

**TÉCNICAS DA PESQUISA OPERACIONAL APLICADAS A
UM PROBLEMA DE COBERTURA DE ARCOS**

ANDREIA SMIDERLE

**TÉCNICAS DA PESQUISA OPERACIONAL APLICADAS A
UM PROBLEMA DE COBERTURA DE ARCOS**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Ciências. Curso de Pós Graduação em Métodos Numéricos em Engenharia – Programação Matemática. Setores de Tecnologia e de Ciências Exatas, Universidade Federal do Paraná.

Orientadora: Prof.^a Dr.^a Maria Teresinha Arns Steiner

Co-orientador: Prof. Dr. Celso Carnieri

Curitiba, 12 de Novembro de 2001.

ANDREIA SMIDERLE

**TÉCNICAS DA PESQUISA OPERACIONAL APLICADAS A
UM PROBLEMA DE COBERTURA DE ARCOS**

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre no Curso de Pós-Graduação em Métodos Numéricos em Engenharia – Programação Matemática da Universidade do Paraná, pela Comissão formada pelos professores:

Maria Teresinha Arns Steiner
Profª Drª Maria Teresinha Arns Steiner
Dpto. de Matemática, UFPR

Celso Carnieri
Prof. Dr. Celso Carnieri
Programa de Pós-Graduação em Métodos
Numéricos em Engenharia, UFPR

Sergio F. Mayerle
Prof. Dr. Sergio F. Mayerle
Programa de Pós-Graduação em Engenharia
de Produção, UFSC

Curitiba, 12 de Novembro de 2001.

Dedico este trabalho ao meu esposo **Sandro**,
meu eterno amor.

“A vida cotidiana é a vida de todo homem.
Todos vivem, sem nenhuma exceção...”

Agnes Heller

AGRADECIMENTOS

A Deus, por tudo que sou e tenho.

À minha família, em especial ao meu marido Sandro pelo amor inabalável apesar de todos os momentos de minha ausência. Aos meus pais, Saly e Orides, pelo apoio, dedicação e preocupação que sempre tiveram em todos os momentos de minha vida. Ao meu irmão, Alcir e a sua família, por toda motivação e carinho ao longo de minha carreira acadêmica.

À professora Maria Teresinha Arns Steiner, pelo carinho e pela orientação para realização deste trabalho.

Ao professor Celso Carnieri pela co-orientação e ao professor Sergio F. Mayerle pelas valiosas sugestões.

Aos colegas, pela amizade formada durante o curso, e em especial ao amigos: Janecler, Rogério, Donizetti, Gilson e Marcelo pelos muitos dias compartilhados em estudos.

Aos amigos Maristela e Claudionei, pelo carinho, apoio e constante ajuda.

À minha ex-professora, hoje colega, Adriana Auzani, pelo carinho que dispensou para a revisão gramática deste trabalho.

Aos amigos Rafael Hentz e Patric Zago pelas dicas de programação que muito me ajudaram.

À empresa Sanepar pelo fornecimento dos dados utilizados neste trabalho.

Ao Centro Federal de Educação Tecnológica do Paraná (CEFET – Unidade de Pato Branco – PR), em especial ao professor Loreci Zanardini pelo esforço em formar esta turma, e UFPR pela oportunidade de realização deste trabalho.

SUMÁRIO

LISTA DE FIGURAS	ix
LISTA DE FLUXOGRAMAS	x
LISTA DE QUADROS	xi
RESUMO.....	xii
ABSTRACT.....	xiii
CAPÍTULO I	
1. INTRODUÇÃO	1
1.1. Objetivos do Trabalho	1
1.2. Importância do Trabalho	1
1.3. Limitações do Trabalho	2
1.4. Estrutura do Trabalho	2
CAPÍTULO II	
2. REVISÃO DE LITERATURA	4
CAPÍTULO III	
3. DESCRIÇÃO DO PROBLEMA REAL	8
3.1. Introdução	8
3.2. Descrição da Situação Atual com a Utilização do Roteamento Existente	10
CAPÍTULO IV	
4. ALGUMAS NOÇÕES BÁSICAS SOBRE GRAFOS	16
4.1. Introdução	16
4.2. Definições	17
4.2.1. Ciclos e Circuitos Hamiltonianos	19
4.2.2. Ciclos Eulerianos	20
4.2.2.1. Determinação de um Caminho Euleriano	20
4.3. Representação de Grafos	21
4.3.1. Matriz de Adjacência	21
4.3.2. Matriz de Incidência.....	22
4.3.3. Matriz de Custos	23
CAPÍTULO V	
5. MODELOS DE ROTEAMENTO DE VEÍCULOS	24
5.1. Introdução	24
5.2. Problema de Roteamento de Veículos.....	25
5.3. Formulação para o Problema Geral de Roteamento de Veículos	25
5.4. Estratégias de Solução para os Problemas de Roteamento.....	27
5.5. Problemas Clássicos Básicos de Roteamento	27
5.5.1. Os Problemas de Cobertura de Nós	28
5.5.1.1. O Problema do Caixeiro Viajante (PCV)	28
5.5.1.1.1. As Origens do Problema	28
5.5.1.1.2. Importância Atual do PCV	30
5.5.1.1.3. Formulação da Programação Matemática para o PCV	30

5.5.1.1.4. Procedimentos de solução para os problemas de cobertura de nós: algoritmos exatos, relaxações e algoritmos aproximativos ou heurísticos	32
5.5.2. Os Problemas de Cobertura de Arcos	33
5.5.2.1. O Problema do Carteiro Chinês (PCC).....	33
5.5.2.1.1. Histórico	33
5.5.2.1.2. A Importância do Modelo do Carteiro Chinês para o Problema de Roteamento de Veículos.....	35
5.5.2.1.3. O Problema do Carteiro Chinês em Grafos Não-Orientados	36
5.5.2.1.4. O Problema do Carteiro Chinês em Grafos Orientados.....	41
5.5.2.1.5. O Problema do Carteiro Chinês Capacitado	42
5.6. Uma técnica para determinação de caminhos mínimos entre nós de um grafo	45

