo cru, e procura atender os pedidos de óleos lubrificantes básicos e parafinas feitos pelo mercado nacional, durante um período pré-fixado. São produzidos ainda em cada unidade diversos resíduos para a petroquímica.

Tanto os óleos lubrificantes como as parafinas estão classificados em 7 tipos, por ordem crescente de viscosidade:

Óleos Leves	1	spindle	(SP)
	2	neutro leve	(NL)
	3	neutro médio	(NM)
	4	neutro pesado	(NP)
Óleos Pesados	5	bright stock	(BS)
	6	cilindro 1	(C1)
	7	cilindro 2	(C2)

Atualmente apenas os 5 primeiros tipos de óleo são produzidos. Somente estes, portanto, constarão no modelo.

Os processos industriais estão classificados em 7 categorias:

- 1 destilação
- 2 desasfaltação
- 3 extração de aromáticos
- 4 desparafinação
- 5 desoleificação da parafina
- 6 hidrogenação do óleo desparafinado
- 7 hidrogenação da parafina dura

Os parques de tanques na saída de cada processo são:

- 1 óleos destilados
- 2 óleos desasfaltados
- 3 óleos refinados
- 4 óleos desparafinados
- 5 parafinas duras
- 6 óleos básicos
- 7 parafinas produto

Há 2 modalidades de processamento: contínua e bloqueada. A modalidade contínua utiliza um único tipo de óleo como entrada e o fraciona em diversos subprodutos. O processo pára apenas por defeito ou manutenção. Na modalidade bloqueada são utilizados alternativamente (não simultaneamente) diversos tipos de óleo, logo periodicamente ocorre uma troca do tipo de óleo processado.

Os processos 1 e 2 atualmente funcionam na modalidade contínua. Os demais são bloqueados.

Dada a seguinte terminologia:

- OE óleo na entrada
 PE parque de tanques da entrada
 OS óleo na saída
 PS parque de tanques na saída

PROCESSO 1				PROCESSO 2				PROCESSO 3			
OE	PE	OS	PS	OE	PE	OS	PS	OE	PE	OS	PS
CRU		SP	1					SP	1	SP	3
		NL	1					NL	1	NL	3
		NM	1					NM	1	NM	3
		NP	1					NP	1	NP	3
		BS/C2	1	BS/C2	1	BC/C2	2	BS/C2	2	BS	3
		Cl	1	Cl	1	Cl	2	Cl	2	Cl	3

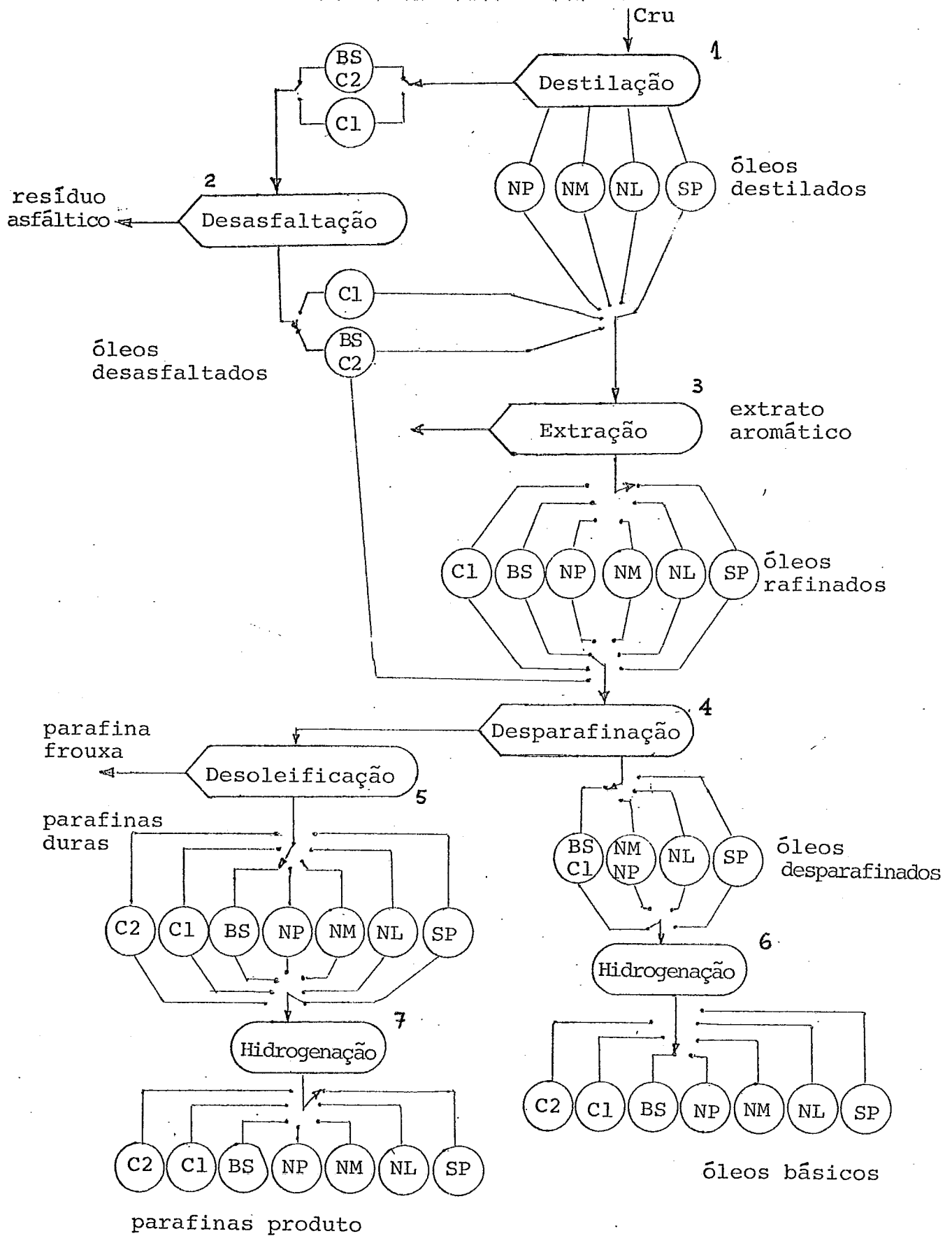
PROCESSOS 4-5				PROCESSOS 4-6				PROCESSO 7			
OE	PE	OS	PS	OE	PE	OS	PS	OE	PE	OS	PS
SP	3	SP	5	SP	3	SP	6	SP	5	SP	7
NL	3	NL	5	NL	3	NL	6	NL	5	NL	7
NM	3	NM	5	NM	3	NM	6	NM	5	NM	7
NP	3	NP	5	NP	3	NP	6	NP	5	NP	7
BS	3	BS	5	BS	3	BS	6	BS	5	BS	7
C1	3	C1	5	C1	3	C1	6	C1	5	C1	7
BS/C2	2	C2	5	BS/C2	3	C2	6	C2	5	C2	7

Já que não são produzidos os óleos 6 e 7, só existem alternativas de escolha sobre 3 processos independentes: 3, 4 e 7. O processo 4 funciona em sincronismo com os processos 5 e 6 elaborando o mesmo tipo de óleo, pois o parque de tanques 4 funciona apenas como pulmão, e não possui capacidade de armazenamento suficiente.

