UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO

UM ESTUDO DE VIABILIDADE DE IMPLANTAÇÃO DE ESTAÇÕES DE TRANSFERÊNCIA DE LIXO

JOSÉ FERREIRA LIMA FILHO

CAMPINA GRANDE - PARAÍBA
AGOSTO DE 1981

L732e Lima Filho, José Ferreira.
Um estudo de viabilidade de implantação de estações de transferência de lixo / José Ferreira Lima Filho. - Campina Grande : 1981.
43 f.

Dissertação (Mestrado em Sistemas e Computação) - Universidade Federal da Paraíba, Centro de Ciências e Tecnologia,1981.
"Orientação : Prof. Dr. Ulrich Seip".
Referências.

1. Lixo - Estações de Transferência - Implementação. 2. Estudo de Viabilidade. 3. Dissertação - Sistemas e Computação. I. Seip, Ulrich. II. Universidade Federal da Paraíba - Campina Grande (PB). III. Título

CDU 628.46(043)

UM ESTUDO DE VIABILIDADE DE IMPLANTAÇÃO DE
ESTAÇÃO DE TRANSFERÊNCIA DE LIXO

JOSÉ FERREIRA LIMA FILHO

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DO CURSO DE PÓS-GRADUAÇÃO EM SISTEMAS E COMPUTAÇÃO DA UNIVERSIDADE FEDERAL DA PARAÍBA COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS (M.Sc.).

Aprovada por:

Ulrich Seip

ULRICH SEIP - Ph.D.
- Presidente -

EVILSON DE ARAÚJO BARROS - M.Sc.

EDSON PACHECO PALADINI - M.Sc.

CAMPINA GRANDE
ESTADO DA PARAÍBA - BRASIL
AGOSTO - 1981

A meus pais e irmãos pelo incentivo e apoio sempre demonstrados, sem os quais não seria possível a realização deste trabalho.

A minha esposa Teresinha e a meu filho Weslley pela compreensão e dedicação.

AGRADECIMENTOS

Ao Professor ULRICH SEIP pela orientação, atenção incentivo e amizade, sempre demonstrado durante a realização deste trabalho.

Aos Professores do DSC e DME que de uma forma ou de outra prestaram sua colaboração.

A todos os colegas de curso pela amizade sempre demonstrada.

A todas as pessoas que de uma forma ou de outra contribuíram para o êxito deste trabalho, em particular aos funcionários do NPD.

RESUMO

O problema considerado neste trabalho é o de localização ótima de facilidades, mais precisamente, o problema de estação de transferência na coleta de lixo. O esforço principal é para determinar a localização ótima das estações de transferência de lixo de modo que minimize o custo total do sistema, incluindo os custos de construção e manutenção das estações de transferência e também os custos de transporte.

O modelo usado neste trabalho é de programação linear, isto é, por linearização dos custos de transporte, construção e manutenção (sistema). Observando que para cada seleção da possível estação de transferência, o problema se reduz a encontrar um fluxo a custo mínimo, sendo usado para isto o algoritmo "out-of-kilter", apropriado para este tipo de problema. Para reduzir o número de possíveis seleções, usa-se o algoritmo "branch-and-bound", para evitar a enumeração, se possível.

Os resultados obtidos mostram que a construção das estações de transferência pode ser útil no caso em que a cidade seja suficientemente grande.

ABSTRACT

The problem considerad in this work is the optimal facility location, more particulary, the Transfer Stations in solid waste collection problems. The major aim is to determine the optimal location of the transfer stations in such a way as to minimize the overall cost of the system including the construction and operating cost of the transfer stations as well as the transportation costs before and the construction.

The model used in this study is solved by linear programming this is, by linearization of the construction, maintenance, and transportation costs involved. Observing that for each selection of possible transfer stations the problem reducer to finding a minimal cost flow, it's choosen as appropriate out-of-kilter algorithm and, in order to reduce the number of possible selection, we choose a brnach and bound algorithm to avoid enumeration if possible.

The results show that construction of transfer stations is useful in case that the city is sufficiently large.

INDICE

INTRODUÇÃO

Este trabalho consiste no estudo do problema de localização de facilidades no setor público. Mais precisamente é considerado o problema da coleta do lixo sob o aspecto da introdução de estações de transferência de lixo, para diminuir os custos da coleta.

Os modelos usados para a coleta do lixo podem ser eficientes, sem contudo minimizar os recursos alocados para esse fim. Esses modelos têm como base, em geral, uma divisão da cidade em setores, aos quais são designados recursos humanos e materiais (garis, motoristas, caminhões, etc) para a coleta. De uma forma geral, o lixo é coletado em cada setor e enviado ao depósito final (aterro) como mostra a figura 1.1 . Isto pode ser viável se a distância entre o ponto final da coleta e o depósito final (aterro) for pequena, pois os veículos coletores são geralmente de pequena capacidade. Quando esta distância for muito grande é conveniente a introdução de estações, já que, a medida que a distância aumenta, o custo se eleva. Para a transferência do lixo destas estações para o aterro final devem ser usados meios de transporte de maior capacidade. Por exemplo: caminhões grandes, carretas ou outros veículos apropriados.

As estações para a transferência de lixo podem ser de tipos diferentes: podem ser só do tipo plataforma ou podem também ter capactador de lixo ou instalações mais sofisticadas. Também os meios de transporte para o aterro final podem ser de tipos mais ou menos sofisticados.

A determinação para a instalação de estações de transferência e os meios adequados de transporte para o aterro final depende de diversos fatores, tais como: a distância, o volume de lixo acumulado num setor, os veículos disponíveis para a coleta, a possibilidade da instalação de uma estação numa área da cidade, o custo da instalação em relação ao orçamento disponível, etc.

Uma definição mais precisa deste problema pode ser dada da seguinte forma: é dada uma cidade dividida em setores, onde para cada setor são designados recursos humanos e materiais para a coleta de lixo, de onde:

a) É necessário designar possíveis locais para a construção de estações de transferência de lixo;
b) Deve-se determinar, de uma forma geral, em quais dos locais as possíveis estações de transferência de lixo devem ser construidas;
c) É preciso determinar quais os setores que deverão enviar lixo para cada estação.

É claro que a última decisão é feita de modo que o custo da coleta do lixo com o uso destas estações de transferência (denotado por ETL) seja minimizado.

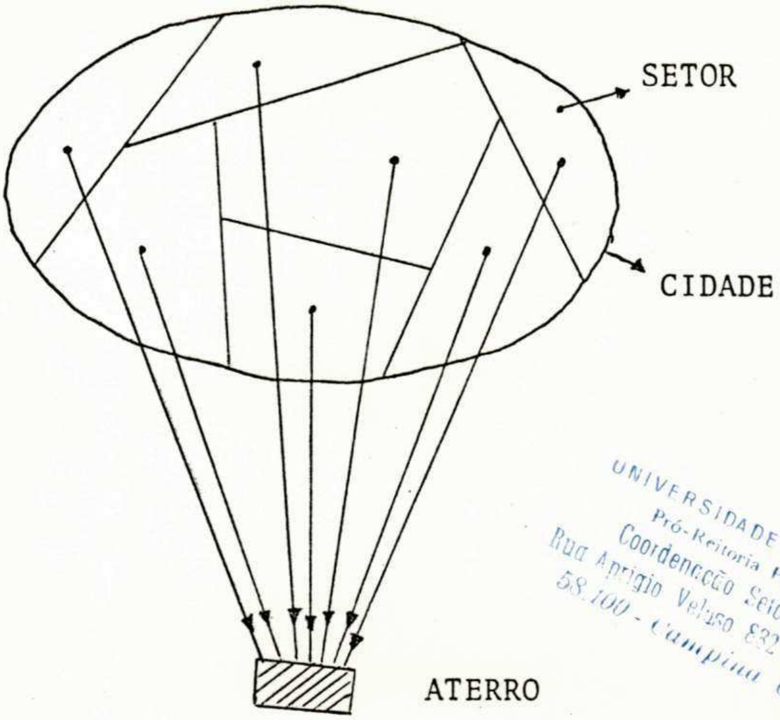


Fig. 1.1 - Modelo comumente usado.

O objetivo deste trabalho é estabelecer um modelo viável que satisfaça os ítens a, b e c.

Desta forma, os principais objetivos para a implantação de estações de transferência de lixo serão:

a) Minimização do custo de transporte;

b) Diminuição do custo total;

c) Melhor serviço devido a maior eficiência;

d) Melhor aproveitamento e menor desgaste dos veículos coletores.

No capítulo 1 são inicialmente introduzidas as variáveis e os dados necessários para o estabelecimento do modelo linear que representa o problema aqui estudado. A seguir, estabelece-se a função objetiva e o modelo, com as explicações necessárias. No capítulo 2 são dadas noções gerais da teoria dos grafos necessárias para o entendimento dos algoritmos "out-of-kilter" [05, 08], e "branch-and-bound" [04] que são apresentados no capítulo 3. Considerando que este trabalho é de natureza prática, foi tomada a cidade do Recife para aplicação do modelo aqui desenvolvido, com parâmetros estimados. Os resultados dessa experiência são abordados no capítulo 4, e no capítulo 5 é apresentada a conclusão deste trabalho.

CAPÍTULO I

ASPECTOS GERAIS DO PROBLEMA

1.1 - Introdução

No modelo desenvolvido neste trabalho, considera-se a cidade dividida em setores e a existência de aterros finais onde o lixo coletado é armazenado.

Para discretizar a situação, cada setor é considerado concentrado num ponto central e por conseguinte, as distâncias entre os setores e entre os setores e os aterros finais podem ser consideradas como conhecidas.

Como em cada problema, deve-se em primeiro lugar, definir o tipo da enumeração a usar, neste trabalho os setores são enumerados de 1 até m e os aterros finais de m+1 até m+n. Isto significa que são considerados m setores e n aterros finais.

Também são considerados como conhecidos os setores que possivelmente podem abrigar uma estação de transferência de lixo. Supõe-se que estes setores aparecerão na enumeração criada como os últimos setores, isto é: supondo-se que em p dos m seto res ($p \leq m$) podem ser possivelmente instaladas estações de transferência de lixo, então estes setores correspondem na enumeração, aos números de p+1 até m.