CAPÍTULO VI

6. DESCRIÇÃO DOS MÉTODOS PROPOSTOS NA RESOLUÇÃO DO PROBLEMA	47
6.1. Introdução	47
6.2. O Problema de Localização de Facilidades	47
6.2.1. Introdução.....	47
6.2.2. O Problema das P-Medianas.....	49
6.2.3. Formulação Matemática para o Problema das P-Medianas	49
6.2.4. Algoritmo das P-Medianas de Teitz e Bart	50
6.2.5. Procedimentos Básicos do Algoritmo das P-Medianas de Teitz e Bart	52
6.3. Algoritmos Genéticos.....	53
6.3.1. Introdução.....	53
6.3.2. Histórico	54
6.3.3. Aplicações dos Algoritmos Genéticos	55
6.3.4. Funcionamento dos Algoritmos Genéticos	56
6.3.5. O Processo de Reprodução.....	59
6.3.6. O Sistema de Representação e Codificação	60
6.3.7. Geração da População Inicial	61
6.3.8. Os Operadores Genéticos	62
6.3.8.1. Operadores de Recombinação (Cruzamento)	62
6.3.8.2. Operadores de Mutação	64
6.3.9. Aplicação dos Operadores Genéticos.....	66
6.3.10. A Função Aptidão	67
6.3.11. Os Esquemas de Seleção	67
6.3.12. Convergência, Diversidade Populacional e Nichos	68
6.3.13. Algoritmos Genéticos Paralelos.....	69
6.4. O Algoritmo Genético Utilizado Neste Trabalho	70
6.4.1. Algoritmo Genético para o Problema das P-Medianas	70
6.4.1.1. Descrição do Algoritmo Proposto	71
6.5. Algoritmo Para Formação dos <i>Clusters</i> (Agrupamentos)	72

CAPÍTULO VII

7. APLICAÇÃO DAS TÉCNICAS APRESENTADAS À SOLUÇÃO DO PROBLEMA PROPOSTO – IMPLEMENTAÇÃO COMPUTACIONAL	74
7.1. Introdução	74
7.2. Etapas da Implementação Computacional.....	75
7.3. A Implementação Computacional e Obtenção dos Resultados	77

7.3.1. O Cadastramento dos Dados – Definição do Grafo Inicial	77
7.3.2. A Definição das 12-Medianas	78
7.3.3. Formação dos <i>Clusters</i>	81
7.3.4. Roteamento dos Subgrupos	84
7.3.4.1. Formulação do Problema do Carteiro Chinês	84
7.3.4.2. Algoritmo do Carteiro Chinês	86
CAPÍTULO VIII	
8. ANÁLISE DOS RESULTADOS NUMÉRICOS, CONCLUSÕES FINAIS E SUGESTÕES PARA TRABALHOS FUTUROS.....	89
8.1. Análise dos resultados	89
8.1.1. Fase I – Formação dos Subgrupos de Atendimento	89
8.1.2. Fase II – Roteamento dos Subgrupos	90
8.2. Conclusões	91
8.3. Sugestões para Trabalhos Futuros.....	92
REFERÊNCIAS BIBLIOGRÁFICAS 93	
ANEXO I	
FORMULAÇÃO DO PROBLEMA DO CARTEIRO CHINÊS PARA A ROTA 272	98
APÊNDICE I	
EXEMPLOS DE APLICAÇÃO DOS ALGORITMOS UTILIZADOS NO TRABALHO	111
APÊNDICE II	
AS COORDENADAS GEOGRÁFICAS DOS PONTOS DE PARADA.....	128

LISTA DE FIGURAS

Figura 3.1 – Mapa da Cidade de Pato Branco – PR (Destacando-se o Grupo 11).....	11
Figura 3.2 – Visualização das Rotas Existentes Atualmente	13
Figura 3.3 – Visualização da Sequência da Rota 11005	14
Figura 4.1 – Representação Geométrica de um Grafo $G(V,E)$, onde $V = \{v_1, v_2, v_3, v_4, v_5\}$ e $E = \{v_1v_2, v_1v_3, v_2v_4, v_3v_4, v_1v_1\}$	17
Figura 4.2 – Representação Geométrica de $V = \{x, y, z, v, w\}$ e $E = \{xy, xv, yv, yw, vz, vw, vw, zw\}$, onde as Arestas vw são Paralelas.	18
Figura 4.3 – Representação de Grafos não Dirigidos, onde em (a) tem-se um Grafo Conexo e em (b), um Grafo não Conexo.....	19
Figura 4.4 – Representação Geométrica de um Circuito Hamiltoniano	20
Figura 4.5 – Representação de um Ciclo Euleriano.....	20
Figura 4.6 – Representação Geométrica do Grafo G e sua Matriz de Adjacência.	22
Figura 4.7 – Representação Geométrica do Grafo G Dirigido e Sua Matriz de Incidência.....	22
Figura 4.8 – Representação Geométrica do Grafo G Não-Dirigido e Sua Matriz de Incidência ..	23
Figura 4.9 – Representação Geométrica do Grafo G Valorado e sua Matriz de Custos.....	23
Figura 5.1 – Possíveis Estratégias Para Solução de PRV	27
Figura 5.2 – Jogo de Hamilton	29
Figura 5.3 – Uma Solução do Jogo de Hamilton.....	29
Figura 5.4 – Visualização de Königsberga, e as Sete Pontes sobre o Rio Pregel.....	34
Figura 5.5 – Representação em Grafo do Problema de Euler, 1736, Representando a Situação do Problema das Sete Pontes.....	34
Figura 5.6 – Grafo Qualquer em (A) e seu Caminho Euleriano em (B).....	35
Figura 5.7 – Processo de Obtenção de $G_i(E^*)$	38
Figura 5.8 – Obtenção de K_4	39
Figura 5.9 – Solução Final.....	40
Figura 5.10 – Exemplo de um PCC Inviável.....	42
Figura 6.1 – Estrutura Básica de um Algoritmo Genético	58
Figura 6.2 – Exemplo de Cruzamento Uniforme	63
Figura 6.3 – Exemplo de Cruzamento com 1-Partição	63
Figura 6.4 – Exemplo de Cruzamento com 2-Partições.....	64
Figura 6.5 – Exemplo de Mutação por Troca Simples.....	65
Figura 6.6 – Exemplo de Mutação por Inversão.....	65
Figura 6.7 – Exemplo de Mutação por Translocação	65
Figura 7.1 – Pontos que Definem a Área de Atendimento Estudada.....	78
Figura 7.2 – Destaque das 12-Medianas em Relação aos Demais Pontos de Parada	81
Figura 7.3 – Determinação dos 12 <i>Clusters</i> Definidos pelo Algoritmo M_Dep_Rz.....	83
Figura 7.4 – Recuperação da Rota 272 – Algoritmo do Carteiro Chinês.....	88
Figura A.1 – Mapa do <i>Cluster</i> 272	105
Figura A.2 – Grafo do <i>Cluster</i> 272	106
Figura A.3 – Mensagem de Solução (Software LINGO).....	107
Figura A.4 – Grafo G	120
Figura A.5 – Grafo $G-(M^*)$	121
Figura A.6 – Grafo G	123
Figura A.7 – Grafo G e Matriz de Distância C	125
Figura A.8 – Mensagem de Solução (Software LINGO).....	126