A programação mais crítica é a das unidades 3 e 4 que constituem 2 pontos de estrangulamento do fluxo através do sistema. O processo 7 possui uma capacidade de produção bastante superior ao volume de óleo que precisa processar, sua programação portanto, é mais folgada.

Um esquema simplificado da planta de unidades de processamento e parques de tanques está na página seguinte.

PLANTA SIMPLIFICADA DO SISTEMA



O objetivo do trabalho é determinar um roteiro para o funcionamento do sistema, ou seja, determinar o tipo de óleo, a vazão por hora e a duração do processamento para cada unidade bloqueada do sistema de modo a satisfazer os pedidos de produtos finais pelo mercado nacional.

Devem ser obedecidas as seguintes restrições:

- necessidades mínimas de produção de qualquer tipo de óleo
- a capacidade de armazenamento dos tanques
- a impossibilidade de produzir mais de um óleo num processo bloqueado simultaneamente
- a proibição de se fazer uma troca entre dois óleos que tenham viscosidades muito diferentes entre si, para evitar que um resto do óleo antigo na unidade contamine o novo
- a necessidade de paralisar algum processo para manutenção

Deve-se evitar, também, a produção de óleo em excesso numa unidade (superior à capacidade de armazenamento) pois esse excesso é retirado do sistema como combustível que possui um preço de venda bastante inferior ao de um lubrificante.

A duração de uma campanha do processamento varia, geralmente, de 3 a 5 dias e deve-se evitar trocas muito frequentes de óleos (pois estas exigem uma parada e consequente resfriamento da unidade)

A vazão diária por um processo possui um limite máximo (vazão máxima de entrada). Por questão de eficiência, deve-se tentar produzir a esse nível.

5. SOLUÇÃO PROPOSTA

5.1. Descrição do Modelo

Foi fixada a vazão de entrada em cada processo, no seu valor máximo e discretizado o período de processamento, de modo que o modelo precisa apenas decidir qual o tipo de óleo a ser produzido em cada processo, durante uma unidade de tempo.

Construiu-se então um modelo de programação linear mista, cujo equacionamento contém restrições referentes a:

- a) capacidades dos tanques
- b) bloqueamento dos processos (exclusividade na operação)
- c) qualidade dos óleos produzidos
restrições sobre a diferença dos índices dos óleos envolvidos numa troca
- d) frequência máxima de troca de óleo
- e) necessidades mínimas e limites máximos de produção para cada tipo de óleo

e tem como objetivo:

- a) minimizar as importações (demanda não atendida)
- b) minimizar a instabilidade no funcionamento dos processos causada pelas trocas de óleo
- c) produzir o óleo de maior valor em cada processo
- d) minimizar a produção de excesso (superior à capacidade de armazenamento)

Dado que o processo 7 possui uma capacidade excedente de produção, sua programação não é crítica e não será feita pelo modelo.

5.2. Simbologia

PARÂMETROS

ARQLUB	arquivo onde é gravado o equacionamento
CRU	vazão de óleo cru na entrada do sistema
DADOVAR	arquivo de onde são lidos os dados variáveis
I	índice de tipo de processo
J	índice de tipo de óleo
K	índice de dia de programação
PERI	período da programação (em unidades de tempo)
ROW	nome da restrição
UNI	unidade de tempo (nº inteiro de dias)
VAR	nome da variável
ACA(I,J,1)	coeficiente da variável do fluxo de entrada no tanque de óleo J no parque I
ACA(I,J,2)	coeficiente da variável do fluxo de saída no tanque de óleo J no parque I
ACA(I,J,3)	coeficiente da variável do fluxo de saída no tanque de óleo J no parque I
CEXC(I,J)	custo de excesso do óleo J no tanque do parque I (por m ³)
CUSPAR(I)	custo de parada do processo I (por dia)
CUSTRO(I)	custo de troca de óleo no processo I (por unidade de diferença de índice dos óleos envolvidos na troca)
CPEROL(J)	custo de óleo básico J não fornecido ao mercado no período (por m ³)
CPERPA(J)	custo da parafina J não fornecida ao mercado no período (por m ³)

EF(I,J)	eficiência do processo I com o óleo J
IPATRO(I)	índice da variável associada a parada no processo I, nas equações de troca
JCO(I)	menor índice de óleo nos tanques do parque I
JCB(I)	menor índice de óleo no processo bloqueado I
JFI(I)	maior índice de óleo no processo bloqueado I
L(I,K)	0 processo I parado no dia K 1 processo I funcionando no dia K
OL2(J)	nome do óleo J
OLOR(J)	tipo de óleo processado no último dia da programação anterior
PMIN(I,J)	fração mínima de óleo J no tanque do parque I
PFIMIN(I,J)	" " " " " " " no último dia
PARQ6(I)	nome do parque de tanques I
PROC3(I)	nome do processo I
PROC6(I)	nome do processo I
QAT(I,J,K)	quantidade de óleo J no parque I no dia K
QI(I,J)	quantidade de óleo J no parque I no início da programação
SOL(J)	retirada média diária de óleo básico J
SPA(J)	retirada média diária
T(I,J)	capacidade do tanque de óleo J no parque I
VZ(I,J)	vazão de entrada diária do óleo J no processo I

Restrições

UNI(I,K)	unicidade de processamento
	I = 3, 4
	K = 1, _____, PERI

- CAP(I,J,K) capacidade dos tanques
 I = 1, 2, 3, 6, 7
 J = JCO(I), ____, 5
 K = 1, ____, PERI
- ETRO(I,K) equação da diferença entre os índices dos óleos numa troca
 I = 3, 4
- EMPO(I,K) equação do módulo da diferença acima, quando esta é positiva
- EMNE(I,K) equação do módulo da diferença acima quando esta é negativa
- RTRO(I,K) restrições para o módulo da diferença acima