Com relação a esta enumeração, precisa-se para o modelo aqui estudado, dos seguintes dados:

a) A quantidade de lixo coletado por semana, no i-ésimo setor, denotado por QL_i (em toneladas por semana, $i=1,\ldots,m$);

b) A capacidade de absorção de lixo por semana, do i-ésimo aterro final, denotado por CAP_i (em toneladas por semana, $i=1,\ldots,m+n$);

c) Tanto as distâncias entre o i-ésimo setor e algum outro setor, que possivelmente abriga uma estação de transferência de lixo, como a distância entre o i-ésimo setor e todos os aterros finais. Estas distâncias são denotadas por d_{ij} (em quilômetros, $1 \leq i \leq m$, $p+1 \leq j \leq m+n$);

d) O custo de transporte de uma tonelada por quilômetro, do lixo transportado do i-ésimo setor para a estação, denotado por δ_i (em cruzeiros por tonelada e quilômetro, $i=1,\ldots,m$).

Para a aplicação do modelo, precisa-se também de valores estimados para os seguintes parâmetros:

e) O limite inferior <u>LIM</u> para a capacidade de cada estação de transferência de lixo (em toneladas por semana);

f) A constante α , que exprime o custo de construção da estação para a transferência de uma tonelada de lixo por semana, se existir alguma esta ção de transferência de lixo (em cruzeiros por (tonelada por semana));

g) A constante β que exprime o custo da manutenção da estação, por semana, para a transferência de uma tonelada de lixo por semana, se existir alguma estação de transferência de lixo está estabelecida (em cruzeiros por tonelada);

h) O custo de transporte, por quilômetro e tonelada de lixo transportado, da i-ésima estação de transferência de lixo para o aterro final, denotado por $\bar{\delta}_i$ (em cruzeiros por tonelada e quilômetro, $i=1,\ldots,m$);

i) O horizonte de planejamento S (em semanas) que designa o tempo até o qual os custos adicionais que ocorrem com a construção e a manutenção das estações de transferência de lixo precisam ser amortizados.

1.2 - Descrição do Modelo

Com estes dados e parâmetros, pode-se formular o modelo linear como segue:

Notação: $i=1,\ldots,p$ designa os pontos fontes que não podem ser uma ETL;

$p+1,\ldots,m$ designa os pontos fontes que podem ser uma ETL;

$m+1,\ldots,m+n$ os aterros finais.

Dados: QL_i para $1 \leq i \leq m$ em toneladas por semana;

CAP_i para $m+1 \leq i \leq m+n$ em toneladas por semana;

δ_i para $1 \leq i \leq m$ em cruzeiros por tonelada e quilômetro;

d_{ij} para $1 \leq i \leq m$ e $p+1 \leq j \leq m+n$ em qulômetros;

e os parâmetros estimados

$\bar{\delta}_i$ para $p+1 \leq i \leq m$ em cruzeiros por tonelada e quilômetro;

α em cruzeiros por tonelada por semana;

β em cruzeiros por tonelada;

LIM em toneladas por semanas;

S em semanas.

Objetivo:

$$\text{MIN } \frac{CTOT}{S} = \left(\frac{\alpha}{S} + \beta\right)\sum_{i=p+1}^{m} C_i + \sum_{j=p+1}^{m}\sum_{i=1}^{m} c_{ij}x_{ij} + \sum_{j=m+1}^{m+n}\sum_{i=p+1}^{m} \bar{c}_{ij}\bar{x}_{ij}$$

Sujeito a:

$$0 \leq C_i = \begin{cases} 0 & \text{para } 1 \leq i \leq p \\ \begin{cases}\text{ou } 0 \\ \text{ou } 0 < LIM \leq C_i \leq \sum_{i=1}^{m} QL_i\end{cases} & \text{para } p+1 \leq i \leq m \\ CAP_i & \text{para } m+1 \leq i \leq m+n \end{cases}$$

$$0 \leq x_{ij} = \begin{cases} QL_i & \text{para } i=j \text{ e } 1 \leq i \leq m \\ 0 & \text{para } i \neq j \text{ e } C_i > 0 \\ 0 & \text{para } i \neq j \text{ e } C_j = 0 \end{cases}$$

$$0 \leq c_{ij} = \begin{cases} \delta_i d_{ij} & \text{para } 1 \leq i \leq m \text{ e } p+1 \leq j \leq m+n \\ \text{arbitrário mas fixado, caso contrário} \end{cases}$$

$$0 \leq \bar{x}_{ij} = 0 \quad \text{para} \begin{cases} m+1 \leq i \leq m+n \\ 1 \leq j \leq m \\ C_i = 0 \end{cases}$$

$$0 \leq \bar{c}_{ij} = \begin{cases} \bar{\delta}_i d_{ij} & \text{para } p+1 \leq i \leq m \text{ e } m+1 \leq j \leq m+n \\ \text{arbitrário mas fixado, caso contrário.} \end{cases}$$

$$\sum_{i=1}^{m} x_{ii} = \sum_{i=m+1}^{m+n} x_{ii}$$

$$x_{ii} = \sum_{j=p+1}^{m+n} x_{ij} \quad \text{se } C_i = 0$$

$$x_{ii} = \sum_{j=1}^{m} x_{ji} + \sum_{j=p+1}^{m} \bar{x}_{ji} \leq C_i \quad \text{para} \quad m+1 \leq i \leq m+n$$

$$\sum_{i=1}^{m} x_{ij} = \sum_{i=m+1}^{m+n} \bar{x}_{ji} \leq C_j \quad \text{para} \quad p+1 \leq j \leq m \text{ e } C_j > 0$$

1.3 - Considerações sobre o modelo

Note-se que:

a) As equações $c_{ij} = \delta_i d_{ij}$ e $\bar{c}_{ij} = \bar{\delta}_i d_{ij}$ indicam que os custos por tonelada de lixo transportado do i-ésimo setor são proporcionais à distância do transporte;

b) Os parâmetros α e β, relativos aos custos de construção e de manutenção respectivamente, surgem da necessidade de linearizar o modelo. O parâmetro α é o fator de proporcionalidade entre o custo da construção e a capacidade de uma estação de transferência de lixo, e o parâmetro β é o fator de proporcionalidade entre o custo da manutenção e a capacidade da ETL. Portanto, pode-se afirmar que o custo da construção de uma estação de transferência de lixo de capacidade C_i é $\alpha \,.\, C_i$, e o custo da manutenção por semana, de uma estação de transferência de lixo de capacidade de C_i é $\beta . C_i$.

c) x_{ij} exprime a quantidade de lixo coletado, transportado, entregue por semana, entre os setores e as estações de transferência de lixo, e entre os setores e os aterros finais, com custos de transporte por quilômetro e tonelada, δ_i , $\bar{x}_{ij}$ designa a quantidade de lixo transportado da estação de transferência para os aterros finais, com o custo de transporte por quilômetro e tonelada, $\bar{\delta}_i$. Supõe-se que cada $\bar{\delta}_i$ é menor que δ_i pois $\bar{\delta}_i$ representa o custo de transporte por quilômetro e tonelada que ocorre com o uso dos novos meios de transporte de capacidade muito maior.

d) As restrições em forma de equação entre os x_{ij} ou $\bar{x}_{ij}$ representam as condições necessárias que são associadas a qualquer problema de fluxo. Neste trabalho, leva-se em consideração o fluxo de lixo dos setores para os aterros finais, usando ou não estações de transferência. Finalmente, as restrições em forma de inequação entre x_{ij} e $\bar{x}_{ij}$ e as capacidades C_i representam os limites superiores para algum fluxo admissível de modo que nenhuma das estações de transferência e nenhum aterro final obtém mais que a sua capacidade de lixo por semana, isto é, a quantidade de lixo que chega em uma estação ou em um aterro final deve ser menor ou igual à sua capacidade.

Como já foi observado acima, o problema do modelo estudado neste trabalho é um problema da construção de um fluxo com custo mínimo, mas como os setores nos quais podem ser estabelecidas estações de transferência estão associados com a descontinuidade entre a capacidade zero (se a decisão é de não estabele cer uma estação de transferência) e a capacidade mínima LIM (se a decisão é estabelecer uma estação de transferência) o problema será decomposto em vários modelos lineares relativamente às possíveis decisões de estabelecer ou não estações de transferência.

Consequentemente são usados dois algoritmos para resolver o problema:

a) O algoritmo "out-of-kilter" para calcular um fluxo a custo mínimo;

b) Um algoritmo "branch-and-bound" que, partindo da solução do algoritmo "out-of-kilter", elimina as soluções desnecessárias.

Antes de apresentar estes algoritmos, no próximo capítulo serão vistas as noções da teoria dos grafos necessárias para entender o funcionamento desses algoritmos.

CAPÍTULO II

ALGUNS ASPECTOS DA TEORIA DE GRAFOS

2.1 - Digrafos

Um digrafo D = (N,A) é uma estrutura composta de um conjunto finito N de elementos, chamados nós e de um conjunto A de pares ordenados de nós, chamados arcos. Como exemplo, a figura 2.1 mostra um digrafo no qual os arcos são representados por setas.

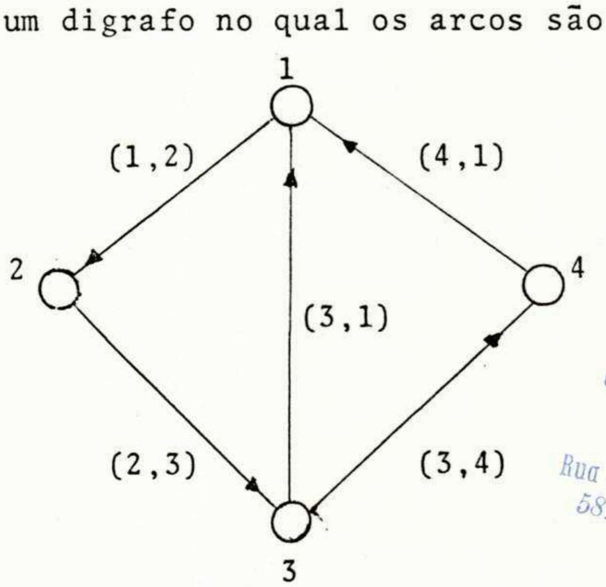


Fig. 2.1 - Exemplo de um digrafo

2.2 - Matriz de Adjacência de um digrafo

Dado um digrafo D = (N,A) define-se "matriz de adjacência" de D = (N,A), como a matriz $A = (a_{ij})$ cujos elementos são definidos por:

$a_{ij} = 1$ se o arco (i,j) pertence a D;

$a_{ij} = 0$ caso contrário.

A matriz de adjacência do digrafo da figura 2.1 é:

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Matriz de adjacência

2.3 - Caminho

Dado um digrafo D = (N,A), define-se um caminho em D como sendo qualquer sequência finita de arcos do tipo (i_0, i_1) $(i_1, i_2)\ldots(i_{k-1}, i_k)(i_k, i_{k+1})$. Isto é, em um caminho, o nó terminal de um arco é igual ao nó inicial do arco seguinte. Quando $i_{k+1} = i_0$ o caminho é dito fechado. Na figura 2.1 a sequência (4,1)(1,2) e (2,3) representa um caminho, o que não ocorre com a sequência (4,1)(3,1)(2,3).