LISTA DE FLUXOGRAMAS

Fluxograma 7.1 – Fases de Resolução do Problema	75
Fluxograma 7.2 – Resumo das Etapas de Resolução	76

LISTA DE QUADROS

Quadro 3.1 – População Urbana Atendida pela Sanepar	8
Quadro 3.2 - Total de Ligações de Água, Esgoto e População Atendida	9
Quadro 3.3 – Comparação entre Sistema Convencional de Faturamento e o Utilizado pela Sanepar.....	10
Quadro 3.4 – Distância Percorrida, em metros, pelos Leituras, nas Rotas Existentes Atualmente	12
Quadro 7.1 – Resultados Numéricos das Simulações para o Problema das 12-Medianas	79
Quadro 7.2 – As 12-Medianas Encontradas em Cada uma das 5 Simulações.	80
Quadro 7.3 – Formação dos 12 <i>Clusters</i> , Utilizando o Algoritmo M_Dep_Rz.....	82
Quadro 7.4 – Resultados Obtidos pelo Software LINGO.....	85
Quadro 7.5 - Distância Percorrida, em metros, pelos Leituras, nas Rotas Obtidas pelo Algoritmo do Carteiro Chinês	86
Quadro A.1 – Coordenadas Geográficas.....	111
Quadro A.2 – Distâncias Euclidianas	111

RESUMO

Neste trabalho é proposta uma solução para o trabalho de leitura dos medidores das contas de água dos clientes, efetuado por uma empresa de saneamento básico do município de Pato Branco, utilizando algumas técnicas da Pesquisa Operacional.

A metodologia aqui apresentada divide o trabalho em duas fases distintas: a primeira onde são formados os grupos de atendimento para cada leiturista; e a segunda fase, onde propõe-se a melhor rota a ser seguida, de modo a minimizar a distância percorrida por cada trabalhador.

Na primeira fase, compara-se a meta-heurística do algoritmo genético com a heurística clássica de Teitz e Bart para a definição de 12-medianas, que em seguida servem como sementes para a formação dos 12 grupos de atendimento dos leiturstas, obtidos através do algoritmo de designação de Gillett e Johnson.

O roteamento de cada um destes grupos caracteriza um problema de cobertura de arcos, que é resolvido na segunda fase, utilizando a formulação matemática do Problema Carteiro Chinês e o Algoritmo do Carteiro Chinês.

O objetivo principal do trabalho é obter, de forma rápida e eficiente, o caminho otimizado de percorrida do leiturista para que se gaste menos tempo com o seu deslocamento, oferecendo um atendimento com maior qualidade aos usuários desses serviços.

ABSTRACT

This study proposes a solution for customers' water meter bill reading task, executed by a basic sanitation company of Pato Branco, through the utilization of some Operational Research techniques.

The methodology presented here divides the work in two distinct phases: the first, where the service groups for each meter reader are formed and; the second, where the best route to be followed is proposed, in order to minimize the service distance for each server.

In the first phase, the meta-heuristic of the genetic algorithm is compared to the Teiz and Bart's classical heuristics. This defines twelve medians that will serve as seeds to form the twelve groups of meter reader service, obtained through the Gillett and Johnson's designation algorithms.

The routing of each group characterizes an arc covering problem, which is solved in the second phase by using the Chinese Postman Problem mathematical form and the Chinese Postman Algorithms.

The main purpose of the work is to obtain, quickly and efficiently, the optimized service distance for the meter reader so that less time is spent on his covering cycle and better quality service is offered to the users of this system.

CAPITULO I

1. Introdução

Serviços do setor público, como por exemplo, varredura de ruas, coleta de lixo, roteamento de carteiros, inspeção de linhas de água, eletricidade ou gás, são realizados a partir de uma grande utilização de recursos humanos, visto que, para a execução dessas tarefas é necessário haver a caminhada ao longo do trecho trabalhado.

Assim, a proposta deste trabalho relaciona-se à leitura das ligações de água, no que se refere a melhoria do nível de atendimento prestado pela Companhia de Saneamento do Paraná (Sanepar).

1.1 Objetivos do Trabalho

Os objetivos do trabalho são:

- a) Estabelecer a área de atendimento para leitura das ligações de água para cada leiturista, levando em consideração que a quantidade de faturas emitidas seja, aproximadamente, a mesma para todos eles.
- b) Determinar, para cada leiturista, o percurso a ser feito dentro de sua área de atendimento, adequando-o as características do trabalho do leiturista.

1.2 Importância do Trabalho

Os serviços de saneamento básico oferecidos pela Sanepar, são operacionalizados pelos agentes ou leiturstas, que fazem a leitura, digitam os dados no microcomputador, emitem a fatura e a entregam ao cliente; além disso, é função do leiturista, conferir dados

do logradouro, prestar informações diversas, vender serviços como consertos dos mais diversos e verificar as condições de instalação da ligação predial de água (conjunto formado pelo ramal predial e o cavalete) e hidrômetro dos locais visitados.

Por isso a necessidade de otimizar, ao máximo, o caminho de percorrida do leiturista para que se gaste menos tempo com o seu deslocamento, oferecendo um atendimento com maior qualidade aos usuários desses serviços.

1.3 Limitações do Trabalho

Uma limitação do trabalho é o fato de ter sido considerada apenas uma determinada área do município de Pato Branco para estudo, e não a cidade toda. Além disso, não seria viável o cadastramento dos 2.932 clientes desta área no mapa digitalizado. Desta forma, 774 pontos representarão estes clientes, como será detalhado no capítulo VII.

1.4 Estrutura do Trabalho

O presente trabalho está estruturado em sete capítulos, além desta introdução.

No segundo capítulo a revisão de literatura vem fundamentada em relatos de trabalhos relacionados ao problema abordado.

O terceiro capítulo destina-se a descrição real do problema, detalhando as características do mesmo, bem como a metodologia utilizada, atualmente, para solucioná-lo.

O quarto capítulo é dedicado, exclusivamente, ao estudo de grafos, no que tange ao problema abordado.

O problema de roteamento de veículos é definido no quinto capítulo, bem como as estratégias de solução do mesmo.

A descrição dos métodos propostos para solução do problema real é apresentada no sexto capítulo.

No capítulo sete fez-se a implementação computacional das técnicas citadas, incluindo os resultados numéricos obtidos.

