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

ROTAS PARA DISTRIBUIÇÃO DE BENS E SERVIÇOS - PROPOSTA DE UM SISTEMA DE INFORMAÇÃO

HELÉDIA CALIL BUENO DA COSTA

CAMPINA GRANDE - PARAÍBA
AGOSTO - 1982

C837r Costa, Heledia Calil Bueno da
Rotas para distribuicao de bens e servicos : proposta de um sistema de informacao / Heledia Calil Bueno da Costa. - Campina Grande, 1982.
132 f. : il.

Dissertacao (Mestrado em Ciencias) - Universidade Federal da Paraiba, Centro de Ciencias e Tecnologia.

1. Sistema de Informacao - 2. Processamento de Dados - 3. Dissertacao I. Brucker, Peter Joachim Siegfried, Dr. II. Universidade Federal da Paraiba - Campina Grande (PB) III. Título

CDU 004.421(043)

ROTAS PARA DISTRIBUIÇÃO DE BENS E SERVIÇOS - PROPOSTA
DE UM SISTEMA DE INFORMAÇÃO

HELÉDIA CALIL BUENO DA COSTA

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:

Peter J. S. Brucker

PETER JOACHIM SIEGFRIED BRUCKER - Ph.D.
- Presidente -

ANTONIO ILDEFONSO DE ALBUQUERQUE MELO - Ph.D.

MARIO TOYOTARO HATTORI - M.Sc.

CAMPINA GRANDE
ESTADO DA PARAÍBA - BRASIL
AGOSTO - 1982

Para
Rafael e Miguel

RESUMO

Neste trabalho é proposto um sistema de informação para a determinação de rotas para a distribuição de bens e serviços.

A fim de possibilitar uma implementação eficiente dos algoritmos envolvidos na solução do problema, foi realizado um estudo sobre os objetos que deveriam ser representados no computador, e as estruturas de dados que poderiam ser utilizadas nessa representação.

Foram levantados todos os problemas e restrições relacionadas com a aplicação real do sistema, visando sua utilização prática.

Efetuou-se algumas aplicações práticas do sistema utilizando dados reais referentes às cidades de Fortaleza e Aracaju, e foi realizada uma análise dos resultados obtidos.

ABSTRACT

The purpose of this work is an information system to determine routs for the goods and public services distribution.

In order to allow an efficient implementation of the algorithms involved in the problem resolution, a study on the objects that should be represented in the computer and the data structure that could be used in this representation was made.

All the problems and restrictions concerned with the actual aplication of the system, aiming to its practical usage were set up.

Some practical aplications of the system were carried out by using real data refering to the cities of Fortaleza and Aracaju, and an analysis of the obtained results was made.

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

Meus sinceros agradecimentos:

À Universidade Federal de São Carlos, que me propiciou a oportunidade de realizar este trabalho;

Ao Professor Dr. Peter Joachin Siegfried Brucker, pela relevante orientação;

Aos Professores Mario Toyotaro Hattori e Antonio Ildefonso de Albuquerque Melo, pelas valiosas contribuições apresentadas;

A toda equipe do Núcleo de Processamento de Dados, pelo apoio e boa vontade durante a implementação dos algoritmos;

E a todos que contribuiram direta ou indiretamente para a realização deste trabalho.

ÍNDICE

CAPÍTULO I

1. INTRODUÇÃO

A distribuição de bens e serviços em muitas cidades brasileiras não é efetuada de forma eficiente e econômica. Quando a distribuição exige a utilização de um veículo, como é o caso de coleta de lixo, a ineficiência do sistema pode resultar em altos custos, devido ao desperdício de combustível e a utilização inadequada do veículo e da mão-de-obra disponível para o serviço.

O sistema usado com mais frequência é a divisão da cidade em zonas que possam ser servidas por um único veículo e a rota para percorrer essas zonas é determinada empiricamente, resultando, geralmente, em custos elevados.

O objetivo desse trabalho é propor um sistema de informação para a determinação de rotas, visando solucionar eficientemente o problema da distribuição de bens e serviços em cidades de médio e grande portes. Esse sistema é aplicável a zonas previamente definidas e foi dividido basicamente em três partes:

1. Entrada das informações sobre uma zona pré-determinada e construção do grafo inicial para o qual será fornecida uma rota;
2. Transformação do grafo inicial num grafo de Euler e determinação de uma rota para esse grafo;
3. Impressão da rota.

Podem-se considerar basicamente duas maneiras distintas de realizar a distribuição de bens e serviços: com ou sem a utilização de veículos.

Nesse trabalho soluciona-se o problema de determinação de rotas tanto para a distribuição motorizada como para a não motorizada, embora se enfoque principalmente o primeiro caso. Esse enfoque se deve a uma série de fatores práticos e que restringem as opções de tráfego tais como mão-única, mão-dupla, contra-mão, ruas com tráfego intenso, etc... Esses problemas não existem para a distribuição não motorizada.

Os algoritmos heurísticos para resolver o problema de determinação de rotas baseiam-se no problema do carteiro chinês e foram detalhadamente descritos nos trabalhos de Santos

[2] e Costa [1].

Santos implementou parte desses algoritmos (transformação do grafo inicial num grafo de Euler) de forma insatisfatória, utilizando estruturas de dados inadequadas para a representação dos dados no computador e execução de vários passos manualmente, que tornaram impossível sua aplicação prática.

A fim de possibilitar uma implementação eficiente dos algoritmos foi realizado um estudo sobre os objetos que deveriam ser representados, as operações a que eram submetidas pelos procedimentos e todas as estruturas de dados que poderiam ser utilizadas para sua representação no computador. Essas estruturas foram analisadas sob os seguintes aspectos:

- espaço de memória necessário;
- tempo de processamento
- facilidade de manipulação dos dados.

A estrutura escolhida foi a que forneceu a melhor combinação desses três fatores.

Foram estudados também outros pontos críticos na implementação efetuada por Santos e realizadas algumas modificações visando evitá-los.

Com a implementação do algoritmo para a determinação de uma rota num grafo de Euler, descrito no trabalho de Costa (1), tornou possível utilizar o sistema em situações reais.

Durante o desenvolvimento do sistema procurou-se considerar todos os problemas e restrições relacionados com a sua aplicação prática. Esses aspectos são descritos no capítulo II desse trabalho.

O capítulo III é dedicado aos objetos envolvidos no problema e às estruturas de dados para sua representação no computador.

Os algoritmos para a determinação da rota são descritos no capítulo IV.

No capítulo V é descrita a implementação dos algoritmos, e são efetuadas algumas aplicações com dados reais referentes às cidades de Fortaleza e Aracaju, e realizada uma análise dos resultados obtidos.

CAPÍTULO II

2. UMA SOLUÇÃO DO PROBLEMA

2.1. Introdução

Nesse capítulo será descrito um sistema de informação para a determinação de rotas, com aplicação à distribuição de bens e serviços.

O problema de coleta de lixo será utilizado como exemplo. O procedimento usado pode ser estendido a outros problemas semelhantes como entrega automática de botijões de gás ou entrega de água mineral engarrafada. Considera-se como "serviço-útil", o fato do coletor de lixo estar se movendo pela rua enquanto recolhe o lixo, e denomina-se como "serviço-inútil", o fato do coletor estar se movendo pela rua sem coletar lixo.

O objetivo principal do sistema consiste em obter-se uma rota que passe por todas as ruas de uma zona pré-determinada da cidade pelo menos uma vez, minimizando a distância percorrida pelo coletor em "serviço-inútil".

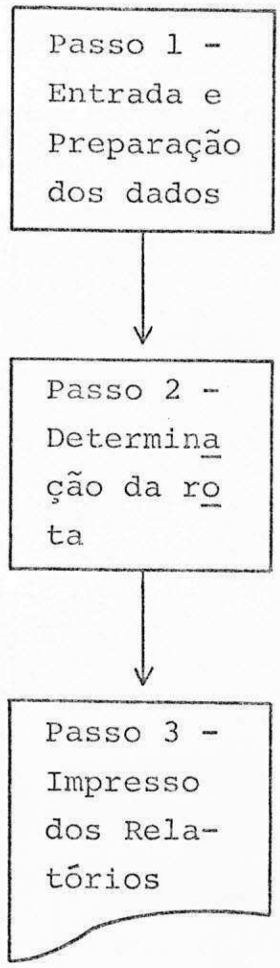
Nesse sistema os custos podem ser considerados de duas formas diferentes:

a) Os custos são lineares com relação à distância. Nesse caso é razoável admitir que minimizando a distância minimiza-se os custos.
b) Além da distância, os custos são determinados por outros fatores, tais como condições de trânsito e estado da rua. Nesse caso os custos serão calculados levando-se em conta esses fatores.

Para a aplicação do sistema, a cidade deverá ser previamente dividida em zonas. Para que a rota fornecida pelo sistema seja a melhor possível, é necessário que o zoneamento da cidade também tenha sido otimizado. Nesse trabalho assume-se que as zonas fornecidas para a determinação de uma rota, tenhem sido divididas da melhor maneira possível.

2.2. Fluxo Geral do Sistema

O sistema para fornecimento de rotas pode ser dividido em três passos. Um diagrama desse sistema é visto a seguir.



2.3. Entradas e Saídas do Sistema

Entende-se por USUÁRIO do sistema a empresa, prefeitura ou qualquer outra entidade que precise efetuar a distribuição de algum bem ou serviço numa determinada zona, e solicite ao sistema o fornecimento da rota para percorrê-la. INFORMANTE é a pessoa designada pelo USUÁRIO para responsabilizar-se pela obtenção e fornecimento das informações necessárias para a determinação da rota. O RESPONSÁVEL pelo sistema é a pessoa responsável pelo recebimento das informações enviadas pelo INFORMANTE, operação do sistema, e também pela entrega ao INFORMANTE das saídas fornecidas pelo sistema.

2.3.1. Entradas

A coleta das informações sobre as características de cada trecho das ruas de uma zona é o procedimento que demanda maior quantidade de tempo. Deve ser conduzida uma pesquisa sobre cada zona para fornecer essas informações. Nesse trabalho considera-se que essas informações são coletadas e fornecidas pelo USUÁRIO.

Nesse sistema são utilizados diversos formulários para a entrada dos dados, que aqui serão descritos suscintamente e seus modelos serão apresentados no Apêndice I.

Define-se atualização de um arquivo como a inclusão ou exclusão de um registro, ou alteração das informações contidas num registro.

2.3.1.1. Formulário de Cadastro de Usuário

Objetivo: fornecer informações sobre um USUÁRIO. É utilizado para a atualização do arquivo CADASTRO DE USUÁRIO.

Preenchido por: USUÁRIO e RESPONSÁVEL pelo sistema.

2.3.1.2. Formulário para Cadastro de Zona

Objetivo: fornecer informações sobre uma zona para a qual será fornecida uma rota. É utilizado para a atualização do arquivo de CADASTRO DE ZONAS de um determinado usuário.

Preenchido por: INFORMANTE.

2.3.1.3. Formulário para Cadastro de Ruas

Objetivo: informar quais são as ruas que compõe uma determinada zona. Utilizado para a criação e atualização do arquivo CADASTRO DE RUAS.

Preenchido por: INFORMANTE.

2.3.1.4. Formulário para Cadastro de Esquinas

Objetivo: informar quais são as esquinas (nós) que pertencem a cada uma das ruas de uma zona. Utilizado para criar ou atualizar o arquivo CADASTRO DE ESQUINAS.

Preenchido por: INFORMANTE.

2.3.1.5. Formulário para Cadastro de Trechos

Objetivo: fornecer as informações necessárias sobre os trechos de todas as ruas de uma zona. Utilizado para criar ou atualizar o arquivo CADASTRO DE TRECHOS.

Preenchido por: INFORMANTE.

2.3.1.6. Formulário para Cadastro de Restrições de Trânsito

Objetivo: informar quais os trechos de uma determinada zona possuem restrições de trânsito. Utilizado para a criação ou atualização do arquivo CADASTRO DE RESTRIÇÃO.

Preenchido por: INFORMANTE.

OBS.: No Apêndice II foi incluído um manual de utilização do sistema.

2.3.2. Relatórios de Saída do Sistema

Os relatórios de saída do sistema são aqui descritos suscintamente, e seus modelos são apresentados no Apêndice I.

2.3.2.1. Relação dos Usuários Cadastrados no Sistema

a) Finalidade:
   - controle dos usuários cadastrados no sistema;
   - fornecer todas as informações necessárias sobre cada um dos usuários.
b) Características Gerais:
   - emissão por solicitação do RESPONSÁVEL pelo sistema;
   - destinatário: RESPONSÁVEL pelo sistema;
   - ordenação: ordenado alfabeticamente por nome de USUÁRIO.

2.3.2.2. Relação das Zonas Cadastradas

a) Finalidade:
   - controle do cadastro das zonas;
   - informar quantas e quais são as zonas cadastradas por um determinado USUÁRIO.
b) Características Gerais:
   - emissão·

- por solicitação do INFORMANTE;
- por solicitação do RESPONSÁVEL pelo sistema;
- destinatário:
  - INFORMANTE;
  - RESPONSÁVEL pelo sistema;
- ordenação: ordem crescente de código da zona.

2.3.2.3. Relação das Ruas Cadastradas numa zona

a) Finalidade: informar quantas e quais são as ruas cadastradas numa determinada zona.

b) Características Gerais:
- emissão:
  - por solicitação do INFORMANTE;
  - por solicitação do RESPONSÁVEL pelo sistema;
  - quando uma nova zona for cadastrada no sistema;
  - quando houver atualização das ruas cadastradas numa zona;
- destinatário:
  - INFORMANTE;
  - RESPONSÁVEL pelo sistema;
- ordenação: ordenado alfabeticamente por nome de rua.

2.3.2.4. Rota para uma Zona

a) Finalidade: indicar a rota que deve ser seguida para percorrer todas as ruas de uma determinada zona com o mínimo custo possível.

b) Características Gerais:
- emissão:
  - quando for cadastrada uma nova zona;
  - quando houver alguma atualização das informações sobre uma zona;
  - quando for solicitado pelo INFORMANTE;
- Destinatário:
  - INFORMANTE.

2.4.Arquivos do Sistema

Os registros dos arquivos são aqui descritos suscintamente, e seus modelos serão apresentados no Apêndice I.

Os arquivos contendo as informações necessárias para a geração das saídas do sistema são os seguintes:

- cadastro de Usuários;
- cadastro de Zonas;
- cadastro de Ruas;
- cadastro de Esquinas;
- cadastro de Trechos;
- cadastro de Restrições;
- grafo;
- MATRIZ CHECK;
- rota.

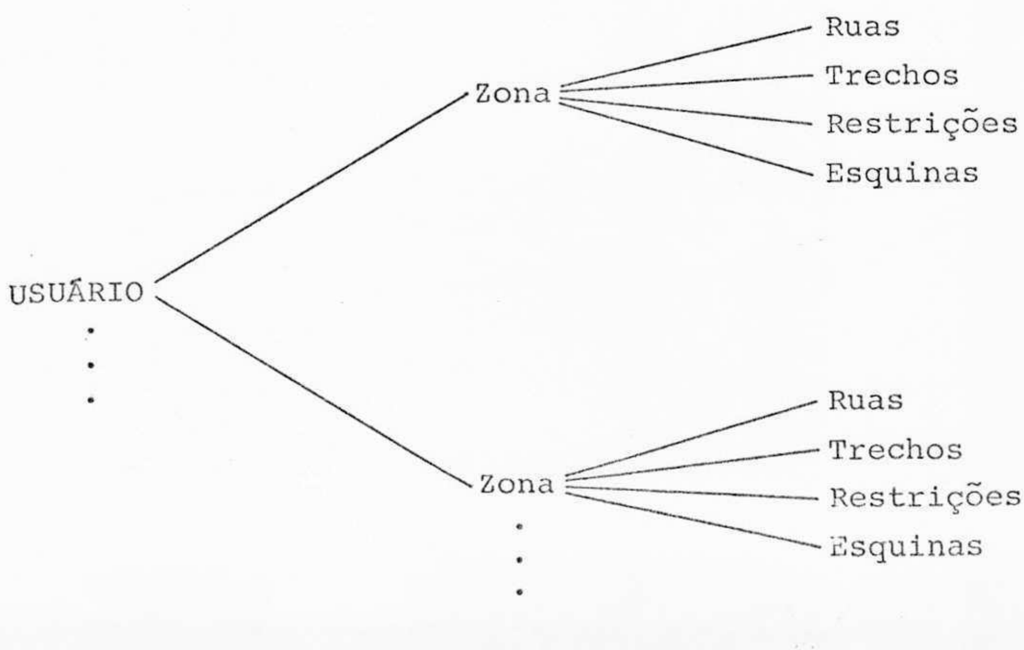
Para os cadastros, o procedimento para a criação e atualização é praticamente idêntico. Os dados entram em cartões, são criticados e caso estejam corretos, os arquivos são atualizados e gravados em fita magnética.

Optou-se pelo armazenamento em fita magnética pelas seguintes razões:

- custo baixo;
- baixa frequência de utilização dos arquivos.

A cada processamento do sistema, os arquivos serão copiados temporariamente em disco, utilizados, e recopiados em fita, liberando novamente a área do disco.

2.4.1. Relacionamento entre as Informações



2.4.2. Arquivo Cadastro de Usuários

Objetivo: conter todas as informações necessárias sobre cada um dos USUÁRIOS.
Armazenamento: fita magnética.
Ordenação: ordem crescente de código de USUÁRIO.

2.4.3. Arquivo Cadastro de Zonas

Objetivo: conter informações sobre as zonas cadastradas por um determinado USUÁRIO.
Armazenamento: fita magnética.
Ordenação: ordem crescente de código de zona.

2.4.4. Arquivo Cadastro de Esquinas

Objetivo: conter todas as esquinas pertencentes a cada uma das ruas de uma zona. Essas informações serão utilizadas na crítica dos dados e na impressão da ROTA.
Armazenamento: fita magnética.
Ordenação: ordem crescente de código de rua.

2.4.5. Arquivo Cadastro de Ruas

Objetivo: conter todas as ruas de uma zona e seus respectivos códigos.
Armazenamento: fita magnética.
Ordenação: ordem crescente de código de rua.

2.4.6. Arquivo Cadastro de Trechos

Objetivo: conter informações sobre os trechos das ruas de uma zona. É a partir desse arquivo que será construído o grafo inicial.
Armazenamento: fita magnética.

2.4.7. Arquivo Cadastro de Restrições

Objetivo: conter informações sobre os trechos com restrição de trânsito existente numa zona.

Ordenação: ordem crescente de nó inicial do trecho com restrição.

Armazenamento: fita magnética.

2.4.8. Arquivo Grafo

Objetivo: conter o grafo inicial que representa a zona para a qual será fornecida uma ROTA.

Armazenamento: disco magnético (temporário).

Ordenação: na ordem crescente de:

1) nó inicial no trecho;
2) nó final do trecho.

2.4.9. Arquivo MATRIZ CHECK

Objetivo: conter a matriz que será utilizada na verificação de erros no GRAFO e na impressão da ROTA.

Armazenamento: disco magnético (temporário).

O formato e utilização dessa matriz serão descritos no Capítulo III.

2.4.10. Arquivo Rota

Objetivo: conter o conjunto de trechos na ordem em que compõem a rota para percorrer a zona.

Armazenamento: disco magnético (temporário).

2.5. Fluxograma do Sistema

a) Entrada de dados

a.1) Atualiza os Cadastros

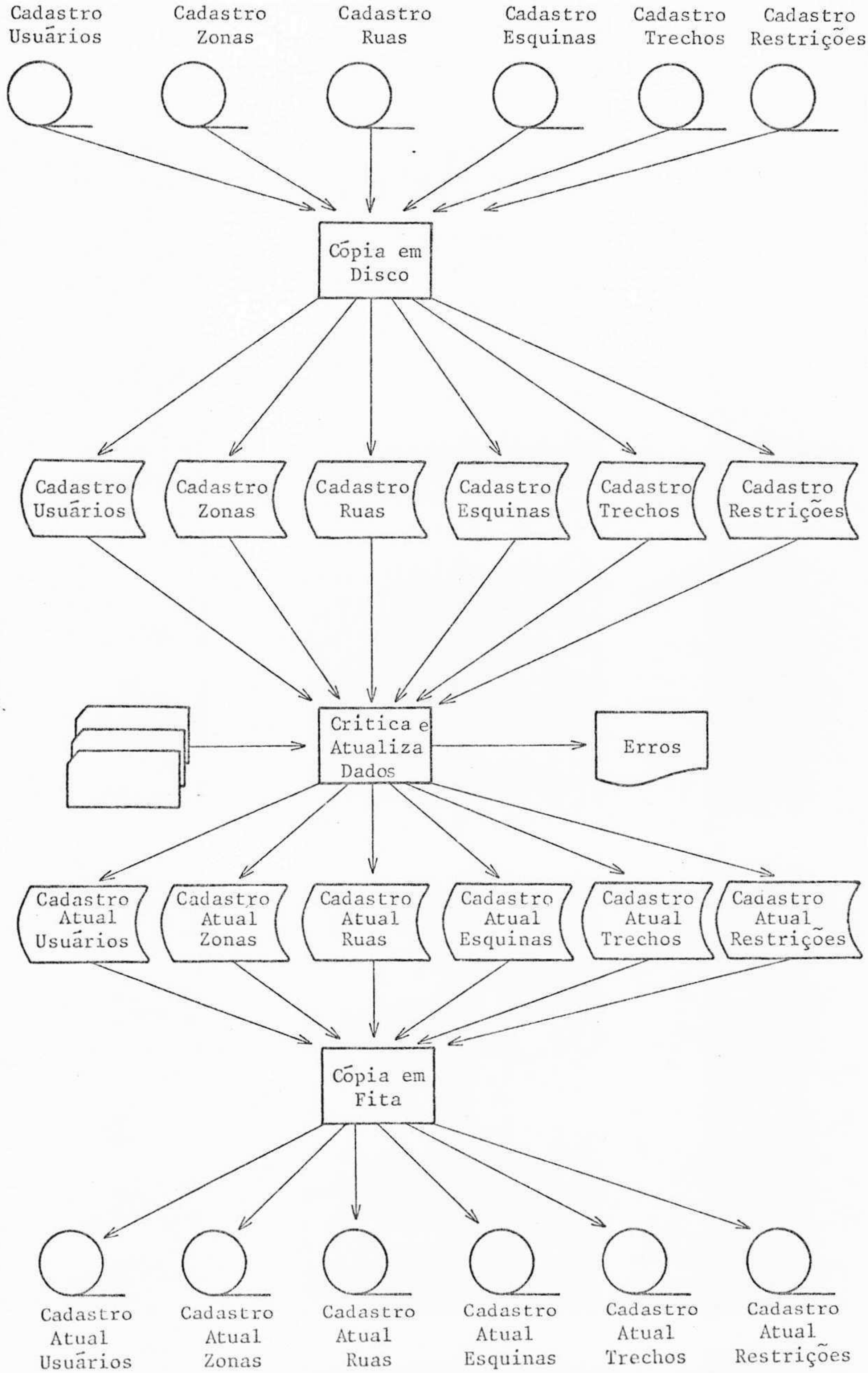


Figura 2.1.

a.2) Prepara o Grafo Inicial.
Constrói e Verifica os possíveis erros do grafo inicial.

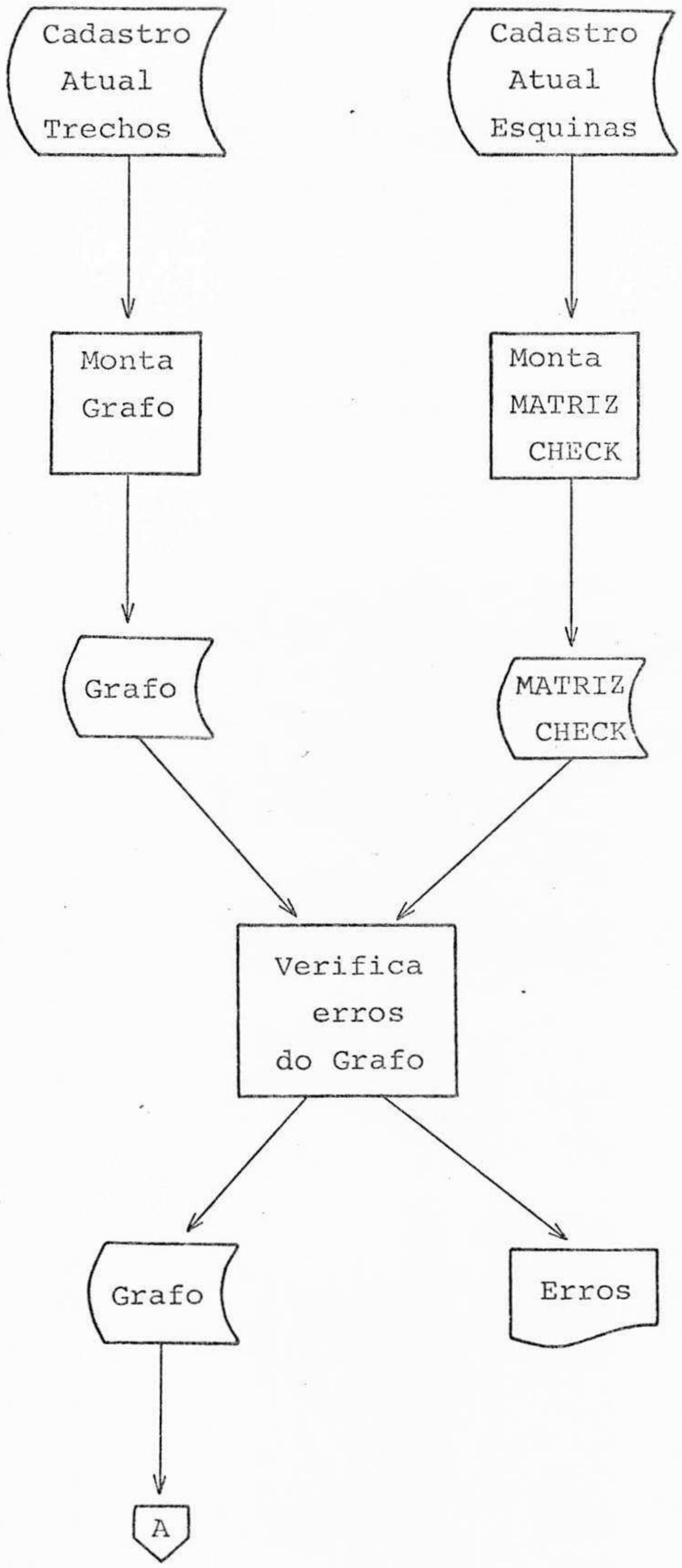
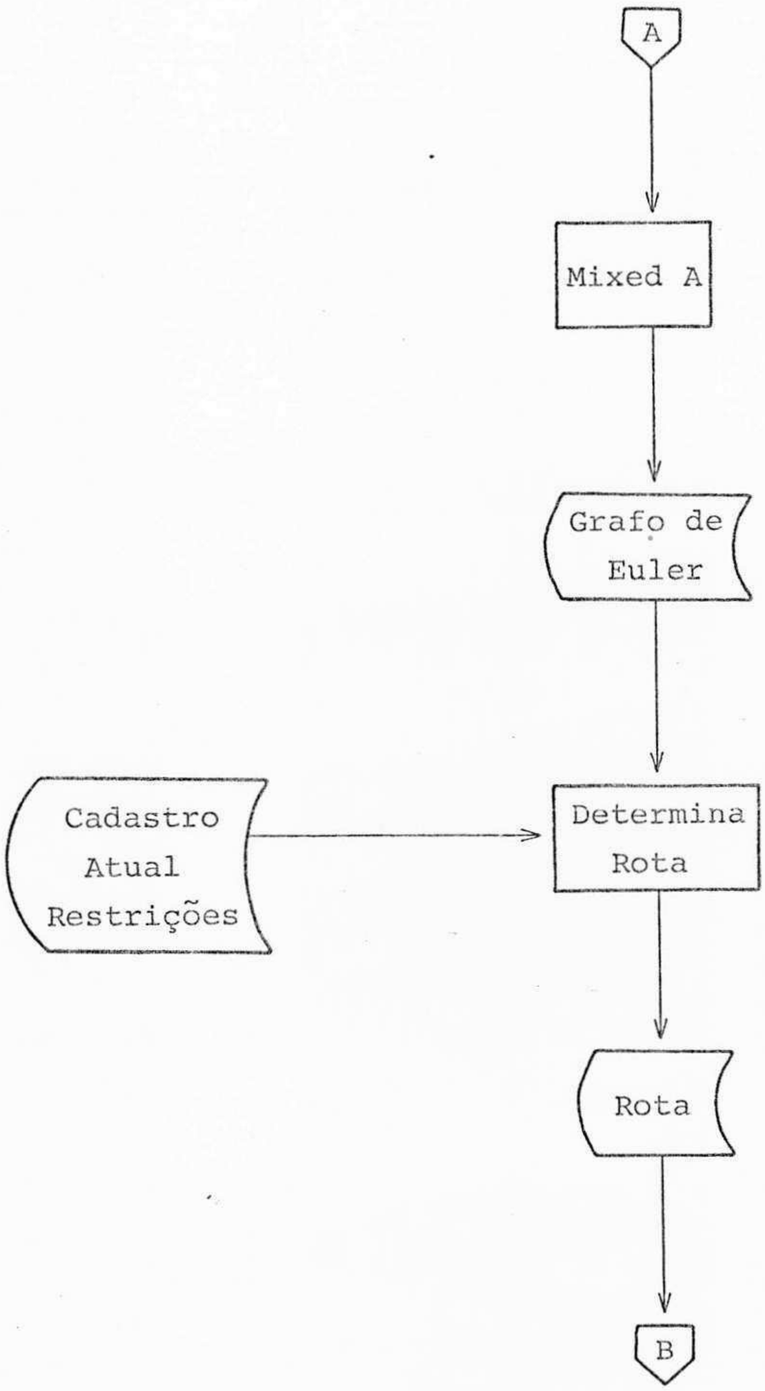
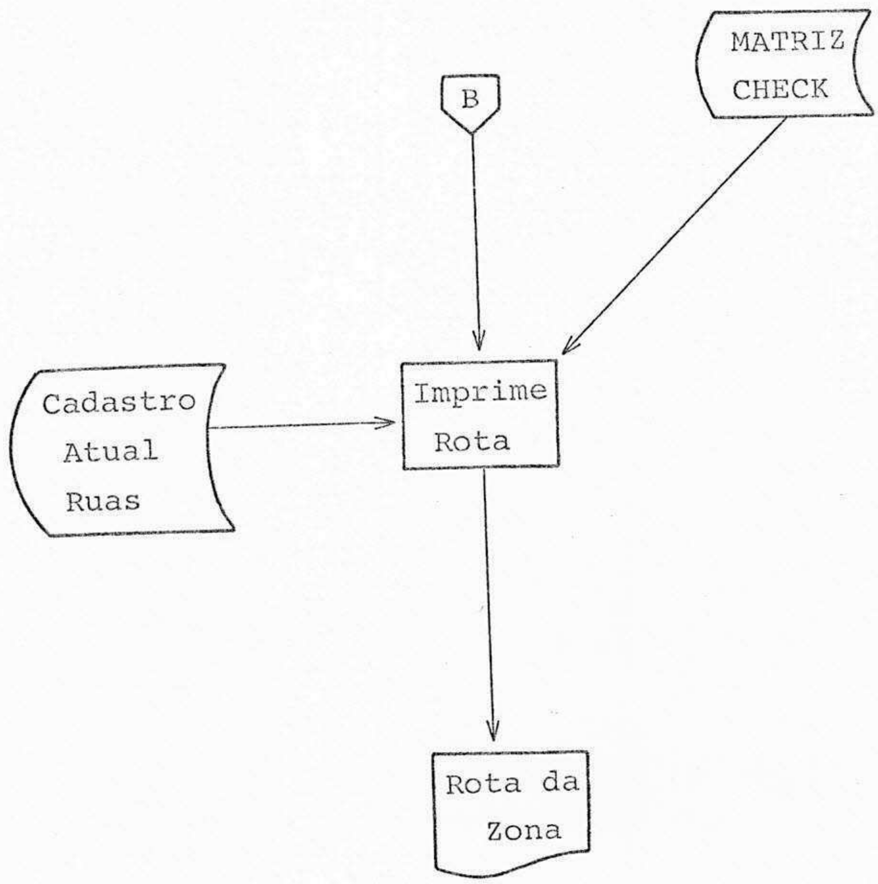