Variáveis

- X(I,J,K) variáveis de decisão 0 - 1 $\left\{ \begin{array}{l} = 0 \text{ não produzir} \\ = 1 \text{ produzir} \end{array} \right.$
 I = 3, 4 processo
 J = JCB(I), ____, 5 óleo
 K = 1, ____, PERI dia
- EXC(I,J,K) quantidade em excesso no tanque (não negativa)
 I = 1, 2, 3, 6, 7 parque
 J = JCO(I), ____, 5 óleo
 K = 1, ____, PERI dia
- PERO(J) quantidade média diária não fornecida ao mercado do óleo básico J (não negativa)
 J = 1, ____, 5
- PERP(J) quantidade média diária não fornecida ao mercado da parafina J
 J = 1, ____, 5 (não negativa)

DTRO(I,K) diferença entre os índices dos óleos envolvidos
 numa troca. (irrestrita em sinal)

I = 3, 4

K = 1, ____, PERI

MODT(I,K) módulo da diferença acima

KEY(I,K) = 0 não há troca no início do intervalo K
 = 1 há

VAZ(2,5,k) vazão do óleo 5 no processo 2 no intervalo K

QAT(I,J,K) estoque operacional nos tanques

I= 1,2,3,6, 7 parque

J= JCO(I), __5 óleo

K= 1, ____, PERI fim de intervalo K

5.3. Equacionamento

Restrições de exclusividade

Como a duração do processamento, somente um óleo pode ser produzido durante cada intervalo de tempo.

Há uma variável $X(I,J,K)$ do tipo 0 - 1 associada a produção do óleo J no processo I no intervalo K (vale 1 quando o óleo J é produzido). Como apenas os processos 3 e 4 estão sujeitos a alternativas de decisão, temos para um horizonte de $PERI$ intervalos de tempo, $2 * PERI$ restrições da forma:

$$\sum_J X(I,J,K) \leq 1$$

$$X(I,J,K) \in \{0,1\}$$

$$I = 3,4$$

$$K = 1, \dots, PERI$$

$$J = 1, \dots, 5$$

Se $L(I,K) = 0$ (processo parado para manutenção)
então $X(I,J,K) = 0$ para todo J

Restrições de armazenamento

A equação de conservação do fluxo para os tanques de todos os parques durante cada intervalo de tempo, indica que o estoque de óleo J ao fim do intervalo K é igual ao estoque ao fim do intervalo $K - 1$, mais o volume fornecido ao tanque menos o volume retirado no intervalo K .

O número de tanques para os parques 1, 2, 3, 6, 7 é respectivamente 5, 1, 5, 5, 5

Tanques do parque 1

$$\begin{aligned} \text{QAT}(1, J, K) &= \text{QAT}(1, J, K-1) + \text{CRU} * \text{EF}(1, 5) * \text{L}(1, K) * \text{UNI} - \\ &\quad - \text{VZ}(3, J) * \text{UNI} * \text{X}(3, J, K) - \text{EXC}(1, J, K) \end{aligned}$$

J = 1, ____, 4

$$\begin{aligned} \text{QAT}(1, 5, K) &= \text{QAT}(1, 5, K-1) + \text{CRU} * \text{EF}(1, 5) * \text{L}(1, K) * \text{UNI} - \\ &\quad - \text{UNI} * \text{VAZ}(2, 5, K) \end{aligned}$$

Tanques do parque 2

$$\begin{aligned} \text{QAT}(2, 5, K) &= \text{QAT}(2, 5, K-1) + \text{EF}(2, 5) * \text{UNI} * \text{VAZ}(2, 5, K) - \text{VZ}(3, 5) * \\ &\quad * \text{UNI} * \text{X}(3, 5, K) - \text{EXC}(2, 5, K) \end{aligned}$$

Tanques do parque 3

$$\begin{aligned} \text{QAT}(3, J, K) &= \text{QAT}(3, J, K-1) + \text{VZ}(3, J) * \text{EF}(3, J) * \text{UNI} * \text{X}(3, J, K) - \\ &\quad - \text{VZ}(4, J) * \text{UNI} * \text{X}(4, J, K) - \text{EXC}(3, J, K) \end{aligned}$$

J = 1, ____, 5

Tanques do parque 4

Funcionam apenas como pulmão entre os processos 4 e 6 que operam em sincronismo.

Tanques do parque 5

Foram acrescentados aos tanques respectivos do parque 7, já que não será feita a programação do processo 7.

Tanques do parque 6

$$QAT(6,J,K) = QAT(6,J,K-1) + VZ(4,J) * EF(4,J) * EF(6,J) * UNI * \\ * X(4,J,K) - SOL(J) * UNI + UNI * PERO(J) - EXC(6,J,K)$$

A conservação do fluxo nesses tanques inclui um volume variável PERO(J) de importação quando não for possível atender à demanda fixa SOL(J).

Tanques do parque 7

$$QAT(7,J,K) = QAT(7,J,K-1) + VZ(4,J) * (1 - EF(4,J)) * EF(5,J) * UNI * \\ * X(4,J,K) - SPA(J) * UNI + UNI * PERP(J) - EXC(7,J,K)$$

Restrições de capacidade

$$0 \leq QAT(I,J,K) \leq T(I,J)$$

$$K = 1, \dots, PERI$$

$$I = 1, 2, 3, 6, 7$$

$$J = JCO(I), \dots, 5$$

Restrições de estabilidade

Quanto maior a frequência e a amplitude das trocas de óleo, maior é a contaminação do óleo antigo sobre o novo, prejudicando sua qualidade. Foi então penalizado na função objetivo, o módulo da diferença entre os índices dos óleos envolvidos numa troca.