Finalmente, o capítulo oito, traz a análise dos resultados, as conclusões finais e as sugestões para trabalhos futuros.

CAPITULO II

2. Revisão de Literatura

Este capítulo faz referência a alguns trabalhos relacionados aos problemas abordados nesta pesquisa: localização de facilidades, formação de agrupamentos, algoritmos genéticos, problema das p-medianas e utilização do Problema do Carteiro Chinês nos Problemas de Roteamento de Veículos.

Os problemas de localização de facilidades constituem uma importante classe de problemas dentro da Otimização Combinatória e, pela sua importância, têm recebido considerável atenção de pesquisadores de diversas áreas. Seu estudo, neste trabalho, torna-se necessário porque, a partir da localização da facilidade (depósito fictício para o leitorista), será formado o grupo de atendimento (*cluster*) para o mesmo. Assim, citam-se alguns autores que publicaram trabalhos nesta área.

LOBO, 1998, cita WEBER como o pioneiro da era moderna. Em um trabalho publicado em 1909, buscava a localização de uma indústria visando a minimização dos custos de transportes da matéria prima e do produto até o mercado consumidor.

TEITZ e BART [1968] propuseram um método aproximado, com bons resultados para a época, para se encontrar a mediana de um grafo ponderado. O método, que procura a solução para o problema através de troca de vértices, a partir de uma solução inicial, é descrito e utilizado neste trabalho (capítulo VI e capítulo VII, adiante).

BEZERRA, 1995, utilizou o modelo das p-medianas para a localização de postos de coleta para apoio ao escoamento de produtos extrativistas, num estudo de caso aplicado ao babaçu, no estado do Piauí, visando racionalizar o trabalho de transporte entre o local de coleta e o local de processamento. Para a otimização do problema foram utilizados três algoritmos: o algoritmo de Floyd para determinar as distâncias mínimas

entre cada nó que compõem a rede viária, o algoritmo das p-medianas para localização dos postos de coleta, e um algoritmo genético para encontrar as distâncias e o roteiro para os problemas do caixeiro viajante. Para avaliar as soluções obtidas neste problema de implantação dos postos de coleta, foi feita uma análise de viabilidade econômica.

ROSA, 1996, apresenta um estudo de caso com duas abordagens para o problema de localização de Centros de Distribuição Domiciliada (CDDs) da ECT. A primeira, trata do problema de cobertura de conjuntos que procura minimizar o número de CDDs (facilidades) necessários para atender toda a cidade satisfazendo padrões de serviços fixos. E a segunda abordagem traz o Problema das P-medianas (“Mini-sum”) cujo objetivo é minimizar o percurso médio dos carteiros de ida e volta dos CDDs as suas regiões de entrega.

CORRÊA, 2000, desenvolve o estudo de um algoritmo que otimiza a designação de candidatos ao vestibular para os locais de provas mais próximos de suas residências. Para resolver o problema das p-medianas capacitado são utilizadas, por ele, duas heurísticas modernas adaptadas ao problema: A primeira é baseada em um algoritmo genético que utiliza os operadores genéticos usuais e um operador heurístico chamado “hipermutação direcionada”. A segunda heurística proposta é baseada em busca tabu e usa memória de curto e de longo prazo para controlar a busca.

Os algoritmos genéticos surgem como uma meta-heurística e vêm sendo aplicados com sucesso nos mais diversos problemas de otimização como localização de facilidades e roteamento de veículos. Segundo DIAS e BARRETO [1998] esta técnica destaca-se pela simplicidade de operação, facilidade de implementação e por ser aplicável em situações onde não se conhece o modelo matemático ou se este for impreciso. Alguns trabalhos destacam-se nesta área:

MAYERLE, 1994, apresenta um algoritmo genético para solução do problema do caixeiro viajante. A partir de uma população aleatória inicial são realizadas operações de cruzamento e mutação, fazendo com que novos indivíduos sejam gerados e incorporados à população inicial. Tal qual na natureza, os indivíduos com as características mais fortes sobrevivem e, consequentemente, participam do processo de geração de descendentes. A fim de demonstrar a eficiência do algoritmo proposto, um conjunto de problemas,

variando-se o número de nós, é resolvido e, os resultados obtidos, são comparados com outras técnicas de solução.

NUNES, 1998, mostra uma abordagem a um problema real de roteamento de veículos, objetivando racionalizar o sistema de transporte dos funcionários de uma empresa, de forma a reduzir a quantidade de veículos utilizados e minimizar o caminho total percorrido. Utilizando os algoritmos genéticos ele encontrou as *p*-medianas (*p* pontos notáveis dentre os diversos pontos de parada) para determinação da semente dos *clusters* e também utilizou os algoritmos genéticos para construção de uma seqüência para coleta dos funcionários em cada um dos *clusters*. A formação dos *clusters* foi determinada a partir da utilização de uma variação do algoritmo de designação de Gillet e Johnson.

SAMPAIO, 1999, elaborou um modelo que busca a localização ideal de escolas do ensino fundamental, visando minimizar o caminho percorrido pelos alunos desde suas residências até a escola mais próxima. A metodologia proposta por ela utiliza as técnicas de *Simulated Annealing* e Algoritmo Genético comparativamente, considerando a localização da residência dos alunos um fator fundamental na determinação da posição ou ampliação de uma escola.

O Problema do Carteiro Chinês caracteriza o problema de cobertura de arcos, que deve no problemas de roteamentos de veículos sobre uma rede viária percorrer todos os arcos exatamente uma única vez. São problemas aplicados como roteamento de carteiros, varredura de ruas, vistorias de ligações de água, luz e gás. O seu estudo pode ser destacado por alguns autores:

COSTA, 1997, propôs uma solução para o problema de entrega de correspondências realizada pelos serviços postais, dividindo esse trabalho em duas fases: a primeira, que determinou a divisão de uma região da cidade em sub-regiões (áreas de atendimento) e a segunda, determinou, para cada carteiro, o roteiro de entrega das correspondências. Foram utilizadas heurísticas em ambas as fases, com destaque para a segunda, onde utilizou-se o Algoritmo do Carteiro Chinês como proposta de solução para o problema de cobertura de arcos.

STERN e DROR, 1978, aplicaram as técnicas do Algoritmo do Carteiro Chinês, no estudo das rotas dos Leituristas de Medidores Elétricos, num problema de roteamento de arcos orientados. O resultado apresentou uma redução de 40% do número de trabalhadores empregados, mostrando a aplicabilidade deste estudo.