Figura 2.2.

b) Determina a Rota



c) Imprime a Rota



CAPÍTULO III

3. ESTRUTURAS DE DADOS PARA A REPRESENTAÇÃO DOS OBJETOS NO COMPUTADOR

A representação e identificação dos dados no computador é um dos assuntos mais importantes desse trabalho.

A escolha da estrutura de dados mais adequados para a representação dos objetos envolvidos no problema, é de fundamental importância para a eficiência da implementação de qualquer algoritmo. A opção por ser uma estrutura inadequada pode ocasionar graves problemas na manipulação dos dados, na execução de programas, e na utilização do espaço de memória.

Os métodos e técnicas que serão descritos nesse capítulo, visam solucionar o problema específico de implementação dos algoritmos para a determinação de rotas, para a distribuição de bens e serviços, mas podem ser aplicadas a uma gama variada de problemas especialmente, os que utilizam grafos na sua modelagem.

Para solucionar o problema de determinação de rotas são utilizados os seguintes objetos de representação:

3.1. Grafos

Os grafos envolvidos na solução do problema têm associados a cada arco ou ramo as seguintes informações:

- Nó inicial;
- Nó final;
- Custo;
- Sentido (do arco);
- Capacidade;
- Fluxo.

Além disso, a cada nó do grafo são associadas as seguintes informações:

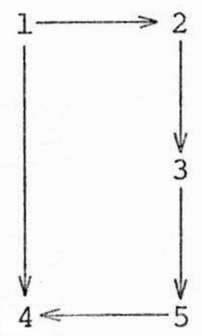
- Grau;
- Déficit.

A seguir são apresentados todos os tipos de grafos utilizados nesse trabalho.

3.1.1. Grafos Orientados (Direcionados)

Definição: Um grafo orientado $G = (N, A)$ consiste de um conjunto finito e não vazio de nós N e um conjunto de arcos A. Os arcos são pares ordenados (V, W) onde V é chamado origem ou nó inicial, e W é chamado de extremidade ou nó final. Diz-se também que o arco (V, W) é de V para W, e que o nó W é adjacente com o nó V.

Ex.:



$G = (N, A)$

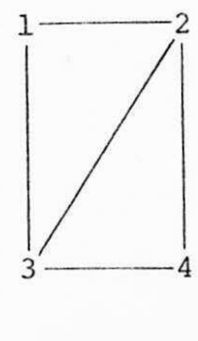
$N = (1, 2, 3, 4, 5)$

$A = \{(1,2),(1,4),(2,3),(3,5),(5,4)\}$

3.1.2. Grafos não Orientados (Não Direcionados)

Definição: Um grafo não orientado $G = (N, E)$ consiste de um conjunto finito e não vazio de nós N e de um conjunto de ramos E. Os ramos são pares ordenados de nós distintos, denotados por {V,W}. Assume-se que $\{V,W\} = (V,W)^*, (W,V)^*$, onde * indica que esses dois arcos representam um ramo. A ocorrência de dois arcos paralelos e com sentidos opostos será indicada por (V,W), (W,V), sem o *.

Ex.:



$G = (N, E)$

$N = (1, 2, 3, 4)$

$E = \{\{1,2\},\{1,3\},\{2,3\}\{2,4\},\{3,4\}\}$

3.1.3. Grafos Mistos

Um grafo mixto $G = (N, E, A)$, consiste de um conjunto finito e não vazio de nós N, de um conjunto de ramos E, e um conjunto de arcos A, onde $A \cap E = \phi$.

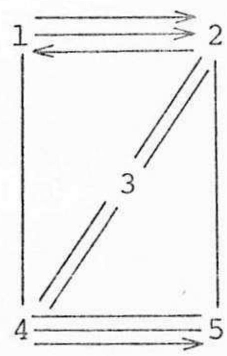
Ex.:

G = (N, E, A)
E = {{1,2},{2,4},{3,4}}
A = {(1,3)}

3.1.4. Grafos Mistos com Arcos e Ramos Paralelos

Um grafo misto com arcos e ramos paralelos G=(N,E,A), consiste de um conjunto de nós N, um conjunto de ramos E e um conjunto de arcos A, tal que o conjunto E pode conter dois ou mais ramos iguais (E={{V,W},{V,W},...}), e o conjunto A pode conter dois ou mais arcos repetidos com o mesmo sentido, ou com sentido contrário (A={(V,W),(V,W),(W,V),...}).

Ex.:



E = {{1,4},{2,5},{2,3},{2,3},
{3,4},{3,4},{4,5},{4,5}}
A = {(1,2),(1,2),(2,1),(4,5)}

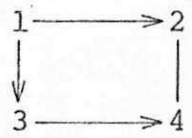
3.2. Operações Realizadas com os Grafos

Durante o processamento são realizadas diversas operações com os grafos, transformando-os sucessivamente até a obtenção da solução desejada. Essas operações são as seguintes:

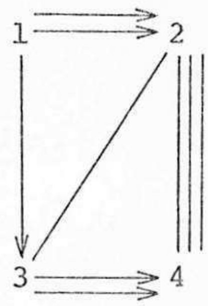
3.2.1. Adição

Essa operação consiste na inclusão de um novo ramo {V,W} ou arco (V,W), no grafo. Essa operação acrescenta um novo elemento ao conjunto de ramos E ou ao conjunto de arcos A. Ocasionalmente pode ocorrer o acréscimo de um novo nó ao conjunto N.

Ex.:



G = (N,E,A)
N = {1, 2, 3, 4}
A = {(1,2),(1,3),(3,4)}
E = {{2,4}}

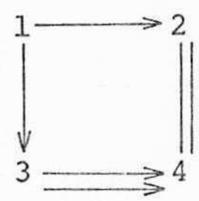


G = (N,E,A)
N = {1, 2, 3, 4}
A = {(1,2),(1,2),(1,3),
(3,4),(3,4)}
E = {{2,3},{2,4},{2,4},{2,4}}

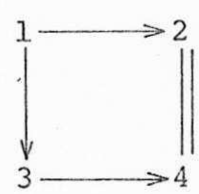
3.2.2. Remoção

Essa operação consiste em retirar um arco (V,W) do conjunto de arcos A ou um ramo {V,W} do conjunto de ramos E. Quando ocorrer esta operação, um desses conjuntos será diminuído de um elemento. Essa operação é a inversa de adição.

Ex.:



G = (N,E,A)
N = {1, 2, 3, 4}
A = {(1,2),(1,3),(3,4),(3,4)}
E = {{2,4},{2,4},{2,4}}

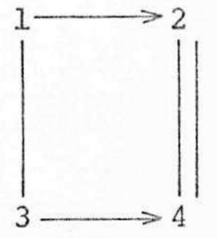


G = (N,E,A)
N = {1, 2, 3, 4}
A = {(1,2),(1,3),(3,4)}
E = {{2,4},{2,4}}

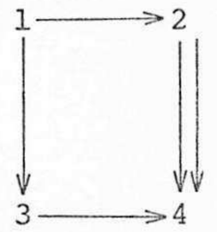
3.2.3. Transformação

Essa operação consiste em mudar um arco (V,W) ε A em um ramo {V,W}εE, ou vice-versa. Nesse caso um dos conjuntos será diminuído de um elemento e o outro sera aumentado de um elemento. Através dessa operação é possível transformar um grafo orientado, num grafo não orientado, ou vice-versa.

Ex.:



G = (N,E,A)
N = {1, 2, 3, 4}
E = {{1,3},{2,4},{2,4}}
A = {(1,2),(3,4)}



G = (N,A)
N = {1, 2, 3, 4}
A = {(1,2),(1,3),(3,4),
(2,4),(2,4)}

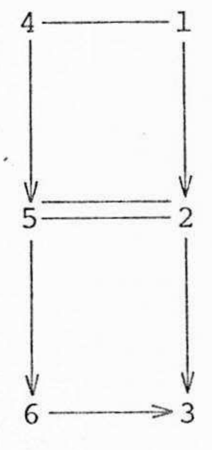
3.2.4. Alteração

Essa operação muda as informações sobre um arco ou ramo, por exemplo, mudança de capacidade de um arco.

3.2.5. Redução

Essa operação transforma um grafo misto num subgrafo não orientado. Nesse caso o conjunto de arcos A torna-se vazio, e o conjunto de nós N pode ser diminuído de alguns elementos.

Ex.:



G = (N,E,A)

N = {1, 2, 3, 4, 5, 6}

A = {(1,2),(2,3),(4,5),
(5,6),(6,3)}

E = {{1,4},{2,5},{2,5}}

G = (N,E)
N = {1, 2, 4, 5}
E = {{1,4},{2,5},{2,5}}

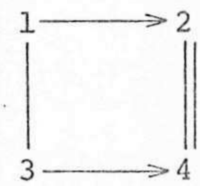
3.2.6. Grau de um Nó

Um ramo {V,W}, ou um arco (V,W), é dito incidente com um nó k se k = V ou k = W.

Num grafo misto um nó k pode ser incidente tanto com arcos como com ramos. A soma do número de arcos e ramos incidentes com um nó k é definido como grau do nó k, cuja representação será grau(k).

Define-se como grau de entrada do nó k, representado por entra(k), o número de arcos cujo nó final é k. O número de arcos cujo nó inicial é k, é definido como grau de saída de k, e representado por sai(k).

Ex.:



grau(2) = 3
entra(2) = 1
sai(2) = 0

3.2.7. Deficit de um Nó

Para cada nó V de um grafo misto define-se deficit de V como entra(V) - sai(V), e denota-se por def(V). No grafo do exemplo anterior o deficit do nó 2 é:

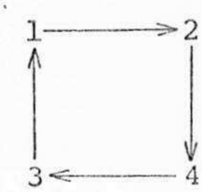
def(2) = entra(2) - sai(2) = 1

3.2.8. Caminho, Rota e Cadeia

Num grafo misto G = (N,E,A) uma sequência finita S = e(1), e(2), ..., e(N), onde e(k) = (V(k), V(k + 1)) ou e(k) = {V(k), V(k + 1)}, é chamada de caminho do nó V(1) para o nó V(N). Quando V(1) = V(N), e este caminho inclui todos os ramos e todos os arcos do grafo, diz-se que este caminho é uma rota.

Cadeia é um caminho ou uma sequência finita S de arcos e ramos, onde o sentido dos arcos não é considerado.

Ex.:



caminho de 1 para 4

1 ⟶ 2

2 ⟶ 4

rota com v(1) = v(N) = 3

3 ⟶ 1

1 ⟶ 2

2 ⟶ 4

4 ⟶ 3

3.3. Representação Matricial de Grafos

A partir de agora, um ramo {i,j} será sempre representado por dois arcos (i,j)* e (j,i)*. O * diferencia um ramo {i,j}, de dois arcos paralelos com sentidos contrários (i,j) e (j,i). Assim, ramos e arcos serão tratados da mesma forma.

Ex.:

1 —— 2 E = {(1,2)*, (2,1)*}

A = { }

1 $\rightleftarrows$ 2 A = {(1,2),(2,1)}

E = { }

3.3.1. Matriz de Incidência

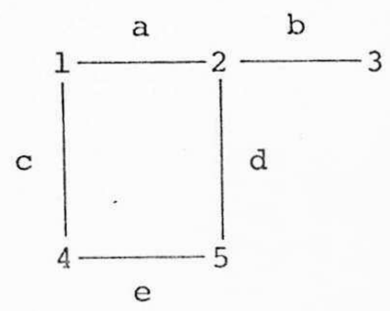
Seja um grafo G = (N,E,A) com N nós e r arcos. A matriz de incidência de G é uma matriz A = [a(i,j)] com N linhas e r colunas, onde as linhas correspondem aos nós, e as colunas correspondem aos arcos.

O conteúdo da matriz é definido da seguinte forma:

a(i,j) = 1 - se o arco j é incidente com o nó i;

a(i,j) = 0 - caso contrário.

Ex.:



Matriz de Incidência

	a	b	c	d	e
1	1	0	1	0	0
2	1	1	0	1	0
3	0	1	0	0	0
4	0	0	1	0	1
5	0	0	0	1	1

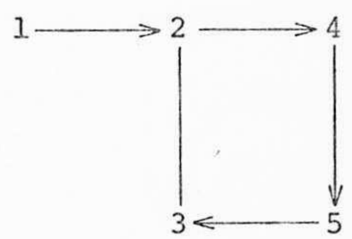
3.3.2. Matriz de Adjacência

Seja um grafo G = (N,E,A) sem ramos ou arcos paralelos. A matriz de adjacência X = [x(i,j)] de ordem N é construída da seguinte maneira:

x(i,j) = 1 - se há um arco de nó i para o nó j;

x(i,j) = 0 - caso contrário

Ex.:



Matriz de Adjacência

	1	2	3	4	5
1	0	1	0	0	0
2	0	0	1*	1	0
3	0	1*	0	0	0
4	0	0	0	0	1
5	0	0	1	0	0

3.3.3. Matriz dos Custos das Cadeias de Menor Custo

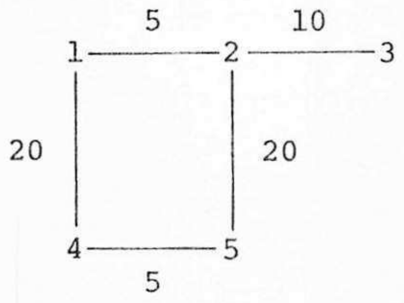
Seja G = (N,E,A) um grafo com N nós e r arcos. A matriz dos custos das cadeias de menor custo, C de ordem N é definida por:

c(i,j) = y - se existe uma cadeia de menor custo y entre os nós i e j;

c(i,j) = infinito - caso contrário, e

c(i,i) = infinito.

Ex.:



Matriz dos Custos das Cadeias de Menor Custo

	1	2	3	4	5
1	inf	5	15	20	25
2	5	inf	10	25	20
3	15	10	inf	35	30
4	20	25	35	inf	5
5	25	20	30	5	inf

3.3.4. Matrizes Esparsas

Uma matriz é dita esparsa quando possui muitos elementos nulos. Não existe uma definição precisa de quando uma matriz é esparsa ou não; é um conceito que se pode reconhecer intuitivamente.

Ex.:

	1	2	3	4	5	6
1	14	0	0	0	0	15
2	0	0	0	0	0	0
3	0	0	0	0	8	0
4	9	0	0	0	0	0
5	0	0	5	0	0	11
6	0	0	0	0	0	3

Figura 3.1

Pode-se notar que a matriz desse exemplo contém uma quantidade muito grande de elementos com valor zero. Apenas 7 dos 36 elementos da matriz não são nulos, Essa matriz é considerada esparsa.

É fácil de notar também que as matrizes de adjacência e de incidência para a representação de grafos são geralmente esparsas.

Quando a matriz que está sendo utilizada é esparsa, pode-se economizar muito espaço de memória e tempo de computação, se somente os termos diferentes de zero forem explicitamente armazenados [3].

Cada elemento da matriz é unicamente caracterizado por sua posição numa linha i e numa coluna j. Armazena-se então a matriz como uma lista de triplas, da seguinte forma:

(i,j,valor)

Ex.: A lista de triplas para representar a matriz esparsa da figura 3.1, seria a seguinte:

i	j	valor
1	1	14
1	6	15
3	5	8
4	1	9
5	3	5
5	6	11
6	6	3

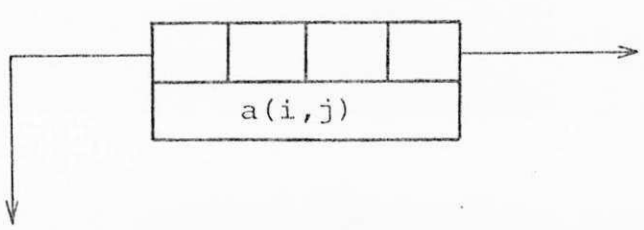
Essa forma de representação sequencial, apesar de suas qualidades econômicas é muito rígida, dificultando as operações com a matriz. Uma estrutura mais geral para o problema é a lista encadeada [3].

Assim, cada coluna da matriz esparsa será representada por uma lista circular encadeada com um nodo-cabeça. E cada linha também irá representar um termo não nulo da matriz, da seguinte forma:

<table>
<tr><td>BAIXO</td><td>LINHA</td><td>COLUNA</td><td>DIREITA</td></tr>
<tr><td colspan="4">VALOR</td></tr>
</table>

O campo BAIXO indica o próximo elemento não nulo da coluna e o campo DIREITA indica o próximo elemento não nulo da linha.

Nodo que esquematiza o termo a(i,j):



A representação da matriz da figura 3.1 por esse método, é vista na figura 3.2.

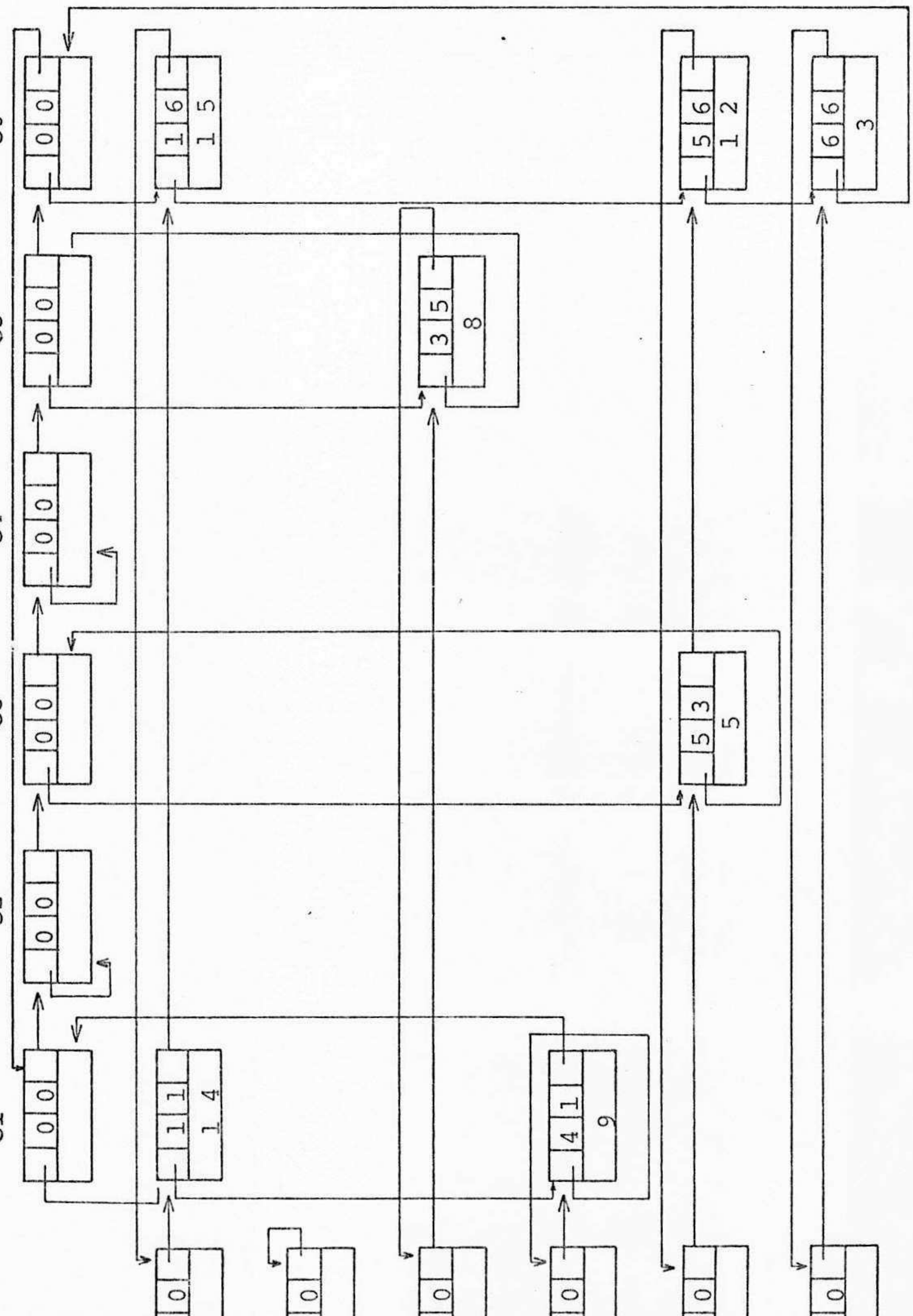


Figura 3.2

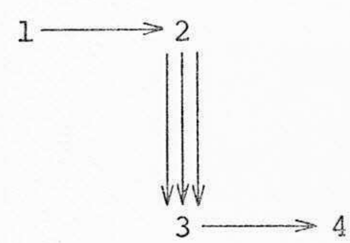
Os nodos-cabeça foram caracterizados como C1 - C6.

3.4. Representação não Matricial de Grafos

3.4.1. Representação Linear

Um grafo G = (N,E,À) também pode ser representado por um conjunto de fórmulas lineares [6]. Essas fórmulas são obtidas através da representação de cada arco por um operador, por exemplo, um * aplicado aos rótulos de nós. Assim, o arco (1,2) é representado por *12. Um grafo utilizando essa técnica seria representado da seguinte maneira:

Ex.:



*12***2333*34

OBS.: No caso de arcos paralelos, o nó inicial é indicado uma única vez, e o nó final é repetido tantas vezes quantos sejam os arcos.

3.4.2. Lista de Arcos

Outra forma não matricial de representação de grafos frequentemente urilizada é uma lista de todos os arcos de um grafo G = (N,E,A) como pares de nós [5].

Ex.: O grafo G = (N,E,A)

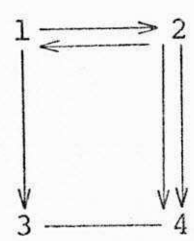


Figura 3.3.

Pode ser representado como o seguinte conjunto de pa res ordenados:

(1,2), (2,1), (1,3), (2,4), (2,4), (3,4)*, (4,3)*.

3.4.3. Listas de Sucessores

Esse método consiste em se representar cada nó k ε N, do grafo G = (N,E,A) por um vetor cujos elementos são os nós que imediatamente sucedem o nó k [3].

O grafo do exemplo da figura 3.3 teria a seguinte representação com essa estrutura:

1: 2,3
2: 1,4,4
3: 4*
4: 3*

3.4.4. Dois Vetores

Uma variação das listas de arcos é a representação do grafo G = (N,E,A) por dois vetores [3], denominados F = {f(1), f(2), ..., f(r)}, e H = {h(1, h(2), ..., h(r)}, onde r é o número de arcos de um nó. Assim, o i-ésimo arco e(i) vai do nó f(i) até o nó h(i).

Para o grafo da figura 3.3 os dois vetores seriam os seguintes:

F = (1,2,1,3*,4*,2,2)
H = (2,1,3,4*,3*,4,4)

Um esquema melhor desse método [3], consiste em um vetor V = {v(1), v(2), ..., v(n)} e um vetor NOF = {nof(1), nof(2), ..., nof(r)}, onde r é o número de arcos do grafo G = (N,E,A). O valor de v(1) indica a posição, no vetor NOF, onde está armazenado o nó final do primeiro arco incidente com o nó i. Os nós finais dos outros arcos incidentes com o nó 1 estarão sequencialmente dispostos em NOF. Assim, o grafo da figura 3.3 seria representado da seguinte maneira:

NO	V
1	1
2	3
3	6
4	7

NOF	
3	1
2	2
4	3
1	4
4	5
4*	6
3*	7

3.4.5. Listas de Adjacência

Essa estrutura consiste na representação dos nós (k1, ..., kN), adjacentes com um nó i, através de uma lista encadeada [5]. Assim um grafo G = (N,E,A) é representado por N listas encadeadas, uma para cada nó i. Cada nodo pertencente a uma lista i, representa um nó adjacente com o nó i. Os nodos serão compostos pelo menos dois campos: NOF e ENC, onde NOF contém o nó adjacente com i, e ENC indica a posição onde está armazenado o próximo nó adjacente com i. ENC = 0 indica que não existem mais nós adjacentes com o nó i.