2.4 - Digrafo Conexo

Um digrafo $D = (N,A)$ é dito conexo (fortemente conexo) se para cada par de nós i_k e i_j existe um caminho de i_k para i_j e um caminho de i_j para i_k.

A figura 2.1 representa um digrafo conexo pois existe caminhos de qualquer nó para outro qualquer.

2.5 - Digrafo Pesado

Na prática, os nós de um digrafo geralmente representa pontos fixos (locais) e os arcos representam as ligações entre os nós. Portanto, em alguns problemas, aos nós e aos arcos são associados números que são chamados respectivamente, pesos sobre os nós e pesos sobre os arcos. Os pesos de um nó podem representar a capacidade deste nó e o peso de um arco podem representar o comprimento deste arco ou custo de transporte sobre ele.

Define-se um digrafo pesado (ponderado) como sendo um digrafo $D = (N, A, v{:}N \to R^+ , c{:}A \to R^+)$ onde v associa a cada nó um número real, não negativo, e c associa a cada arco um número real, não negativo, isto é, D é pesado sobre os nós e sobre os arcos.

2.6 - Digrafo Fundado

Diz-se que $D = (N,A)$ é um digrafo fundado se não existe em D nenhum caminho fechado.

Quando num digrafo fundado $D = (N,A)$ existem dois nós i_0 e i_s de modo que:

i) i_0 não é nó terminal de nenhum arco de D;

ii) Existe um caminho de i_0 para qualquer $i_k \in N$;

iii) i_s não é nó inicial de nenhum arco de D;

iv) Existe um caminho de qualquer nó $i_k \in N$ para i_s, diz-se que i_0 é o nó fonte e i_k o nó foz, do digrafo $D = (N,A)$.

CAPÍTULO III

DESCRIÇÃO DOS ALGORITMOS

3.1 - Algoritmo Out-of-Kilter

O algoritmo que será apresentado, a seguir, determina uma circulação de fluxo a custo mínimo num digrafo. Considere c_{ij} o custo de transporte associado a cada unidade de fluxo de um nó i para um nó j, x_{ij} o fluxo correspondente ao arco (i,j). Sendo L_{ij} e S_{ij} os respectivos limites, inferiores e superiores, de fluxo sobre o arco (i,j), o modelo matemático associado ao problema de circulação de fluxo a custo mínimo é escrito como se segue:

$$\text{Min } Z = \sum_{i=1}^{m} \sum_{j=1}^{m} c_{ij} x_{ij}$$

$$(1) \qquad \sum_{j=1}^{m} x_{ij} = \sum_{j=1}^{m} x_{ji} \qquad \text{para todo } i=1,\dots,m$$

$$(2) \qquad 0 \leq L_{ij} \leq x_{ij} \leq S_{ij} \qquad \text{para todo } i=1,\dots,m$$

A restrição (1) designa que em cada nó do digrafo, a quantidade de fluxo que chega é igual à quantidade que sai; por isso, o modelo é chamado de modelo de circulação. Supõe-se que o custo c_{ij} e os limites inferiores e superiores, L_{ij} e S_{ij}, são números inteiros.

Se os limites em (2) não forem incluidos no modelo de circulação, as restrições do problema dual para cada (i,j) serão dadas por:

(3) $$U_i - U_j \leq c_{ij}$$

O algoritmo "out-of-kilter" é do tipo primal-dual e usa um custo relativo definido como sendo a diferença entre c_{ij} e $(U_i - U_j)$, que será denotado por $\bar{c}_{ij}$.

(4) $$\bar{c}_{ij} = c_{ij} - U_i + U_j$$

Inicia-se o método com um fluxo x_{ij}, inteiro, e que satisfaça a restrição de circulação (1), e valores inteiros arbitrários para as variáveis duais U_k e os custos relativos. Como a não existência de fluxo satisfaz a (1), inicia-se o processo com $x_{ij} = 0$ e $U_k = 0$.

Os demais passos são:

Passo 1: Se cada arco (i,j) satisfaz a uma das condições

1) $\bar{c}_{ij} > 0$ e $x_{ij} = L_{ij}$;

2) $\bar{c}_{ij} = 0$ e $L_{ij} \leq x_{ij} \leq S_{ij}$;

3) $\bar{c}_{ij} < 0$ e $x_{ij} = S_{ij}$,

então todos os arcos estão "in-kilter" e a solução é ótima. Pare. Caso contrário, vá para o passo 2.

Passo 2: Calcule o número de "out-of-kilter" (NOK_i), quando possível, de acordo com:

1) $NOK_1 = L_{ij} - x_{ij}$ se $\bar{c}_{ij} > 0$ e $x_{ij} < L_{ij}$

2) $NOK_2 = \bar{c}_{ij}(x_{ij}-L_{ij})$ se $\bar{c}_{ij} > 0$ e $x_{ij} > L_{ij}$

3) $NOK_3 = L_{ij} - x_{ij}$ se $\bar{c}_{ij} = 0$ e $x_{ij} < L_{ij}$

4) $NOK_4 = x_{ij} - S_{ij}$ se $\bar{c}_{ij} = 0$ e $x_{ij} > S_{ij}$

5) $NOK_5 = \bar{c}_{ij}(x_{ij}-S_{ij})$ se $\bar{c}_{ij} < 0$ e $x_{ij} < S_{ij}$

6) $NOK_6 = x_{ij} - S_{ij}$ se $\bar{c}_{ij} < 0$ e $x_{ij} > S_{ij}$

selecione um arco (i_0 , j_0) com o maior NOK_i .

Passo 3: Se o arco (i_0 , j_0) satisfaz uma das condições ímpares do passo 2, rotule j_0 com $(\wedge)$.
Caso contrário: rotule i_0 com $(\wedge)$.

Passo 4: Faça K_0 igual ao nó não rotulado.
Se existe um caminho do nó rotulado para o nó não rotulado do arco (i_0 , j_0) vá ao passo 5.
Caso contrário vá ao passo 10.

Passo 5: Faça K igual ao nó rotulado.

Passo 6: Rotule com $(\wedge)$ todos os nós terminais dos arcos (k,h) iniciando em k que não são rotulados e os quais satisfazem a

$$1) \begin{cases} \bar{c}_{kh} > 0 \text{ e } x_{kh} < L_{kh} \\ c_{kh} \leq 0 \text{ e } x_{kh} < S_{kh} \end{cases}$$

Rotule todos os nós iniciais dos arcos (h,k) chegando em k que não sejam rotulados e os quais satisfazem a:

$$2 \begin{cases} \bar{c}_{hk} \geq 0 \quad e \quad x_{hk} > L_{hk} \\ \bar{c}_{hk} < 0 \quad e \quad x_{hk} > S_{hk} \end{cases}$$

Passo 7: Se $h = K_0$ vá ao passo 8.

Caso contrário, volte ao passo 5.

Passo 8: No caminho encontrado, faça

a) $\Delta_i = L_{kh} - x_{kh}$ se $x_{kh} < L_{kh}$

b) $\Delta_i = S_{kh} - x_{kh}$ se $x_{kh} < S_{kh}$

c) $\Delta_i = L_{hk} - x_{hk}$ se $x_{hk} > L_{hk}$

d) $\Delta_i = S_{hk} - x_{hk}$ se $x_{hk} > S_{hk}$

Faça: $\Delta = \mathrm{Min}(\mathrm{abs}(\Delta_i))$;

Passo 9: Incremente o fluxo x_{ij} pela quantidade Δ sobre os arcos (k,h) que satisfazem a condição 1 do passo 6 e decremente o fluxo pela quantidade Δ sobre os arcos (h,k) que satisfazem a condição 2 do passo 6. Volte ao passo 1.

Passo 10: Faça:

$C_1 = \mathrm{Min}\ \{\bar{c}_{ij} > 0$ / i é rotulado e j não é rotulado e $x_{ij} \leq S_{ij}\}$

$C_2 = \mathrm{Min}\ \{\bar{c}_{ij} < 0$ / i não é rotulado e j é rotulado e $x_{ij} > L_{ij}\}$

$\bar{C} = \mathrm{Min}(C_1\ ,\ -C_2\ ,\ \infty)$

Passo 11: Se C_1 e C_2 não são definidos, isto é, os conjuntos são vazios, o problema não tem solução. Pare.

Caso contrário, faça:

$U_k = U_k + \bar{C}$ para todo nó k rotulado e revise os custos relativos de acordo com:

$\bar{c}_{ij} = c_{ij} - U_i + U_j$ para todo i e todo j.

Passo 12: Se o arco (i_0, j_0) está em "out-of-kilter", volte ao passo 4.

Caso contrário, volte ao passo 1.

3.2 - Algoritmo Branch-and-Bound

Como existe uma restrição sobre a capacidade mínima para que se possa construir estações de transferência de lixo e como este problema é de minimização de custos e o problema não tem somente a restrição de que a capacidade mínima de cada estação seja maior ou igual a zero. Observamos que uma solução encontrada pela aplicação do algoritmo "out-of-kilter" não precisa satisfazer a estas restrições sobre as capacidades mínimas para decisão de instalar alguma ETL. Como estas restrições precisam ser também satisfeitas, então será introduzido um processo adicional.

Para cada possibilidade que se pode escolher algum local para serem construidas estações de transferência de lixo, determinamos pelo algoritmo "out-of-kilter" a solução ótima correspondente. Chamamos as soluções ótimas, que satisfazem também as restrições sobre as capacidades mínimas das estações, de soluções admissíveis e entre as soluções admissíveis escolhemos uma que tem o menor custo.

É importante frisar que, não necessariamente este processo precisa pesquisar todas as possibilidades. Se na pesquisa é encontrada uma solução admissível, então não será necessário examinar alguma outra possibilidade com menos estações para cons truir que a já pesquisada, pois cada possibilidade deste tipo terá um custo maior. Este resultado obviamente corresponde a um processo do tipo "branch-and-bound", que será apresentado a seguir em forma de passos.

Supondo-se que existem P possíveis locais onde podem ser construidas estações de transferência de lixo, define-se uma seleção com P - I elementos dos P elementos dados, como sendo uma das possibilidades que obtém-se quando elimina-se I locais, isto é, quando admite-se apenas a construção de P - I ETL. Se uma seleção fornece uma solução que satisfaz as restrições sobre as capacidades mínimas das ETL então diz-se que esta solução é admissível.