EGLESE e MURDOCK, 1991, desenvolveram um software para auxiliar a limpeza de ruas com veículos vassoura. Este estudo foi aplicado no noroeste da Inglaterra, com a intenção de desenvolver uma rota a ser seguida por cada varredor cuja distância total percorrida fosse minimizada, sujeita a quantidade de trabalho e o tempo hábil em cada dia.

GHIANI e IMPROTA, 2000, apresentaram uma variante do clássico problema do Carteiro Chinês, o problema do Carteiro Chinês Hierárquico, cujos arcos são divididos em *clusters* e a relação precedente é definida nos *clusters*. Sua aplicação prática pode ser vista nos controles de neve e gelo nas ruas e estradas.

O tratamento dado pelos pesquisadores aos problemas de localização de facilidades e de roteamento de veículos é bastante diversificado, variando não apenas nos algoritmos utilizados, mas também no tipo de tratamento dado às particularidades próprias de cada problema.

CAPITULO III

3. Descrição do Problema Real

3.1 Introdução

A Companhia de Saneamento do Paraná (Sanepar), empresa estadual de economia mista, é responsável pelas ações de saneamento básico em quase todo o Estado do Paraná, Brasil.

O saneamento básico do Paraná, no início da década de 60, tinha um índice de atendimento muito baixo. Apenas 8,3% da população era servida por rede de abastecimento de água, e apenas 4,1% com rede de esgoto. Das 221 sedes municipais da época, 19 possuíam todos os serviços e 37 somente o de água. Das 20 cidades mais populosas do Estado, segundo o censo de 1950, apenas 11 tinham serviço de água.

Hoje 98% da população urbana é abastecida com água tratada, são mais de 7 milhões de habitantes, distribuídos entre as 619 localidades atendidas pela Sanepar, o que pode ser visualizado no quadro 3.1 abaixo:

Quadro 3.1 – População urbana atendida pela Sanepar

Número de ligações de água do Estado	1.846.609
Número de ligações de esgoto do Estado	600.226
População urbana atendida com água	7.361.594
População urbana atendida com esgoto	2.799.100
Nível de atendimento com água	98,59%
Nível de atendimento com esgoto	37,46%
Localidades atendidas com água	619
Localidades atendidas com esgoto	133

A Sanepar, unidade de Pato Branco, centraliza o atendimento das cidades: Chopinzinho, Clevelândia, Coronel Vivida, Itapejara do Oeste, Mangueirinha, Mariópolis, Palmas, Pato Branco, São João e Vitorino. A seguir, tem-se o quadro 3.2 com o total de ligações de água, esgoto e a população atendida pelos serviços da Sanepar nesta unidade.

Quadro 3.2 - Total de ligações de água, esgoto e população atendida

Local	Número de Ligações		Nível de Atendimento		
	Ligações de Água	Ligações De Esgoto	Água	Esgoto	Esgoto Tratado
Chopinzinho	2.846	668	98,99%	22,67%	87,30%
Clevelândia	3.883	1.157	99,24%	30,07%	100,00%
Coronel Vivida	3.850	922	98,83%	23,44%	100,00%
Itapejara do Oeste	1.440		98,42%		
Mangueirinha	1.820	730	99,09%	39,15%	100,00%
Mariópolis	1.011		98,35%		
Palmas	7.109	911	99,09%	13,48%	100,00%
Pato Branco	14.097	5.832	99,27%	47,73%	100,00%
São João	1.320	375	99,07%	25,90%	100,00%
Vitorino	905		98,79%		
Total	38.281	10.595	99,10%	30,50%	99,36%

As ligações podem ser classificadas como:

- água (somente água);
- esgoto (água é conseguida através de recurso ou fonte própria);
- água e esgoto.

Estas ligações atendem usuários dos seguintes grupos:

- residenciais;
- comerciais;
- industriais;
- poder público;
- utilidade pública.

Após a distribuição da água, existe o trabalho de faturar os consumos. A empresa Mercado é uma empresa terceirizada que realiza esta atividade em Pato Branco, usando um sistema de processamento e emissão de fatura no próprio local do consumo. A

utilização de microcomputadores portáteis reduz, em muito, os custos operacionais se comparada aos sistemas convencionais de faturamento, conforme pode-se observar no quadro 3.3.

Quadro 3.3 – Comparação entre Sistema convencional de faturamento e o utilizado pela Sanepar

Sistema convencional	Sistema atual utilizado pela Sanepar
1 O funcionário coleta os dados apresentados pelo medidor.	1 O leiturista digita os dados colhidos do medidor em seu equipamento portátil. A fatura é emitida e entregue ao cliente na mesma hora.
2 A informação é levada até a sede da empresa, onde é feita a digitação, consistência, processamento e a impressão da fatura.	
3 O funcionário retorna ao endereço dias depois, para entregar a fatura ao cliente.	