Usando essa estrutura para armazenar o grafo da figura 3.3 resultaria no seguinte:

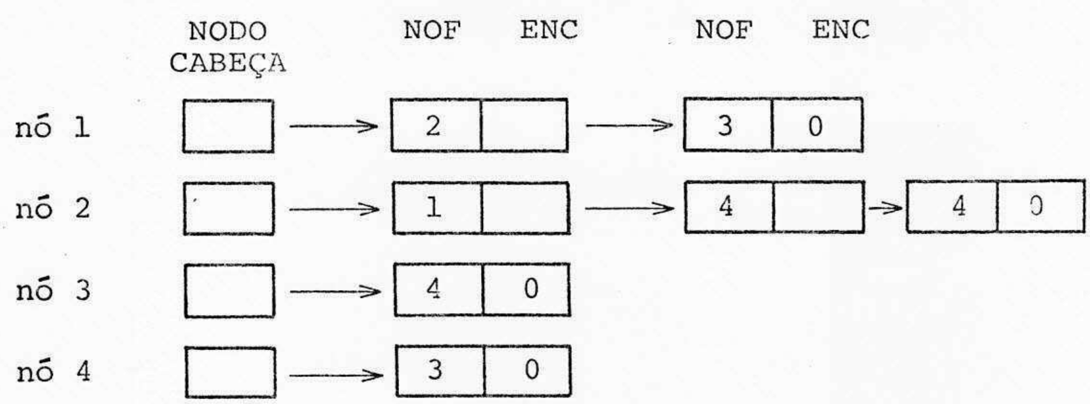


Figura 3.4.

A representação tabular dessa estrutura pode ser feita através da utilização de três vetores: NC(i) (i=1,N) onde são armazenados os nodos-cabeça; NOF(j) (j=1,r) onde são armazenados os nós adjacentes com o nó i; e ENC(j) (j=1,r) cujo conteúdo indica a posição do próximo nó adjacente com o nó i.

	NC	NOF	ENC	ARCOS
1	1	2	2	(1,2)
2	3	3	0	(1,3)
3	6	1	5	(2,1)
4	4	3	0	(4,3)*
5		4	7	(2,4)
6		4	0	(3,4)*
7		4	0	(2,4)

Essa representação deve ser entendida da seguinte forma: a lista de adjacência do nó 1 começa na posição 1 (NC(1)=1), NOF(1)=2 indica a existência do arco (1,2). ENC(1) = 2 indica que o próximo nó adjacente com o nó 1 está armazenado na posição 2. NOF(2) = 3 indica que existe um arco (1,3), e assim por diante. ENC(2) = 0 significa fim da lista de adjacência do nó 1.

Na representação de grafos por listas de adjacência, cada ramo {i,j} tem duas entradas, uma na lista do nó i, (i,j)*, e outra na lista do nó j, (j,i)*.

3.4.6. Listas de Adjacências Multiencadeadas

Em algumas situações torna-se necessário, como no caso do algoritmo ROTA, marcar um arco como já tendo sido examinado. Isto pode ser resolvido transformando-se as listas de adjacência em multilistas de adjacência [4]. Essa estrutura consiste de nodos, que podem ser compartilhados por diversas listas. Assim, cada arco continuará a ser representado por um único nodo, mas esse nodo poderá pertencer a duas ou mais listas. O esquema do nodo será então o seguinte:

NOF	ENC1	ENC2		ENCN
			.....	

Onde o campo NOF contém o nó adjacente com o nó i, o campo ENC1 é o encadeamento de uma lista para o nó i, ENC2 é o encadeamento de outra lista para o nó i, e assim por diante.

Uma nova lista de adjacência, utilizando os mesmos nodos da lista da figura 3.4, pode ser o seguinte:

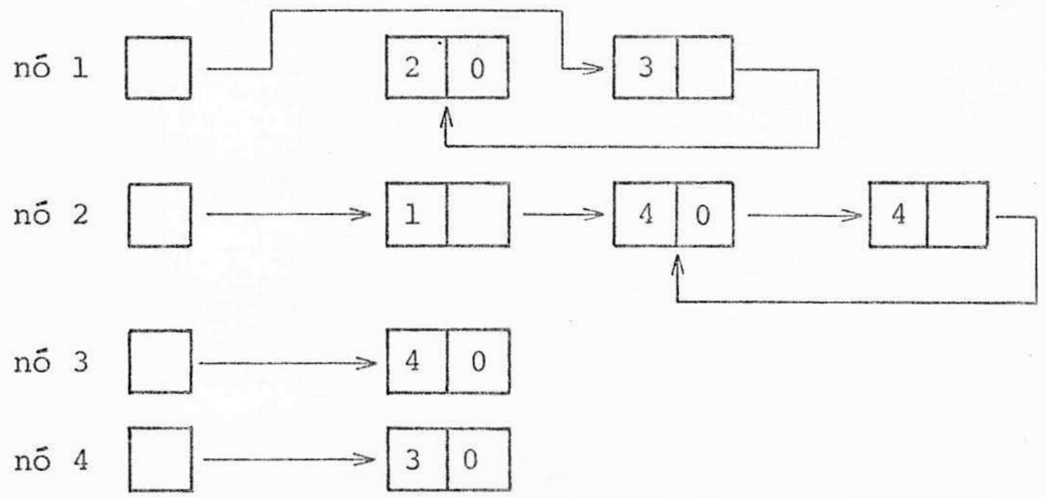


Figura 3.5.

Com a seguinte representação tabular:

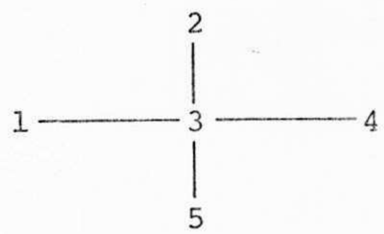
	NC	NOF	ENC	ARCO
1	2	2	0	(1,2)
2	3	3	1	(1,3)
3	6	1	7	(2,1)
4	4	3	0	(4,3)*
5		4	0	(2,4)
6		4	0	(3,4)*
7		4	5	(2,4)

Assim as representações tabulares correspondentes às figuras 3.4 e 3.5, podem ser fundidas formando uma multilista de adjacência.

	NC1	NC2	NOF	ENC1	ENC2	ARCO
1	1	2	2	2	0	(1,2)
2	3	3	3	0	1	(1,3)
3	6	6	1	5	7	(2,1)
4	4	4	3	0	0	(4,3)*
5			4	7	0	(2,4)
6			4	0	0	(3,4)*
7			4	0	5	(2,4)

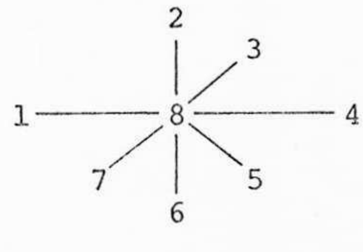
3.5. Discussão das Estruturas de Dados para a Representação de Grafos no Computador

Na solução do problema de rotas para a distribuição de bens e serviços, os grafos são utilizados para representar zonas de uma cidade. Uma análise prática dessas zonas mostra que uma esquina é formada, geralmente, pelo cruzamento de apenas duas ruas. A representação de uma esquina comum num grafo seria a seguinte:



Conclui-se que o número de arcos incidentes com um nó é em média 4.

A prática mostra também que o número de arcos incidentes com um nó, geralmente, não ultrapassa o valor 7.



Essas observações são úteis para o dimensionamento do espaço a ser reservado em função do número de arcos.

3.5.1. Matrizes

A maneira mais frequente de representação de grafos no computador é a matriz, isto porque matrizes constituem-se em mecanismo de fácil manipulação. Além disso, em muitas aplicações de teoria dos grafos, as matrizes também se tornam formas naturais de expressão do problema. Outro incentivo à utilização de matrizes é a correspondência entre algumas propriedades da teoria dos grafos e das matrizes.

3.5.1.1. Matriz de Adjacência

O espaço necessário para se representar um grafo, utilizando matriz de adjacência é de N^2 palavras, considerando-se que se utilizem apenas valores inteiros, e que um número inteiro seja representado em uma palavra de memória.

No caso de um grafo composto de 100 nós, seria necessária a reserva de 10.000 palavras. Considerando-se ainda que as matrizes de adjacência são geralmente esparsas, conclui-se que a representação de grafos por esse método causa um indesejável desperdício de memória. Além disso, é impossível representar-se grafos contendo arcos paralelos nesse tipo de estrutura. Outra desvantagem é a impossibilidade de diferenciar-se um ramo (i,j)*, (j,i)*, de dois arcos paralelos com sentidos contrários (i,j), (j,i).

Considere agora um problema trivial, como calcular o número de arcos existentes num determinado grafo G. Usando matriz de adjacência, um algoritmo para solucionar esse problema tem que pesquisar N^2 elementos para obter a resposta.

Através dessa estrutura é fácil representar os custos para percorrer os arcos do grafo ou qualquer peso que se queira atribuir aos arcos.

3.5.1.2. Matriz dos Custos das Cadeias de Menor Custo

A matriz dos custos das cadeias de menor custo no problema específico de determinação de rotas, é definida para um grafo não orientado, portanto, essa matriz é simétrica. Nesse caso só é necessário armazenar a parte triangular superior da matriz, com um total de $N(N-1)/2$ elementos ao invés de N^2.

3.5.2. Representação Linear

Èsse método apesar de utilizar um espaço pequeno e permitir a representação de arcos paralelos, torna difícil a operação com o grafo, além de não permitir a representação do custo para percorrer os arcos.

3.5.3. Lista de Arcos

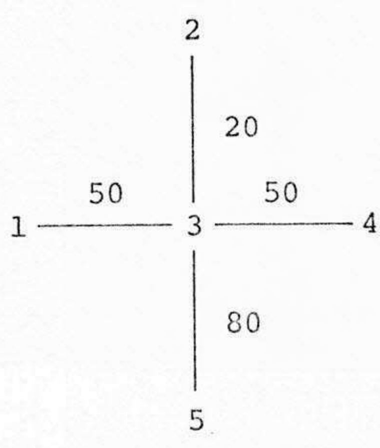
Lista de arcos é uma forma muito conveniente de entrada de dados, mas não é indicada para o armazenamento, e manipulação posteriores dos dados.

3.5.4. Lista de Sucessores

A representação de um grafo pelo método da lista de sucessores, se comparado com o de matrizes de adjacência, traz uma economia de memória considerável. Essa estrutura também permite a representação de arcos paralelos e de custos dos arcos. Para incluir a representação dos custos nessa estrutura, basta considerar-se o nó sucessor como um par (nó sucessor , custo).

Considerando-se que o número médio de arcos por nó é 4, o espaço de memória necessário para armazenar um grafo de 100 nós e os custos para se percorrer os seus arcos seria de 800 palavras, isto é, 100 vetores com 8 posições cada um, quatro para representar os nós sucessores, e quatro para o custo dos arcos.

Ex.:



	1	2	3	4	5	6	7	8
1:	3	50						
2:	3	20						
3:	1	50	2	20	4	50	5	80
4:	3	50						
5:	3	80						

Nota-se claramente que esse método, levando-se em conta apenas a média de ruas por esquina, não permite a representação de mais de 4 arcos incidentes com cada nó. Nesse caso, a reserva de espaço deve prever o pior caso, isto é, considerar a possibilidade de ocorrência de 7 arcos incidentes com cada nó.

Ex.:

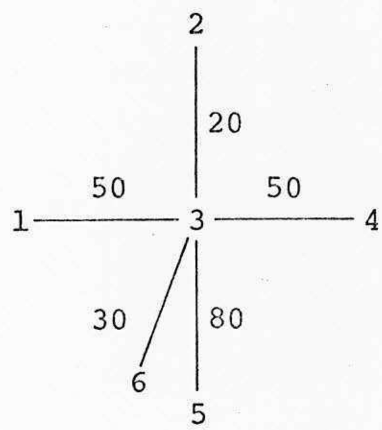


Figura 3.6.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1:	3	50												
2:	3	20												
3:	1	50	2	20	4	50	5	80	6	30				
4:	3	50												
5:	3	80												
6:	3	30												

Para se representar o grafo da figura 3.6, é necessária a reserva de N vetores de 14 posições cada um, isto é, 14N posições. No caso de um grafo de 100 nós, o espaço total reservado seria de 1.400 palavras, uma economia muito grande, se comparada com as matrizes de adjacência.

O algoritmo para verificação do número de arcos existentes num grafo G, utilizando essa estrutura, teria que pesquisar r + N elementos, onde r é o número de arcos do grafo e r <<(N^2).

Essa estrutura apesar de melhorar bastante a ocupação do espaço de memória, ainda ocasiona desperdícios. É necessária a previsão do pior caso, que implica na reserva de espaço ocioso. Nesse caso torna-se necessário a reserva de 14N posições, apesar de serem utilizadas, em média, apenas 8N. Portanto, tem-se aproximadamente (14N - 8N) palavras de memória ociosas.

3.5.5. Dois Vetores

Esse método permite um melhor aproveitamento do espaço, comparado com estruturas anteriormente discutidas. Possibilita também a representação de pesos, de arcos paralelos, e a diferenciação entre (i,j)*, (j,i)* e (i,j), (j,i). Para incluir os custos basta considerar-se mais um vetor CUSTO = {custo(1), custo(2), ..., custo(r)}, cujas posições indicarão o custo do arco.

Nessa estrutura a reserva de espaço para o armazenamento do grafo pode ser feito para o número médio de arcos por nó. Assim diminui-se o espaço ocioso em relação à estrutura anterior. São necessárias N+2(4N) = 9N palavras, isto é, um vetor V com N posições, e dois vetores NOF e CUSTO com 4N posições cada um.

Seria necessário pesquisar r+1 elementos, para determinar o número de arcos de um grafo G representado por essa estrutura.

A utilização desse método para a representação de grafos no computador, apesar de muito eficiente em termos de ocupação de memória, é muito rígida, não permitindo facilidades para operações com o grafo. A inclusão de um novo arco, por exemplo, torna-se uma operação difícil e dispendiosa em termos de tempo de processamento.

3.5.6. Listas de Adjacência

Essa estrutura apesar de um pouco menos eficiente em termos de utilização de espaço, permite grande facilidade de operação com o grafo, exigindo em compensação mais tempo de processamento. Os métodos citados anteriormente, com exceção da matriz de adjacência, são muito rígidos, apresentando dificuldades quando alguma alteração for necessária.

A representação dos custos nessa estrutura se faz através da inclusão de mais um campo em cada nodo. Pode-se também incluir mais um campo para diferenciar um ramo de dois arcos paralelos.

NOF	CUSTO	ENC

Os nodos disponíveis são mantidos em uma lista separada. Qualquer lista poderá utilizá-los, quando for necessário.

O algoritmo para a verificação do número de arcos , caso seja utilizada essa estrutura, precisará pesquisar r+N elementos.

Nesse tipo de estrutura é necessária a reserva de N+3(4N) = 13N, isto é, um vetor V com N posições, e três vetores NOF, CUSTO e ENC com 4N elementos cada um. Para um grafo contendo 100 nós, são necessárias 400 palavras a mais que a estrutura de DOIS VETORES, mas em compensação possibilita uma flexibilidade muito maior de operação com grafos.

Conclui-se assim, que a estrutura de dois vetores é a mais eficiente quando for necessário apenas consultar os dados sem alterá-los; as listas de adjacências são melhores quando se fazem operações com o grafo; e as listas de ramos são as mais indicadas para a entrada de dados.

CAPÍTULO IV

4. DETERMINAÇÃO DA ROTA

Neste capítulo serão descritos os procedimentos necessários para a determinação de uma rota para uma zona.

Sempre que houver referência aos arquivos, a nomenclatura FDA indicará a condição de Fim De Arquivo.

4.1. Prepara o Grafo Inicial (Figura 2.2)

O grafo que representa uma zona para a qual será determinada uma rota é montado a partir das informações contidas nos arquivos CADASTRO DE TRECHOS e CADASTRO DE RESTRIÇÕES.

4.1.1. Monta o Grafo

Entrada: arquivo CADASTRO DE TRECHOS.
Saída: arquivo GRAFO.

Passo 1: Lê registro do arquivo CADASTRO TRECHOS.
Passo 2: Se é FDA. Então vá para o passo 7.
Passo 3: Se o trecho não tem coleta
Então vá para o passo 1.
Passo 4: Se o trecho é intrasitável
Então vá para o passo 1.
Passo 5: Armazena as informações sobre o trecho no arquivo GRAFO.
Passo 6: Vá para o passo 1.
Passo 7: Ordena o arquivo GRAFO. PARE.

Um exemplo para ilustrar esse algoritmo é visto a seguir.

Supondo-se a seguinte zona:

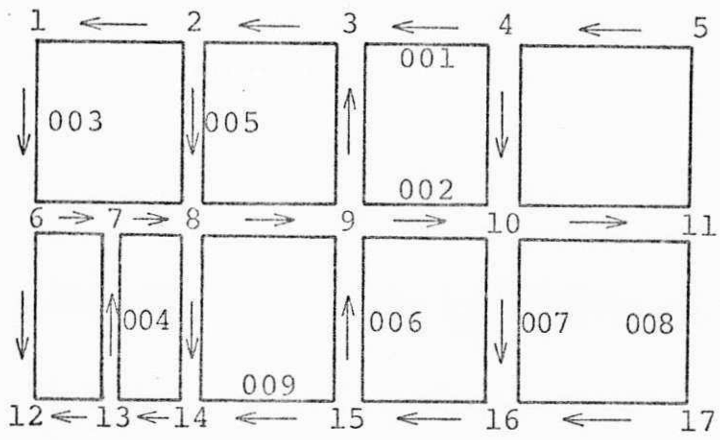


Figura 3.1

Para a qual as informações necessárias já foram coletadas e criticadas, dando origem ao seguinte arquivo CADASTRO DE TRECHOS.

ARQUIVO CADASTRO DE TRECHOS

COD.RUA	NOINIC	NOFIM	DIST	SENT	COND	ESTAD	COL
001	2	1	60	1	1	1	1
001	3	2	80	1	1	1	1
001	4	3	80	1	1	1	1
001	5	4	80	1	1	1	1
002	6	7	30	1	1	1	1
002	7	8	30	1	1	1	1
002	8	9	80	1	1	1	1
002	9	10	80	1	1	1	1
002	10	11	80	1	1	1	1
009	13	12	30	1	1	1	1
009	14	13	30	1	1	1	1
009	15	14	80	1	1	1	1
009	16	15	80	1	1	1	1
009	17	16	80	1	1	1	1
003	1	6	60	1	1	1	1
003	6	12	60	1	1	1	1
004	13	7	60	1	1	1	1
005	2	8	60	1	1	1	1
005	8	14	60	1	1	1	1
006	9	3	60	1	1	1	1
006	15	9	60	1	1	1	1
007	4	10	60	1	1	1	1
007	10	16	60	1	1	1	1
008	17	11	60	2	1	1	1
008	11	5	60	2	1	1	1

Esse arquivo é utilizado como entrada do algoritmo que MONTA O GRAFO, e o arquivo GRAFO resultante é o seguinte:

ARQUIVO GRAFO

NOINIC	NOFIM	CUSTO	SENTID
1	6	60	1
2	1	60	1
2	8	60	1
3	2	80	1
4	3	80	1
4	10	60	1
5	4	80	1
5	11	60	2
6	7	30	1
6	12	60	1
7	8	30	1
8	9	60	1
8	14	60	1
9	3	60	1
9	10	80	1
10	11	80	1
10	16	60	1
11	5	60	2
11	17	60	2
13	7	30	1
13	12	30	1
14	13	30	1
15	9	60	1
15	14	60	1
15	15	80	1
16	15	80	1
17	11	60	2
17	16	80	1

Figura 3.2.

4.1.2. Monta Matriz Check

Objetivo: Construir a MATRIZ CHECK a partir do arquivo CADASTRO DE ESQUINAS. Essa matriz será utilizada na crítica do grafo e impressão da ROTA.

A matriz CHECK tem a seguinte forma:

a) Os índices das linhas representam as ruas;
b) Os índices das colunas representam os nós;
c) O conteúdo da matriz (0 e 1) indica se um nó pertence (1), ou não (0), a uma determinada rua.

Essa matriz foi concebida para auxiliar na crítica ao grafo. Será utilizada na verificação dos dados que entram via CADASTRO DE TRECHOS.

Além de detectar erros na entrada dos dados, a MATRIZ CHECK também será utilizada na impressão da ROTA, indicando todos os cruzamentos entre as ruas.

Algoritmo:

Entrada: arquivo CADASTRO DE ESQUINAS.

Saída: arquivo MATRIZ CHECK.

OBS.: NO(i) = 0 indica fim da lista de nós da rua.

Passo 1: Zera a MATRIZ CHECK.

Passo 2: Lê registro do CADASTRO DE ESQUINA.

Passo 3: Se é FDA
Então armazena a MATRIZ CHECK; PARE.

Passo 4: i = 1
Enquanto NO(i) diferente de 0, Faça
Início
MATRIZ CHECK(CODRUA, NO(i)) = 1
i = i + 1
Fim.

Passo 5: Vá para o passo 2.

Ex.: O CADASTRO DE ESQUINAS da Zona da Figura 3.1 é:

ARQUIVO CADASTRO DE ESQUINAS

CODRUA	NO(1)	NO(2)	NO(3)	NO(4)	NO(5)	NO(6)	NO(7)
001	1	2	3	4	5	0	
002	6	7	8	9	10	11	0
003	1	6	12	0			
004	7	13	0				
005	2	8	14	0			
006	3	9	15	0			
007	4	10	16	0			
008	5	11	17	0			
009	12	13	14	15	16	17	0

A MATRIZ CHECK resultante do algoritmo monta MATRIZ CHECK seria a seguinte:

MATRIZ CHECK

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0
3	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
4	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
5	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
6	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0
7	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0
8	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1
9	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

4.1.3. Verifica Erros do Grafo

Objetivo: evitar a entrada de erros no grafo inicial.

Uma primeira verificação que deve ser feita, é assegurar-se a unicidade de numeração de cada nó. Além disso, precauções devem ser tomadas para garantir que todos os nós do grafo tenham ao menos um arco entrando e um saindo dele. De outra forma, poderia ocorrer que a rota chegasse numa intersecção e não pudesse sair dela. Caso ocorra essa condição, e não seja constatado erro na orientação dos arcos, pode-se solucionar o problema adicionando-se à zona trechos retirados da zona vizinha, visto que esse problema só ocorrerá em ruas limítrofes. Esses trechos adicionais terão serviço inútil. Deve ser verificado também se o número total de nós está dentro do limite estabelecido.

Nesse passo verificações devem ser feitas para detectar um nó desconectado, ou ramo codificado errôneamente. A MATRIZ CHECK será utilizada para verificação desses erros.

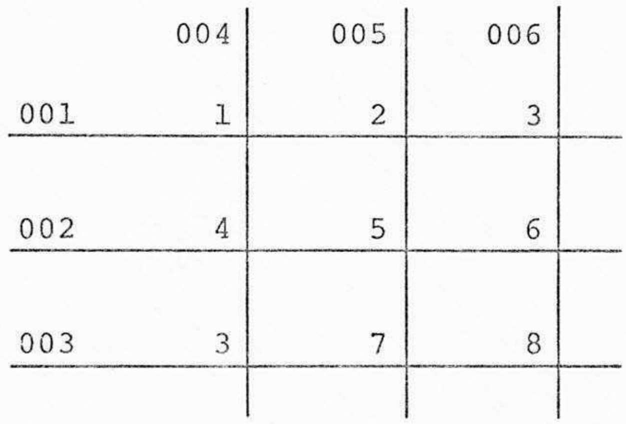
Os erros apontados devem ser corrigidos para remover as discrepâncias, e a CRÍTICA deve ser então repetida. Na conclusão desse passo, todos os erros devem ter sido removidos, e um grafo G = (N,E,A,c) misto e conexo, deve ter sido especificado, onde N é o conjunto de arcos, e c é o custo para se percorrer um arco ou ramo.

O procedimento da identificação dos erros pode ser feito da seguinte forma:

a) Calcula-se o número total de nós, e verifica-se se está dentro do limite máximo especificado;
b) Para identificar a existência de um nó desconectado, basta pesquisar sequencialmente os vetores NOINIC e NOFIM do arquivo GRAFO (exemplo na figura 3.2), e verificar se todos os nós estão presentes em pelo menos um deles. Caso todos os nós estejam presentes, pelo menos uma vez nos dois vetores, é uma indicação de que todos os nós possuem ao menos um arco entrando e um arco saindo dele;
c) Duas ruas com o mesmo código ou a tentativa de utilizar trechos de uma rua não cadastrada na zona é identificada na CRÍTICA DOS DADOS (Apêndice I);
d) Para verificar a unicidade de numeração de cada nó, utiliza-se a MATRIZ CHECK da seguinte forma:

Admite-se que na prática duas ruas de uma zona só se cruzam uma única vez num determinado nó. Assim, duas linhas da MATRIZ CHECK só devem possuir simultaneamente o valor 1 numa única coluna, caso contrário é uma indicação de que existem dois nós com a mesma numeração.

Ex.: Supondo-se a seguinte zona:



Com a seguinte MATRIZ CHECK:

	1	2	3	4	5	6	7	8
001	1	1	1	0	0	0	0	0
002	0	0	0	1	1	1	0	0
003	0	0	1	0	0	0	1	1
004	1	0	1	1	0	0	0	0
005	0	1	0	0	1	0	1	0
006	0	0	1	0	0	1	0	1

A coluna 3 contem 4 valores 1, indicando que há dois nós com o número 3.

e) Um arco codificando errôneamente, por exemplo, o arco (1,2) transformado em (1,22), por um erro no preenchimento do formulário ou de perfuração dos dados, pode ser identificado conferindo-se todos os trechos do grafo com a MATRIZ CHECK.

Como as informações contidas no CADASTRO DE ESQUINAS são utilizadas para verificação do CADASTRO DE TRECHOS, sugere-se, para maior confiabilidade do sistema, que os formulários para a entrada dos dados nesses dois arquivos, sejam preenchidos por pessoas diferentes, supondo-se que a possibilidade de duas pessoas cometerem o mesmo erro é pequena.

4.2. Algoritmos para Transformação do Grafo Inicial num Grafo de Euler (Mixed A).

4.2.1. Algoritmos para Determinação dos Menores Caminhos

Seja C = c(i,j) a matriz custo de um grafo G=(N,E,A,c) com custos não negativos.

O problema consiste em encontrar um caminho de i para j, de tal forma que os custos para percorrer esse caminho sejam mínimos.

O algoritmo de DIJKSTRA [9] é o de maior eficiência para se determinar o caminho mínimo entre dois nós s e t, bem como o caminho mínimo entre um nó s e todos os outros nós $i \in N$.

Nesse trabalho o algoritmo de DIJKSTRA é utilizado para resolver os dois tipos de problemas citados anteriormente.

Em geral esse método se baseia na atribuição de rótulos temporários aos nós. O rótulo de um nó contem o limite superior do custo do caminho de s para um outro nó. Esses rótulos são continuamente reduzidos por um procedimento iterativo, e em cada iteração, exatamente um rótulo temporário p se torna permanente, indicando o custo exato do menor caminho entre os nós s e p.

Define-se T(xi) como um conjunto de nós tal que $(xi,xj) \in A$.

<u>Algoritmo de DIJKSTRA</u> [9]

Seja l(xi) o rótulo do nó xi.

Passo 1: Faca l(s) = 0 e marque o seu rótulo como permanente.
Faca l(xi) = infinito para todos os xi diferentes de s e marque seus rótulos como temporários.
p = s

Passo 2: Para todos os $xi \in T(p)$ com rótulos temporários faca:
$l(xi) = \min\{l(xi),\ l(p) + c(i,j)\}$

Passo 3: Dentre todos os nós com rótulos temporários encontre x' de modo que $l(x') = \min\{l(xi)\}$

Passo 4: Marque o rótulo xi' como permanente e faca $p = xi'$

Passo 5: (i) (Se somente o caminho de s para t é desejado)
Se $p = t$
Então $l(p)$ é o custo do caminho mínimo entre s e t PARE.
Senão vá para o passo 2.
(ii) (Se um caminho de s para todos os outros nós for desejado)
Se todos os nós têm rótulos permanentes.
Então os rótulos são os custos dos caminhos mínimos. PARE.
Senão vá para o passo 2.