Quando a transição é de um óleo para o estado "parada do processo", ou vice versa, o custo associado corresponde ao aquecimento ou desaquecimento. Como esse custo é menor para os óleos mais leve. Ou seja, para esse modelo, parar um processo correspondente a passar para o óleo tipo 1

Equações de diferença entre os índices (1º dia)

$$DTRO(I,1) = \left\{ \sum_{JA=1}^5 JA * X(I,JA,1) - OLOR(I) + IPATRO(I) * \right. \\ \left. * \left[1 - \sum_{JB=1}^5 X(I,JB,1) \right] \right\} * L(I,1) * L(I,0)$$

I = 3,4

Equações de diferença entre os índices (os dias seguintes)

$$DTRO(I,K) = \left\{ \sum_{JA=1}^5 JA * X(I,JA,K) - \sum_{JB=1}^5 JB * X(I,JB,K-1) \right. \\ \left. + IPATRO(I) * \left[1 - \sum_{JC=1}^5 X(I,JC,K) - 1 + \sum_{JE=1}^5 X(I,JE,K-1) \right] \right\} * \\ * L(I,K) * L(I,K-1)$$

I = 3,4
K = 2, ____, PERI

$$MODT(I,K) \geq DTRO(I,K) \quad I = 3,4$$

$$MODT(I,K) \geq -DTRO(I,K) \quad K = 1, ____, PERI$$

Como MODT possui um custo na função objetivo,

quando $DTRO \geq 0$ então $MODT = DTRO$

quando $DTRO < 0$ então $MODT = -DTRO$

Restrições sobre a amplitude das trocas

Não podem ser feitas trocas entre óleos com viscosidades muito diferentes entre si (diferença de índices superior a 2).

$$\begin{aligned} \text{MODT}(I,K) \leq & 2 * \text{KEY}(I,K) + 4 * \left[1 - \sum_{\text{JA}=1}^5 \text{X}(I,\text{JA},K) \right] + 4 * \\ & * \left[1 - \sum_{\text{JB}=1}^5 \text{X}(I,\text{JB},K-1) \right] \quad \begin{array}{l} I = 3,4 \\ K = 2, _, \text{PERI} \end{array} \\ \text{MODT}(I,1) < & 2 * \text{KEY}(I,K) + 4 * \left[1 - \sum_{\text{JA}=1}^5 \text{X}(I,\text{JA},1) \right] \quad I = 3,4 \end{aligned}$$

Restrições sobre a frequência das trocas

A duração mínima de uma campanha de processamento é de 2 dias

$$\begin{aligned} \sum_{\text{KA}} \text{KEY}(I,\text{KA}) \leq & 1 \quad \begin{array}{l} I = 3,4 \\ K = (\text{LMAX} + 1), _, \text{PERI} \\ \text{KA} = (K - \text{LMAX}), _, K \\ \text{LMAX} = \text{inteiro de } \left(\frac{2}{\text{UNI}} \right) \end{array} \\ \text{KEY}(I,K) \in & \{0,1\} \end{aligned}$$

Limites máximos de produção

$$\begin{aligned} \sum_K \text{X}(I,J,K) \leq & \frac{\text{MAXDIA}(I,J)}{\text{UNI}} \quad \begin{array}{l} I = 3,4 \\ J = 1, _, 5 \\ K = 1, _, \text{PERI} \end{array} \end{aligned}$$

Limites mínimos de produção

$$\sum_K X(I,J,K) \geq \text{MINDIA}(I,J)/\text{UNI}$$

$$I = 3,4$$

$$J = 1, _, 5$$

$$K = 1, _, \text{PERI}$$

Função objetivo

O objetivo do modelo é minimizar:

a) custos de formação de excesso

$$\sum_K \sum_I \sum_J \text{CEXC}(I,J) * \text{EXC}(I,J,K)$$

$$I = 1,2,3,6,7$$

$$J = \text{JCO}(I), _, 5$$

$$K = 1, _, \text{PERI}$$

b) custos de produção dos óleos

$$\sum_K \sum_I \sum_J \text{VALOL}(I,J) * \text{UNI} * X(I,J,K)$$

$$I = 3,4$$

$$J = 1, _, 5$$

$$K = 1, _, \text{PERI}$$

c) custo de troca de óleo

$$\sum_I \sum_K \text{CUSTRO}(I) * [\text{MODT}(I,K) + \text{KEY}(I,K)]$$

$$I = 3,4$$

$$K = 1, _, \text{PERI}$$

d) custo de importação da demanda não atendida de óleos básicos e parafinas

$$\sum_J \text{UNI} * \text{PERI} * [\text{CPROL}(J) * \text{PERO}(J) + \text{CPERPA}(J) * \text{PERP}(J)]$$

$$J = 1, _, 5$$

5.4. Algoritmo Utilizado

Método Branch - and - Bound

Dado um problema de programação linear mista (PPLM), procura-se primeiro resolvê-lo como um problema de programação linear contínua (PPL), isto é, onde todas as variáveis são consideradas como contínuas. Esse problema contínuo é chamado PPL origem.

O método pesquisa, então, a obtenção de soluções inteiras, considerando cada PPL como um nó.

Isso é feito em 2 passos principais (a partir do nó origem):

1. (Seleção de novo nó) Seleciona-se o próximo nó a ser expandido de uma lista de nós candidatos, que possuem solução não integralizada com valor de função objetivo (FO) melhor que a melhor solução inteira obtida até o momento. O nó escolhido tem o melhor valor de F.O. do ramo expandido.
2. (Expansão do nó) Introduzem-se 2 novas restrições no PPL associado a esse nó, de modo a obrigar determinada variável inteira não integralizada (que não assume valor inteiro) a sê-la, e resolve-se esses PPL criando novos nós.

O algoritmo pára, quando nenhum nó puder ser incluído na lista de nós candidatos à expansão, porque todos os ramos terminam em nós inteiros ou inviáveis ou possuem valor de F.O. pior do que a melhor solução inteira encontrada.

Há diversas variantes para esse algoritmo que dependem de:

- nº de soluções inteiras expandíveis desejado
- tempo de execução
- exigência de solução ótima ou não

- existência de uma estimativa heurística para a solução inteira a partir de um nó
- regras para a seleção de novos nós

O problema estudado que possui 10 * PERI variáveis 0 - 1 permite a expansão de até $2^{10*PERI}$ nós inteiros (ignorando todas as restrições).

Dado o grande número de soluções expandíveis, foi necessário estabelecer restrições a expansão dos nós.

- a) foi criado um limite de tempo para a busca
- b) somente são candidatos a expansão, nós com valor de custo pelo menos 2X cruzeiros melhor que o custo da melhor solução inteira obtida até o momento, onde X é proporcional ao período de programação e corresponde a cerca de 2% do custo da solução contínua.

Quando a melhor solução inteira encontrada está numa faixa próxima da solução contínua, os nós candidatos à expansão (abertos) são desprezados se não forem X cruzeiros melhores que o custo da melhor solução inteira.