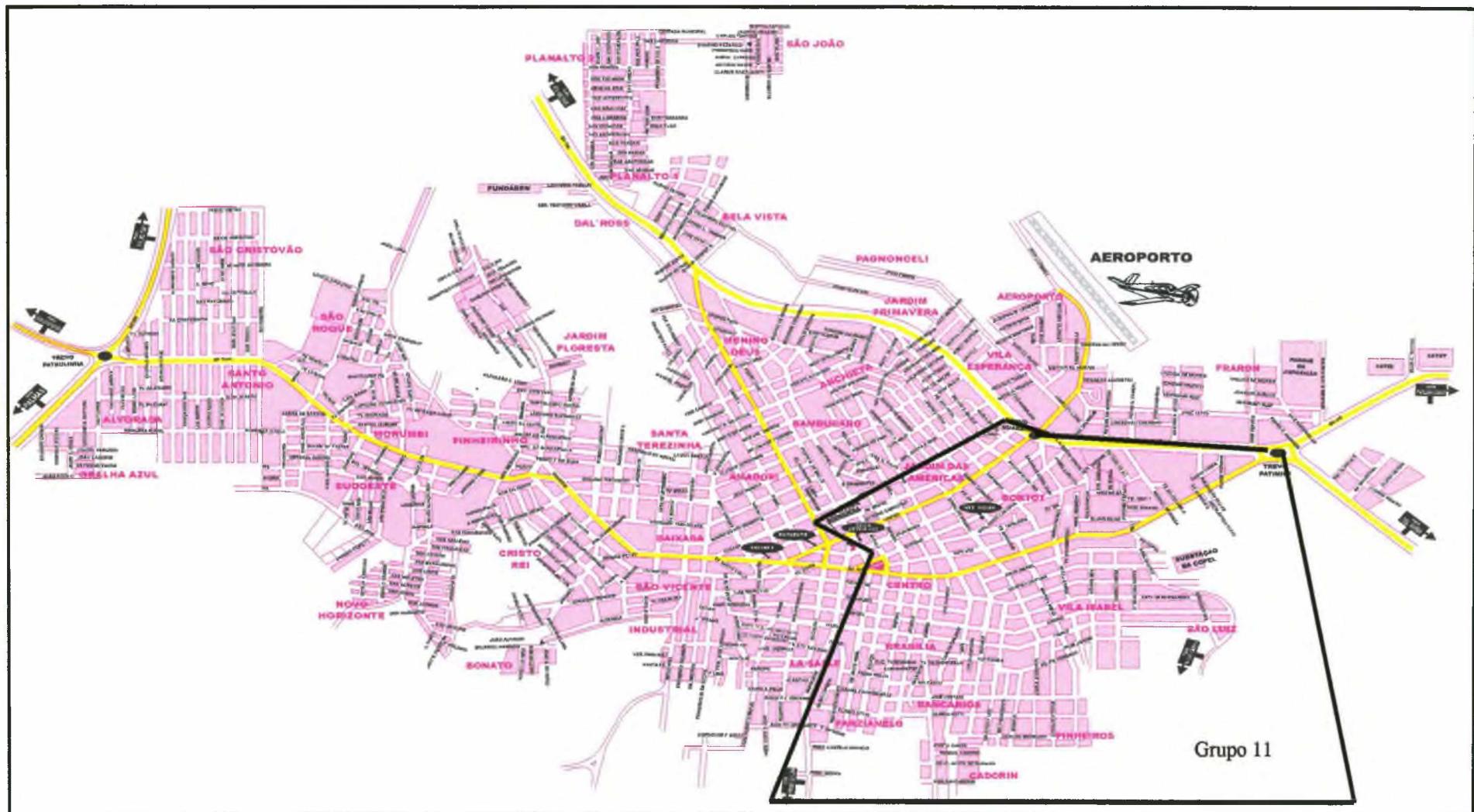
3.2 Descrição da Situação Atual com a Utilização do Roteamento Existente no Município de Pato Branco

Em Pato Branco, existem 14.097 ligações de água. Para a realização da leitura das contas de água, a cidade está dividida, atualmente, em 07 grupos, de onde considerou-se, para fins de estudo neste trabalho, o grupo 11 que é formado pelos bairros: Centro (parte), Parzianelo, Cadorin, Bancários, Pinheiros, Brasília, Vila Isabel, São Luiz, Bortot, Trevo da Guarani (Parte).

Este grupo foi escolhido por apresentar características de centro e bairros. Apresenta trechos com vários clientes e também longos trechos com poucos clientes.

O mapa da cidade de Pato Branco, com o grupo 11 em destaque, encontra-se na Figura 3.1.

Figura 3.1 – Mapa da cidade de Pato Branco – PR (Destacando-se o grupo 11)



Esse grupo possui 2.932 ligações e está dividido, atualmente, em 12 rotas para os leituristas. A distância percorrida e o número de ligações que devem ser visitadas por cada leiturista nas rotas já existentes estão distribuídas conforme informações contidas no quadro 3.4.

Quadro 3.4 – Distância percorrida, em metros, pelos leituristas nas rotas existentes atualmente.

Rota	Distância percorrida (m)	Número de ligações
11005	9.512	240
11010	8.791	248
11015	7.639	251
11020	7.816	249
11025	8.674	253
11030	9.553	222
11035	8.182	243
11040	11.180	253
11045	8.103	251
11050	7.620	241
11055	8.063	268
11060	10.660	213
Total	105.793	2932

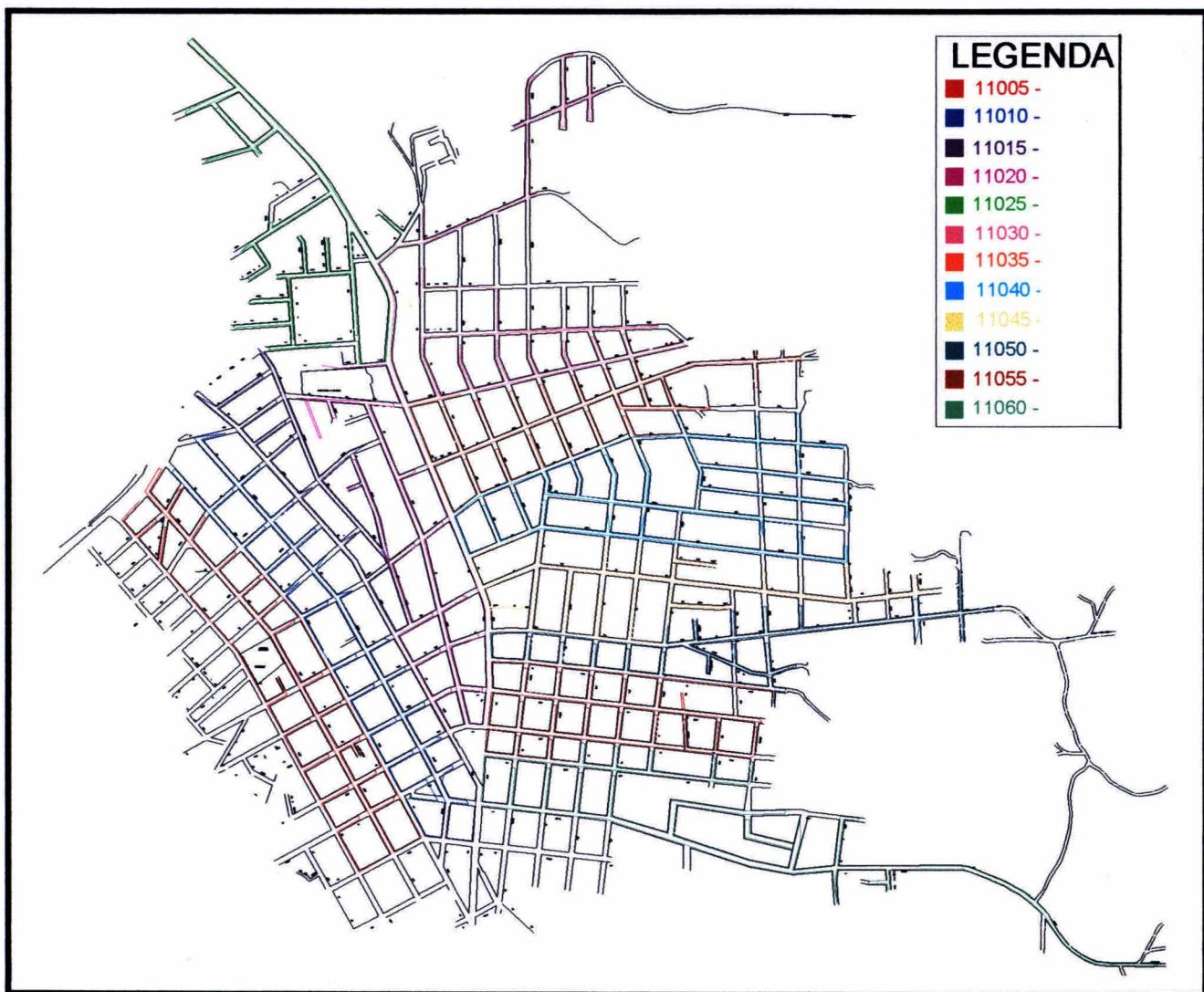
De acordo com as informações contidas no quadro acima, tem-se que para efetuar a leitura de 2.932 ligações, a distância total percorrida é 105.793 metros. Doze leituristas realizam esse trabalho ao longo de um dia.