4.2.2. Casamento num Grafo Completo

Seja $G = (N,E)$ um grafo completo com um conjunto de nós N e um conjunto de ramos E. Chama-se casamento ao subconjunto $M \subset E$, tal que não existem dois ramos em M que sejam incidentes com um mesmo nó. Se o grafo $G = (N,E,c)$ com custos não negativos, com $c(i,j)$ representando o custo para se percorrer cada ramo $\{i,j\}$, pode-se associar a cada $M \in \bar{M}$ o custo:

$$c(M) = \sum_{\{i,j\} \in M} c(i,j)$$

Agora, seja $C = e1, e2, \ldots,$ onde $ek = \{i(k), i(k+1)\}$, um caminho num grafo $G = (N,E)$. Diz-se que um caminho é elementar se todos os nós $i(1), \ldots, i(n)$ são dois a dois distintos. Então observa-se que um caminho desse tipo fornece dois casamentos de nós em G:

a) todos os ramos ej do caminho C com j par;

b) todos os ramos ej do caminho C com j ímpar.

Esta observação é usada agora para aumentar um casamento maximal como segue: um caminho elementar C é dito alternativo, se os seus ramos estão alternativamente em M. Se além disso, C fundir dois nós não incidentes com os ramos de M, diz-se que C é um caminho alternativo aumentado. É claro que um caminho alternativo aumentado sempre tem um número ímpar de ramos, e se $C = e1, \ldots, e(2N+1)$, então todos os ramos ej, com j par pertencem a M, e todos os ramos ej, com j ímpar não pertencem a M. Mas os ramos ej com j ímpar, juntamente com todos os ramos de M, que não pertencem a C, formam um casamento cujo número de ramos, é igual ao número de ramos de M mais 1. O novo casamento obtido dessa maneira será denotado por Mc, e sua cardinalidade por $|Mc|$, que conforme foi observado, é igual a $|M| + 1$.

Denotando-se por $\mathcal{C}(M)$ o conjunto de todos os caminhos alternativos aumentados com respeito ao casamento M, associe-se o custo c(Mc) para o casamento correspondente Mc, com c(Mc) definido por:

$$c(Mc) = \sum_{\{i,j\} \in M/C} c(i,j) + \sum_{\{i,j\} \in C/M} c(i,j)$$

O conjunto de todos os casamentos perfeitos será denotado por $\overline{M}p$.

O algoritmo que se segue tem como objetivo a construção de um casamento maximal com custo mínimo.

<u>Algoritmo de DERIGS</u> [11]

Passo 1: Faca $M = \phi$

Passo 2: Se $M \in \overline{M}p$: PARE, M é otimal

Passo 3: Determine $\overline{C}$ e $\mathcal{C}(M)$, de modo que para $M\overline{c}$, com $c(M\overline{c}) \leq c(Mc)$ para todos os $\overline{c} \in (M)$.

Passo 4: Faca $M = M\overline{c}$
vá para o passo 2.

4.2.3. Fluxos Gerais com Custo Mínimos

O algoritmo que será apresentado tem como finalidade principal o equilíbrio dos nós, isto é, transformar o grafo de modo que cada nó tenha o grau de entrada igual ao grau de saída. Para estabelecer esse equilíbrio será usado o problema de fluxos gerais com custos mínimos.

Definições: associa-se com o grafo G = (N,A,c,u) o seguinte problema de fluxo em rede:

Minimize $\sum_{e \in A} c(e) * x(e)$

Sujeito a:

$$\sum_{e \,\varepsilon\, sai(i)} x(e) - \sum_{e \,\varepsilon\, entra(i)} x(e) = def(i)$$

para todos os i ε N

x(e) ⩾ 0

Algoritmo de BUSACKER e GOWEN [10]

Passo 1: Faca

Defina: $a(i,j) = \begin{cases} 1 & \text{se o nó inicial de } j = i \\ -1 & \text{se o nó final de } j = i \\ 0 & \text{caso contrário} \end{cases}$

-A = {-1, ..., -p}

def(i) = deficit no nó i

q = número de arcos

p = número de nós

mzero = 0

nzero = 0

V+ = (i/def(i) > 0)

V- = (i/def(i) < 0)

x(j) = 0 para todos os j = 1, ..., q

Passo 2: Se V+ = φ: PARE (x é otimal)

Se V- diferente de φ

Então escolha i0 ε V+

Se mzero = 0

Então m = def(i0)

Passo 3: A' = {j ε A/x(j) < b(j)}

A'' = {-j ε -A/0 < x(j)}

Passo 4: Aplique o algoritmo de Dijkstra ao grafo o rientado D = (N,A' ∪ A'', c) e determine o vetor l = (li, ..., ln), onde l(i) designa a distância entre i0 para i, sendo l(i) = infinito quando não existe caminho de i0 para i.

Passo 5: Faca N' = i/l(i) < infinito)

Passo 6: Se V ∩ N' = φ : PARE (problema sem solução)

Senão escolha il ε (V- ∩ N')/l(i) é mínimo

Escolha um caminho mínimo w de i0 para il.

Se nzero = 0 : n = def(il)

Passo 7: E(j) = b(j) - x(j) para j ε w

$E(j) = x(j)$ para $j \in w$

$Epis = \min \{E(j)/j \in w$ ou $-j \in w,\ m,\ n\}$

Passo 8: $x(j) = x(j) + Epis$ se $j \in w$

$x(j) - Epis$ se $-j \in w$

$x(j)$ caso contrário

Passo 9: $m = def(i0) - \sum_{j=1}^{q} a(i0,j) * x(j)$

$n = -def(i1) + \sum_{j=1}^{q} a(i1,j) * x(j)$

Passo 10: mzero = 1

nzero = 1

Se $m = 0$

Então $V+ = V+ - \{i0\}$

Se $n = 0$

Então $V- = V- - \{i1\}$

Vá para o passo 2.

4.2.4. GP - Variação Minimal

Este algoritmo faz o casamento de nós de grau ímpar repetindo tantos arcos quantos forem necessários, para transformá-los em grau par. Para todos os outros nós de G o grau permanece par.

Entrada: um grafo $G = (N,E,A,c)$

Saída: um grafo G com grau(i) par para todos os nós $i \in N$.

Passo 1: encontre um conjunto $N' \subset N$ de nós com grau ímpar em G.

Passo 2: Para todo $i,j \in N'$ encontre a cadeia de mínimo custo entre i e j e o custo correspondente $c(i,j)$.

Passo 3: Encontre um casamento de mínimo custo M, no grafo completamente não orientado, utilizando o conjunto de nós N' e os custos $c(i,j)$.

Passo 4: Para cada $\{i,j\} \in M$ duplique em G os ramos e arcos que pertencem à cadeia mínima entre i e j.

4.2.5. ES - Variação Minimal

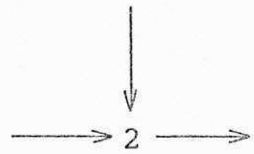
Entrada: um grafo G = (N,E,A,c)

Saída: um grafo G com grau de entrada = grau de saída para todo $i \in N$.

Definições:

a) para cada nó $i \in N$ define-se deficit de i por def(i) = entra(i) - sai(i). Denota-se por entra(i), o número de arcos que têm em 1 seu nó terminal, e sai(i) o número de arcos que têm i por nó inicial.

Ex.:



entra(2) = 2
sai(2) = 1
def(2) = 1

b) além disso associam-se duas funções ao grafo G = (N,A,c,u), onde N é o conjunto de nós, A é o conjunto de arcos, $c: A \to R$ é a função custo, e $u: A \to R$ é a função capacidade.

Passo 1: Associe a cada arco $e \in A$ a capacidade u(e) = infinito e faça o custo c(e) igual ao custo do grafo original.

Passo 2: Substitua cada ramo $e = \{i,j\} \in E$ por um arco $e = (i,j)$.

Passo 3: Calcule o deficit de cada nó.

Passo 4: Se def(i) = 0 para todo $i \in N$: PARE

Passo 5: Para cada arco $\vec{e} = (i,j)$ criado no passo 2, crie dois novos arcos $\overleftarrow{e} = (j,i)$ e $e' = (j,i)$. Além disso faca:

$c(\vec{e}) = c(\overleftarrow{e}) = c(e)$

$c(e') = 0$

$u(\vec{e}) = u(\overleftarrow{e}) = \text{infinito}$

$u(e') = 2.$

Passo 6: Aplique o algoritmo de fluxo.

Passo 7: Substitua cada arco $e \in A$ por $x(e) + 1$ arcos paralelos a e.

Passo 8: Para cada ramo $e \in E$ faca:

Se $x(e') = 0$

Então substitua e por um arco paralelo a $\vec{e}$.

Se x(e') = 1

Então mantenha o ramo.

Se x(e') = 2

Então substitua e por um arco paralelo a $\overleftarrow{e}$.

Passo 7: Adicione $x(\overrightarrow{e})$ cópias paralelas a $\overrightarrow{e}$ e $x(\overleftarrow{e})$ cópias paralelas a $\overleftarrow{e}$.

4.2.6. Algoritmo Mixed 1A

Entrada: um grafo G = N,E,A,c).

Saída: um grafo G'' = (N,E,A,c) com grau de entrada igual ao grau de saída e grau de cada nó par.

Passo 1: Aplique o algoritmo GP - Variação Minimal ao grafo com saída G'.

Passo 2: Aplique o algoritmo ES - Variação Minimal ao grafo G' com saída G''.

4.2.7. Agoritmo Mixed 2A

Entrada: um grafo G = (N,E,A,c).

Saída: um grafo G'' = (N,E,A,c) com grau de entrada igual ao grau de saída e com grau de cada nó par.

Passo 1: Aplique o algoritmo ES - Variação Minimal ao grafo G, com saída G' = (N,E,A,c).

Passo 2: Aplique o algoritmo GP - Variação Minimal ao grafo parcial G* = (N,E,c) com saída G''.

4.2.8. Algoritmo Mixed A

Entrada: Arquivo GRAFO (um grafo G = (N,E,A,c) misto e conexo).

Saída: Uma rota de mínimo custo contendo todos os arcos e ramos de G (arquivo ROTA).

Passo 1: Aplique o algoritmo Mixed 1A em G.

Passo 2: Aplique o algoritmo Mixed 2A em G.

Passo 3: Compare as soluções de Mixed 1A e Mixed 2A e escolha a de menor custo.

Para ilustrar o funcionamento dos algoritmos é dado o seguinte exemplo:

Dado o grafo com custo 130 representado pelo seguinte diagrama:

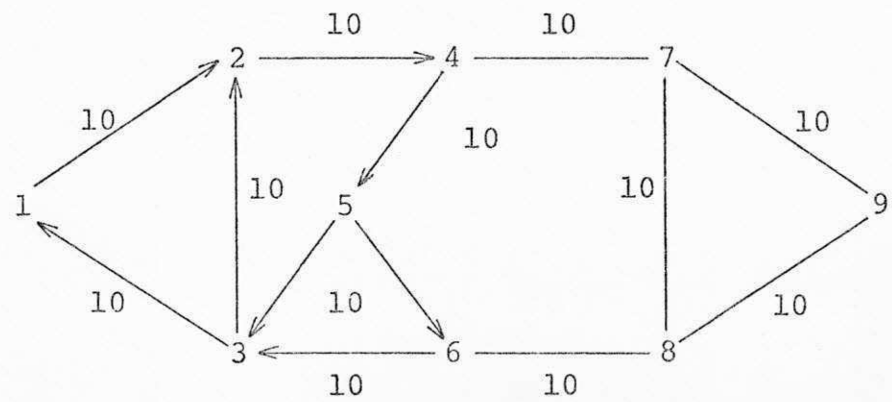


Figura 3.3.

Esse grafo tem o seguinte ARQUIVO GRAFO.

ARQUIVO GRAFO

NOINIC	NOFIM	CUSTO	SENTID
1	2	10	1
2	4	10	1
3	1	10	1
3	2	10	1
4	5	10	1
4	7	10	2
5	3	10	1
5	6	10	1
6	3	10	1
6	8	10	2
7	4	10	2
7	8	10	2
7	9	10	2
8	6	10	2
8	7	10	2
8	9	10	2
9	7	10	2
9	8	10	2

Aplicando o Algoritmo Mixed A a esse grafo tem-se o seguinte:

a) Construção do Mixed 1A correspondente:

a.1) GP - Variação Minimal.

Passo 1:

N

NO	GRAU
1	2
2	3
3	4
4	3
5	3
6	3
7	3
8	3
9	2

N' = {2, 4, 5, 6, 7, 8}

Passo 2:

MATRIZ CUSTO DAS CADEIAS MÍNIMAS (N')

	2	4	5	6	7	8
2	0	10	20	20	20	30
4	10	0	10	20	10	20
5	20	10	0	10	20	20
6	20	20	10	0	20	10
7	20	10	20	20	0	10
8	30	20	20	10	10	0

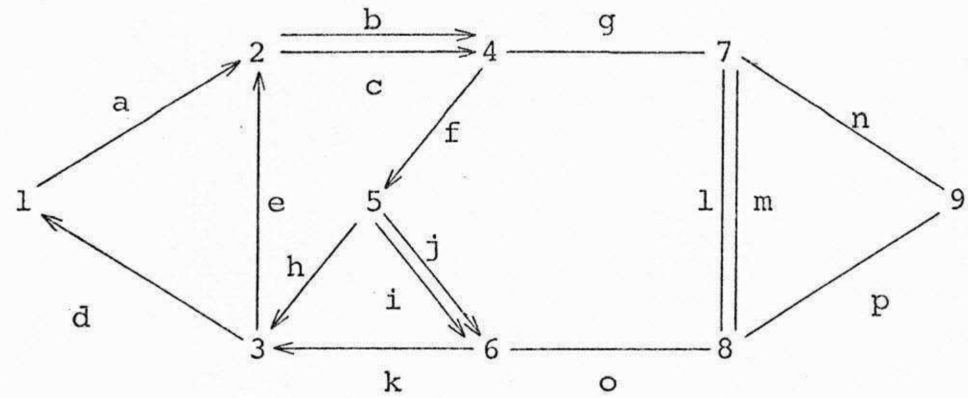
Casamento de Mínimo Custo

M

2 —— 4
5 —— 6
7 —— 8

{i,j} ε M	CADEIA MINIMAL DE i para j
{2,4}	2 —— 4
{5,6}	5 —— 6
{7,8}	7 —— 8

O grafo seguinte é o resultado da transformação efetuada pelo algoritmo GP - Variação Minimal. Esse resultado será utilizado como entrada para o passo 2 do algoritmo Mixed 1A, que é a aplicação do algoritmo ES - Variação Minimal.

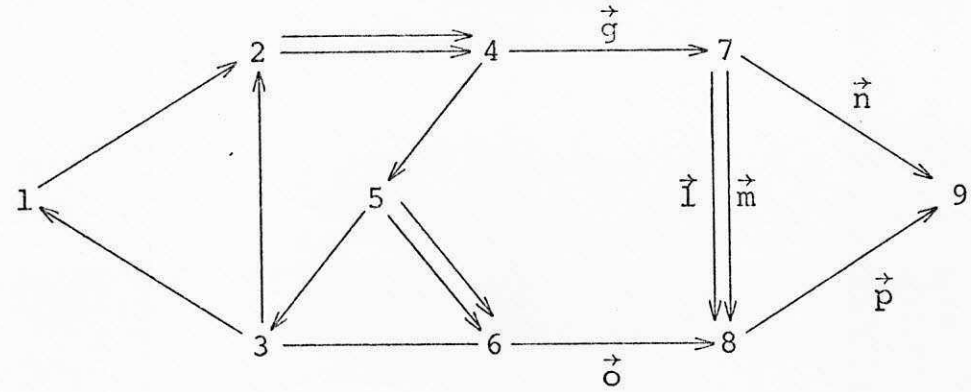


a.2) ES - Variação Minimal

Passo 1: A = {a,b,c,d,e,f,h,i,j,k}

u(a) = infinito	c(a) = 10
u(b) = infinito	c(b) = 10
u(c) = infinito	c(c) = 10
u(d) = infinito	c(d) = 10
u(e) = infinito	c(e) = 10
u(f) = infinito	c(f) = 10
u(h) = infinito	c(h) = 10
u(i) = infinito	c(i) = 10
u(j) = infinito	c(j) = 10
u(k) = infinito	c(k) = 10

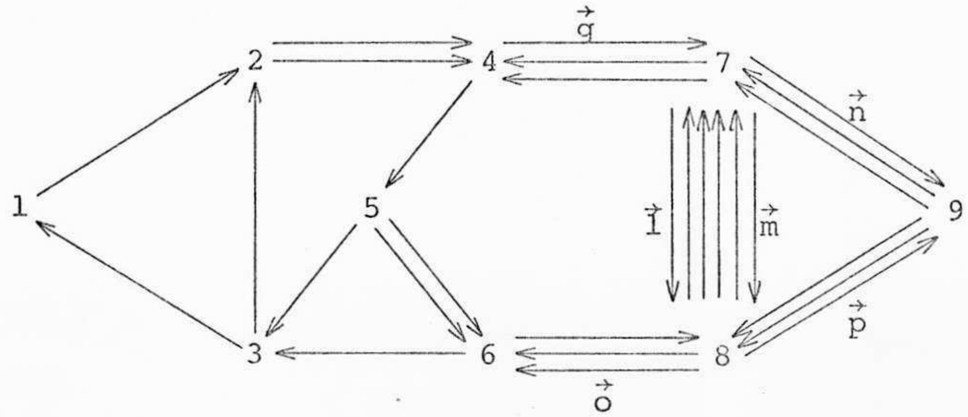
Passo 2: E = {g,l,m,n,o,p}



Passo 3:

NO	DEFICIT
1	0
2	0
3	0
4	0
5	-2
6	0
7	-2
8	2
9	2

Passo 4:

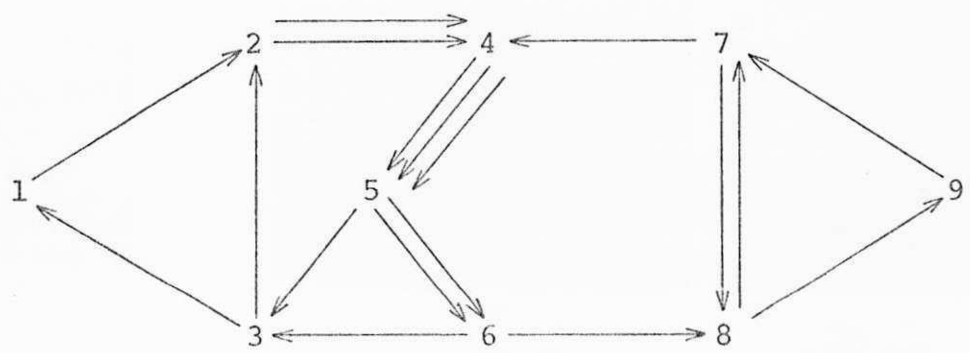


$u(\vec{g}) = inf$	$c(\vec{g}) = 10$
$u(\overleftarrow{g}) = inf$	$c(\overleftarrow{g}) = 10$
$u(g') = 2$	$c(g') = 0$
$u(\vec{l}) = inf$	$c(\vec{l}) = 10$
$u(\overleftarrow{l}) = inf$	$c(\overleftarrow{l}) = 10$
$u(l') = 2$	$c(l') = 0$
$u(\vec{m}) = inf$	$c(\vec{m}) = 10$
$u(\overleftarrow{m}) = inf$	$c(\overleftarrow{m}) = 10$
$u(m') = 2$	$c(m') = 0$
$u(\vec{n}) = inf$	$c(\vec{n}) = 10$
$u(\overleftarrow{n}) = inf$	$c(\overleftarrow{n}) = 10$
$u(n') = 2$	$c(n') = 0$
$u(\vec{o}) = inf$	$c(\vec{o}) = 10$
$u(\overleftarrow{o}) = inf$	$c(\overleftarrow{o}) = 10$
$u(o') = 2$	$c(o') = 0$
$u(\vec{p}) = inf$	$c(\vec{p}) = 10$
$u(\overleftarrow{p}) = inf$	$c(\overleftarrow{p}) = 10$
$u(p') = 2$	$c(p') = 0$

Passo 6: Saída do algoritmo de fluxo

ARCO	X
a	0
b	0
c	0
d	0
e	0
f	2
$\overrightarrow{g}$	0
$\overleftarrow{g}$	0
g'	2
h	0
i	0
j	0
k	0
$\overrightarrow{l}$	0
$\overleftarrow{l}$	0
l'	2
$\overrightarrow{m}$	0
$\overleftarrow{m}$	0
m'	0
$\overrightarrow{n}$	0
$\overleftarrow{n}$	0
n'	2
$\overrightarrow{o}$	0
$\overleftarrow{o}$	0
o'	0
$\overrightarrow{p}$	0
$\overleftarrow{p}$	0
p'	0

Passo 8 e Passo 9: Grafo resultante do algoritmo Mixed 1A, com custo = 180.



b) Construção do Mixed 2A

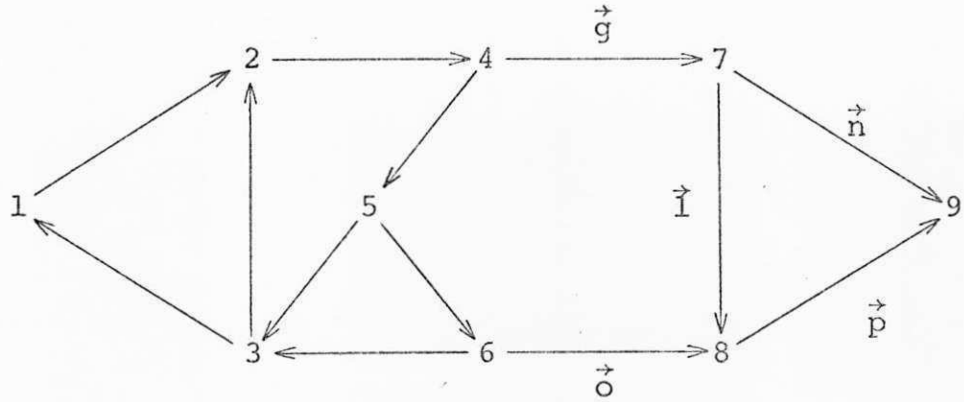
b.1) Passo 1: ES - Variação Minimal

Entrada: grafo G (figura 3.2)

Passo 1: A = {a,b,d,e,f,h,i,k}

u(a) = inf	c(a) = 10
u(b) = inf	c(b) = 10
u(d) = inf	c(d) = 10
u(e) = inf	c(e) = 10
u(f) = inf	c(f) = 10
u(h) = inf	c(h) = 10
u(i) = inf	c(i) = 10
u(k) = inf	c(k) = 10

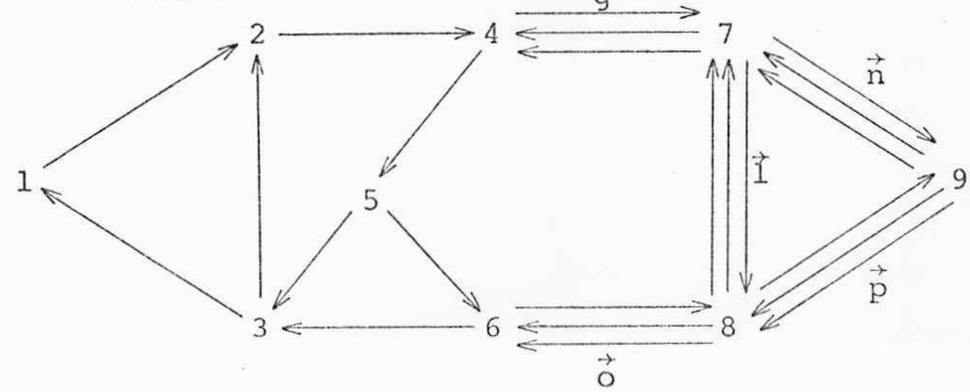
Passo 2: E = {g,l,m,o,p}



Passo 3:

NO	DEFICIT
1	0
2	1
3	0
4	-1
5	-1
6	-1
7	-1
8	1
9	2

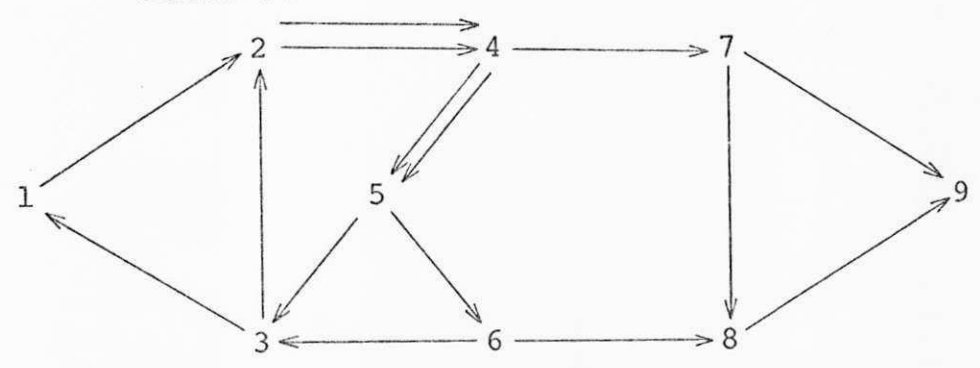
Passo 5:



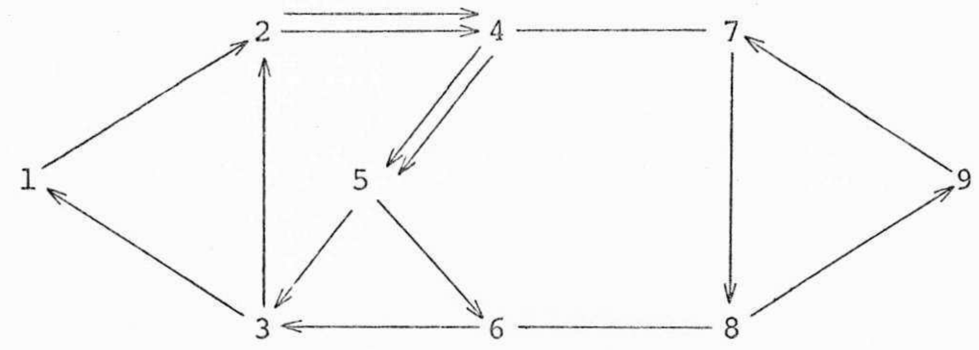
Passo 6: Saída do algoritmo do fluxo

ARCO	X
a	0
b	1
d	0
e	0
f	1
$\vec{g}$	0
$\overleftarrow{g}$	0
g'	1
h	0
i	0
k	0
$\vec{l}$	0
$\overleftarrow{l}$	0
l'	0
$\vec{m}$	0
$\overleftarrow{m}$	0
m'	2
$\vec{o}$	0
$\overleftarrow{o}$	0
o'	1
$\vec{p}$	0
$\overleftarrow{p}$	0
p'	0

Passo 7:



Passo 8 e Passo 9: Grafo resultante do algoritmo ES - Variação Minimal, que servirá como entrada para o passo 2.



b.2) GP - Variação Minimal

Entrada: Grafo G =

4 —— 7

6 —— 8

Passo 1:

NO	GRAU
4	1
6	1
7	1
8	1

$N' \subset N = \{4,6,7,8\}$

Passo 2:

MATRIZ DOS CUSTOS DAS CADEIAS DE MENOR CUSTO

	4	6	7	8
4	0	inf	10	inf
6	inf	0	inf	10
7	10	inf	0	inf
8	inf	10	inf	0

Passo 3: Casamento Ótimo

M

4 —— 7

7 —— 8

Passo 4: Grafo resultante do algoritmo Mixed 2A, com custo = 170

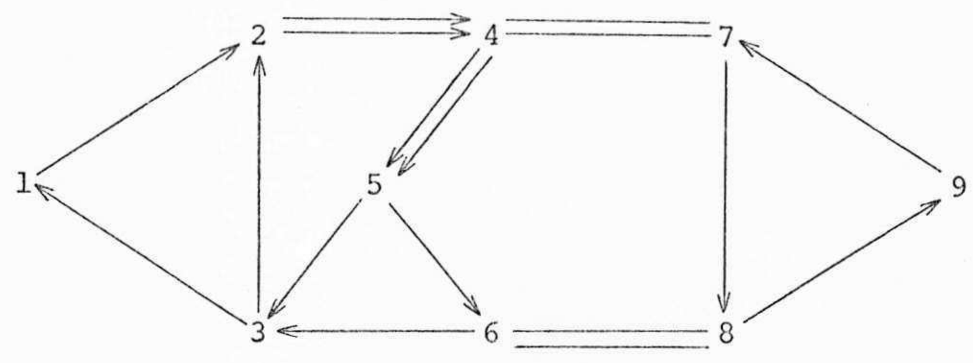


Figura 3.4

c) Passo 3: O grafo resultante do algoritmo Mixed 2A fornece o melhor resultado. O custo para percorrer o grafo resultante do algoritmo Mixed 1A é de 180, enquanto que para percorrer o grafo que resulta do algoritmo Mixed 2A, o custo é 170.