6. RESULTADOS

O modelo foi programado e testado para diversos conjuntos de dados e diversos períodos. Alguns desses testes estão relacionados abaixo:

PER DIAS	NRES	NVAR	NVINT	DENS %	TCPU MIN	ITER	F.O.	OBS	
4	139	377	40	1,69	0,12	287	-1574.	Sol. Contínua	
					0,48	591	- 949.	Sol. Inteira	
					0,56	642	-1369.	"	
					0,73	751	-1409.	"	
					0,98	928	—	Fim	
8	275	741	80	0,89	0,43	634	-16317.	Sol.Contínua	
					1,84	1477	-14411.	Sol.Inteira	
					35,4	19932	—	Fim	
7	240	642	63	0,99	0,25	488	- 3008.	Sol.Contínua	
					1,20	1070	- 2375.	Sol.Inteira	
					1,50	1232	- 2855.	"	
					4,69	2975	—	Fim	
7	242	651	70	1,00	0,34	535	-13003.	Sol.Contínua	
					7,28	4937	- 7423.	Sol.Inteira	
					15,80	10226	- 9273.	"	
					não encerrou a busca	16,40	10567	-10534.	"
					60,00	36797	—	estouro tempo	
7	240	649	70	1,01	0,32	548	-15632.	Sol.Contínua	
					4,13	3043	- 8577.	Sol.Inteira	
					5,76	4106	-10428.	"	
					não encerrou a busca	50,00	33078	—	estouro tempo

PER	NRES	NVAR	NVINT	DENS	TCPU	ITER	F.O.	OBS
DIAS				%				
15	505	1325	165	0,50	1,81	1451	-2851.	Sol. Contínua
					5,22	2515	-1087.	Sol. Inteira
					6,87	3009	-1831.	"
	não encerrou a busca				30,00	9982	—	estouro tempo
10	339	919	100	0,72	0,84	934	-20458.	Sol. Contínua
					3,90	2417	-18105.	Sol. Inteira
	não encerrou a busca				59,00	—		estourou tempo
10	339	909	90	0,71	0,82	900	- 3717.	Sol. Contínua
					9,47	5056	- 2991.	Sol. Inteira
					12,58	6472	- 3199.	"
					18,73	9091	- 3287.	"
					50,0	21883	—	estouro tempo

7. AVALIAÇÃO DOS RESULTADOS

Teoricamente, pode-se provar que o algoritmo branch and bound conduz à solução ótima inteira (quando existe) num número finito de passos (tempo finito). Na prática, no entanto, a duração da busca ultrapassou todos os limites pre-fixados quando o número de variáveis 0-1 era superior a 50, devido ao crescimento exponencial do nº de nós gerados.

Para acelerar a busca, foram desprezadas todas as soluções de custo próximo (no entorno de 1%) às já obtidas e foi reduzida a região viável a uma faixa bem próxima à solução contínua obtida inicialmente, determinada pelo nº máximo e mínimo de dias de produção para cada tipo de óleo.

Tais providências acarretaram que não se poderia mais provar a otimalidade da solução, dado que a solução ótima poderia estar fora dos limites estabelecidos, e não conseguiram reduzir a contento a aleatoriedade da duração da busca.

Justifica-se o abandono de soluções muito próximas as já encontradas devido à uma indefinição nos custos unitários constantes na função objetivo.

8. CONCLUSÕES

O modelo construído fornece soluções que levam em conta todos os critérios de decisão utilizados na programação tradicional, podendo pois substituir o processo anterior.

A incerteza quanto ao fim da busca utilizando o programa MIP da IBM, para modelos com mais de 50 variáveis inteiras sugere a utilização de algum algoritmo mais específico para a programação 0-1.

Foi usado durante algum tempo o método de Benders para programação mista com enumeração implícita para as variáveis 0-1. Como este programa se encontra ainda em fase de testes, deve-se esperar a correção do mesmo para poder-se obter soluções a um custo mais previsível.

9. REFERÊNCIAS BIBLIOGRÁFICAS

1. Tomlin, J.A. "Branch and Bound methods for integer and nonconvex programming". Integer and Nonlinear Programming Amsterdam, North Holland Publishing Company 1970
Chapter 21 pag 437 - 450
2. Greenberg, H. "Integer Programming"
Academic Press pag 177
3. Manual IBM "Mathematical Programming System Extended/370 and Mixed Integer Programming/370" Program Reference Manual SH19 - 1099 - 1.

```

//MIPP9004 JOB (9004,-590,230,12),'593-2 MICHAEL 2755',
//      MSGI FVEL=(1,1),REGION=576K
//JS01 EXEC PGM=P020701A
//STEPLIB DD DSN=B.TEST.CARGA.PESOPCO.N01,DISP=SHR
//SYSIN DD DSN=CRJ.A0207.STAB001.U0230,DISP=SHR
//BCDOUT DD DSN=&&A0207001,UNIT=SYSDA,
//      SPACE=(CYL,(1,1)),DCB=(RECFM=FB,LRECL=80,BLKSIZE=4080),
//      DISP=(NEW,PASS)
//SYSPRINT DD SYSOUT=A
//JS02 EXEC MP5X370,MIP=,COND=(4,LT),TIME=1440
//MPSCOMP.SYSIN DD *

```

```

PROGRAM
INITIALZ
MOVE(XPNAME,'DTILUB')
MOVE(XDATA,'DADOS')
MOVE(XOBJ,'OBJ')
MOVE(XRHS,'LIM')
MOVE(XBOUND,'BOUND')
CONVERT('SUMMARY')
XNODES=1800
SETUP
MVADR(XDENFS,FIM1)
OPTIMIZE
SOLUTION
MXFNLOG=0
XDEL TM=60
MVADR(XDEDEL TM,ARQ)
MVADR(XDCPRINT,INT)
MVADR(XMXOVFL,ARQ)
MIXSTART('COST')
MIXFLOW
ARG MIXSAVE
MIXSTATS
GOTO(FIM2)
INT MXMDROP=MXMDROP-70.
IF((MXMDROP-MXMBESTF).GT.700.,SIGA)
MXMDROP=MXMDROP+35.
SOLUTION
SIGA CONTINUE
FIM1 SOLUTION
FIM2 EXIT
PEND

```

```

/*
//MPSEXEC.MIXWORK DD UNIT=SYSDA,SPACE=(CYL,(6,2))
//MPSEXEC.PROBFILE DD DSN=&&A0207002,UNIT=SYSDA,
//      SPACE=(CYL,6.,CONTIG),
//      DISP=(NEW,DELETE)
//MPSEXEC.SYSUDUMP DD DUMMY
//MPSEXEC.SYSIN DD DSN=*.JS01.BCDOUT,DISP=(OLD,DELETE)
//