Passo 0: Define-se P como sendo o número de possíveis ETL. Faça $I = 0$;

Passo 1: Encontre todas as $\binom{P}{P - I}$ seleções de P - I elementos dentre os P elementos dados;

Passo 2: Aplique o algoritmo "out-of-kilter" com respeito as seleções encontradas no passo 1 e que não seja subconjunto de nenhuma seleção admissível já encontrada. Vá para o passo 3. Se não existe nenhuma seleção desse tipo vá para o passo 4;

Passo 3: Guarde as seleções correspondentes a solução admissível e o custo correspondente. Faça $I = I + 1$ volte ao passo 1;

Passo 4: Dentre todas as seleções e soluções admissíveis encontradas escolha uma de menor custo. Pare.

CAPÍTULO IV

RESULTADOS COMPUTACIONAIS

Foi incluido neste trabalho a codificação de um programa em linguagem WATFIV para resolver o problema de circulação de fluxo a custo mínimo (algoritmo "out-of-kilter") e o algoritmo "branch-and-bound" para selecionar os melhores locais a custo mínimo. A codificação é baseado nos algoritmos apresentados no capítulo 3. Os mesmos foram implementados para resolver o problema de localização de estações de transferência de lixo. O algoritmo "branch-and-bound" só será ativado se, na primeira execução do algoritmo "out-of-kilter" existe um arco que representa uma possível ETL não é zero nem maior que a capacidade mínima prescrita para viabilidade de construção da estação.

O algoritmo foi testado com alguns dados estimados como sejam: custos de construção, manutenção e de transporte, e outros dados reais obtidos da cidade do Recife, sejam: a divisão da cidade em 59 setores, 5 locais possíveis a serem instaladas estações de transferência e o local do aterro final, os quais estão representados no mapa por 1,...,56, E1,...,E5 e A respectivamente.

A montagem do digrafo foi feita tomando o centro de cada setor, os cinco locais possíveis para instalação de ETL's o aterro final como sendo os nós e a distância entre os setores e o aterro e entre as ETL e o aterro final como sendo os ramos do digrafo. A figura resultante tem 71 nós e 423 ramos como mostra a figura 4.1 .

O teste do algoritmo foi feito considerando um tipo de ETL simples e o horizonte de planejamento (S) igual a 1 ano. O local escolhido pelo algoritmo foi E1 como mostra o mapa no Anexo II.

Os dados de entrada do programa são os seguintes:

a) Um cartão, especificando o número de nós e o número de ramos. Um outro com a capacidade mínima necessária para construir uma ETL e o custo associado à construção e manutenção da mesma. Estes dados são perfurados no mesmo cartão para um ou mais locais; os mesmos são perfurados em formato livre.

b) A cada ramo é associado seis valores: Os dois primeiros são os nós terminais do ramo; o terceiro e o quarto, o limite inferior e superior respectivamente; o quinto o fluxo, e o sexto representa o custo correspondente a esse ramo. Esses valores são também perfurados em formato livre.

Com esses dados de entrada, o programa determina inicialmente uma circulação de fluxo a custo mínimo. Se o fluxo que passa através de cada ramo que representa uma ETL não for zero ou menor que a capacidade mínima necessária para construir uma ETL, o algoritmo "branch-and-bound" é ativado e é feita a operação de abrir ou fechar uma facilidade, até que o fluxo seja ze ro ou maior ou igual à capacidade mínima acima citado.

A saída do programa são os locais de implantação de ETL's e seu custo total com solução otimizado.

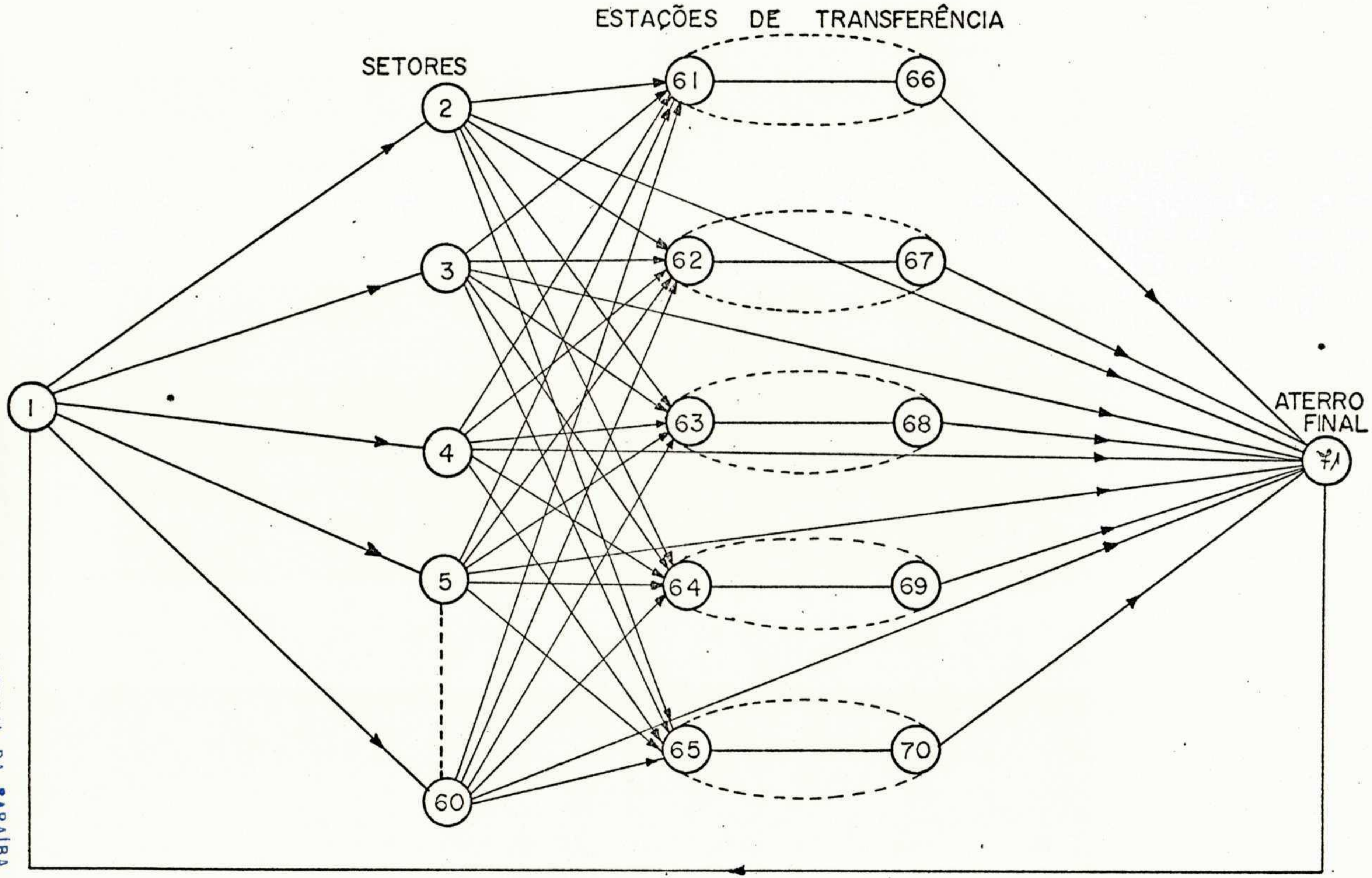

ESTAÇÕES DE TRANSFERÊNCIA
SETORES
ATERRO FINAL
1
2
3
4
5
60
61
62
63
64
65
66
67
68
69
70
71

fig 4.1

CAPÍTULO V

CONCLUSÃO

Neste trabalho foi considerado o problema de distribuição de serviço público. O esforço principal foi concentrado na otimização do problema de localização de estações de transferência de lixo. Os algoritmos aqui implementados servem também para determinar a localização de facilidades no setor privado, por exemplo, a localização de armazém.

É bom frisar que os algoritmos aqui implementados fornecem soluções ótimas relativa à formulação linear do problema; mas, devido à complexidade do problema na realidade esta formulação linear, não reflete precisamente a realidade. Pode ser dito que as técnicas apresentadas podem ser utilizadas como instrumentos para encontrar uma boa solução para o problema.

ANEXO I

PROGRAMA

O programa apresentado neste anexo está implementado no sistema IBM/370-145 na linguagem de programação WATFIV.

```
$JOB          JOSE,TIME=45,PAGES=28
$[illegible]
C        ****-----------------------------------------------------------*****
C        ****     UNIVERSIDADE FEDERAL DA PARAIBA  (UFPB)               ****
C        ****     DEPARTAMENTO DE SISTEMA E COMPUTACAO                  ****
C        ****************** TITULO ***************************************
C        ****     UM ESTUDO DE VIABILIDADE DE IMPLANTACAO DE ESTACAO****
C        ****     DE TRANSFERENCIA DE LIXO (ETL)                        ****
C        ****     JOSE FERREIRA LIMA FILHO                              *****
C        ****                                                           *****
C        -----------------------------------------------------------------
C     ****  SUBROTINA QUE VERIFICA O CASO QUE O ARCO SE ENCONTRA   ****
C        -----------------------------------------------------------------
         SUBROUTINE CASOS(CUSBA,CUSTO,VARDU,FLUXO,LIMIF,CAPAC,CASO,K,
     *                INDEX)
         REAL CUSTO(423),CUSBA(423),VARDU(71)
         INTEGER 2 CASO(423),INDEX(423,2),LIMIF(423),CAPAC(423),FLUXO(423
     *   ),K,I,J
         I=INDEX(K,1)
         J=INDEX(K,2)
C     ****  CALCULA O CUSTO RELATIVO                                ****
         CUSBA(K)=CUSTO(K)+VARDU(I)-VARDU(J)
         IF(CUSBA(K).GT.0)THEN DO
            IF(FLUXO(K).EQ.LIMIF(K))THEN DO
               CASO(K)=1
            END IF
            IF(FLUXO(K).LT.LIMIF(K))THEN DO
               CASO(K)=4
            END IF
            IF(FLUXO(K).GT.LIMIF(K))THEN DO
               CASO(K)=5                                                 CASC01
            END IF
          END IF
         IF(CUSBA(K).EQ.0)THEN DO
            IF(LIMIF(K).LE.FLUXO(K).AND.FLUXO(K).LE.CAPAC(K))THEN DO
               CASO(K)=2
            END IF                                                       CASC02
            IF(FLUXO(K).LT.LIMIF(K))THEN DO
               CASO(K)=6
            END IF                                                       CASC02
            IF(FLUXO(K).GT.CAPAC(K))THEN DO
               CASO(K)=7                                                 CASC02
            END IF
          END IF
         IF(CUSBA(K).LT.0)THEN DO
            IF(FLUXO(K).EQ.CAPAC(K))THEN DO
               CASO(K)=3
            END IF
            IF(FLUXO(K).LT.CAPAC(K))THEN DO
               CASO(K)=8
            END IF
            IF(FLUXO(K).GT.CAPAC(K))THEN DO
               CASO(K)=9
            END IF
          END IF
         RETURN                                                          CASC04
         END                                                             CASC04
C        -----------------------------------------------------------------
C        SUBROTINA QUE DETERMINA O MENOR ELEMENTO ENTRE DOIS DELES
C        -----------------------------------------------------------------
```

```
      SUBROUTINE MIN(IA,AH,LD)
      INTEGER 2 IA,AH,IK,LD                                          MIN100??
      IF(IA.LE.AH)GO TO 30                                           MINI0030
      IK=AH                                                          MIN100??
      AH=IA                                                          MINI0050
      IA=IK
30    LD=IA
      RETURN                                                         MIN10???
      END
```