4.3. Determinação de uma Rota num Grafo de Euler

Serão apresentado agora os algoritmos que determinam uma rota num grafo de Euler [1].

Algoritmo que monta a Arvore Máxima

Esse algoritmo utiliza dois conjuntos:

- T: conjunto de nós que compõem a árvore;
- R: conjunto contendo o nó i0, e os nós que pertencem à árvore e que possuem ramos incidentes;
- A: conjunto dos arcos incidentes com um nó i;
- E: conjunto dos ramos incidentes com um nó i.

Algoritmos:

a) Subrotina Estende (k,T,R)

```
1: FILA ← {k}
2: WHILE (FILA ≠ Ø) DO
      Begin
3:       i ← Primeiro da FILA
4:       Elimine i da FILA
5:       WHILE (A(i) ≠ Ø) DO
6:         Begin
7:           Escolha um arco (h,i) ε  A(i)
```

```
8:        IF (h não é incidente com
              nenhum arco de T)
              THEN Begin
9:                    T ← T ∪ {(h,i)}
10:                   Adicione h na FILA
11:                   IF (E(h) ≠ Ø)
12:                      THEN  R ← R ∪ {h}
                   End
            End
     End
```

b) Árvore Máxima

```
1: FOR i ← 1, N DO
     Begin
2:     E(i) ← conjunto dos ramos incidentes
              com o nó i
3:     A(i) ← conjunto dos arcos incidentes
              com o nó i
     End
4: T ← Ø
5: R ← {i0}
6: ESTENDE (i0,T,R)
7: WHILE (Existir um r   R, com E(r) ≠ Ø) DO
     Begin
8:     R ← R \ {r}
9:     j ← r
10:    WHILE (existir k ≠ r, com {j,k} ε E(j)) DO
         Begin
11:        Elimine {j,k} de E(j) e E(k)
12:        IF (k não é incidente com algum arco em T)
             THEN Begin
13:                   T ← T ∪ {k,j}
14:                   IF (E(k) ≠ Ø)
15:                      THEN R ← R ∪ {k}
16:                   ESTENDE (k,T,R)
                   End
17:        j ← k
         End
   End
```

Algoritmo ROTA

A notação utilizada nesse algoritmo será a seguinte:

- -ARBOR(i): conjunto de arco ou ramo associado ao nó i;
- -RAMOS(i): conjunto de ramos incidentes com o nó i, excetuando-se o ramo pertencente à árvore, se existir;
- -ARCOS(i): conjunto de arcos que saem do nó i, exceto o que pertence à árvore, se existir.

c) Subrotina Montalista(i)

```
1: WHILE (exitir j  ε  RAMOS(i)) DO
     Begin
2:     LISTA(i) ← j
       RAMOS(i) ← RAMOS(i) \ {j}
     End
4:  WHILE (existir k  ε  ARCOS(i)) DO
      Begin
5:      LISTA(i) ← k
6:      ARCOS(i) ← ARCOS(i) \ {j}
      End
7:    IF (i ≠ nó inicial)
8:        THEN  LISTA(i) ← ARBOR(i)
```

d) Algoritmo Rotafinal

```
1: FOR i ← 1, N  DO
2:     MONTALISTA(i)
3: i ← nó inicial
4: WHILE(LISTA(i) ≠ Ø)  DO
    Begin
5:    ROTA ← Elemento (i,j) que encabeça a LISTA(i)
6:    Elimine (i,j) da LISTA(i)
7:    IF ((i,j) é ramo)
8:       THEN Elimine (j,i) da LISTA(i)
9:    i ← j
    End
10: Armazena a ROTA
```

Ex.: A rota determinada para o grafo da figura 3.4 seria a seguinte:

ROTA	
NÓ INICIAL	NÓ FINAL
1	2
2	4
4	7
7	4
4	5
5	6
6	8
8	9
9	7
7	8
8	6
6	3
3	2
2	4
4	5
5	3
3	1

4.4. Impressão da Rota

Objetivo: imprimir a ROTA de uma zona.

Entrada: ARQUIVO ROTA.

Saída: RELATÓRIO ROTA DE UMA ZONA.

Algoritmo:

```
1: Lê registro do ARQUIVO ROTA
2: CODRUA ← 001
3: WHILE (MATRIZ CHECK (CODRUA, NO INICIAL) = 1
          e MATRIZ CHECK (CODRUA, NO FINAL) = 1
4:    DO CODRUA ← CODRUA + 1
5: While (não é FDA)  DO
     Begin
6:     CODRUA ANTERIOR ← CODRUA
7:     Lê registro do ARQUIVO ROTA
8:     CODRUA ← 001
9:     WHILE (MATRIZ CHECK (CODRUA, NO INICIAL) = 1
              e MATRIZ CHECK (CODRUA, NO FIANL) = 1
10:       DO CODRUA ← CODRUA + 1
11:    IF (CODRUA ANTERIOR = CODRUA)
          THEN Begin
12:              Procura no arquivo CADASTRO DE RUAS os nomes
                 das ruas cujos códigos são CODRUA ANTERIOR e
                 CODRUA.
```

```
13:                 Imprime os nomes das ruas
                End
        End
```

CAPÍTULO V

5. IMPLEMENTAÇÃO DOS ALGORITMOS

Para a implementação dos algoritmos descritos nos capítulos II e IV, sugere-se a utilização das linguagens COBOL e FORTRAN IV. Essas linguagens são indicadas devido às suas características. A linguagem COBOL permite fácil acesso a arquivos, manipulação de grandes volumes de dados, e fornece facilidades para a crítica e consistência dos dados e impressão de relatórios. A linguagem FORTRAN IV por sua vez, é mais eficiente quando estão envolvidos no problema grande quantidade de cálculos. Além disso, essas duas linguagens permitem maior portabilidade ao sistema.

Sugere-se que os seguintes algoritmos sejam implementados em COBOL:

- CRÍTICA E ATUALIZAÇÃO DOS DADOS;
- IMPRIME A ROTA;

Esses algoritmos são bastante simples de implementar.

Os algoritmos que devem ser implementados em FORTAN IV são os seguintes:

- MONTA O GRAFO;
- MONTA A MATRIZ CHECK;
- VERIFICA ERROS GRAFO;
- MIXED A;
- ROTA.

Desses algoritmos, os mais complexos e de implementação mais difícil, são sem dúvida alguma MIXED A e ROTA, que envolvem grande quantidade de cálculos e estruturas complexas para a representação dos dados no computador.

Nesse trabalho foram implementados os algoritmos MIXED A e ROTA no computador IBM 370/145 do NPD/UFPb, na linguagem FORTRAN IV.

Nesse capítulo são descritas as estruturas de dados utilizadas na implementação desses algoritmos, são apresentadas algumas aplicações com dados e é efetuado uma análise dos resultados obtidos.

5.1. Descrição das Estruturas de Dados Escolhidas

Para a representação dos grafos no computador optou-se pela utilização de LISTAS DE ADJACÊNCIA, devido ao grande número de operações sofridas pelos grafos através dos algoritmos, até a obtenção da solução desejada. Essa estrutura mostrou-se mais adequada ao problema, permitindo um bom aproveitamento da memória, fácil operação com os grafos, e possibilitou a obtenção de bons resultados também em termos de tempo de computação.

Cada nodo da lista de adjacência terá o seguinte formato:

NOFIM	CUSTO	DIREC	ENCAD

Na implementação os nodos foram armazenados na forma de vetores, assim:

- PONTNO: vetor de N posições (N é o número de nós do grafo), cujos índices i representam os nós, e cada elemento j = PONTNO(i), é um apontador para a posição onde se encontram as informações sobre o primeiro arco incidente com o nó i. PONTNO(1) indica a posição do primeiro arco incidente com o nó 1;

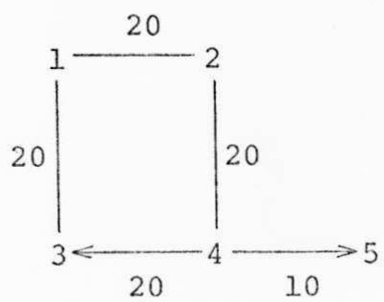
- NOFIM: vetor onde cada elemento NOFIM(j) representa um nó adjacente com um nó i, formando o arco (i, NOFIM (i)).

- CUSTO: vetor onde cada elemento CUSTO(j) representa o custo para percorrer o arco (i, NOFIM(j)).

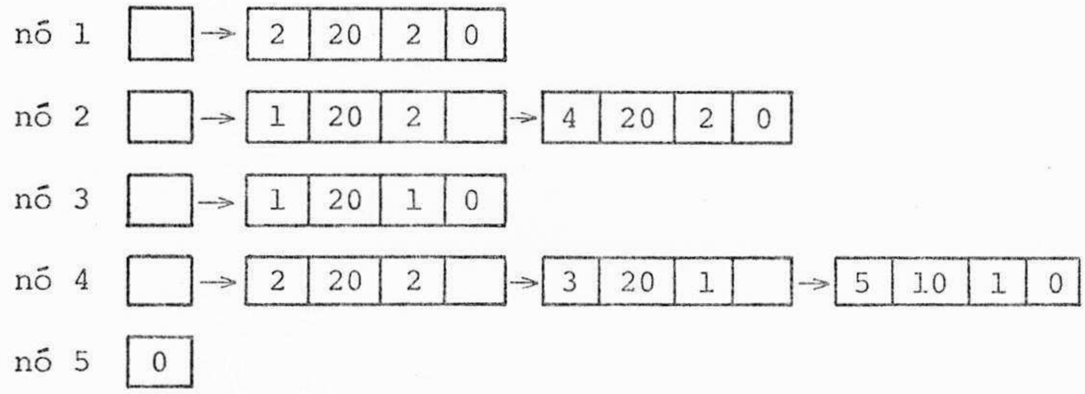
- DIREC: vetor onde cada elemento DIREC(j) indica a direção (sentido) do arco (i, NOFIM(j)). DIREC(j) = 1 indica um arco, DIREC(j) = 2 um ramo.

- ENCAD: vetor onde cada elemento ENCAD(j) indica a posição do próximo arco incidente com o nó i. ENCAD(j) = 0 indica fim da lista do nó i.

Ex.:



LISTA DOS NÓS



Representação Tabular dessa estrutura

NÓ	PONTNO	NOFIM	CUSTO	DIREC	ENCAD
1	1	2	20	2	0
2	2	1	20	2	3
3	4	4	20	2	0
4	5	1	20	1	0
5	0	2	20	2	6
6		3	20	1	7
7		5	10	1	0

Deve-se entender essa estrutura da seguinte forma: NÓ indica o número do nó. Assim, as informações sobre o primeiro arco incidente com o nó 1 encontram-se na posição 1 (PONTNO (1) = 1). Então tem-se que o nó final desse arco é o 2 (NOFIM (1) = 2), com custo 20 (CUSTO(1) = 20). DIREC(1) = 2 indica sentido duplo, isto é, indica que é um ramo. Finalmente ENCAD(1)= 0 indica o final da lista do nó i.

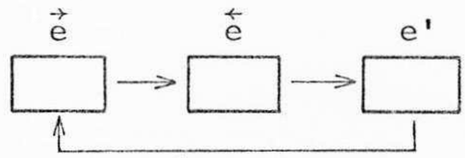
O nó 4 tem um arco na posição 5 (PONTNO(4) = 5). NOfim(5) = 2 indica que esse é o ramo {4,2}. Sabe-se que é um ramo porque DIREC(5) = 2, e que tem custo 20 porque CUSTO(5)=20. Finalmente, ENCAD(5) = 6 indica que o próximo arco incidente com o nó 4 encontra-se na posição 6, e assim por diante.

O grafo inicial terá N nós e r arcos.

Armazenou-se a parte triangular superior da matriz dos custos das cadeia de menor custo c(i,j) coluna por coluna , num vetor denominado CC com N(N-1)/2 posições, da seguinte forma:

c(1,2), c(1,3), c(2,3), c(1,4), c(2,4), c(2,3), ...

Para a implementação do algoritmo ES - Variação Minimal, foram incluídos mais três campos em cada nodo da lista de adjacência: capacidade (U), fluxo (X) e um novo campo (ELIN) para encadear os arcos $\vec{e}$, $\overleftarrow{e}$ e e' da seguinte forma:



Assim, o esquema dos nodos da lista de adjacência para esse algoritmo passa a ser o seguinte:

NOF	CUSTO	DIREC	U	X	ENCAD	ELIN

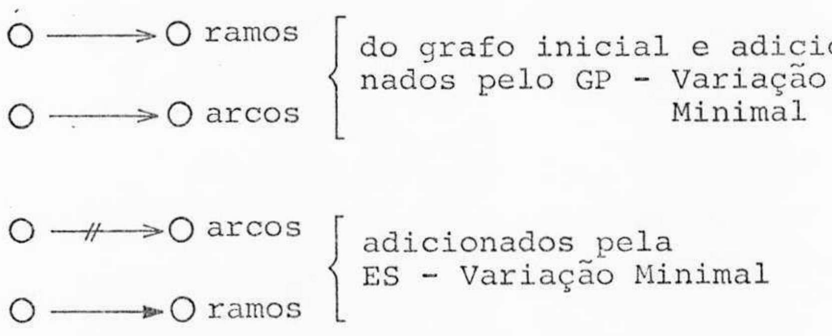
5.2. Aplicações

Para verificar a eficiência da implementação dos algoritmos MIXED A e ROTA, foram utilizados dados reais pertencentes às cidades de Fortaleza e Aracaju.

As aplicações foram realizadas levando-se em consideração apenas as distâncias dos trechos.

Para cada uma das aplicações são apresentados o grafo inicial, os grafos resultantes dos algoritmos MIXED 1A e MIXED 2A, os tempos gastos no processamento e as rotas que foram determinadas.

A seguinte simbologia é utilizada nos diagramas dos grafos:



Com a solução do problema de determinação de rotas de custo mínimo é obtida através de algoritmos heurísticos, realizou-se uma análise desses resultados em relação a solução ótima do problema.

Estimativas de Frederickson [13] mostram que no pior caso, o custo de uma rota encontrada através da utilização dos algoritmos MIXED 1 e MIXED 2 é, teoricamente, menor ou igual a 5/3 de uma rota ótima. Os algoritmos usados nesse trabalho são uma variante de MIXED 1 e MIXED 2, para os quais as estimativas de Frederickson também são válidas.

Esse limite de 5/3 é teórico, e sua validade em aplicações práticas ainda não está provada, embora os resultados obtidos através de aplicação com dados reais sejam geralmente menores que esse limite. Esse fato pode ser comprovado pelos resultados obtidos nesse trabalho.

Como já foi visto, a condição para a existência de uma rota de Euler num grafo misto é que todos os nós do grafo possuam grau par e grau de entrada igual ao grau de saída. Portanto, uma rota é obtida através da combinação entre uma GP - Variação Minimal e uma ES - Variação Minimal no grafo inicial.

Apesar da solução obtida ser heurística, é possóvel determinar um intervalo de variação para a solução ótima do problema da seguinte forma:

Define-se:

PROBLEMAS:

P = Obter um grafo, tal que todos os seus nós possuam grau par e grau de entrada igual ao grau de saída;

Pgp = Obter um grafo, tal que todos os seus nós possuam grau par;

Pes = Obter um grafo, tal que todos os seus nós possuam grau de entrada igual ao grau de saída;

CUSTOS:

Cotm = Custo da solução ótima do problema;

Cheu = Custo de solução heurística obtida;

Cgp = Custo da solução da GP - Variação Minimal;

Ces = Custo da solução da ES - Variação Minimal;

Tem-se que:

1) Cotm $\leqslant$ Cheu

2)

SOLUÇÃO DO PROBLEMA	ALGORITMO EFICIENTE	CUSTO DA SOLUÇÃO
P	Não existe	Cheu
Pgp	GP-Var.Min	Cgp
Pes	ES-Var.Min	Ces

3) A solução ótima é uma solução possível para o Pgp, então:

$$Cgp \leq Cotm$$

4) A solução ótima é uma solução possível para o Pes, então:

$$Ces \leq Cotm$$

Portanto, o limite de variação para a solução ótima é definido por:

$$(MAX\ \{Cgp,\ Ces\}) \leq Cotm \leq Cheu$$

Pode-se assim estabelecer a seguinte razão:

$$\frac{Cheu}{Cotm} \leq \frac{Cheu}{(MAX\{Cgp, Ces\})}$$

5.3. Resultados Obtidos

5.3.1. Zona 001

Grafo inicial: 32 nós, 61 arcos.

MIXED 1A

Custo do grafo inicial	= 560
Aumento no custo pela GP - Var. Min.	= 100
Cgp	= 600
Aumento do custo pela ES - Var. Min.	= 280
Custo do grafo MIXED 1A	= 940

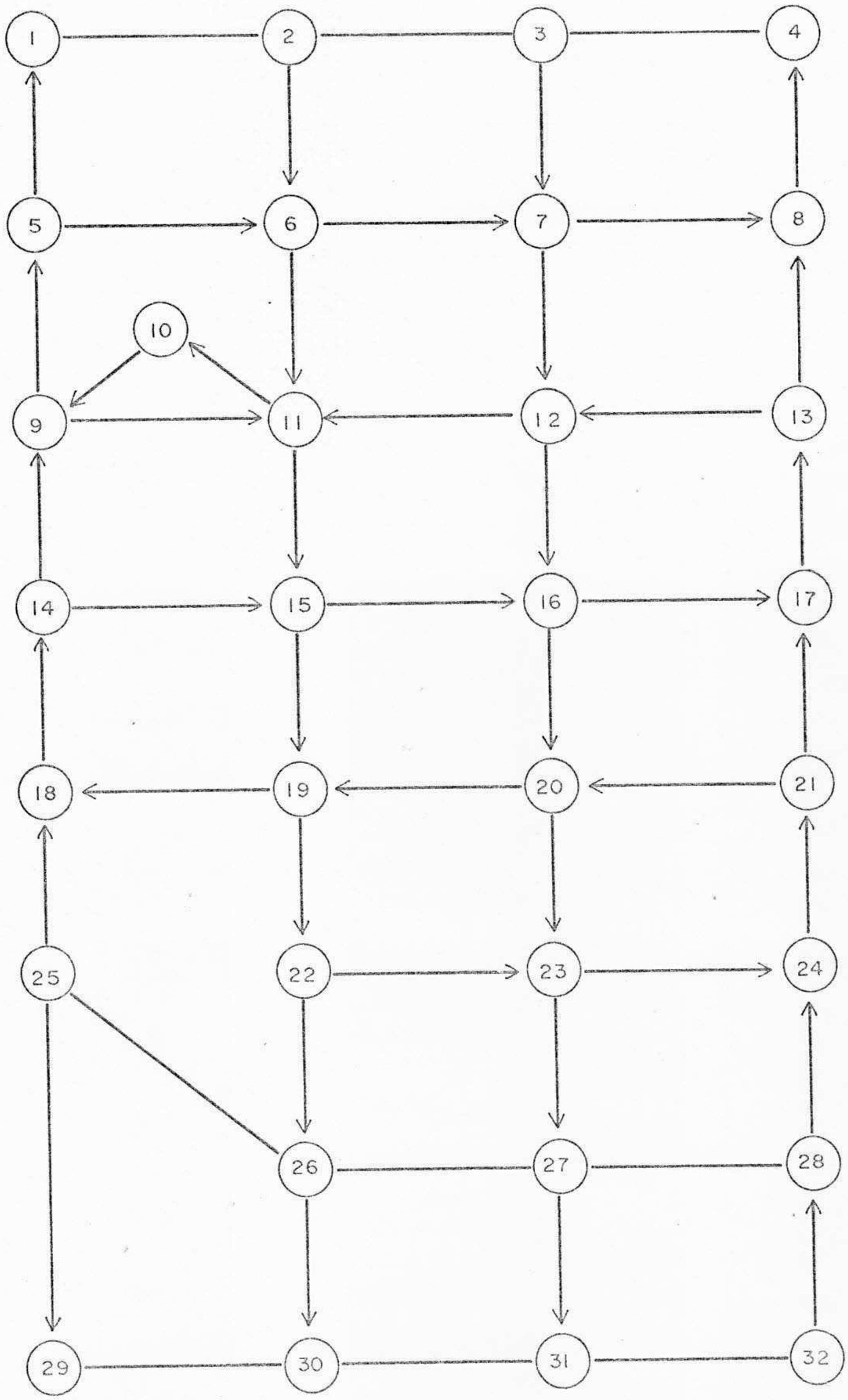
MIXED 2A

Custo do grafo inicial	= 560
Aumento no custo ES - Var. Min.	= 220
Ces	= 780
Aumento no custo pela GP - Var. Min.	= 0
Custo do grafo MIXED 2A	= 780

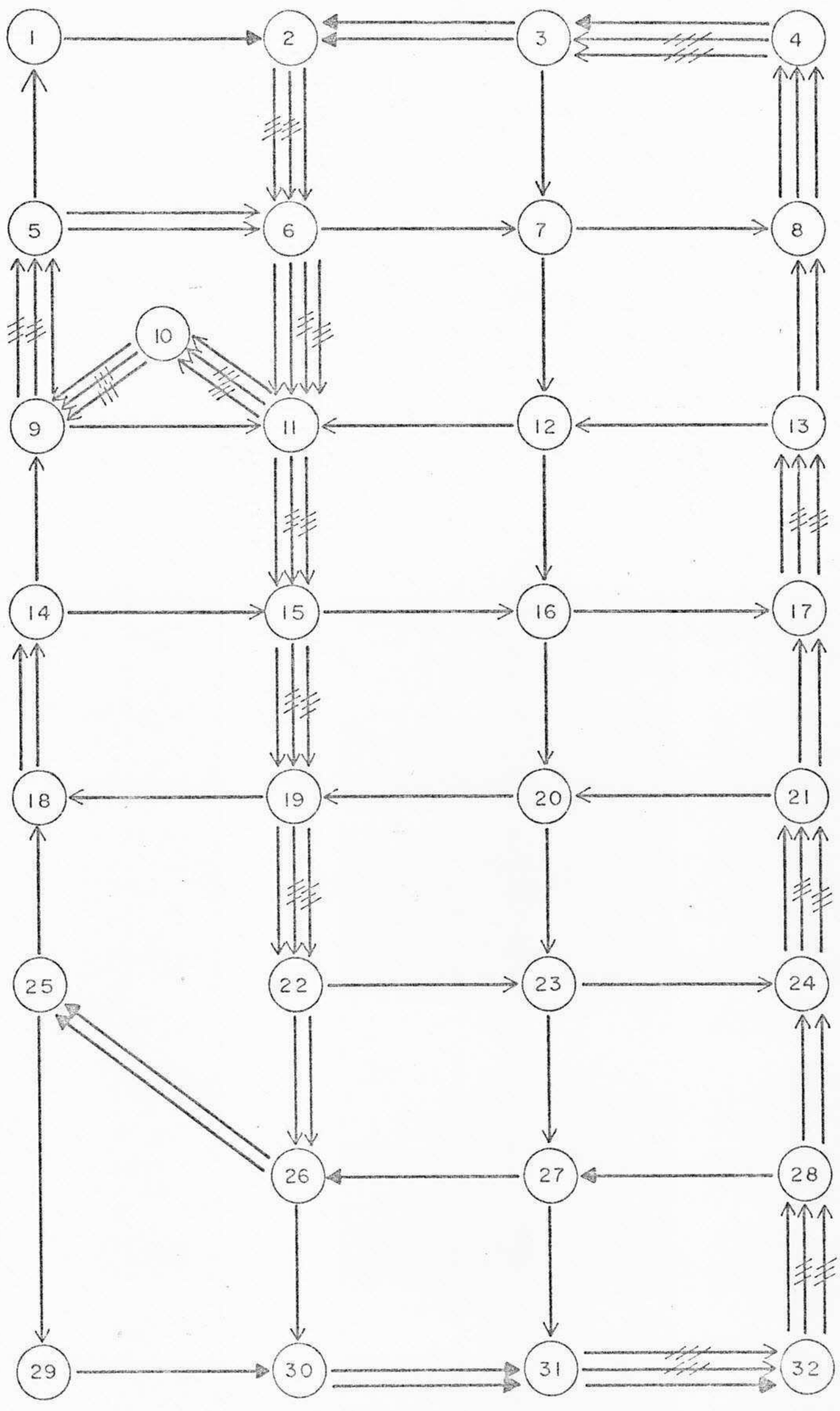
O algoritmo MIXED 2A fornece a melhor solução.
Limites de variação da solução ótimo:
Cheu = 780 Cgp = 660 Ces = 780
MAX {660, 780} = 780
$780 \leqslant Cotm \leqslant 780 \rightarrow Cotm = 780$
Portanto a solução encontrada em MIXED 2A é ótima.

TEMPO DE PROCESSAMENTO

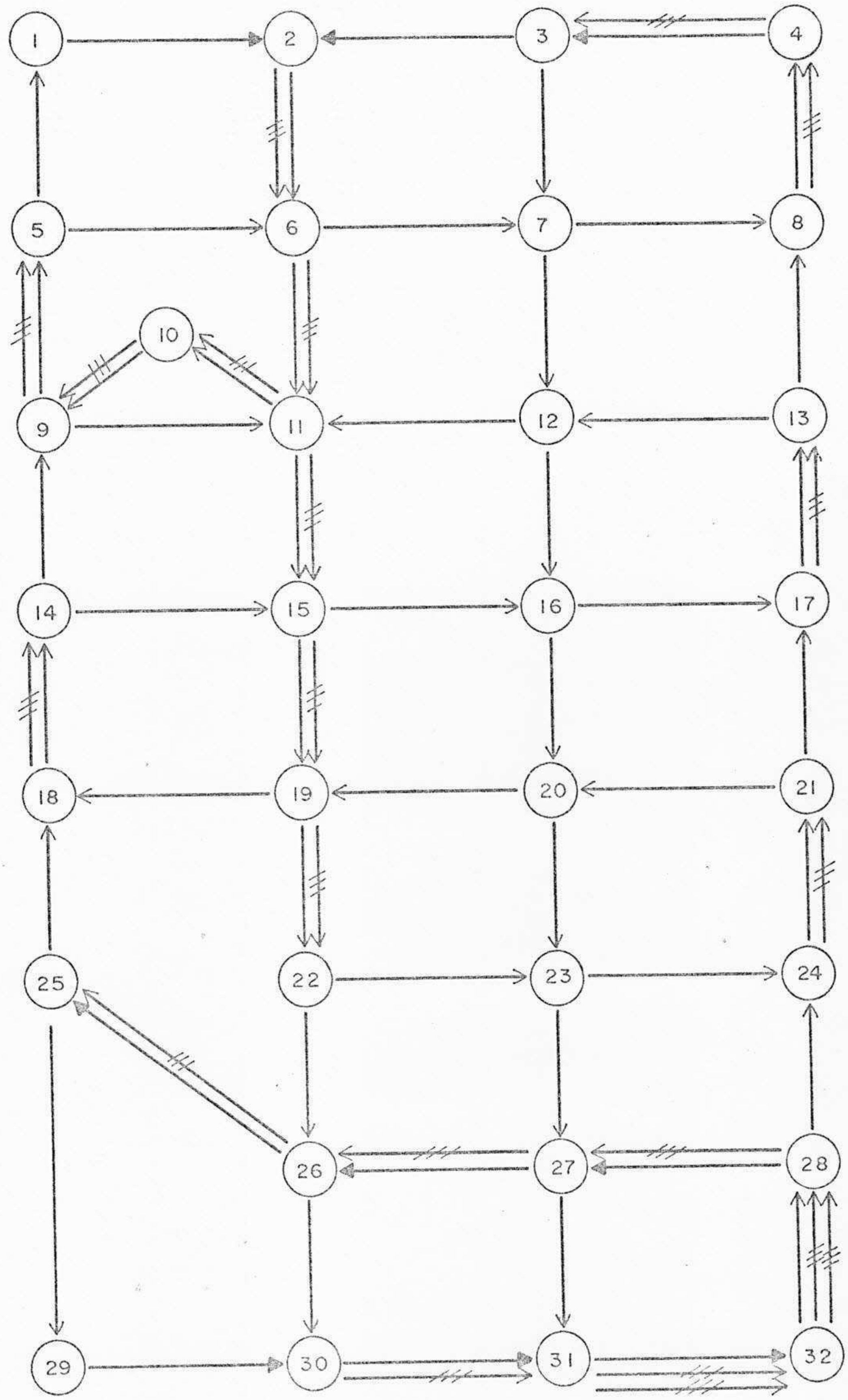
MIXED 1A	= 5,48 seg
MIXED 2A	= 2,03 seg
ROTA	= 1,26 seg
TOTAL	= 8,77 seg



ZONA 001 - GRAFO INICIAL



RESULTADO FORNECIDO PELO "MIXED 1 A"



RESULTADO FORNECIDO PELO "MIXED 2A"

Zona 001 - Rota fornecida para o grafo resultante do "MIXED 2A".