```



```

/*
CUSTO DE FUNCIONAMENTO DE UM PROCESSO ( POR DIA )
=====*/
CUSPAR(7) DEC FIXED (8,0) ,
000046
000047
000048
000049
000050
/*
FRACAO MINIMA DE OLEO NUM TANQUE
( POR PROCESSO, DURANTE O PERIODO )
=====*/
RMIN(7,7) DEC FIXED (6,3)
INIT ((7)(15)0.0,(2)0.00) ,
000051
000052
000053
000054
000055
000056
/*
FRACAO MINIMA DE OLEO NUM TANQUE
( POR PROCESSO, NO ULTIMO DIA )
PFIMIN(7,7) DEC FIXED (6,3)
INIT ((7)(15)0.003,(2)0.00) ,
000057
000058
000059
000060
000061
/*
VALOR DA PRODUCAO DO OLEO J NO PROCESSO I
=====*/
DCL VALOL(7,7) DEC FIXED(9,3) ,
000062
000063
000064
000065
/*
CUSTO DE OLEO EM EXCESSO PARA F.C.C. (PCR M3)
=====*/
CFXC(7,7) DEC FIXED (9,3)
000066
000067
000068
000069
/*
CUSTO UNITARIO DE OLEO BASICO NAO FORNECIDO NO PERIODO
=====*/
DCL CPEROL(7) DEC FIXED(9,3) ,
000070
000071
000072
000073
/*
CUSTO UNITARIO DE PARAFINA NAO FORNECIDA NO PERIODO
=====*/
CRERPA(7) DEC FIXED(9,3)
000074
000075
000076
000077
000078
000079
000080
000081
000082
/*
DCL (SPA(7),SOL(7),OI(7,7),OIN(7,7)) DEC FIXED(9,1) ,
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
DCL OL2(8) CHAR(2) INIT('SP','NL','NM','NP','BS','CL','C2','PR') ,
000093
000094
000095
000096
000097
000098
000099
000100
000101
000102
000103
(IE,LMAX) BIN FIXED(4) ,

```


NI(10) CHAR(1) INIT('1','2','3','4','5','6','7','8','9','0'), 0001
 PROC6(7) CHAR(6) INIT('DESTIL','DESASF','EXTRAC','DESPAR',
 'DESOLE','HIDROL','HIDPAR'), 0001
 RAKO6(7) CHAR(6) INIT('DESTIL','DESASF','RAFINA',
 'DESPAR','DESOLE','OLBASI','PARAFP') 0001
 DCL N2(30) CHAR(2) INIT('01','02','03','04','05','06','07','08',
 '09','10','11','12','13','14','15',
 '16','17','18','19','20','21','22',
 '23','24','25','26','27','28','29','30'), 0001
 BR8 CHAR(8) INIT(' ') 0001

DCL 1 NAM , 0001
 2 CAM1 CHAR(8) , 0001
 2 VAZ1 CHAR(6) INIT(' ') , 0001
 2 CAM2 CHAR(8) , 0001
 2 VAZ2 CHAR(58) INIT(' ') , 0001
 1 RES , 0001
 2 VAZ1 CHAR(1) INIT(' ') , 0001
 2 CAM1 CHAR(1) , 0001
 2 VAZ2 CHAR(2) INIT(' ') , 0001
 2 CAM2 CHAR(8) , 0001
 2 VAZ3 CHAR(68) INIT(' ') , 0001
 1 COLB , 0001
 2 VAZ1 CHAR(4) INIT(' ') , 0001
 2 CAM1 CHAR(8) , 0001
 2 VAZ2 CHAR(2) INIT(' ') , 0001
 2 CAM2 CHAR(8) , 0001
 2 VAZ3 CHAR(2) INIT(' ') , 0001
 2 COEF2 PIC '(6)Z9V.(3)9' , 0001
 2 VAZ4 CHAR(3) INIT(' ') , 0001
 2 CAM3 CHAR(8) , 0001
 2 VAZ5 CHAR(2) INIT(' ') , 0001
 2 COEF3 PIC '(6)Z9V.(3)9' , 0001
 2 VAZ6 CHAR(19) INIT(' ') 0001

DCL 1 COLA , 0001
 2 VAZ1 CHAR(4) INIT(' ') , 0001
 2 CAM1 CHAR(8) , 0001
 2 VAZ2 CHAR(2) INIT(' ') , 0001
 2 CAM2 CHAR(8) , 0001
 2 VAZ3 CHAR(2) INIT(' ') , 0001
 2 COEF2 PIC '(6)Z9V.(3)9' , 0001
 2 VAZ4 CHAR(44) INIT(' ') , 0001

1 INTE , 0001
 2 VAZ1 CHAR(4) INIT(' ') , 0001
 2 CAM1 CHAR(6) , 0001
 2 VAZ2 CHAR(4) INIT(' ') , 0001
 2 CAM2 CHAR(8) , 0001
 2 VAZ3 CHAR(17) INIT(' ') , 0001
 2 CAM3 CHAR(8) , 0001
 2 VAZ4 CHAR(33) INIT(' ') , 0001

1 BOU , 0001
 2 VAZ1 CHAR(1) INIT(' ') , 0001
 2 CAM1 CHAR(2) , 0001
 2 VAZ2 CHAR(1) INIT(' ') , 0001
 2 CAM2 CHAR(8) , 0001
 2 VAZ3 CHAR(2) INIT(' ') , 0001
 2 CAM3 CHAR(8) , 0001
 2 VAZ4 CHAR(2) INIT(' ') , 0001
 2 COEF1 PIC '(6)Z9V.(3)9' , 0001


```

END 000220
IF UNI 0.3 THEN 000221
DO 000222
  PUT PAGE 000223
  PUT EDIT ('**** ERRO ****', 000224
    'UNIDADE DE TEMPO INSUFICIENTE', 000225
    'PRECISA SER MAIOR QUE 0.2') 000226
    (3(SKIP(1),X(10),A)) 000227
  CALL PLIRETC(12) 000228
  GO TO FIM 000229
END 000230
LMAX = FLOOR(2.02/UNI-0.2) 000231
IF (LMAX+1) NINT THEN 000232
DO 000233
  PUT PAGE 000234
  PUT EDIT ('**** ERRO ****', 000235
    'PERIODO DE PROGRAMACAO INSUFICIENTE', 000236
    'NUM. INTERVALOS PRECISA SER MAIOR QUE ',(LMAX+1)) 000237
    (3(SKIP(1),X(10),A),F(3,0)) 000238
  CALL PLIRETC(12) 000239
  GO TO FIM 000240
END 000241
PUT SKIP(3) 000242
GET EDIT (CARTAO) (A(80)) COPY 000243
GET EDIT (CARTAO) (A(80)) COPY 000244
GET EDIT (CARTAO) (A(80)) COPY 000245
CARTON = CARTAO 000246
DO J=1 TO 7 000247
  NPOS(I) = 80 000248
  DO J=JCB(I) TO 5.8 000249
    NPOT = INDEX(CARTON,OL2(J)) 000250
    IF NPOT = 0 THEN NPOT = -80 000251
    IF NPOT NPOS(I) THEN DO 000252
      NPOS(I) = NPOT 000253
      OLOR(I) = J 000254
    END 000255
  END 000256
  IF NPOS(I) = 80 THEN DO 000257
    PUT PAGE 000258
    PUT EDIT(' *** ERRO *** ', 000259
      'OLEO DESCONHECIDO NO PROCESSO ',PROC6(I)) 000260
      (SKIP(2),X(10),A,SKIP(1),X(10),A,A(6)) 000261
    CALL PLIRETC(12) 000262
    GO TO FIM 000263
  END 000264
  ELSE DO 000265
    NPOSP2 = NPOS(I)+2 000266
    CARTON = SUBSTR(CARTON,NPOSP2) 000267
  END 000268
END 000269
IF OLOR(5) = OLOR(4) 000270
  THEN DO 000271
    PUT PAGE 000272
    PUT EDIT ('**** ERRO ****', 000273
      'OLEO DESCONHECIDO NO PROCESSO ',PROC6(5)) 000274
      (SKIP(1),X(10),A,SKIP(1),X(10),A,A(6)) 000275
    CALL PLIRETC(12) 000276
    GO TO FIM 000277
  