Observa-se, por exemplo, que o leiturista **11060** efetuou, neste mês, a leitura de 213 ligações. Isso representa quase 20% a menos que o número de leituras efetuadas pelo leiturista **11055**, que realizou 268 leituras. Porém, pode-se notar que a distância percorrida pelo leiturista **11060** foi de 10.660 metros, o que representa 32% a mais que a distância percorrida pelo leiturista **11055**, que foi de 8.063 metros. Assim, conclui-se que,

apesar de ter efetuado menos leituras, gastou praticamente o mesmo tempo, em virtude do tempo gasto com a distância percorrida.

Na Figura 3.2, a seguir, são apresentadas as rotas existentes atualmente.

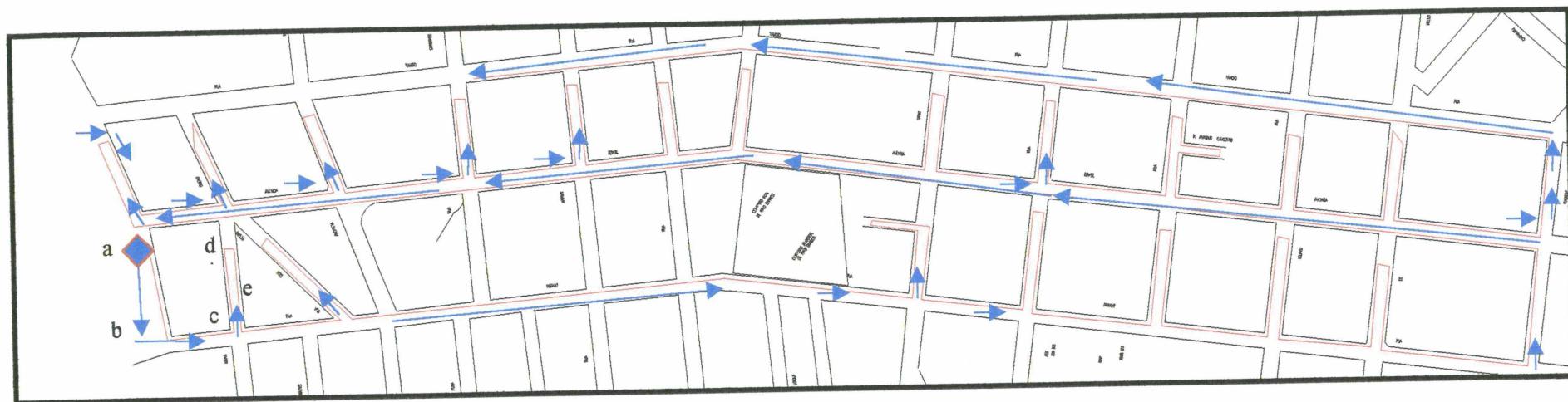
Figura 3.2 – Visualização das rotas existentes atualmente



As rotas existentes atualmente em toda a rede viária do município de Pato Branco, são elaboradas manualmente.

Na figura 3.3 pode-se visualizar a sequência da rota 11005.

Figura 3.3. Visualização da sequência da rota 11005



De acordo com a figura acima, percebe-se que a rota segue sempre pela “esquerda” do leiturista. Ele inicia a rota no ponto **a**, faz as leituras a sua esquerda chega ao ponto **b**, continua até o ponto **c**, fazendo as leituras que existem a sua esquerda, faz a quadra e quando chega ao ponto **d** atravessa a rua retornado ao ponto **c**, assim, o lado **e** passa a ser sua esquerda. O leiturista segue este critério até o final de sua rota.

CAPITULO IV

4. Algumas Noções Básicas Sobre Grafos

4.1 Introdução

Tendo o estudo de grafos como pressuposto importante para o desenvolvimento desta pesquisa, são apresentadas a seguir definições relevantes feitas por estudiosos como CHRISTOFIDES [1975], BOAVENTURA NETO [1979] e RABUSKE [1992].

De acordo com RABUSKE [1992], a teoria dos grafos apresenta-se como ferramenta simples, acessível e poderosa para construção de modelos e resolução de problemas relacionados com arranjos de objetos discretos. A tecnologia possui hoje um grande número de problemas que requerem a construção de sistemas complexos, devido às combinações de seus componentes. Estes problemas abrangem processos industriais, análise de caminho crítico, tática e logística (campo militar), sistemas de comunicação, estudo de transmissão de informações, escolha de uma rota ótima, fluxos em redes, redes elétricas (engenharia elétrica e civil, arquitetura, computação), genética, psicologia, economia, estrutura social, jogos, física, química, tecnologia de computador, antropologia, lingüística, etc.

Baseada na simples idéia de pontos interligados por linhas, a teoria dos grafos combina esses ingredientes básicos em um rico sortimento de formas e as dota com propriedades flexíveis, fazendo, assim, com que esta teoria seja uma ferramenta útil e poderosa em Pesquisa Operacional em geral e, especificamente, em problemas de roteamento.