	Nó inicial	Nó final
1	1	2
2	2	6
3	6	7
4	7	8
5	8	4
6	4	3
7	3	7
8	7	12
9	12	16
10	16	17
11	17	13
12	13	8
13	8	4
14	4	3
15	3	2
16	2	6
17	6	11
18	11	10
19	10	9
20	9	5
21	5	6
22	6	11
23	11	15
24	15	16
25	16	20
26	20	23
27	23	24
28	24	21
29	21	17
30	17	13
31	13	12
32	12	11
33	11	15
34	15	19
35	19	22
36	22	23

	Nó inicial	Nó final
37	23	27
38	27	26
39	26	25
40	25	29
41	29	30
42	30	31
43	31	32
44	32	28
45	28	24
46	24	21
47	21	20
48	20	19
49	19	22
50	22	26
51	26	30
52	30	31
53	31	32
54	32	28
55	28	27
56	27	31
57	31	32
58	32	28
59	28	27
60	27	26
61	26	25
62	25	18
63	18	14
64	14	15
65	15	19
66	19	18
67	18	14
68	14	9
69	9	11
70	11	10
71	10	9
72	9	5
73	5	1

5.3.2. Zona 002

Crafo inicial: 62 nós, 187 arcos.

MIXED 1A

Custo do Grafo Inicial	= 7.564
Aumento no custo pela GP - Var. Min.	= 1.308
Cgp	8.872
Aumento no custo pela ES - Var. Min.	632
Custo do grafo MIXED 1A	9.504

MIXED 2A

Custo do Grafo Inicial	= 7.564
Aumento no custo pela ES - Var. Min.	= 602
Ces	= 8.166
Aumento no custo pela GP - Var. Min.	= 3.061
Custo do grafo MIXED 2A	=11.227

O algoritmo MIXED 1A fornece a melhor solução.
Limites de variação da solução ótima:

Cheu = 9.504 Cgp = 8.872 Ces = 8.166

MAX {Cgp, Ces} = 8.872

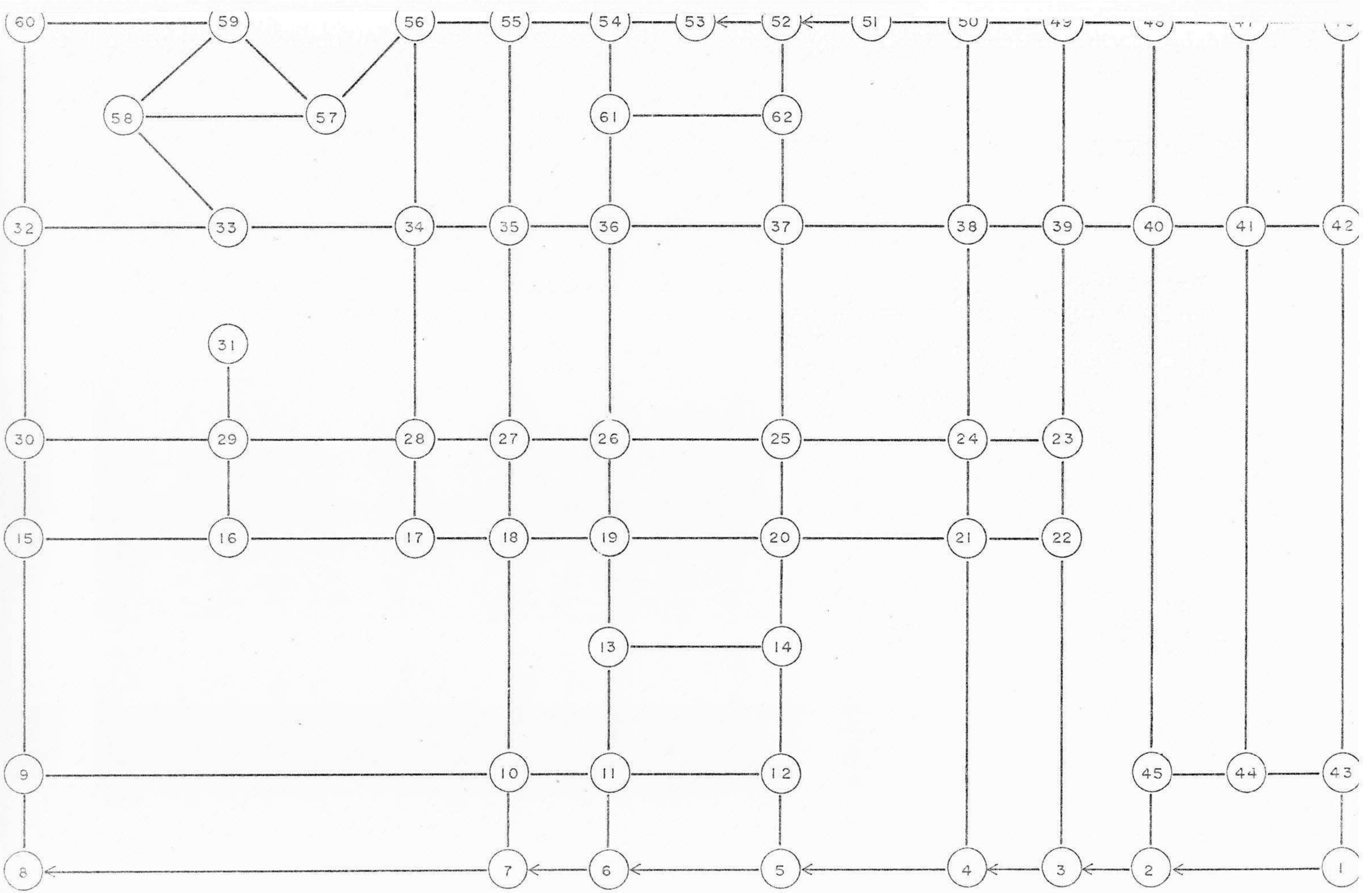
$8.872 \leqslant \text{Cotm} \leqslant 9.504$

$$\frac{\text{Cheu}}{\text{Cgp}} = 1.071$$

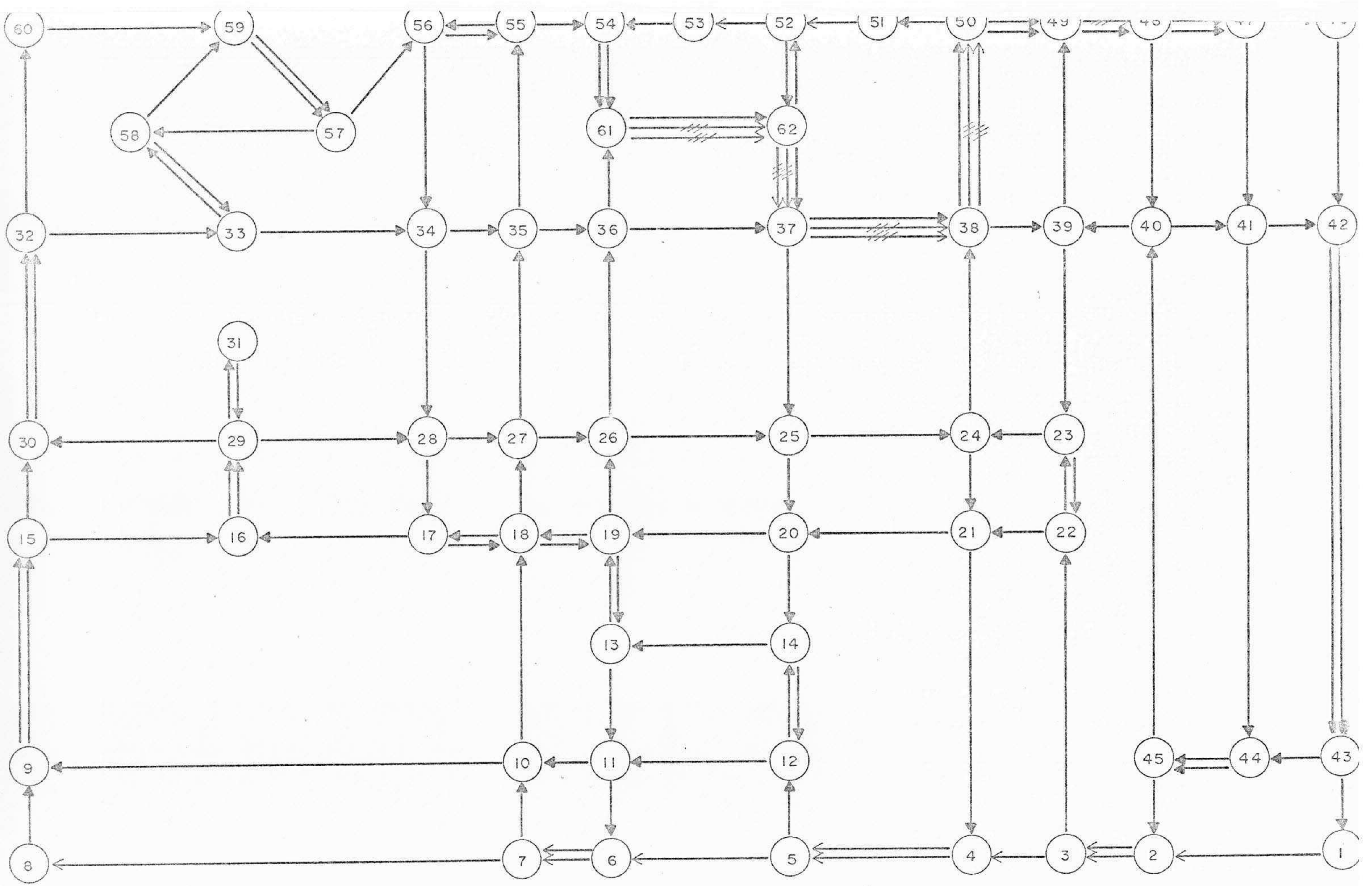
Portanto, a solução obtida é teoricamente, no máximo 7,1% pior que a solução ótima.

TEMPO DE PROCESSAMENTO

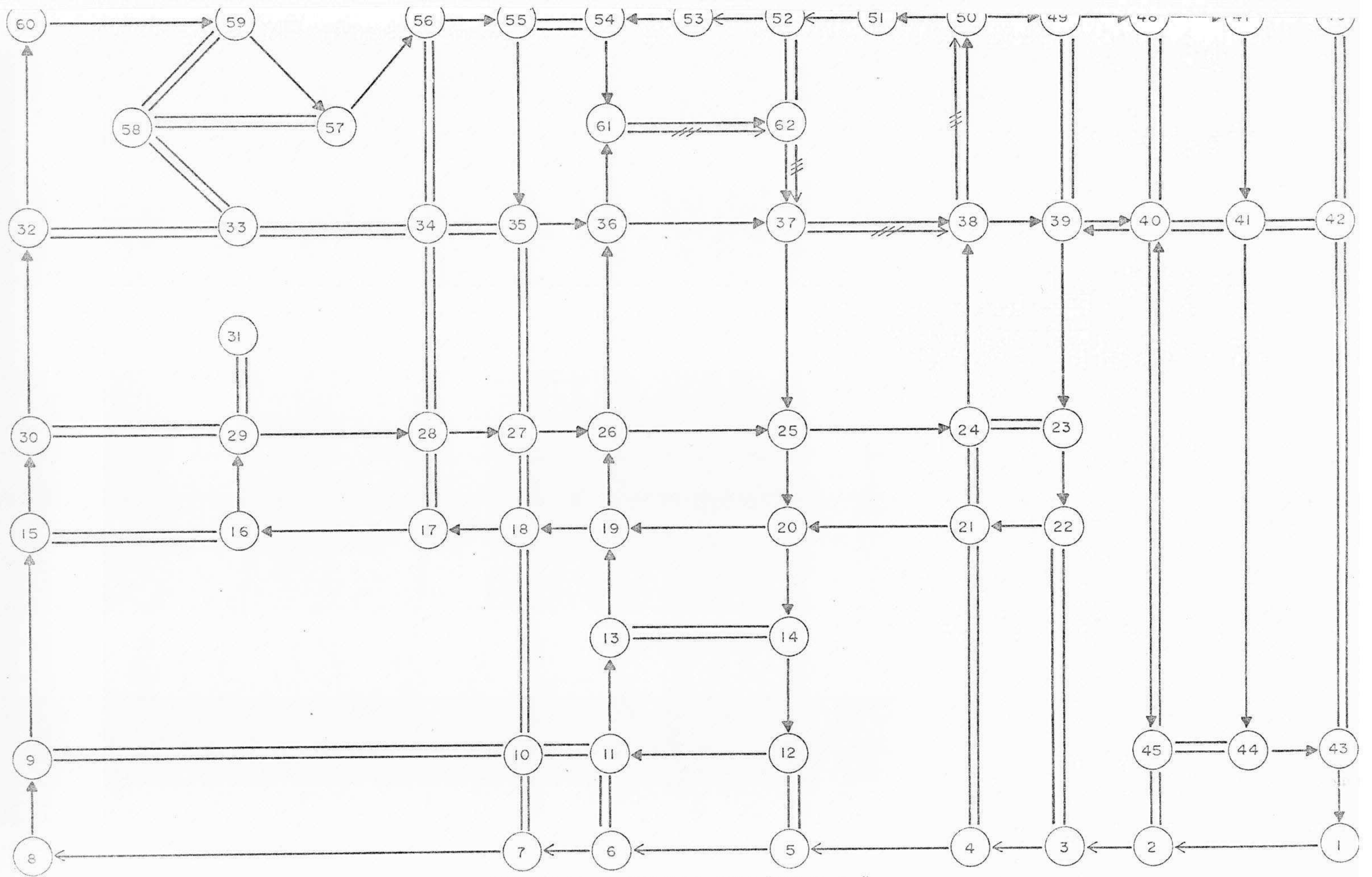
MIXED 1A	=	34,91	seg
MIXED 2A	=	32,79	seg
ROTA	=	3,81	seg
TOTAL	=	71,51	seg



ZONA 002 - GRAFO INICIAL



RESULTADO FORNECIDO PELO "MIXED 1 A"



RESULTADO FORNECIDO PELO "MIXED 2A"

Zona 002 - Rota fornecida para o grafo resultante do "MIXED 1A".

	Nō inicial	Nō final
1	1	2
2	2	3
3	3	4
4	4	5
5	5	6
6	6	7
7	7	8
8	8	9
9	9	15
10	15	30
11	30	32
12	32	60
13	60	59
14	59	57
15	57	58
16	58	59
17	59	57
18	57	56
19	56	55
20	55	56
21	56	34
22	34	28
23	28	17
24	17	16
25	16	29
26	29	30
27	30	32
28	32	33
29	33	58
30	58	33
31	33	34
32	34	35
33	35	55
34	55	54
35	54	61
36	61	62
37	62	37

	Nõ inicial	Nõ final
38	37	25
39	25	20
40	20	14
41	14	12
42	12	11
43	11	6
44	6	7
45	7	10
46	10	9
47	9	15
48	15	16
49	16	29
50	29	31
51	31	29
52	29	28
53	28	27
54	27	35
55	35	36
56	36	61
57	61	62
58	62	37
59	37	38
60	38	50
61	50	49
62	49	48
63	48	47
64	47	46
65	46	42
66	42	43
67	43	44
68	44	45
69	45	2
70	2	3
71	3	22
72	22	21
73	21	4
74	4	5
75	5	12
76	12	14

	Nó inicial	Nó final
77	14	13
78	13	11
79	11	10
80	10	18
81	18	17
82	17	18
83	18	27
84	27	26
85	26	36
86	36	37
87	37	38
88	38	50
89	50	51
90	51	52
91	52	53
92	53	54
93	54	61
94	61	62
95	62	52
96	52	62
97	62	37
98	37	38
99	38	50
100	50	49
101	49	48
102	48	47
103	47	41
104	41	44
105	44	45
106	45	40
107	40	39
108	39	23
109	23	22
110	22	23
111	23	24
112	24	21
113	21	20
114	20	19
115	19	13

	Nõ inicial	Nõ final
116	13	19
117	19	18
118	18	19
119	19	26
120	26	25
121	25	24
122	24	38
123	38	39
124	39	49
125	49	48
126	48	40
127	40	41
128	41	42
129	42	43
120	43	1

5.3.3. Zona 003

Grafo inicial: 104 nós, 300 arcos.

MIXED 1A

Custo do grafo inicial	=	14.407
Aumento do custo pela GP - Var. Min.	=	1.398
Cgp	=	15.805
Aumento no custo pela ES - Var. Min.	=	1.467
Custo do grafo MIXED 1A	=	17.272

MIXED 2A

Custo do grafo inicial	=	14.407
Aumento no custo pela ES - Var. Min.	=	857
Ces	=	15.264
Aumento no custo pela GP - Var. Min.	=	3.962
Custo do grafo MIXED 1A	=	19.226

O algoritmo MIXED 1A fornece a melhor solução.
Limites de variação da solução ótima:

Cheu = 17.272   Cgp = 15.805   Ces = 15.264

MAX = {Cgp, Ces} = 15.805

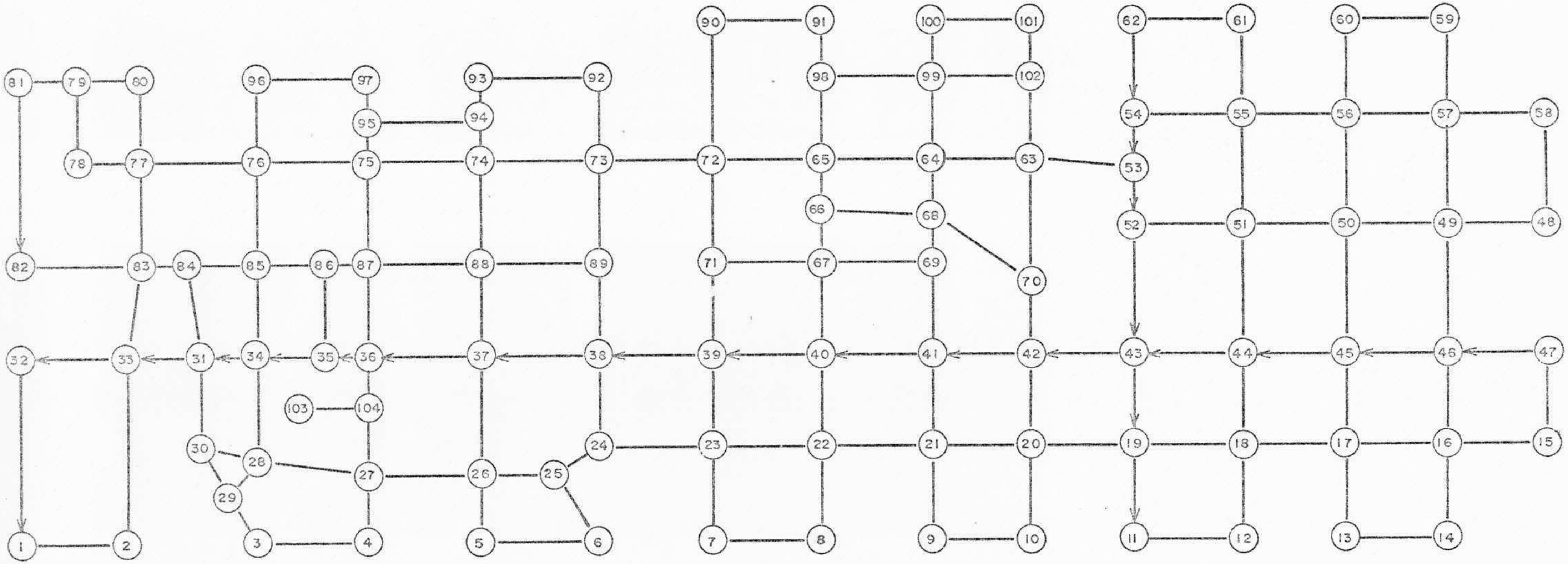
$15.805 \leqslant Cotm \leqslant 17.272$

$$\frac{Cheu}{Cgp} = 1.093$$

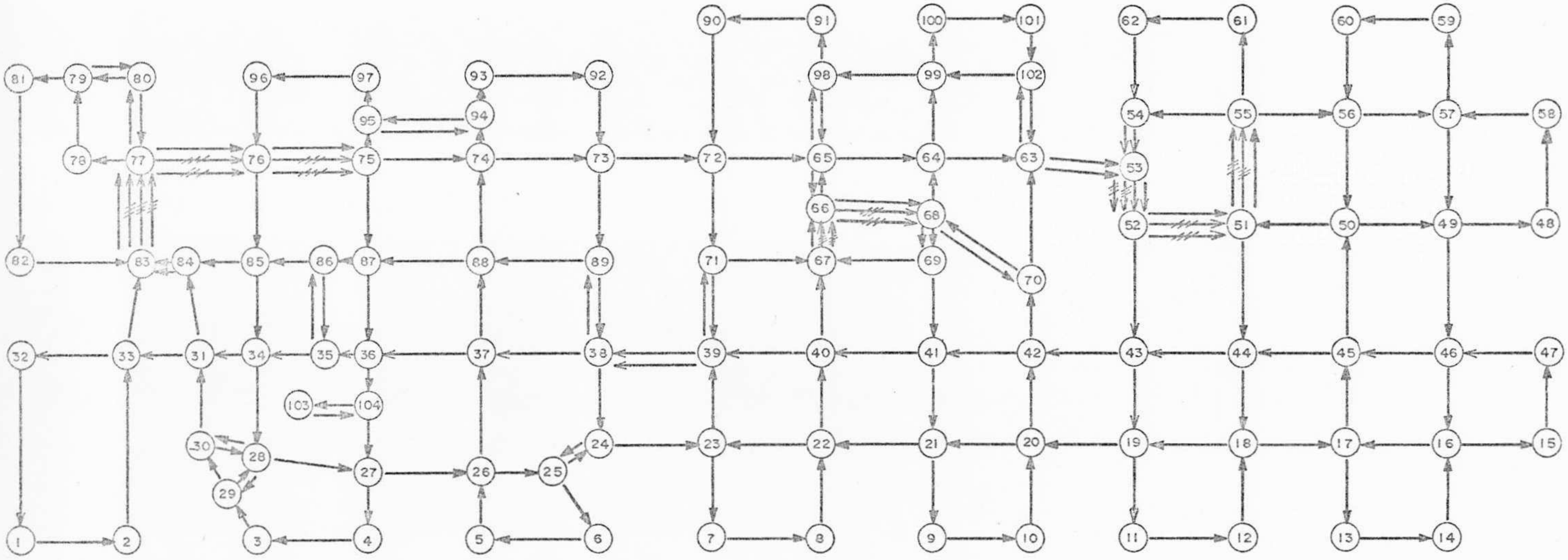
Portanto, a solução obtida é teoricamente, no máximo 9,3% pior que a solução ótima.

TEMPO DE PROCESSAMENTO

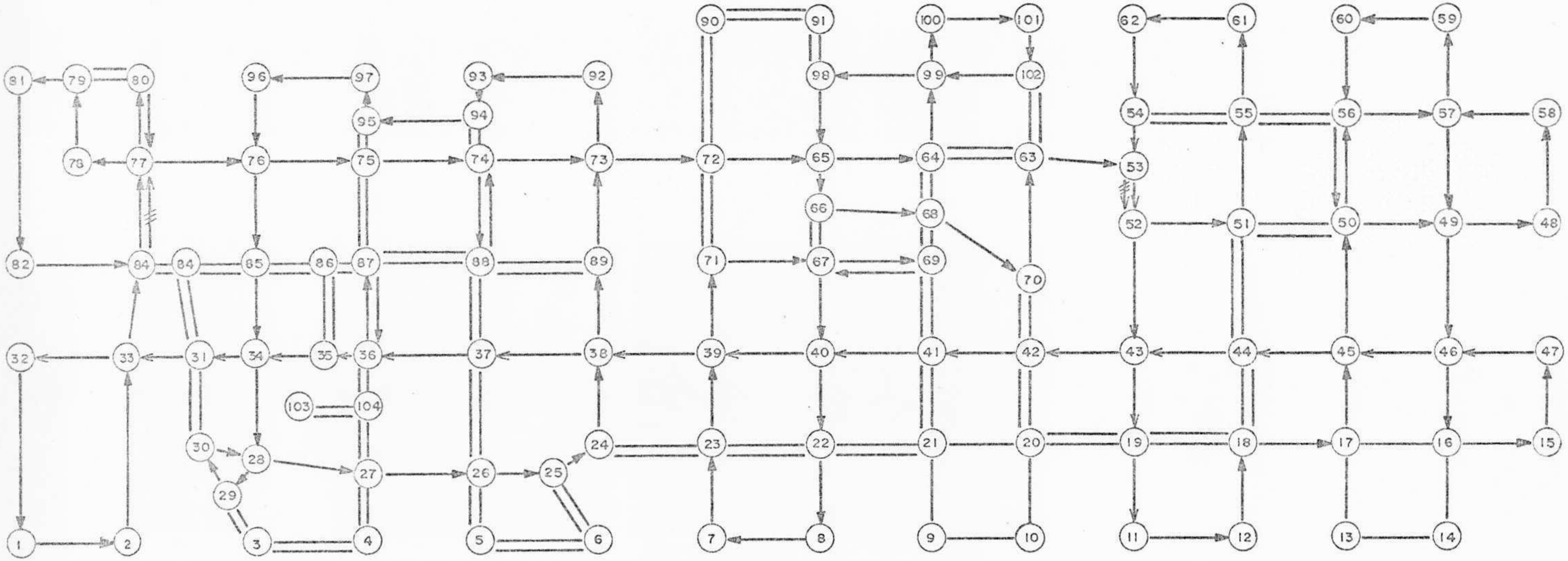
MIXED 1A	=	72,67	seg
MIXED 2A	=	65,12	seg
ROTA	=	4,02	seg
TOTAL	=	141,81	seg



ZONA 003 - GRAFO INICIAL



RESULTADO FORNECIDO PELO "MIXED 1 A"



RESULTADO FORNECIDO PELO "MIXED 2A"

Zona 003 - Rota fornecida para o grafo resultante do "MIXED 1A".