SUBROTINA QUE EXAMINA A SOLUCAO EH ADMISSIVEL

```
      SUBROUTINE EXSOLA (I,J,K,L.CHSOL,CADSOL,VETSOL,APVSOL)
      INTEGER*2 I,J,K,CHSOL,CHSOL1,CADSOL(36),VETSOL(5),APVSOL,MM,NN,L
      CHSOL=0
      DO 900 NN=1,APVSOL
         M=CADSOL(VETSOL(NN))
         IF(M.EQ.1)THEN DO
      IF((CADSOL(VETSOL (NN)+M).EQ.I).OR.(CADSOL(VETSOL (NN)+M).EQ.J).OR.
     *   (CADSOL(VETSOL (NN)+M).EQ.K).OR.(CADSOL(VETSOL(NN)+M).EQ.L))
     *   THEN DO
            CHSOL=1
         RETURN
         END IF
         CHSOL=0
         END IF
         IF(M EQ.2)THEN DO
            MM=1
            IF((CADSOL( VETSOL(NN)+MM).EQ.I .AND.CADSOL(VETSOL(NN)+MM+1).
     *        EQ.J).OR.(CADSOL(VETSOL(NN)+MM).EQ.I.AND.CADSOL(VETSOL(NN)
     *        +MM+1).EQ.K).OR.(CADSOL(VETSOL(NN)+MM).EQ.J.AND.CADSOL(
     *        VETSOL(NN)+MM+1).EQ.K).OR.(CADSOL(VETSOL(NN)+MM).EQ.I.AND.
     *        CADSOL(VETSOL(NN)+MM+1).EQ.L).OR.(CADSOL(VETSOL(NN)+MM).EQ
     *        .J.AND.CADSOL(VETSOL(NN)+MM).EQ.L).OR.(CADSOL(VETSOL(NN)+
     *        MM).EQ K.AND.CADSOL(VETSOL(NN)+MM+1).EQ.L))THEN DO
              CHSOL=1
              RETURN
            END IF
            CHSOL=0
         END IF
         IF(M EQ.3)THEN DO
            MM=1
            IF((CADSOL( VETSOL(NN)+MM).EQ.I .AND.CADSOL(VETSOL(NN)+MM+1).
     *         EQ.J.AND.CADSOL(VETSOL(NN)+MM+2).EQ.K).OR.(CADSOL
     *         (VETSOL(NN)+MM).EQ.I.AND.CADSOL(VETSOL(NN)+MM+1).EQ.J.
     *         AND.CADSOL(VETSOL(NN)+MM+2).EQ.L).OR.(CADSOL(VETSOL(NN)+
     *         MM) EQ. I. AND. CADSOL(
     *       VETSOL(NN)+MM+1).EQ.L.AND.CADSOL(VETSOL(NN)+MM+2).EQ.K
     *       ).OR.(CADSOL(VETSOL(NN)+MM).EQ.J.AND.CADSOL(VETSOL(NN)+MM
     *        +1).EQ.K.AND.CADSOL(VETSOL(NN)+MM+2).EQ.L))THEN DO
            CHSOL=1
              RETURN
          END IF
              CHSOL=0
         END IF
900   CONTINUE
      RETURN
      END
```

```
C     ------------------------------------------------------------------
C          SUBROTINA MUDANCA DA VARIAVEL DUAL
C     ------------------------------------------------------------------

      SUBROUTINE MUDAN(VARDU,NRAMOS,NOS, CAMIN,FLUXO,CAPAC,CUSBA,BETA,
     *                 LIMIF, INDEX)
      REAL DETA,DELTA ,INF,VARDU(71),CUSBA(423),BETA
      INTEGER 2 CAMIN (71),INDEX(423,2),FLUXO(423),LIMIF(423)
     *        .CAPAC(423),K,M,I,N,NOS, NRAMOS
      DELTA=99999999.                                                    MUDA0
      DETA=99999999.                                                     MUDA0
      INF 99999999.                                                      MUDA0
      DO 10 K=1,NRAMOS                                                   MUDA0
         M INDEX(K,1)
         I INDEX(K,2)
         IF(CAMIN(M).EQ.1.AND.CAMIN(I).EQ.0.AND.FLUXO(K).LE.CAPAC(K)
     *      .AND.CUSBA(K).GT.0)THEN DO
           IF(CUSBA(K).LT.DELTA)DELTA=CUSBA(K)
         END IF                                                          MUDA0
           IF(CAMIN(M).EQ.0.AND.CAMIN(I).EQ.1.AND.CUSBA(K).LT.0.AND.
     *        FLUXO(K).GE.LIMIF(K))THEN DO
           IF(-CUSBA(K).LT.DETA)DETA=-CUSBA(K)
         END IF                                                          MUDA0
10    CONTINUE                                                           MUDA0
      BETA=99999999.                                                     MUDA0
      IF(DELTA.LT.BETA)BETA=DELTA                                        MUDA0
      IF(DETA.LT.BETA)BETA=DETA
      IF(BETA.LT.INF)THEN DO                                             MUDA0
        DO 20 K=1,NOS                                                    MUDA0
           IF(CAMIN(K).EQ.0)VARDU(K)=VARDU(K)+BETA
20      CONTINUE                                                         MUDA0
      END IF                                                             MUDA0
      RETURN                                                             MUDA0
      END                                                                MUDA0
C     ------------------------------------------------------------------
C     ****      SUBRUTINA  OUT- OF - KILTER                         *****
C     ------------------------------------------------------------------

      SUBROUTINE OUT(CUSTO,VARDU,LIMIF,FLUXO,CAPAC,INDEX,NRAMOS,NUMEST,
     *              CAPSES,ESTAC,LIMCAP,NOS,IK2,CHAVE)
      INTEGER 2 CASO(423),SAIR(423,71),ENTRA(423,71),NUENT(423),NSAI(423
     *        ),LABEL(71,4),ISCAN(71),CAMIN(71),P,D,V,T,INDEX(423,2)
     *        ,Q,CHAVE,ESTAC(5),NRAMOS,NOS,K,KK,IK2,IC,NUMEST,J,KM,I,
     *        IX(9),LIMIF(423),FLUXO(423),EPS(423),LD,CAPAC(423),IJ,
     *        CAPSES(5),IV,IG,Y,Z,W
      REAL VARDU(71),CUSTO(423),CUSBA(423),BETA,FUN(5),INF
      DATA IX/54,110,164,220,274,330,384,423/
      LIMCAP=
C     ****************------------------------************************
C     ****      CADA LABEL E UMA LINHA COM 4 ELEMENTOS            *****
C     ****         COLUNA 1 O NO A PARTIR DO QUAL SE ROTULA       ****
C     ****         COLUNA 2 NO ROTULADO                           ****
C     ****         COLUNA 3 ARCO CORRESPONDENTES                  ****
C     ****         COLUNA 4 CONTEM 0 SE AUMENTA O FLUXO           ****
C     ****                         1 SE DIMINUI O FLUXO           ****
C     ****************-----------------------*************************
      INF=99999999.
      IF(IK2.EQ.1)GO TO 40
      DO 114 I=1,NRAMOS                                                  JOSL00
         NUENT(I)=0                                                      JOSE00
```

```
14     NSAI(I)=0
0   DO 1 K 1,NRAMOS.
       LIMA=0
       KK=1
       IF(IK2.EQ.1)GO TO 10
       Y=INDEX(K,1)
       Z=INDEX(K,2)
       J=NSAI(Y)+1
       NSAI(Y)=J
       V=NUENT(Z)+1
       NUENT(Z)=V
       SAIR(Y.J)=K
       ENTRA(Z,V)=K
)      CALL CASOS(CUSBA,CUSTO,VARDU,FLUXO,LIMIF,CAPAC,CASO,K,INDEX)
  ***                   SUBROTINA ROTULAR                           *****
    CONTINUE
    IK2=1
    WHILE(KK.LE.NRAMOS)DO
  ****  VERIFICA SE EXISTE ALGUM ARCO EM OUT-OF-KILTER    ****
      WHILE(CASO(KK).GT.3)DO
        K=1
        WHILE(K.LE.NOS)DO
          CAMIN(K)=0
          ISCAN(K)=0
          LABEL(K,2)=0
          K=K+1
        END WHILE
        IF(CASO(KK).EQ.4)THEN DO
          LABEL(1,1)=INDEX(KK,1)
          LABEL(1,2)=INDEX(  ,  )
          LABEL(1,3)=KK
          LABEL(1,4)=0
          EPS(1)=LIMIF(KK)-FLUXO(KK)
          P=INDEX(KK,2)
          CAMIN(P)=1
          META=INDEX(KK,1)
        END IF
        IF(CASO(KK).EQ.6.OR.CASO(KK).EQ.8)THEN DO
          LABEL(1,1)=INDEX(KK,1)
          LABEL(1,2)=INDEX(KK,2)
          LABEL(1,3)=KK
          LABEL(1,4)=0
          EPS(1)=CAPAC(KK)-FLUXO(KK)
          META=INDEX(KK,1)
          P=INDEX(KK,2)
          CAMIN(P)=1
        END IF
        IF(CASO(KK).EQ.5.OR.CASO(KK).EQ.7)THEN DO
          LABEL(1,1)=INDEX(KK,2)
          LABEL(1,2)=INDEX(KK,1)
          LABEL(1,3)=KK
          LABEL(1,4)=1
          EPS(1)=FLUXO(KK)-LIMIF(KK)
          META=INDEX(KK,2)
          P=INDEX(KK,1)
          CAMIN(P)=1
        END IF
        IF(CASO(KK).EQ.9)THEN DO
          LABEL(1,1)=INDEX(KK,2)
          LABEL(1,2)=INDEX(KK,1)
```

```
      LABEL(1,3)=KK
      LABEL(1,4)=1
      EPS(1)=FLUXO(KK)-CAPAC(KK)
      META=INDEX(KK,2)
      P=INDEX(KK,1)
      CAMIN(P)=1
   END IF
   IC=1
   J=1
   ISCAN(1)=LABEL(1,2)
   WHILE(LABEL(IC,2).NE.META.AND.ISCAN(J).NE.0)DO
      T=1
      Y=ISCAN(J)
      WHILE(T.LE.NSAI(Y).AND.LABEL(IC,2).NE.META)DO
         K=SAIR(Y,T)
         D=INDEX(K,2)
         IF(CAMIN(D).EQ.0)THEN DO
         IF (CUSBA(K).GT.0.AND.FLUXO(K).LT.LIMIF(K))THEN DO
               IC=IC+1
               LABEL (IC,1)=INDEX(K,1)
               LABEL (IC,2)=INDEX(K,2)
               LABEL (IC,3)=K
               LABEL (IC,4)=0
               IV=LIMIF(K)-FLUXO(K)
               IG=EPS(J)
               CAMIN (D)=1
               CALL MIN(IV,IG,LD)
               EPS(IC)=LD