```

```

                END                                000278
        OLOR(6) = OLOR(4)                          000279
        OLOR(7) = OLOR(4)                          000280
        PUT SKIP(3)                                000281
        GET EDIT (CARTAO) (A(80)) COPY             000282
        GET EDIT (CARTAO) (A(80)) COPY             000283
        GET EDIT (CARTAO) (A(80)) COPY             000284
        GET STRING (CARTAO) LIST ((MAXDIA(3,J) DO J=1 TO 5)), 000285
        GET EDIT (CARTAO) (A(80)) COPY             000286
        GET STRING (CARTAO) LIST ((MINDIA(3,J) DO J=1 TO 5)), 000287
        PUT SKIP(2)                                000288
        GET EDIT (CARTAO) (A(80)) COPY             000289
        GET EDIT (CARTAO) (A(80)) COPY             000290
        GET EDIT (CARTAO) (A(80)) COPY             000291
        GET STRING (CARTAO) LIST ((MAXDIA(4,J) DO J=1 TO 5)), 000292
        GET EDIT (CARTAO) (A(80)) COPY             000293
        GET STRING (CARTAO) LIST ((MINDIA(4,J) DO J=1 TO 5)), 000294
        DO I=3,4                                    000295
        DO J=1 TO 5                                  000296
            IF MAXDIA(I,J) - MINDIA(I,J)
            THEN DO                                  000297
                PUT PAGE                            000298
                PUT EDIT('*** ERRO NUS LIMITES MAXIMUS E MINIMOS ', 000300
                    'DE DIAS DE PROCESSAMENTO')    000301
                    (SKIP(2),X(10),A,SKIP(1),X(10),A) 000302
                CALL PLIRETC(12)                    000303
                GO TO FIM                            000304
            END                                      000305
        END                                         000306
    END                                           000307
    PUT SKIP(2)                                    000308
    GET EDIT (CARTAO) (A(80)) COPY                000309
    PUT SKIP(1)                                    000310
    GET EDIT (CARTAO) (A(80)) COPY                000311
    DO I=1,2,3,4,7                                 000312
        GET EDIT (CARTAO) (A(80)) COPY            000313
        GET STRING (CARTAO) LIST((L(I,K) DO K=0 TO NINT)), 000314
    END                                             000315
    /*                                             000316
    /*                                             000317
    TESTE NOS INDICES L(I,K) DE FUNCIONAMENTO    000318
    =====*/                                     000319
    DO I=1,2,3,4,7                                 000320
        DO K=0 TO NINT                              000321
            IF L(I,K) = 1 & L(I,K) = 0            000322
            THEN DO                                  000323
                PUT EDIT (' *** ERRO FUNCIONAMENTO NEM 0 NEM 1 ') 000324
                    (SKIP(8),X(20),A)              000325
                CALL PLIRETC(12)                    000326
                GO TO FIM                            000327
            END                                      000328
        END                                         000329
    END                                           000330
    PUT SKIP(3)                                    000331
    GET EDIT (CARTAO) (A(80)) COPY                000332
    PUT SKIP(1)                                    000333
    GET EDIT (CARTAO) (A(80)) COPY                000334
    DO I=1,2,3,5,6,7                               000335

```


END 000452

/* 000453

000454

000455

000456

000457

===== */ 000458

GRAVACAO DA MATRIZ 000458

NAM,CAM1 = 'NAME' 000460

NAM,CAM2 = 'DADD' 000461

WRITE FILE (BCDOUT) FROM(NAM) 000462

NAM,CAM1 = 'ROWS' 000463

NAM,CAM2 = 'BR 8' 000464

WRITE FILE (BCDOUT) FROM(NAM) 000465

RES,CAM1 = 'N' 000466

RES,CAM2 = 'OBJ' 000467

WRITE FILE (BCDOUT) FROM(RES) 000468

/* 000469

RESTRIC0ES DE EXCLUSIVIDADE 000470

===== */ 000471

DO K=1 TO NINT 000472

DO I=3,4 000473

IF L(I,K) = 1 THEN DO 000474

RES,CAM1 = 'L' 000475

RES,CAM2 = 'UNI' N1(I) N2(K) ' ' 000476

WRITE FILE (BCDOUT) FROM (RES) 000477

END 000478

END 000479

END 000480

/* 000481

000482

RESTRIC0ES DE CAPACIDADE 000483

===== */ 000484

DO K=1 TO NINT 000485

DO I=1,2,3,6,7 000486

DO J=JCO(I) TO 5 000487

RES,CAM1 = 'E' 000488

RES,CAM2 = 'CAP' N1(I) N1(J) N2(K) ' ' 000489

WRITE FILE (BCDOUT) FROM (RES) 000490

END 000491

END 000492

END 000493

/* 000494

RESTRIC0ES DE TROCAS DE CLEC 000495

===== */ 000496

DO I=3,4 000497

DO K=1 TO NINT 000498

IF L(I,K) = 1 & L(I,K-1) = 1 000499

THEN DO 000500

RES,CAM1 = 'G' 000501

RES,CAM2 = 'EMPO' N1(I) N2(K) ' ' 000502

WRITE FILE (BCDOUT) FROM (RES) 000503

RES,CAM2 = 'EMNE' N1(I) N2(K) ' ' 000504

WRITE FILE (BCDOUT) FROM (RES) 000505

RES,CAM1 = 'L' 000506

RES,CAM2 = 'RTRO' N1(I) N2(K) ' ' 000507

WRITE FILE (BCDOUT) FROM (RES) 000508

RES,CAM1 = 'E' 000509


```

COLA.COE2 = CPERPA(J)*UNI*NINT      00062
WRITE FILE (BCDOUT) FROM (COLA)      00062
DO K=1 TO NINT      00062
  COLA.CAM2 = 'CAP' N1(I) N1(J) N2(K) ' '      00062
  COLA.COE2 = -UNI      00062
  WRITE FILE (BCDOUT) FROM (COLA)      00062
END      00062
END      00062
/*      00062