	Nō inicial	Nō final
1	1	2
2	2	33
3	33	83
4	83	77
5	77	76
6	76	75
7	75	74
8	74	94
9	94	93
10	93	92
11	92	73
12	73	72
13	72	65
14	65	64
15	64	63
16	63	53
17	53	52
18	52	51
19	51	55
20	55	56
21	56	57
22	57	59
23	59	60
24	60	56
25	56	50
26	50	49
27	49	48
28	48	58
29	58	57
30	57	49
31	59	46
32	46	16
33	16	15
34	15	47
35	47	46
36	46	45

	Nó inicial	Nó final
37	45	50
38	50	51
39	51	55
40	55	61
41	61	62
42	62	54
43	54	53
44	53	52
45	52	51
46	51	55
47	55	54
48	54	53
49	53	52
50	52	51
51	51	44
52	44	18
53	18	17
54	17	13
55	13	14
56	14	16
57	16	17
58	17	45
59	45	44
60	44	43
61	43	19
62	19	11
63	11	12
64	12	18
65	18	19
66	19	20
67	20	42
68	42	70
69	70	63
70	63	102
71	102	63
72	63	53
73	53	52
74	52	43
75	43	42

	Nó inicial	Nó final
76	42	41
77	41	21
78	21	9
79	9	10
80	10	20
81	20	21
82	21	22
83	22	40
84	40	67
85	67	66
86	66	65
87	65	98
88	98	65
89	65	66
90	66	68
91	68	64
92	64	99
93	99	100
94	100	101
95	101	102
96	102	99
97	99	98
98	98	91
99	91	90
100	90	72
101	72	71
102	71	67
103	67	66
104	66	68
105	68	69
106	69	67
107	67	66
108	66	68
109	68	70
110	70	68
111	68	69
112	69	41
113	41	40
114	40	39

	Nō inicial	Nō final
115	39	38
116	38	24
117	24	25
118	25	24
119	24	23
120	23	7
121	7	8
122	8	22
123	22	23
124	23	39
125	29	71
126	71	39
127	39	38
128	38	89
129	89	88
130	88	74
131	74	73
132	73	89
133	89	38
134	38	37
135	37	88
136	88	87
137	87	86
138	86	85
139	85	84
140	84	83
141	83	77
142	77	76
143	76	75
144	75	95
145	95	94
146	94	95
147	95	97
148	97	96
149	96	76
150	76	75
151	75	87
152	87	36
153	36	104

	Nõ inicial	Nõ final
154	104	103
155	103	104
156	104	27
157	27	26
158	26	25
159	25	6
160	6	5
161	5	26
162	26	37
163	37	36
164	36	35
165	35	86
166	86	35
167	35	34
168	34	28
169	28	27
170	27	4
171	4	3
172	3	29
173	29	28
174	28	29
175	29	30
176	30	28
177	28	30
178	30	31
179	31	84
180	84	83
181	83	77
182	77	78
183	78	79
184	79	81
185	81	82
186	82	83
187	83	77
188	77	80
189	80	79
190	79	80
191	80	77
192	77	76

	Nō inicial	Nō final
193	76	85
194	85	34
195	34	31
196	31	33
197	33	32
198	32	1

5.3.4. Zona 004

Grafo inicial: 143 nós, 283 arcos.

MIXED 1A

Custo do grafo incial	=	14.237
Aumento no custo pela GP - Var. Min.	=	1.781
Cgp	=	16.018
Aumento no custo pela ES - Var. Min.	=	7.206
Custo do grafo MIXED 1A	=	23.224

MIXED 2A

Custo do grafo inicial	=	14.237
Aumento no custo pela ES - Var. Min.	=	5.289
Ces	=	19.526
Aumento no custo pela GP - Var. Min.	=	761
Custo do grafo MIXED 2A	=	20.287

O algoritmo MIXED 2A fornece a melhor solução.
Limites de variação da solução ótima:

Cheu = 20.287 Cgp = 16.018 Ces = 19.526

MAX {Cgp, Ces} = 19.526

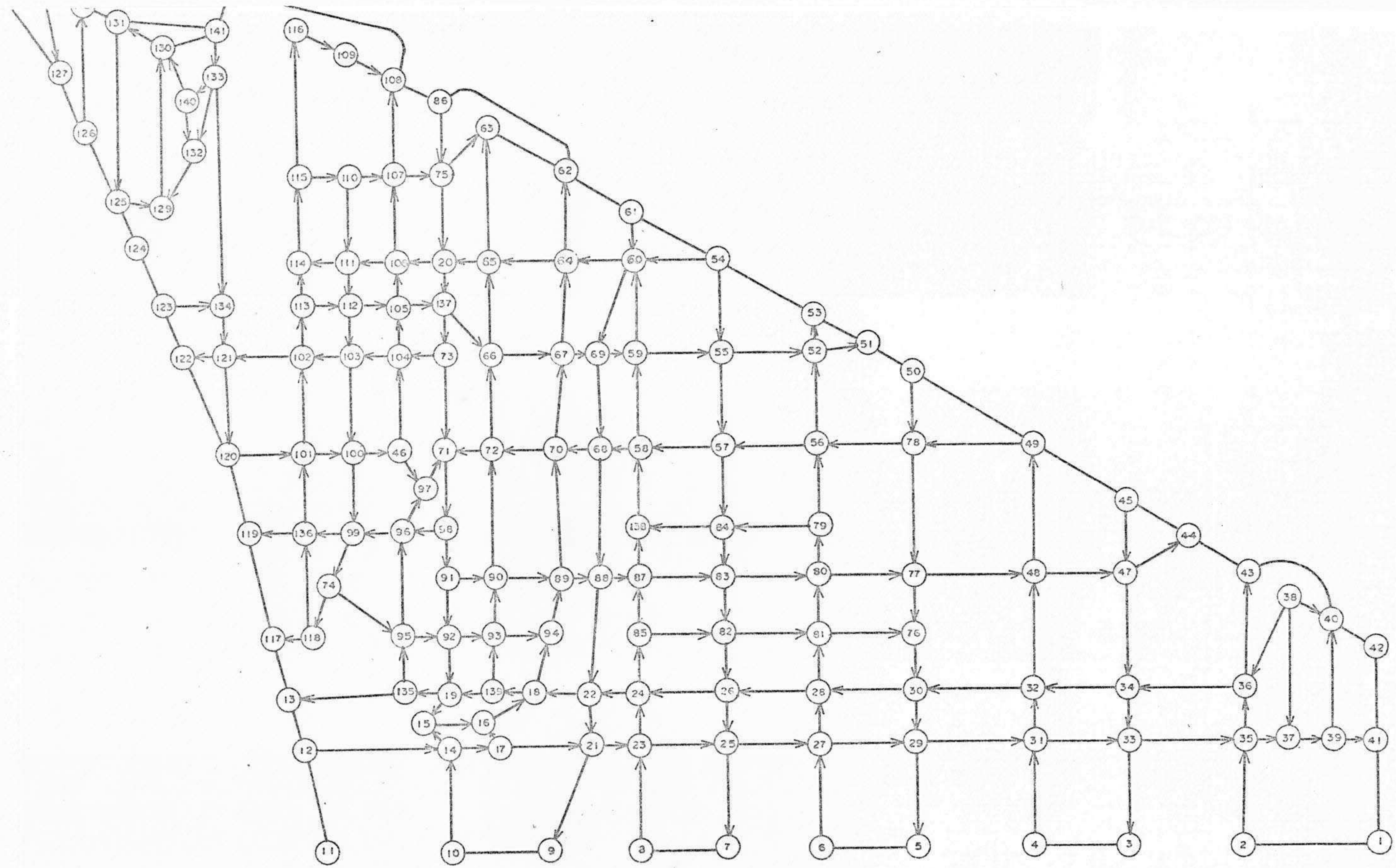
$19.526 \leqslant \text{Cotm} \leqslant 20.287$

$\frac{\text{Cheu}}{\text{Ces}} = 1.039$

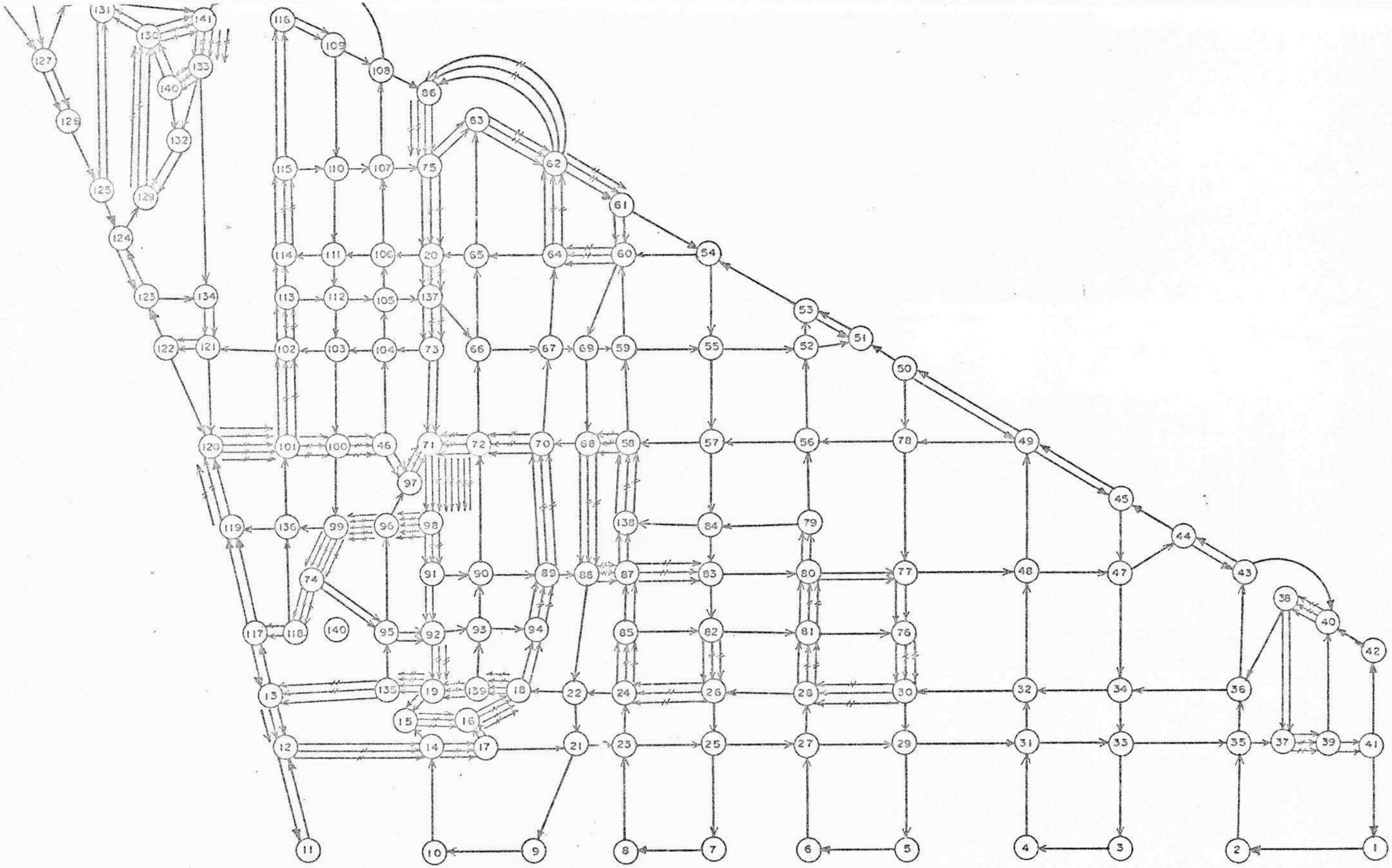
Portanto a melhor solução encontrada é teoricamente, no máximo 3,9% pior que a solução ótima.

TEMPO DE PROCESSAMENTO

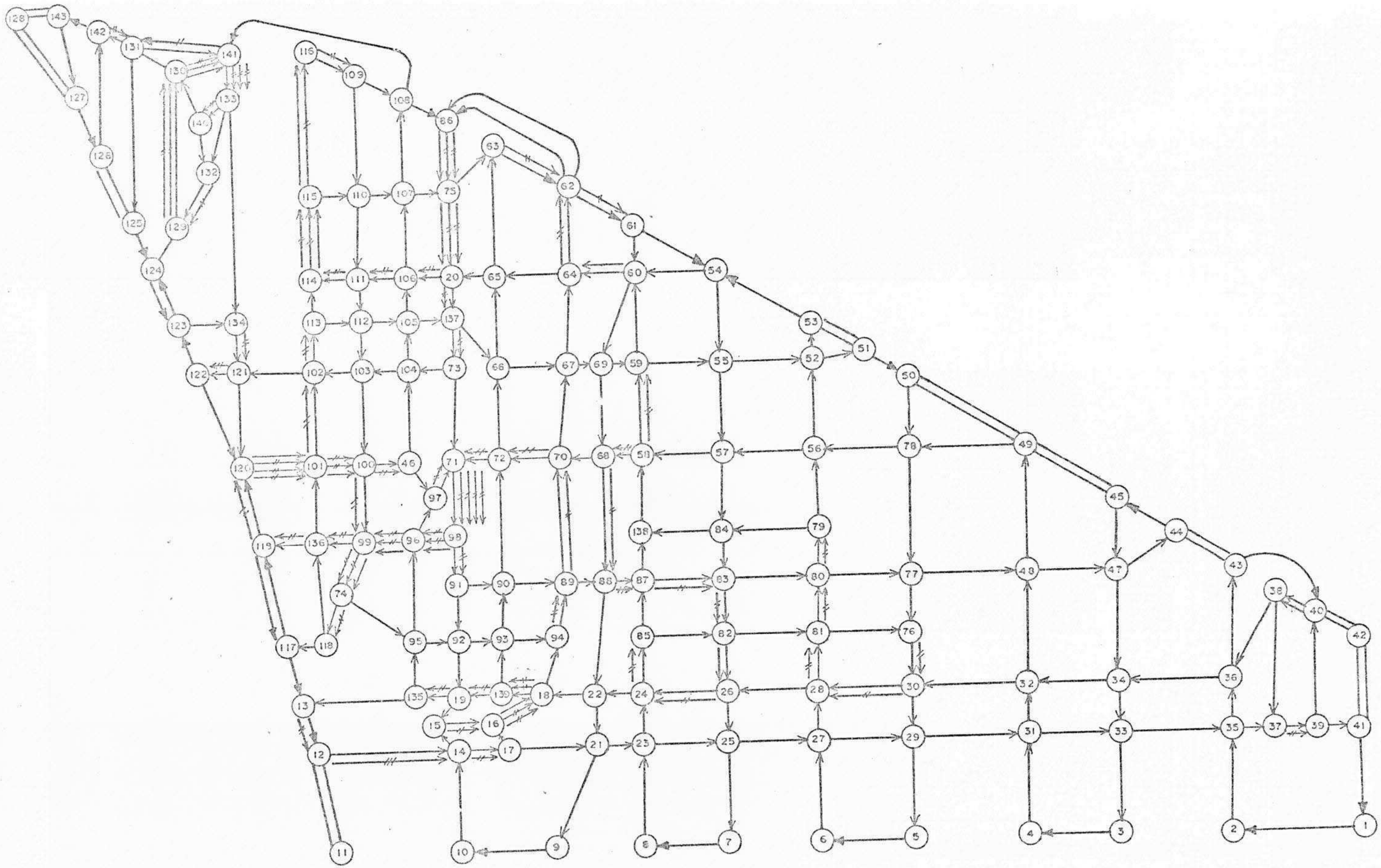
MIXED 1A	=	210,73	seg
MIXED 2A	=	88,74	seg
ROTA	=	4,54	seg
TOTAL	=	304,01	seg



ZONA 004 - GRAFO INICIAL



RESULTADO FORNECIDO PELO "MIXED 1A"



RESULTADO FORNECIDO PELO "MIXED 2A"

Zona 004 - Rota fornecida para o grafo resultante do "MIXED 2A".

	Nõ inicial	Nõ final
1	1	2
2	2	35
3	35	36
4	36	43
5	43	44
6	44	43
7	43	40
8	40	42
9	42	40
10	40	38
11	38	36
12	36	34
13	34	32
14	32	48
15	48	49
16	49	45
17	45	49
18	49	50
19	50	49
20	49	78
21	78	56
22	56	57
23	57	58
24	58	59
25	59	60
26	60	64
27	64	62
28	62	61
29	61	60
30	60	64
31	64	65
32	65	20
33	20	106
34	106	111
35	111	114
36	114	115
37	115	116
38	116	109
39	109	110
40	110	111
41	111	114
42	114	115

43	115	116
44	116	109
45	109	108
46	108	141
47	141	108
48	108	141
49	141	131
50	131	125
51	125	126
52	126	125
53	125	131
54	131	125
55	125	124
56	124	129
57	129	130
58	130	131
59	131	142
60	142	143
61	143	128
62	128	127
63	127	128
64	128	143
65	143	142
66	142	143
67	143	127
68	127	126
69	126	142
70	142	131
71	131	141
72	141	133
73	133	132
74	132	129
75	129	130
76	130	141
77	141	133
78	133	140
79	140	132
80	132	129
81	129	130
82	130	141
83	141	133
84	133	140
85	140	130
86	130	141
87	141	133
88	133	134
89	134	121
90	121	122
91	122	123
92	123	124
93	124	123
94	123	134
95	134	121
96	121	122
97	122	120
98	120	119
99	119	120
100	120	101
101	101	100
102	100	46

103	46	104
104	104	103
105	103	102
106	102	113
107	113	114
108	114	115
109	115	110
110	110	107
111	107	108
112	108	86
113	86	75
114	75	20
115	20	106
116	106	111
117	111	112
118	112	103
119	103	100
120	100	46
121	46	97
122	97	71
123	71	98
124	98	91
125	91	92
126	92	19
127	19	15
128	15	16
129	16	18
130	18	139
131	139	19
132	19	135
133	135	95
134	95	96
135	96	97
136	97	71
137	71	98
138	98	96
139	96	99
140	99	74
141	74	95
142	95	92
143	92	93
144	93	94
145	94	89
146	89	70
147	70	72
148	72	71
149	71	98
150	98	96
151	96	99
152	99	74
153	74	118
154	118	136
155	136	101
156	101	100
157	100	99
158	99	136
159	136	119
160	119	120
161	120	101
162	101	102

163	102	121
164	121	120
165	120	101
166	101	102
167	102	113
168	113	112
169	112	105
170	105	106
171	106	107
172	107	75
173	75	20
174	20	137
175	137	73
176	73	104
177	104	105
178	105	137
179	137	73
180	73	71
181	71	98
182	98	96
183	96	99
184	99	136
185	136	119
186	119	120
187	120	101
188	101	100
189	100	99
190	99	74
191	74	118
192	118	117
193	117	119
194	119	117
195	117	13
196	13	12
197	12	11
198	11	12
199	12	14
200	14	15
201	15	16
202	16	18
203	18	139
204	139	19
205	19	135
206	135	13
207	13	12
208	12	14
209	14	17
210	17	16
211	16	18
212	18	139
213	139	93
214	93	90
215	90	72
216	72	71
217	71	98
218	98	91
219	91	90
220	90	89
221	89	70
222	70	72

223	72	66
224	66	65
225	65	63
226	63	62
227	62	86
228	86	75
229	75	20
230	20	137
231	137	66
232	66	67
233	67	64
234	64	62
235	62	86
236	86	75
237	75	63
238	63	62
239	62	61
240	61	54
241	54	60
242	60	69
243	69	59
244	59	55
245	55	57
246	57	84
247	84	138
248	138	58
249	58	68
250	68	70
251	70	67
252	67	69
253	69	68
254	68	88
255	88	87
256	87	83
257	83	82
258	82	26
259	26	24
260	24	85
261	85	87
262	87	138
263	138	58
264	58	68
265	68	88
266	88	87
267	87	83
268	83	82
269	82	81
270	81	80
271	80	79
272	79	56
273	56	52
274	52	53
275	53	51
276	51	53
277	53	54
278	54	55
279	55	52
280	52	51
281	51	50
282	50	78

283	78	77
284	77	76
285	76	30
286	30	28
287	28	81
288	81	80
289	80	79
290	79	84
291	84	83
292	83	80
293	80	77
294	77	48
295	48	47
296	47	44
297	44	45
298	45	47
299	47	34
300	34	33
301	33	3
302	3	4
303	4	31
304	31	32
305	32	30
306	30	28
307	28	81
308	81	76
309	76	30
310	30	29
311	29	5
312	5	6
313	6	27
314	27	28
315	28	26
316	26	24
317	24	85
318	85	82
319	82	26
320	26	25
321	25	7
322	7	8
323	8	23
324	23	24
325	24	22
326	22	18
327	18	94
328	94	89
329	89	88
330	88	22
331	22	21
332	21	9
333	9	10
334	10	14
335	14	17
336	17	21
337	21	23
338	23	25
339	25	27
340	27	29
341	29	31
342	31	33

343	33	35
344	35	37
345	37	39
346	39	40
347	40	38
348	38	37
349	37	39
350	39	41
351	41	42
352	42	41
353	41	1

CAPÍTULO VI

CONCLUSÃO

Os resultados obtidos mostram que a combinação dos procedimentos heurísticos MIXED 1A e MIXED 2A, aplicada em dados reais, fornece resultados consideravelmente bons.

Com relação à implementação, esses resultados mostram um importante aumento na eficiência em relação à implementação efetuada por Santos |2|.

As principais diferenças entre as duas implementações estão resumidas no quadro seguite:

IMPLEMENTAÇÃO DE SANTOS	IMPLEMENTAÇÃO ATUAL
Utiliza MATRIZES DE ADJACÊNCIA	Utiliza LISTAS DE ADJACÊNCIA
Processa grafos com no máximo 100 nós	Processa grafos com no máximo 180 nós, podendo esse valor ser aumentado facilmente
Determina e armazena as cadeias de menor custos, e seus custos, entre todos os nós	Determina e armazena apenas cadeias de menor custo entre os nós com grau ímpar. E só determina as cadeias quando necessário
Não determina a rota	Determina a rota
Executa vários passos manualmente	Entrada um grafo misto $G = (N,E,A,c)$ Saída: uma rota

As duas implementações foram processadas no computador IBM 370/145, com sistema operacional OS/VS1 e compilador WATFIV-S. Aplicadas a um mesmo grafo, forneceram os seguintes resultados em termos de tempo de execução:

Grafo inicial: 32 nós, 61 arcos.

TEMPO DE EXECUÇÃO

ALGORITMO	SANTOS	ATUAL
MIXED 1A	377,50 seg	5,48 seg
MIXED 2A	526,76 seg	2,03 seg
	903,97 seg	7,51 seg
ROTA	não faz	1,26 seg
TOTAL	903,97 seg	0,77 seg

Esses resultados vêm comprovar que a escolha de estruturas adequadas para a representação dos dados é de fundamental importância para a implementação eficiente de qualquer algoritmo.

Com a implementação do algoritmo para a determinação de uma rota num gráfo de Euler completou-se a solução do problema, possibilitando sua utilização em situações reais.

Como resultado desse trabalho obteve-se um sistema para solucionar o problema de fornecimento de rotas para a distribuição de bens e serviços, aplicável na resolução daquela classe de problemas reais cuja característica principal é a necessidade de percorrer uma zona pré-determinada de uma cidade, passando por todas as suas ruas pelo menos uma vez. Dentro dessa classe de problemas estão incluídos: entrega automática de botijões de gás, ou de água engarrafada; coleta de lixo; entrega de lixo; entrega de cartas, de contas de luz, etc...

No desenvolvimento do sistema procurou-se considerar e solucionar todos os problemas e restrições que ocorrem em aplicações práticas. Consideraram-se principalmente os problemas ocasionados pela utilização de veículos na distribuição de bens e serviços.

Como complementação desse trabalho, sugere-se a realização de estudos adicionais sobre o zoneamento ideal de uma cidade. Dividir uma cidade em zonas, da melhor maneira possível, é um passo fundamental para que a rota fornecida para uma dessas zona seja também a melhor possível.

APÊNDICE I

1. CÓDIGOS UTILIZADOS NO SISTEMA

1.1. Código de Usuário

Para tornar o código do usuário representativo e de identificação mais fácil, sugere-se o seguinte código alfanumérico de quatro posições.

1	2	3	4
letra			dígito

As letras devem formar um conjunto que represente o melhor possível o nome do usuário, e o dígito é utilizado para diferenciar usuários, cujos conjuntos de letras coincidem.

Ex.:

USUÁRIO	CÓDIGO
Prefeitura Municipal de Fortaleza	FOR1
Prefeitura Municipal de Recife	REC1
Companhia de Distribuição de Água	CDA2
Departamento de Água e Esgotos	DAE1

Esse código será atribuído pelo responsável pelo sistema.

1.2. Código da Zona

Esse código será atribuído pelo prórpio usuário e sugere-se o seguinte procedimento:

a) A cidade deve ser dividida em zonas, de forma a permitir que as rotas determinadas sejam as melhores possíveis;

b) Atribui-se um código numérico de 4 posições a cada uma das zonas, de preferencia sequencialmente.

dígitos

1	2	3	4

OBS.1: Definiu-se um máximo de 180 esquinas por zona baseado em

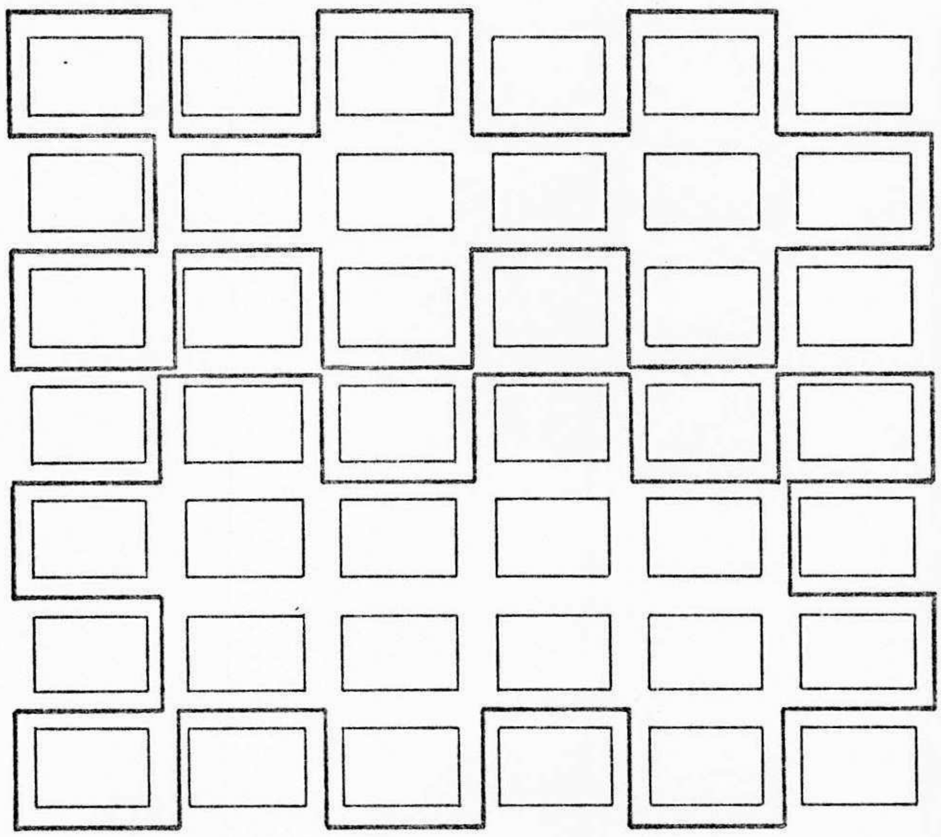
dois pontos principais:

a) Na prática uma zona tem em média 100 esquinas e, em geral não ultrapassa 150. Assim, dimensionar o sistema para 180 esquinas é mais do que razoável.

b) A possibilidade de implementação em equipamentos com capacidade de memória reduzida.

OBS.2: Para se obter o melhor roteiro possível, deve-se delimitar a zona de tal maneira que forme "dentes", isto é, metade da rua limítrofe vai pertencer a uma zona e a outra metade pertencerá à zona vizinha. Com isso diminui-se o "serviço inútil" realizado pelo coletor nas ruas limítrofes.

Ex.:



1.3. Código de Rua

Sugere-se que o código de rua seja numérico, atribuído sequencialmente, e que inicie sempre pelo número 001 para qualquer zona.

O código teria o seguinte formato:

└┘ └┘ └┘
1 2 3
dígito

Ex.:

RUA	CÓDIGO
R. Carlos Augusto dos Anjos	001
R. Princesa Isabel	002
R. Benedito Motta	003
R. Nova Lima	004
R. do Beco	005
R. Santo Antonio	006
R. João da Silva	007
R. Coronel Almeida	008
R. da Alfândega	009

2. PREPARAÇÃO DOS DADOS

A coleta das informações sobre cada um dos trechos de uma determinada zona é o procedimento que demanda maior quantidade de tempo. Deve ser conduzida uma pesquisa sobre cada zona para colher essas informações. Consideraremos aqui que essa tarefa é realizada pelo usuário.

Todos os modelos de formulários que serão apresentados a seguir conterão os seguintes campos em comum:

CT - Código do cartão - será utilizado para identificação do cartão durante o processamento. Essa informação será pré-impressa nos formulários.

CT
01

DATA - Em todos os formulários deve ser indicada a data do preenchimento do formulário.

DIA	MÊS	ANO
|	|	|

FOLHA - As folhas devem ser numeradas pelo usuário.

FOLHA	

VISTOS - Todos os formulários devem ser assinados pelas pessoas encarregadas do seu preenchimento, perfuração e conferência. Essas pessoas devem ainda acrescentar a data em que foram realizadas essas operações.