            END IF
         IF(CUSBA(K).LE.0.AND.FLUXO(K).LT.CAPAC(K))THEN DO
               IC=IC+1
               LABEL (IC,1)=INDEX(K,1)
               LABEL (IC,2)=INDEX(K,2)
               LABEL (IC,3)=K
               LABEL (IC,4)=0
            IV=CAPAC(K)-FLUXO(K)
               IG=EPS(J)
               CAMIN (D)=1
               CALL MIN(IV,IG,LD)
               EPS(IC)=LD
            END IF
         END IF
         T=T+1
      END WHILE
      T=1
      Z=ISCAN(J)
      WHILE(T.LE.NUENT(Z).AND.LABEL(IC,2).NE.META)DO
         K=ENTRA(Z,T)
         D=INDEX(K,1)
         IF(CAMIN(D).EQ.0) THEN DO
            IF(CUSBA(K).GE.0.AND.FLUXO(K).GT.LIMIF(K))THEN DO
               IC=IC+1
                LABEL(IC,1)=INDEX(K,2)
                LABEL(IC,2)=INDEX(K,1)
               LABEL (IC,3)=K
               LABEL (IC,4)=1
                IV=FLUXO(K)-LIMIF(K)
               IG=EPS(J)
               CAMIN(D)=1
               CALL MIN(IV,IG,LD)
```

```
              EPS(IC)=LD
            END IF
            IF(CUSBA(K).LT.0.AND.FLUXO(K).GT.CAPAC(K))THEN DO
              IC=IC+1
              LABEL(IC,1)=INDEX(K,2)
              LABEL(IC,2)=INDEX(K,1)
              LABEL(IC,3)=K
              LABEL(IC,4)=1
              IV=FLUXO(K)-CAPAC(K)
              IG=EPS(J)
              CAMIN(D)=1
              CALL MIN(IV,IG,LD)
              EPS(IC)=LD
            END IF
          END IF
          T=T+1
        END WHILE
        J=J+1
        ISCAN(J)=LABEL(J,2)
      END WHILE
        IF(ISCAN(J).EQ.0)THEN DO
          CALL MUDAN(VARDU,NRAMOS,NOS,CAMIN,FLUXO,CAPAC,CUSBA,BETA,
     *               LIMIF,INDEX)
          IF(BETA.EQ.INF)THEN DO
            LIMA=1
            GO TO 80
          ELSE DO
            DO 50 I=1,NRAMOS
              CALL CASOS(CUSBA,CUSTO,VARDU,FLUXO,LIMIF,CAPAC,CASO,
     *                   I,INDEX)
50          CONTINUE
            GO TO 80
          END IF
        END IF
        P=META
            ALTERACAO DO FLUXO
        DO 60 I=1,IC
          K=IC+1-I
          IF(LABEL(K,2).EQ.P)THEN DO
            KM=LABEL(K,3)
            IF(LABEL(K,4).EQ.1)THEN DO
              FLUXO(KM)=FLUXO(KM)-EPS(IC)
            ELSE DO
              FLUXO(KM)=FLUXO(KM)+EPS(IC)
            END IF
            P=LABEL(K,1)
            CALL CASOS(CUSBA,CUSTO,VARDU,FLUXO,LIMIF,CAPAC,CASO,KM,
     *                 INDEX)
          END IF
60      CONTINUE
80      TIPO=0
      END WHILE
      KK=KK+1
    END WHILE
    IF(LIMA.EQ.1)THEN DO
      WRITE(6,660)
660 FORMAT('1',53X,'SOLUCAO IMPOSSIVEL')
    STOP
    END IF
    SOM=0
```

```
      SOMA=0                                                            BRBC00
      DO 160 I=1,NUMEST
         IF(FLUXO(ESTAC(I)).EQ.0.OR.FLUXO(ESTAC(I)).GE.LIMCAP)GO TO 160
         CHAVE=0                                                        BRBC00
      GO TO 111
 160  CONTINUE                                                          BRBC00
      DO 170 I=1,NRAMOS                                                 BRBC00
 170     SOMA=SOMA+CUSTO(I)*FLUXO(I)
      IF(CHAVE.EQ.1)THEN DO                                             BRBC00
         SOM=SOMA                                                       BRBC01
 999  WRITE(6,777)                                                      IMPR00
 777  FORMAT('1',2X)                                                    IMPR02
      K=1
      DO 222 IK=1,8
         WRITE(6,650)
         NRAMOS=IX(IK)
      WRITE(6,444)(I,INDEX(I,1),INDEX(I,2),LIMIF(I),FLUXO(I),CUSTO(I),
     *             CAPAC(I),I=K,NRAMOS)
         K=NRAMOS+1
 650  FORMAT(3X,120('-'),//2(4X,'NUME',2X,'NOIN',3X,'NOFI',6X,'LIMI'
     *,5X,'FLUXO',6X,'CUSTO',6X,'LIMS')/3X,120('-')/)
 444  FORMAT(2(3(3X,I3),2(2X,I9),2X,F9.0,2),I9)/)
 222  CONTINUE
      END IF                                                            BRBC11
 111  RETURN
      END
C     ----------------------------------------------------------------
C          PROGRAMA PRINCIPAL                                     *****
C     ----------------------------------------------------------------

      INTEGER*2 FLUXO(423),CAPAC(423),ESTAC(5),LIMIF(423),INDEX(423,2),
     *          IX(8),NUMEST,NOS,NRAMOS,I,APCSOL,APVSOL,LIMPES,
     *          ESCMIN,CACMIN,CAPSES(5),VETSOL(5),PTSOL,CONJEF,L
     >          ,CADSOL(36),PTVSOL,J,K,CHSOL,IK2,CHAVE,YI(5)
      REAL SOMA,CUSMIN,CUSTO(423),VARDU(71),FUN(5)
      DATA YI/5*1/
      DATA IX/54,110,164,220,274,330,384,423/                           PRIN00
      CACMIN=0
      READ,NUMEST,(ESTAC(I),CAPSES(I),FUN(I),I=1,NUMEST)
      READ,NRAMOS,NOS,(VARDU(I),I=1,NOS)
      READ,(INDEX(I,1),INDEX(I,2),LIMIF(I),FLUXO(I),CUSTO(I),CAPAC(I),
     *     I=1,NRAMOS)
C           FUN    CUSTO DE CONSTRUCAO                        -------------
C           NUMEST  NUMEROS DE POSSIVEIS ETL          --------------------
C           CADSOL CADEIA DE SOLUCAO                 --------------------
C           NRAMOS NUMEROS DE RAMOS                    ------------------
      CHAVE=1                                                           PRIN01
      IK2=0                                                             PRIN01
      DO 474 I=1,NRAMOS
         LIMIF(I)=0
 474     CAPAC(I)=105
      READ,(LIMIF(I),I=1,59)
      READ,(CUSTO(I),I=60,353)
      READ,(CUSTO(I),I=354,412)
      LIMCAP=26
      SEMAN=52                                                          JPSE0
      SOMA=0.
      WRITE(6,1)
   1  FORMAT('1',3X,87('-'),/20X,'UM ESTUDO DE VIABILIDADE DE IMPLANTACA
     *O DE ESTACAO DE TRANSFEREN-',/20X,'CIA DE LIXO',/4X,87('-'),//20X,
```

```
     *'CASO EM ESTUDO', 9X, 'CINCO LOCAIS POSSIVEIS PARA IMPLANTACAO DE', /
     *42X, 'ESTACAO DE TRANSFERENCIA DE LIXO (ETL)', //20X, 'LEGENDA PARA
     *SOLUCAO   1-SE O LOCAL FOI ESCOLHIDO', //44X, 'O-CASO CONTRARIO', 50X
     *, //33X, 'SOLUCOES ADMISSIVEIS ENCONTRADAS'///)
      DO 454 I=1,59                                                    JOSE01
         CAPAC(I)=LIMIF(I)
454      CUSTO(I)=0
      DO 70 I=1,NUMEST
         CAPAC(ESTAC(I))=CAPSES(I)
         CUSTO(ESTAC(I))=CUSTO(ESTAC(I))+FUN(I)/SEMAN
70    CONTINUE                                                         JOSE01
      DO 113 I=60,353
113      CUSTO(I)=60*CUSTO(I)
      DO 112 I=354,418
112      CUSTO(I)=150*CUSTO(I)
      CALL OUT(CUSTO,VARDU,LIMIF,FLUXO,CAPAC,INDEX,NRAMOS,NUMEST,
     *         CAPSES,ESTAC,LIMCAP,NOS,IK2,CHAVE)
C
C     ----------------------------------------------------------------
C      PESQUISA COM UMA ESTACAO DE TRANSFERENCIA DE LIXO FECHADA
C     ----------------------------------------------------------------
      CONJEF=I
      APCSOL=0
      APVSOL=0
      LIMPES=NUMEST
      CUSMIN=9999999.
      ESCMIN=0
      CACMIN=0
      DO 111 I=1,LIMPES
         CAPAC(ESTAC(I))=0
C      CHAMADA DO ALGORITMO DE OUT-OF-KILTER
         CALL OUT(CUSTO,VARDU,LIMIF,FLUXO,CAPAC,INDEX,NRAMOS,NUMEST,
     *         CAPSES,ESTAC,LIMCAP,NOS,IK2,CHAVE)
C      VERIFICA SE O FLUXO EH VIAVEL
         DO 51 N=1,LIMPES
            IF(FLUXO(ESTAC(N)).GT.0.AND.FLUXO(ESTAC(N)).LT.LIMCAP)
     *         GO TO 100
51       CONTINUE
         APVSOL=APVSOL+1
         APCSOL=APCSOL+1
         VETSOL(APVSOL)=APCSOL
         CADSOL(VETSOL(APVSOL))=CONJEF
         APCSOL=APCSOL+1
         CADSOL(APCSOL)=I
         DO 61 N=1,NRAMOS
            SOMA=SOMA+CUSTO(N)*FLUXO(N)
61       CONTINUE
         IF(SOMA.LT.CUSMIN)THEN DO
           CUSMIN=SOMA
           ESCMIN=I
           CACMIN=VETSOL(APVSOL)
           YI(CADSOL(CACMIN+1))=0
           WRITE(6,3)(YI(II),II=1,NUMEST),CUSMIN
3          FORMAT(20X,' * LOCAIS POSSIVEIS DAS ETL  1   2   3   4   5',/
     *     /23X, 'SOLUCAO ADMISSIVEL',5X,5(2X,I2),//23X,'CUSTO DESTA SO
     *LUCAO',4X,F15.2//)
           YI(CADSOL(CACMIN+1))=1
         END IF
         SOMA=0.
100      CAPAC(ESTAC(I))=CAPSES(I)
```

```
11   CONTINUE
     ---------------------------------------------------------------