```

VARIÁVEIS - MÓDULO DE TROCA

```

===== */      00062
DO I=3,4      00062
  DO K=1 TO NINT      00062
    IF L(I,K) = 1 & L(I,K-1) = 1      00062
      THEN DO      00062
        COLB.CAM1 = 'MODT' N1(I) N2(K) ' '      00062
        COLB.CAM2 = 'EMPO' N1(I) N2(K) ' '      00062
        COLB.COE2 = 1.      00062
        COLB.CAM3 = 'OBJ'      00062
        COLB.COE3 = CLESTRO(I)      00062
        WRITE FILE (BCDOUT) FROM (COLB)      00062
        COLB.CAM1 = 'MGDT' N1(I) N2(K) ' '      00062
        COLB.COE2 = 1.      00062
        COLB.CAM2 = 'RTRO' N1(I) N2(K) ' '      00062
        COLB.CAM3 = 'EMNE' N1(I) N2(K) ' '      00062
        COLB.COE3 = 1.      00062
        WRITE FILE (BCDOUT) FROM (COLB)      00062
      /*      00062

```

VARIÁVEIS DIFERENÇA ENTRE OS ÍNDICES

```

===== */      00062
COLA.CAM1 = 'DTRU' N1(I) N2(K) ' '      00062
COLA.CAM2 = 'ETRU' N1(I) N2(K) ' '      00062
COLA.COE2 = 1.      00062
WRITE FILE (BCDOUT) FROM (COLA)      00062
COLB.CAM1 = 'DTRO' N1(I) N2(K) ' '      00062
COLB.CAM2 = 'EMPO' N1(I) N2(K) ' '      00062
COLB.COE2 = -1.      00062
COLB.CAM3 = 'EMNE' N1(I) N2(K) ' '      00062
COLB.COE3 = 1.      00062
WRITE FILE (BCDOUT) FROM (COLB)      00062

```

END

END

END

/*

VARIÁVEIS DE DECISÃO 0-1

```

===== */      00062
INTE.CAM1 = 'INVIN'      00062
INTE.CAM2 = 'MARKER'      00062
INTE.CAM3 = 'INTORG'      00062
WRITE FILE (BCDOUT) FROM (INTE)      00062
DO K=1 TO NINT      00062
  DO I=3,4      00062
    IF L(I,K) = 1 THEN      00062

```


RESTRIC0ES DE CAPACIDADE

=====*/

```

NTD=0
DO I=1,2,3,6,7
  DO J=JCO(I) TO 5
    QIN(I,J) = QI(I,J)
  FND
FND
DO K=1 TO NINT
  NTD = NTD+L(1,K)
  DO J=1 TO 5
    COLA.CAM2 = 'CAP1' N1(J) N2(K) ' '
    COLA.COE2 = QIN(1,J)+FLOAT(L(1,K))*UNI*CRU*EF(1,J)
    WRITE FILE (BCDOUT) FROM (COLA)
  FND
  DO I=2,3
    DO J=JCO(I) TO 5
      COLA.CAM2 = 'CAP' N1(I) N1(J) N2(K) ' '
      COLA.COE2 = QIN(I,J)
      WRITE FILE (BCDOUT) FROM (COLA)
    FND
  FND
  DO J=JCB(6) TO 5
    COLB.CAM2 = 'CAP' N1(6) N1(J) N2(K) ' '
    COLB.COE2 = QIN(6,J)- UNI*SOL(J)
    COLB.CAM3 = 'CAP' N1(7) N1(J) N2(K) ' '
    COLB.COE3 = QIN(7,J) - UNI*SPA(J)
    WRITE FILE (BCDOUT) FROM (COLB)
  FND
  IF K = 1 THEN
    DO I=1,2,3,6,7
      DO J=JCO(I) TO 5
        QIN(I,J) = 0.
      FND
    FND
  FND
FND
/*

```

RESTRIC0ES DE TROGAS

=====*/

```

DO I=3,4
  DO K=1 TO NINT
    IF L(I,K) = 1 & L(I,K-1) = 1
      THEN DO
        COLA.CAM2 = 'ETRU' N1(I) N2(K)
        IF K = 1
          THEN COLA.COE2 = -OLOR(I)+1PATRU(I)
          ELSE COLA.COE2 = 0
        WRITE FILE(BCDOUT) FROM (COLA)
        COLA.CAM2 = 'RTRU' N1(I) N2(K) ' '
        COLA.COE2 = 4.
        IF K=1 THEN COLA.COE2 = 2.
        WRITE FILE (BCDOUT) FROM (COLA)
        IF K LMAX THEN DO
          COLA.CAM2 = 'FTMA' N1(I) N2(K) ' '
          COLA.COE2 = 1.
          WRITE FILE (BCDOUT) FROM (COLA)
        FND
      FND
  FND

```



```

BOU.CAM3 = 'PERP' N1(J) ' ' 000916
BOU.COEF1 = SPA(J) 000917
WRITE FILE (BCDOUT) FROM (BOU) 000918
END 000919
DO K=1 TO NINT 000920
IF L(2,K) = 1 THEN DO 000921
BOU.CAM3 = 'VAZ25' N2(K) ' ' 000922
BOU.COEF1 = VZ(2,5)*FLOAT(L(2,K)) 000923
WRITE FILE (BCDOUT) FROM (BOU) 000924
END 000925
END 000926
/* 000927
000928
000929

```

FINAL

```

===== */ 000931
NAM.CAM1 = 'ENDATA' 000932
WRITE FILE (BCDOUT) FROM (NAM) 000933
FIM ENG GERLUB 000934

```