Ex.:

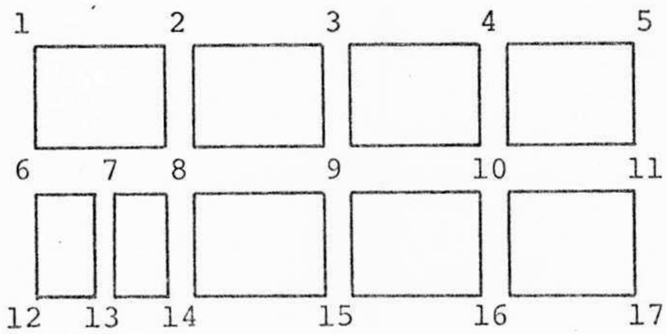
// ________	// ________	// ________
Preenchido	Perfurado	Conferido

Entende-se por trecho, a porção da rua compreendida entre duas esquinas, que de agora em diante serão chamadas de nós.

Procedimentos necessários para o preenchimento do formulário para cada cadastramento de trechos: de posse de um mapa da zona o usuário deve fazer o seguinte:

a) Numerar cada nó sequencialmente, iniciando sempre pelo número 1. Sugere-se para maior clareza e facilidade de preenchimento, que essa numeração seja feita no mesmo sentido em que escrevemos, isto é, da esquerda para a direita, e de cima para baixo.

Ex.:



b) Obter todas as informações necessárias sobre cada trecho:
- DISTÂNCIA em metros entre os dois nós (esquinas);
- SENTIDO do trânsito (mão única, ou mão dupla);
- CONDIÇÕES de tráfego (intenso, regular, pequeno);
- ESTADO de conservação (bom, regular, ruim, intran-

sitável);

- Se deve ou não haver COLETA no trecho (em função do número de casas nele existentes);

OBS.: De todas as informações citadas acima, a única de importância fundamental é a distância do trecho. As demais podem ser suprimidas caso não seja possível obtê-las. Nesse caso a rota da zona será calculada apenas em função da distância. Se for possível fornecer as outras informações a determinação da rota será feita em função de todas elas, obtendo-se possivelmente uma rota melhor.

O usuário pode então iniciar o preenchimento dos formulários.

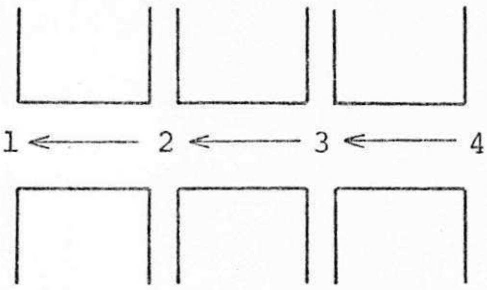
OBS.: a) Caso haja repetição constante da mesma informação numa determinada coluna, o usuário poderá informá-la apenas uma vez e passar um traço na coluna até a linha onde a informação se modifica.

Ex.:

D	D
S	S
S	
S	
S	
S	
S	
S	
S	
S	
S	
S	
S	
S	
N	N
N	N
S	S

b) Caso o sentido do tráfego no trecho seja único, os nós inicial e final devem indicar esse sentido.

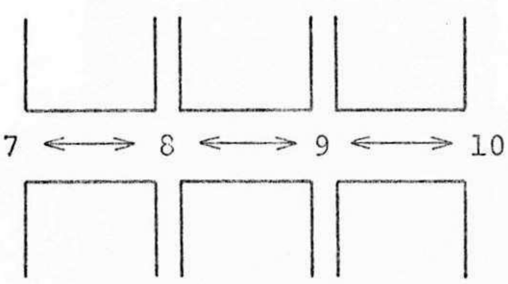
Ex.:



nó inicial	nó final
2	1
3	2
4	3

c) Caso o sentido do tráfego seja duplo, os nós inicial e final podem indicar um dos dois sentidos.

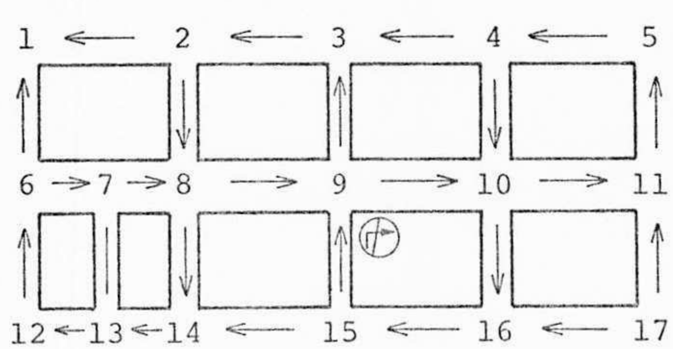
Ex.:



nó inicial	nó final
7	8
9	8
9	10

Entende-se por restrição de trânsito a ocorrência de um aviso de proibido, contornar à direita ou à esquerda, ou seguir em frente, apesar do sentido da rua permitir essa manobra.

Ex.:



Considerando-se que as ruas tenham o sentido indicado no mapa acima e que existe um sinal proibindo contornar à direita na esquina número 9, deve-se considerar como restrito o trecho:

$$\begin{tikzcd} 9 \arrow[r] & 10 \\ 15 \arrow[u] & \end{tikzcd}$$

O nó inicial do trecho com restrição será o 15, o nó com restrição será o 9, e o nó final do trecho com restrição será o 10.

Sugere-se aqui os seguintes modelos de formulários:

ROTAS PARA
DISTRIBUIÇÃO DE BENS E SERVIÇOS

CT
1

FORMULARIO PARA CADASTRO DE USUARIO

SEQ 1 (2) — NOME DO USUARIO (3–35)

NOME DO INFORMANTE (36–68)

SEQ 2 (2) — CARGO DO INFORMANTE (3–35)

ENDEREÇO DO USUÁRIO

RUA (36–62) — NUMERO (63–66)

ANDAR (17–69) — SALA (70–73) — COD. DDD (74–77) — SEQ 3 (2) — FONE (3–9)

BAIRRO (10–34) — CEP. (35–39)

CIDADE (40–67) — ESTADO (68 69)

COD. USUÁRIO (70–73) — DATA: DIA MES ANO (74–79) — OP (80)

OP = I INCLUSÃO
E EXCLUSÃO
A ALTERAÇÃO
D DESATIVAÇÃO
R REATIVAÇÃO

/ / ________ INFORMANTE

/ / ________ RESPONSÁVEL PELO SISTEMA

/ / ________ PERFURADO

/ / ________ CONFERIDO

ROTAS PARA
DISTRIBUIÇÃO DE BENS E SERVIÇOS

CT
2
1

FORMULARIO PARA CADASTRO DE SETOR

COD. USUARIO	COD. SETOR	DIA	MÊS	ANO	OP
2 5	6 9	10		15	16

INICIO ROTA	FIM ROTA
17 19	20 22

OP: I INCLUSÃO
E EXCLUSÃO
A ALTERAÇÕES

/ / PREENCHIDO

/ / PERFURADO

/ / CONFERIDO

ROTAS PARA
DISTRIBUIÇÃO DE BENS E SERVIÇOS

CT
3

FORMULARIO PARA CADASTRO DE RUAS

COD. USUARIO	COD. SETOR	DIA	MÊS	ANO	FOLHA
2 5	6 9				

COD. RUA	NOME DA RUA	OP.
10 12	13 41	42

/ / PREENCHIDO / / PERFURADO / / CONFERIDO

OP - I-INCLUSÃO
E-EXCLUSÃO
A-ALTERAÇÃO

ROTAS PARA
DISTRIBUIÇÃO DE BENS E SERVIÇOS

CT
4

FORMULARIO PARA CADASTRO DE ESQUINAS (NÓS)

COD. USUARIO	COD. SETOR	DIA	MÊS	ANO	FOLHA
2 – 5	6 – 9				

COD. RUA	ESQUINAS (NÓS) PERTENCE A RUA												OP
10 – 12	13 – 15	16 – 18	19 – 21	22 – 24	25 – 27	28 – 30	31 – 33	34 – 36	37 – 39	40 – 42	43 – 45	46 – 48	49

/ / PREENCHIDO

/ / PERFURADO

/ / CONFERIDO

OP - I-INCLUSÃO
A-ALTERAÇÃO

ROTAS PARA DISTRIBUIÇÃO DE BENS E SERVIÇOS

CT
5
1

FORMULARIO PARA CADASTRO DE TRECHOS

COD. USUARIO	COD. SETOR	DIA	MES	ANO	FOLHA
2 – 5	6 – 9				

COD. RUA	NÓ INICIAL	NÓ FINAL	DISTÁCIA	A	B	C	D	E
10 – 12	13 – 15	16 – 18	19 – 22	23	24	25	26	27
10 – 12	13 – 15	16 – 18	19 – 22	23	24	25	26	27

A: SENTIDO DA RUA
1 – ÚNICO
2 – DUPLO

B: CONDIÇÕES TRANS.
1 - FRACO
2 - REGULAR
3 - INTENSO

C: ESTADO DA RUA
1 - BOM
2 - REGULAR
3 - RUIM

D: COLETA
1 – TEM
Ø – NÃO TEM

E: COD. OPERAÇÃO
I – INCLUSÃO
E – EXCLUSÃO
A – ALTERAÇÃO

/ / ________ PREENCHIDO

/ / ________ PERFURADO

/ / ________ CONFERIDO

ROTAS PARA
DISTRIBUIÇÃO DE BENS E SERVIÇOS

CT
6

FORMULARIO PARA CADASTRO DE RESTRIÇÕES DE TRÂNSITO

COD. USUARIO	COD. SETOR	DIA	MES	ANO	FOLHA
2 5	6 9				

NÓ INICIAL	NÓ C/REST.	NÓ FINAL	OP.
10 12	13 15	16 18	19

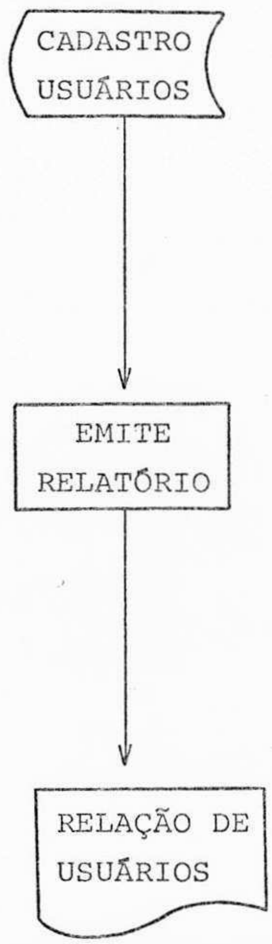
/ / PREENCHIDO
/ / PERFURADO
/ / CONFERIDO

OP- I-INCLUSÃO
E-EXCLUSÃO

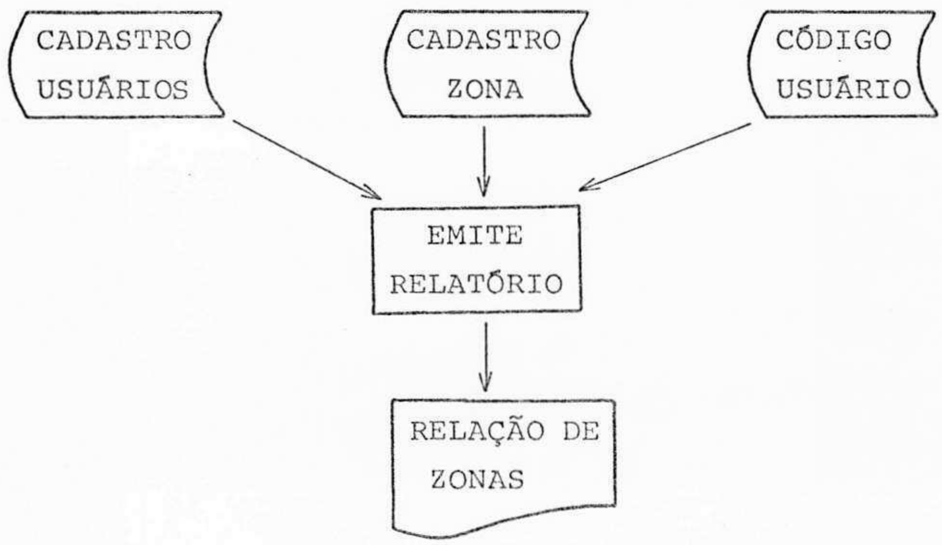
3. RELATÓRIO DE CONTROLE

Além dos relatórios descritos no capítulo II, serão emitidos ainda alguns relatórios para controle seguindo os fluxogramas.

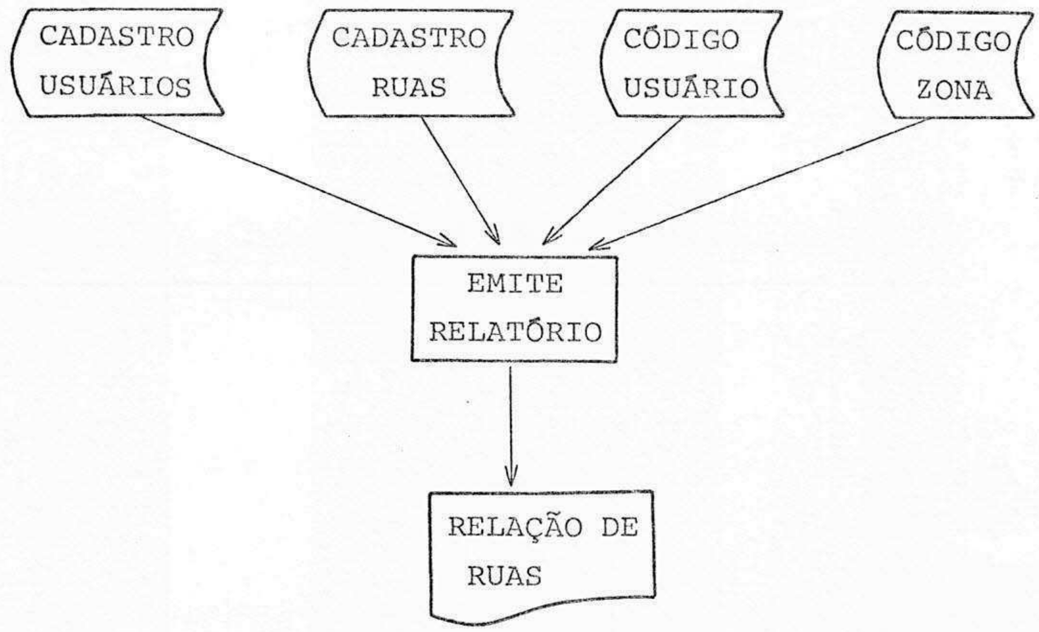
a) Relação de Usuários Cadastrados no Sistema.



b) Relação de Zonas de um Usuário



c) Relação de Ruas Cadastradas numa Zona



A seguir são apresentados os modelos desses relatórios.

```
*****************************************************************************************
*                                                                                       *
*    ROTAS PARA DISTRIBUIÇÃO DE BENS E SERVIÇOS            02/08/82           FOLHA -   1      *
*                                                                                       *
*    RELAÇÃO DE USUÁRIOS CADASTRADOS NO SISTEMA       TOTAL DE USUÁRIOS CADASTRADOS -      3      *
*                                                                                       *
*****************************************************************************************


CÓDIGO  ATIVO             USUÁRIO                                  ENDEREÇO

 DAE1    SIM   DEPARTAMENTO DE ÁGUAS E ESGOTOS S/A    RUA RUI BARBOSA            - 184    23 ANDAR SALA 2344
               ENCAR - JOSÉ SILVA PIMENTEL            13560 - SÃO CARLOS                              SP
               CARGO - GERENTE ADMINISTRATIVO               FONE - (0162)   71-8236
                                                            CADASTRADO EM - 06/06/79

 CDA2    NÃO   COMPANHIA DE DISTRIBUIÇÃO DE ÁGUA      RUA GENERAL OSÓRIO         - 1574    9 ANDAR SALA  904
               ENCAR - MARIA EUGÊNIA PINHEIRO         10000 - SANTO ANDRÉ                             SP
               CARGO - CHEFE DA DISTRIBUIÇÃO                FONE - (0166)  42-3776
                                                            CADASTRADO EM - 12/11/80

 FOR1    SIM   PREFEITURA MUNICIPAL DE FORTALEZA      RUA JOÃO PESSOA            - 2354
               ENCAR - RAFAEL CALIL BUENO DA COSTA    58100 - FORTALEZA                               CE
               CARGO - SECRETARIO DOS TRANSPORTES           FONE - (083) 321-8514
                                                            CADASTRADO EM - 24/02/81
```

```
*****************************************************************************************
*                                                                                       *
*    ROTAS PARA DISTRIBUIÇÃO DE BENS E SERVIÇOS          02/08/82          FOLHA -    1 *
*                                                                                       *
*    USUÁRIO - DEPARTAMENTO DE ÁGUAS E ESGOTOS S/A            CÓDIGO - DAE1             *
*                                                                                       *
*    RELAÇÃO DE ZONAS CADASTRADAS                        TOTAL DE ZONAS -    7          *
*                                                                                       *
*****************************************************************************************
```

CÓDIGO DA ZONA	DATA DE CADASTRAMENTO
001	03/02/79
002	03/04/80
003	05/08/80
004	24/08/80
005	31/08/80
006	10/01/81
007	21/04/81

```
*************************************************************************************************************
*                                                                                                           *
*    ROTAS PARA A DISTRIBUIÇÃO DE BENS E SERVIÇOS                02/08/82            FOLHA -     1          *
*                                                                                                           *
*    USUÁRIO - DEPARTAMENTO DE ÁGUAS E ESGOTOS S/A                   CÓDIGO - DAE 1                         *
*                                                                                                           *
*    RELAÇÃO DE RUAS CADASTRADAS NA ZONA - 0001                  TOTAL DE RUAS DA ZONA -    11              *
*                                                                                                           *
*************************************************************************************************************
```

NOME DA RUA	CÓDIGO DA RUA
SÃO SEBASTIÃO	001
JESUINO DE ARRUDA	002
CARLOS BOTELHO	003
VIRGÍLIO POZZI	004
MARECHAL DEODORO	005
JOSÉ BONIFÁCIO	006
EPISCOPAL	007
ALEXANDRINA	008
SÃO JOAQUIM	009
GEMINIANO COSTA	010
RUI BARBOSA	011

```
*****************************************************************************************************
*                                                                                                   *
*    ROTAS PARA DISTRIBUIÇÃO DE BENS E SERVIÇOS                02/08/82          FOLHA -    1       *
*                                                                                                   *
*    USUÁRIO - DEPARTAMENTO DE ÁGUAS E ESGOTOS                    CÓDIGO - DAE1                     *
*                                                                                                   *
*    ROTA DA ZONA - 0001                     DISTÂNCIA TOTAL A SER PERCORRIDA -        4.753 MS     *
*                                                                                                   *
*****************************************************************************************************
```

	SIGA PELA RUA	ATÉ O CRUZAMENTO COM A RUA	COLETA
1	CARLOS BOTELHO	SÃO SEBASTIÃO	SIM
2	SÃO SEBASTIÃO	VIRGILIO POZZI	SIM
3	VIRGÍLIO POZZI	RUI BARBOSA	SIM
4	RUI BARBOSA	ALEXANDRINA	SIM
5	ALEXANDRINA	EPISCOPAL	SIM
6	EPISCOPAL	MARECHAL DEODORO	SIM
7	MARECHAL DEODORO	CARLOS BOTELHO	SIM
8	CARLOS BOTELHO	JOSÉ BONIFÁCIO	SIM
9	JOSÉ BONIFÁCIO	ALEXANDRINA	SIM
10	ALEXANDRINA	SÃO JOAQUIM	SIM
11	SÃO JOAQUIM	RUI BARBOSA	SIM
12	RUI BARBOSA	CARLOS BOTELHO	SIM

APÊNDICE II

1. ALGORITMOS COMPLEMENTARES

Os algoritmos para o fluxograma da figura 2.1 são os seguintes:

1.1. Copia em Disco

Objetivo: Copiar todos os arquivos que estão em fita, e que serão utilizados temporariamente em disco durante o processamento.

Algoritmo:

```
PARA TODOS OS ARQUIVOS FAÇA:
   Begin
     Lê registro em fita;
     Copia registro em disco;
   End;
```

1.2. Copia em Fita

Objetivo: copiar os arquivos utilizados do disco para a fita.

```
PARA TODOS OS ARQUIVOS FAÇA:
   Begin
     Lê registro em disco;
     Copia registro em fita;
   End;
```

OBS.: Os procedimentos COPIA EM DISCO e COPIA EM FITA, podem ser executados através de comandos de "JCL".

1.3. Crítica e Atualização

Objetivo: fazer a crítica e consistência dos dados, procurando evitar ao máximo a entrada de dados errados no sistema. Caso os dados estejam corretos os arquivos correspondentes serão atualizados.

Visando proporcionar um melhor entendimento do algoritmo, serão descritas a seguir todas as variáveis utilizadas.

- CODUS (Código do Usuário)
- CT (Código do Cartão) =
  1: Cadastramenro de usuário;
  2: Cadastramento de zona;
  3: Cadastramento de ruas;
  4: Cadastramento de esquinas;
  5: Cadastramento de trechos;
  6: Cadastramento de restrições de trânsito.

- CODOP (Código de Operação) =
  I: Inclusão;
  E: Exclusão;
  A: Alteração;
  D: Desativação;
  R: Reativação.

- SITUAC (Situação do Usuário) =
  A: Ativo;
  D: Desativado - indica que o usuário não tem permissão para utilizar o sistema.

Algoritmo:

```
    Lê o primeiro cartão;
    IF CT = 1
       THEN Begin
              Testa se os tipos dos conteúdos
              dos campos estão corretos;
              CASE CODOP:
                  I: IF CODUS já existe
                          THEN Erro 1
                          ELSE Begin
                                 Inclui o usuário;
                                 SITUAC = A;
                               End;
                  E: IF CODUS não existe
                         THEN Erro 2
                         ELSE Begin
                                Remove registro;
                                Remove todos os arquivos
                                 do usuário;
```

```
                D: IF CODUS não existe
                       THEN Erro 2
                       ELSE SITUAC = D;
                R: IF CODUS não existe
                       THEN Erro 2
                       ELSE SITUAC = D;
                A: IF CODUS não existe
                       THEN Erro 3
                       ELSE atualiza os campos indi-
                            cados;
                End;
            End;
    IF SITAC = D
        THEN Begin
               Erro 4;
               STOP
             End;
        ELSE Para todos os cartões de dados faca:
        CASE CT:
          2: Testa compatibilidade dos campos;
             CASE CODOP:
               I: IF CODZON já existe
                      THEN Erro 5
                      ELSE Inclui a zona;
               E: IF CODZON não existe
                      THEN Erro 6
                      ELSE Begin
                               Exclui o registro;
                               Remove os arquivos da
                               zona;
                           End;
               A: IF CODZON não existe
                      THE Erro 6
                      ELSE Atualiza os campos indica
                           dos;
             END;
          3: Testa a compatibilidade dos campos;
             CASE CODOP:
             I: IF NOME já existe
                    THEN Erro 7
```

```
                ELSE IF CODRUA existe
                        THEN Erro 9
                        ELSE Inclui a rua;
          E: IF CODRUA não existe
                THEN Erro 8
                ELSE Begin
                       Remove registro;
                       Exclui a rua;
                       Remove os arquivos relacio-
                       nados com a rua;
                     End;
          A: IF CODRUA não existe
                THEN Erro 8
                ELSE Atualiza os campos indicados;
       END;
    4: Testa a compatibilidade dos campos;
       CASE CODOP:
         I: IF CODRUA existe
               THEN Erro 9
               ELSE Inclui o registro;
         A: IF CODRUA não existe
               THEN Erro 10
               ELSE Altera o registro;
      END;
         ELSE CASE CODOP:
                I: Inclui o registro;
                A: Altera o registro;
    5: Testa a compatibilidade dos campos;
       CASE CODOP:
         I: IF o trecho existe
               THEN Erro 11
               ELSE BEGIN
                      Inclui o registro;
                      IF SENTID = 2
                         THEN Adicione o trecho
                                   (j,i);
                    END;
         E: IF trecho não existe
               THEN Erro 12
               ELSE exclui o registro;
```

```
        A: IF o trecho não existe
              THEN Erro 12
              ELSE Altera o registro;
      END;
    6: Testa compatibilidade dos campos;
      CASE CODOP:
        I: IF trecho existe
              THEN Erro 11
              ELSE Exclui o registro;
        A: IF trecho não existe
              THEN Erro 12
              ELSE Exclui o registro;
      END;
```

TABELA DE ERROS:

ERRO 1 - Erro no código do usuário. Já existe outro usuário com o mesmo código. Inclusão rejeitada;

ERRO 2 - Exclusão, desativação, ou reativação de um usuário não cadastrado no sistema;

ERRO 3 - Alteração das informações de um usuário não cadastrado no sistema;

ERRO 4 - Usuário desativado. Não tem permissão para utilizar o sistema;

ERRO 5 - Erro no código da zona. Já existe uma zona desse usuário cadastrado com o mesmo código. Inclusão rejeitada.

ERRO 6 - Exclusão ou alteração de uma zona, que não está cadastrado no sistema;

ERRO 7 - Erro no nome da rua. Uma rua com o mesmo nome está cadastrada no sistema. Inclusão rejeitada;

ERRO 8 - Exclusão ou alteração de uma rua não cadastrada no sistema;

ERRO 9 - Erro no código da rua. Já existe uma rua com o mesmo código cadastrada no sistema. Inclusão rejeitada;

ERRO 10 - Alteração de dados de uma rua não cadastrada no sistema;

ERRO 11 - Trecho já cadastrado anteriormente. Inclusão rejeitada.

ERRO 12 - Exclusão ou alteração de um trecho de rua não cadastrado.

1.4. EMITE RELATÓRIO

Objetivo: emitir RELAÇÃO DE:

- USUÁRIOS CADASTRADOS;
- ZONAS CADASTRADAS;
- RUAS DE UMA ZONA.

O procedimento para a impressão desses relatórios é simples e idêntico para todos, consistindo basicamente em listar as informações contidas nos arquivos correspondentes.

BIBLIOGRAFIA

[1] COSTA, M.A. Bueno - Métodos para Resolução de Rotas Eu-lerianas em Grafos Mistos com Aplicação na Distribuição de Bens e Serviços. UFPb, Campina Grande, Pb, (1982).

[2] SANTOS, Antonio - Uma Solução Heurística para o Problema do Carteiro Chinês num Grafo Misto com Aplicação à Distribuição de Bens e Serviços. UFPb, Campina Grande, Pb, (1981).

[3] HOROWITZ, E.; Sahni, S. - Fundamentals of Data Structure. Computer Science Press, Inc., Maryland, (1977).

[4] KNUTH, D.E. - The Art of Computer Programming. Addison--Wesley Publishing Company, USA, (1973).

[5] AHO, A.V.; HOPCROFT, J.E. & ULLMAN, J.D. - The Design and Analysis of Computer Algorithms. Addison-Wesley Publishing Company, USA, (1974).

[6] BERZTISS, A.T. - Data Structure Theory and Practice. Academic Press, London, (1975).

[7] DEO, N. - Graph Theory with Application to Engeneering and Computer Science. Prentice Hall, London, (1974).

[8] READ, R.C. - Graph Theory and Computing. Academic Press, New York, (1972).

[9] CHRISTOFIDES, N. - Graph Theory: An Algorithmic Approach. Academic Press, London, (1975).

[10] EISELT, H.A.; FRAGER, V. Helmut - Operations Research Handbook. Walter de Gruyeter, Berlin, (1977).

[11] DERIGS, U. - A Shortest Augmenting Path Method for Solving Minimal Perfect Matching Problems. Report 79-06, Mathematisches Institut der Universität zu Köln, (1979).

[12] HILLIER, F.S.; LIEBERMAN, G.J. - Operation Research, Holden-Day, Inc., San Francisco, (1974).

[13] FREDERICKSON, L.N. - Approximation Algorithms for Some Postman Problems. ACM 26, 3 (July, 1979), 538--554.

[14] BODIN, L.D. - A Detaild Description of a Computer System for the Routing and Scheduling of Streer Sweepers. Computation & Operations Research, vol. 6, p.p. 181--198. Pergamon Press Ltd, (1979).