      PESQUISA COM DUAS ESTACOES DE TRANSFERENCIA FECHADAS
     ---------------------------------------------------------------
     IF(NUMEST.GT.1)THEN DO
     J=0
     L=0
     K=0
     PTVSOL=APVSOL
     CONJEF=CONJEF+1
     LIMPES=LIMPES-1
     DO 222 I=1,LIMPES
      TESTA SE HOUVE ALGUMA SOLUCAO ADMISSIVEL
        IF(PTVSOL.GT.0)THEN DO
      PESQUISA SE A SOLUCAO EH ADMISSIVEL
        CALL EXSOLA(I,J,K,L,CHSOL,CADSOL,VETSOL,APVSOL)
        IF (CHSOL.EQ.1)GO TO 222
        END IF
          CAPAC(ESTAC(I))=0
        JJ=I+1
        DO 82 J=JJ,NUMEST
           IF(PTVSOL.GT.0)THEN DO
              CALL EXSOLA(I,J,K,L,CHSOL,CADSOL,VETSOL,APVSOL)
             IF(CHSOL.EQ.1)GO TO 82
           END IF
             CAPAC(ESTAC(J))=0
      CHAMADA DO ALGORITMO DE OUT-OF-KILTER
           CALL OUT(CUSTO,VARDU,LIMIF,FLUXO,CAPAC,INDEX,NRAMOS,NUMEST,
   *                CAPSES,ESTAC,LIMCAP,NOS,IK2,CHAVE)
      VERIFICA SE O FLUXO EH ADMISSIVEL
           DO 52 N=1,NUMEST
              IF(FLUXO(ESTAC(N)).GT.0.AND.FLUXO(ESTAC(N)).LT.LIMCAP)
   *                .GO TO 80
2          CONTINUE
           ATUALIZACAO DOS APONTADORES
           APVSOL=APVSOL+1
           VETSOL(APVSOL)=APCSOL+1
           CADSOL(VETSOL(APVSOL))=CONJEF
           CADSOL(APCSOL+2)=I
           CADSOL(APCSOL+3)=J
           APCSOL=APCSOL+3
           DO 62 N=1,NRAMOS
              SOMA=SOMA+CUSTO(N)*FLUXO(N)
2          CONTINUE
           IF(SOMA.LT.CUSMIN)THEN DO
             CUSMIN=SOMA
             ESCMIN=I
             CACMIN=VETSOL(APVSOL)
             YI(CADSOL(CACMIN+1))=0
             YI(CADSOL(CACMIN+2))=0
             WRITE(6,3)(YI(II),II=1,NUMEST),CUSMIN
             YI(CADSOL(CACMIN+1))=1
             YI(CADSOL(CACMIN+2))=1
           END IF
           SOMA=0.
0          CAPAC(ESTAC(J))=CAPSES(J)
2       CONTINUE
        CAPAC(ESTAC(I))=CAPSES(I)
22   CONTINUE
     END IF
```

```
IF(NUMEST.GT.2)THEN DO
------------------------------------------------------------
PESQUISA COM TREIS ESTACOES DE TRANSFERENCIA FECHADAS
------------------------------------------------------------
LIMPES=LIMPES-1
J=0
L=0
K=0
CONJEF=CONJEF+1
PTVSOL=APVSOL
LIMP1=NUMEST-1
DO 333 I=1,LIMPES
 TESTA SE HOUVE ALGUMA SOLUCAO ADMISSIVEL
   K=0
   J=0
   IF(PTVSOL.GT.0 )THEN DO
   PESQUISA DA SOLUCAO ENCONTRADA
   CALL EXSOLA(I, J, K, L, CHSOL, CADSOL, VETSOL, APVSOL)
   IF(CHSOL.EQ.1) GO TO 333
   END IF
   CAPAC(ESTAC(I) )=0
   JJ=I+1
   DO 92 J=JJ, LIMP1
      IF(PTVSOL.GT.0)THEN DO
      CALL EXSOLA (I, J, K, L, CHSOL, CADSOL, VETSOL, APVSOL)
      IF(CHSOL.EQ.1)GO TO 92
      END IF
      CAPAC(ESTAC (J ) )=0
      LL=J+1
      DO 83 K=LL, NUMEST
         IF(PTVSOL.GT.0)THEN DO
         CALL EXSOLA(I, J, K, L, CHSOL, CADSOL, VETSOL, APVSOL)
         IF(CHSOL.EQ.1)GO TO 83
         END IF
         CAPAC(ESTAC(K))=0
         CALL OUT(CUSTO, VAROU, LIMIF, FLUXO, CAPAC, INDEX, NRAMOS,
*                 NUMEST, CAPSES, ESTAC, LIMCAP, NOS, IK2, CHAVE)
         DO 72 N=1, NUMEST
            IF(FLUXO(ESTAC(N)).GT.0.AND.FLUXO(ESTAC(N)).LT.LIMCAP)
*              GO TO 98
2        CONTINUE
         APVSOL=APVSOL+1
         VETSOL(APVSOL)=APCSOL+1
         CADSOL(VETSOL(APVSOL))=CONJEF
         CADSOL(APCSOL+2)=I
         CADSOL(APCSOL+3)=J
         CADSOL(APCSOL+4)=K
         APCSOL=APCSOL+4
         DO 42 N=1, NRAMOS
            SOMA=SOMA+CUSTO(N)*FLUXO(N)
2        CONTINUE
         IF(SOMA.LT.CUSMIN)THEN DO
           CUSMIN=SOMA
           ESCMIN=I
           CACMIN=VETSOL(APVSOL)
           YI(CADSOL(CACMIN+1))=0
           YI(CADSOL(CACMIN+2))=0
           YI(CADSOL(CACMIN+3))=0
           WRITE( 6, 3)(YI(II), II=1,NUMEST),CUSMIN
           YI(CADSOL(CACMIN+1))=1
```

```
                 YI(CADSOL(CACMIN+2))=1
                 YI(CADSOL(CACMIN+3))=1
               END IF
 98            CAPAC(ESTAC(K))=CAPSES(K)
               SOMA=0.
 83          CONTINUE
             CAPAC(ESTAC(J))=CAPSES(J)
 92       CONTINUE
          CAPAC(ESTAC(I))=CAPSES(I)
          CAPAC(ESTAC(J))=CAPSES(J)
 333  CONTINUE
      END IF
       IF(NUMEST.GT.3)THEN DO
C     -----------------------------------------------------------------
C     PESQUISA COM QUATRO S ESTACOES DE TRANSFERENCIA FECHADAS
C     -----------------------------------------------------------------
      LIMP2=LIMP1
      K=0
      J=0
      L=0
      LIMPES=LIMPES-1
      CONJEF=CONJEF+1
      LIMP1=LIMP1-1
      PTVSOL=APVSOL
      DO 444 I=1,LIMPES
      J=0
      L=0
      K=0
         IF(PTVSOL.GT.0)THEN DO
         CALL EXSOLA(I,J,K,L,CHSOL,CAPSOL,VETSOL,APVSOL)
         IF(CHSOL.EQ.1) GO TO 444
         END IF
         CAPAC(ESTAC(I))=0
         JJ=I+1
         DO 84 J=JJ,LIMP1
            IF(PTVSOL.GT.0)THEN DO
            CALL EXSOLA(I,J,K,L,CHSOL,CADSOL,VETSOL,APVSOL)
            IF(CHSOL.EQ.1)GO TO 84
            END IF
            CAPAC(ESTAC(J))=0
            LJ=J+1
            DO 74 K=LJ,LIMP2
               IF(PTVSOL.GT.0)THEN DO
                CALL EXSOLA(I,J,K,L,CHSOL,CADSOL,VETSOL,APVSOL)
                IF(CHSOL.EQ.1)GO TO 74
                END IF
                CAPAC(ESTAC(K))=0
                 LI=K+1
                 DO 94 L=LI,NUMEST
                   IF(PTVSOL.GT.0)THEN DO
                   CALL EXSOLA(I,J,K,L,CHSOL,CADSOL,VETSOL,APVSOL)
                   IF(CHSOL.EQ.1)GO TO 94
                   END IF
                   CAPAC(ESTAC(L))=0
                    CALL OUT(CUSTO,VARDU,LIMIF,FLUXO,CAPAC,INDEX,NRAMOS,
     *                   NUMEST,CAPSES,ESTAC,LIMCAP,NOS,IK2,CHAVE)
                   DO 41 N=1,NUMEST
                        IF(FLUXO(ESTAC(N)).GT.0.AND.FLUXO(ESTAC(N)).LT.
     *                    LIMCAP)GO TO 14
 41                CONTINUE
```

```
                APVSOL=APVSOL+1
                VETSOL(APVSOL)=APCSOL+1
                CADSOL(VETSOL(APVSOL))=CONJEF
                CADSOL(APCSOL+2)=I
                CADSOL(APCSOL+3)=J
                CADSOL(APCSOL+4)=K
                CADSOL(APCSOL+5)=L
                APCSOL=APCSOL+5
                DO 43 N=1,NRAMOS
3                  SOMA=SOMA+CUSTO(N)*FLUXO(N)
                IF(SOMA.LT.CUSMIN)THEN DO
                   CUSMIN=SOMA
                   ESCMIN=I
                   CACMIN=VETSOL(APVSOL)
                   YI(CADSOL(CACMIN+1))=0
                   YI(CADSOL(CACMIN+2))=0
                   YI(CADSOL(CACMIN+3))=0
                   YI(CADSOL(CACMIN+4))=0
                   WRITE(6,3)(YI(II),II=1,NUMEST),CUSMIN
                   YI(CADSOL(CACMIN+1))=1
                   YI(CADSOL(CACMIN+2))=1
                   YI(CADSOL(CACMIN+3))=1
                   YI(CADSOL(CACMIN+4))=1
                 END IF
                 SOMA=0.
4            CAPAC(ESTAC(L))=CAPSES(L)
4            CONTINUE
          CAPAC(ESTAC(K))=CAPSES(K)
4         CONTINUE
       CAPAC(ESTAC(J))=CAPSES(J)
       CAPAC(ESTAC(I))=CAPSES(I)
4      CONTINUE
34  CONTINUE
    END IF
    WRITE(6,555)
55  FORMAT(1X,///37X,'SOLUCAO OTIMA',//)
    IF(CACMIN.NE.0)THEN DO
    M=CADSOL(CACMIN)
    DO 811 I=1,M
       YI(CADSOL(CACMIN+I))=0
11  CONTINUE
    WRITE(6,3)(YI(II),II=1,NUMEST),CUSMIN
    ELSE DO
    WRITE(6,414)
4   FORMAT('1','NAO EH VIAVEL INSTALAR ESTACOES DE TRANSFERENCIA DE
   *          LIXO'/)
    END IF
    K=1                                                                   IMPF0020
    DO 334 J=1,8
       WRITE(6,650)                                                       IMPF0040
       N=IX(J)                                                            IMPF0050
    WRITE(6,445)(I,INDEX(I,1),INDEX(I,2),LIMIF(I),FLUXO(I),CUSTO(I),
   *     CAPAC(I),I=K,N)
       K=N+1                                                              IMPF0070
34  CONTINUE
45  FORMAT(2(3(3X,I3),2(2X,I9),2X,F9.0,2),I9)/)
50  FORMAT(3X,120('-'),//2(4X,'NUME',2X,'NOIN',3X,'NOFI',6X,'LIMI',      IMPF0190
   *5X,'FLUXO',6X,'CUSTO',6X,'LIMS'),/3X,120('-')/)                      IMPF0200
    STOP
    END
```

ANEXO II

É apresentado aqui o resultado obtido na aplicação deste trabalho na cidade do Recife.

CASO EM ESTUDO — CINCO LOCAIS POSSIVEIS PARA IMPLANTACAO DE ESTACAO DE TRANSFERENCIA DE LIXO (ETL)

LEGENDA PARA SOLUCAO 1-SE O LOCAL FOI ESCOLHIDO

0-CASO CONTRARIO

SOLUCOES ADMISSIVEIS ENCONTRADAS

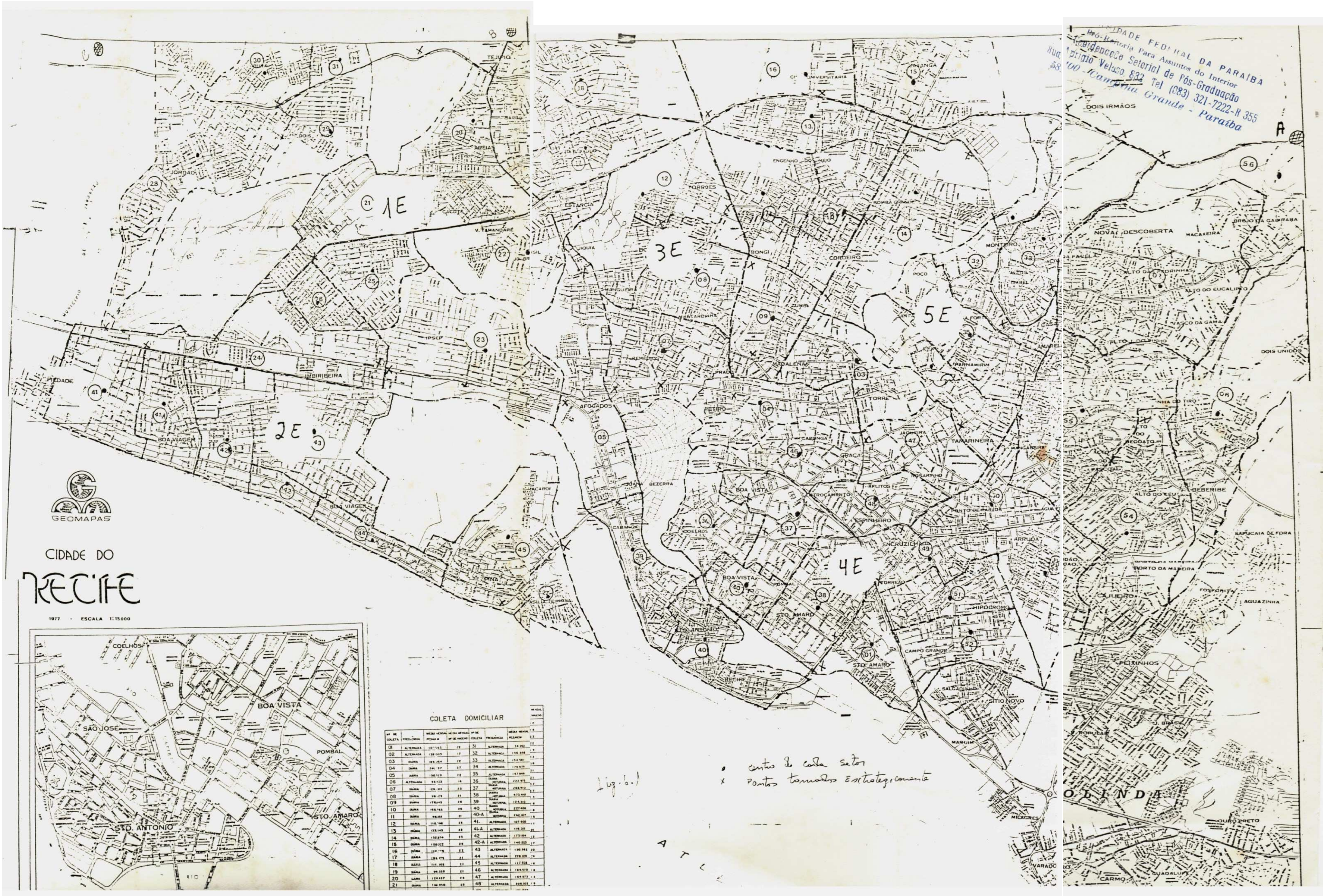
* LOCAIS POSSIVEIS DAS ETL	1	2	3	4	5
SOLUCAO ADMISSIVEL	0	0	0	0	1
CUSTO DESTA SOLUCAO	472192.10				

* LOCAIS POSSIVEIS DAS ETL	1	2	3	4	5
SOLUCAO ADMISSIVEL	0	0	1	0	0
CUSTO DESTA SOLUCAO	460945.90				

* LOCAIS POSSIVEIS DAS ETL	1	2	3	4	5
SOLUCAO ADMISSIVEL	1	0	0	0	0
CUSTO DESTA SOLUCAO	454489.10				

SOLUCAO OTIMA

* LOCAIS POSSIVEIS DAS ETL	1	2	3	4	5
SOLUCAO ADMISSIVEL	1	0	0	0	0
CUSTO DESTA SOLUCAO	454489.10				



COLETA DOMICILIAR

Nº DE COLETA	FREQUÊNCIA	MÉDIA MENSAL PESAGEM	MÉDIA MENSAL Nº DE VIAGENS	Nº DE COLETA	FREQUÊNCIA	MÉDIA MENSAL PESAGEM	MÉDIA MENSAL Nº DE VIAGENS
01	ALTERNADA	107 563	22	31	ALTERNADA	34 202	[illegible]
02	ALTERNADA	138 005	17	32	ALTERNADA	146 898	[illegible]
03	DIÁRIA	185 364	22	33	ALTERNADA	154 581	[illegible]
04	DIÁRIA	241 97	27	34	ALTERNADA	174 575	[illegible]
05	DIÁRIA	190 529	22	35	ALTERNADA	157 849	[illegible]
06	ALTERNADA	93 433	16	36	DIÁRIA NOTURNA	222 975	[illegible]
07	DIÁRIA	129 184	23	37	DIÁRIA NOTURNA	266 910	[illegible]
08	DIÁRIA	98 123	21	38	DIÁRIA NOTURNA	473 410	[illegible]
09	DIÁRIA	178 645	26	39	DIÁRIA NOTURNA	124 310	[illegible]
10	DIÁRIA	165 163	25	40	DIÁRIA NOTURNA	227 406	[illegible]
11	DIÁRIA	98 351	21	40-A	DIÁRIA NOTURNA	262 417	[illegible]
12	DIÁRIA	110 796	23	41	ALTERNADA	107 500	[illegible]
13	DIÁRIA	133 145	22	41-A	ALTERNADA	119 311	[illegible]
14	DIÁRIA	132 974	23	42	ALTERNADA	173 104	[illegible]
15	DIÁRIA	120 322	24	42-A	ALTERNADA	140 225	[illegible]
16	DIÁRIA	114 775	22	43	ALTERNADA	130 982	[illegible]
17	DIÁRIA	155 775	21	44	ALTERNADA	228 129	[illegible]
18	DIÁRIA	111 208	21	45	ALTERNADA	117 558	[illegible]
19	DIÁRIA	94 359	21	46	ALTERNADA	164 570	[illegible]
20	DIÁRIA	124 437	24	47	ALTERNADA	164 573	[illegible]
21	DIÁRIA	144 950	23	48	ALTERNADA	208 102	[illegible]

Fig. 6.1

• centro de cada setor
x pontos tomados estrategicamente

BIBLIOGRAFIA

[01] - COOPER, L.. - Location - Allocation Problems. Operations Research. Vol. 11 (1963), p. 331.

[02] - GRAFINKEL, R.S. and NENHOUSER, G.L.. - Integer Programming. John Willey & Sons (1972).

[03] - HAND A. TAHA. - Integer Programming Theory Aplication, and Computations. Academic Press (1975).

[04] - LAWLER, E.L.. - Combinatorial Optimization: Networks and Matroids. Holt, Rinehart and Winston (1965).

[05] - MARKS, D.H., and JON C. LIEBMAN.. - Location Models: A Solid Waste Colection Example.

[06]- REVELLE, C.S., MARKS, D.H. and LIEBMAN, J.C.. - Private and Public Sector Location Models. Man. Sc. Vol. 16, Nº 11, July 1970, p. 692 - 707.

[07]- WAGNER, H. M.. - Principles of Operations Research. Englewood cliffs, N. J. Prentice - Hall, Inc. 1969.