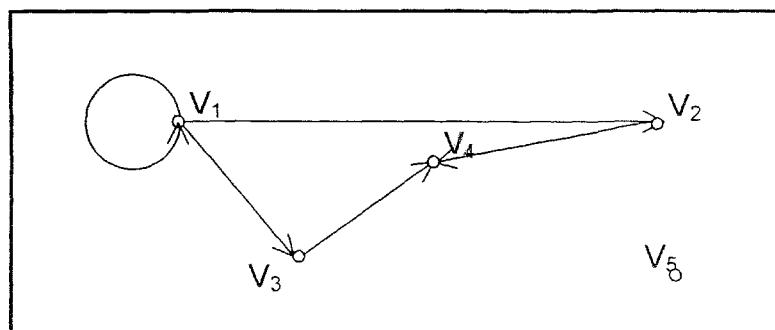
4.2 Definições

Seja V um conjunto finito e não vazio, e E uma relação binária sobre V . Os elementos de V são representados por pontos. O par ordenado $(v,w) \in E$, (ou simplesmente vw), onde $v,w \in V$, é representado por uma linha ligando v a w . Tal representação de um conjunto V e uma relação binária sobre o mesmo, é denominada um grafo $G(V, E)$.

Em outras palavras, um grafo G é definido como sendo um par ordenado (V, E) . Os elementos de V são denominados de **vértices** ou **nós** e os pares ordenados de E , denominados de **arestas** ou **arcos** do grafo. Quando as linhas possuem orientação, usualmente representado por uma seta, essas são chamadas de arcos. Se as linhas não têm orientação, são chamadas de arestas. Uma aresta/arco é dita incidente com os vértices que ela liga. Uma aresta/arco incidente a um único vértice é denominado um laço. Dois vértices são ditos **adjacentes**, se eles estão ligados por uma aresta/arco.

A Figura 4.1 mostra uma representação geométrica do grafo $G(V, E)$, onde v_5 é um vértice isolado (não há aresta incidindo sobre ele) e o arco (v_1, v_1) é um laço.

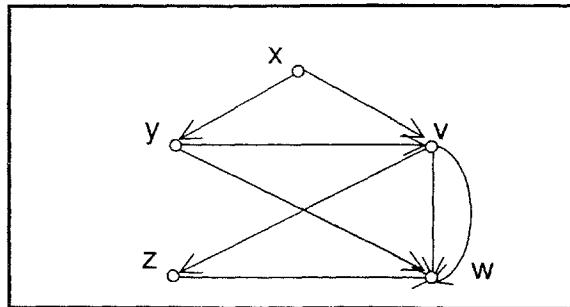
Figura 4.1 – Representação Geométrica de um grafo $G(V, E)$, onde $V = \{v_1, v_2, v_3, v_4, v_5\}$ e $E = \{v_1v_2, v_1v_3, v_2v_4, v_3v_4, v_1v_1\}$.



O cardinal $|V| = n$ é a ordem de G , adotando-se $|E| = m$, sem designação específica.

Duas arestas que incidam sobre o mesmo vértice são ditas adjacentes. Se existem duas arestas $e_i = (v, w)$ e $e_j = (v, w)$, então diz-se que e_i e e_j são arestas paralelas, conforme apresentado na Figura 4.2.

Figura 4.2 – Representação Geométrica de $V = \{x, y, z, v, w\}$ e $E = \{xy, xv, yv, yw, vz, vw, vw, zw\}$, onde as arestas vw são paralelas.



Se um grafo possui laços e/ou arestas paralelas, é denominado de multigrafo, do contrário, diz-se que o grafo é simples.

Um grafo simples, em que cada par distinto de vértices é adjacente, é denominado grafo completo. O grafo completo de n vértices é usualmente representado por K_n , e denominado clique. Todo grafo completo de n vértices possui $m = \binom{n}{2}$ arestas.

Alguns grafos se notabilizam por peculiaridades em suas estruturas. Essas, podem ser bastante úteis na representação de situações reais ou na utilização de algoritmos de solução para os problemas em grafos.

Um grafo é dito dirigido (ou dígrafo), se possuir arcos (linhas com orientação), caso contrário, é dito não dirigido. Claramente, um grafo não dirigido é uma representação de um conjunto de vértices ou nós e uma representação simétrica binária sobre esse conjunto. Em um grafo não dirigido, uma aresta ligando dois vértices v e w pode ser representada por (v,w) o (w,v) indistintamente.

Uma cadeia de arestas, em um grafo, é qualquer sequência não direcionada de arestas, na qual cada aresta é conectada com suas arestas adjacentes através dos seus dois vértices terminais. Se nenhuma aresta aparecer mais que uma vez, tem-se uma cadeia simples. Quando, nesta sequência não direcionada, nenhum vértice figurar mais que uma vez, tem-se uma cadeia elementar.

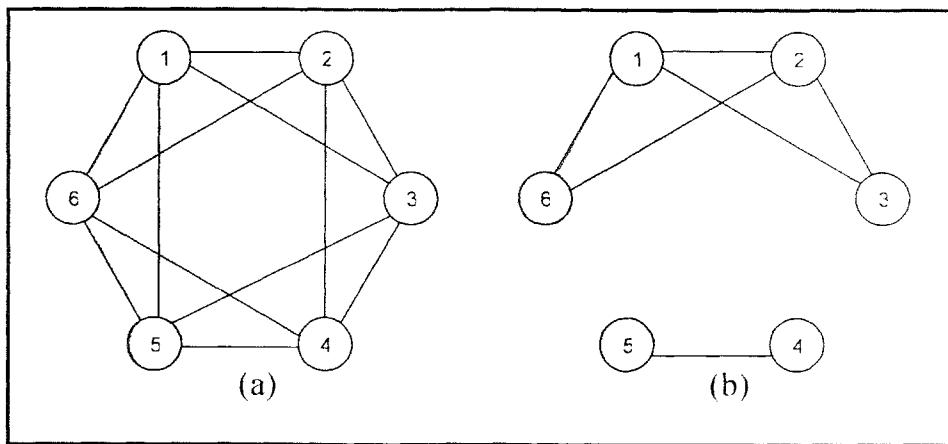
Um caminho, em um grafo dirigido, é qualquer seqüência de arcos onde o vértice final de um, é o vértice inicial do seguinte. Assim, caminho elementar é um caminho no

qual um vértice não figura mais que uma vez. Caminho simples é um caminho no qual um arco não aparece mais que uma vez.

Define-se um ciclo como uma cadeia fechada em um grafo G , ou seja, que inicia e termina no mesmo nó. Quando se tem um grafo G orientado, denomina-se circuito a sequência distinta de arestas que repetem o último nó visitado.

Um grafo G é dito conexo se para todo par de vértices existe, pelo menos, uma cadeia entre eles. Caso contrário, o grafo é dito não conexo. Por exemplo, o grafo da figura 4.3(a) é conexo e o de 4.3(b) é não conexo.

Figura 4.3 – Representação de grafos não dirigidos, onde em (a) tem-se um grafo conexo e em (b), um grafo não conexo.

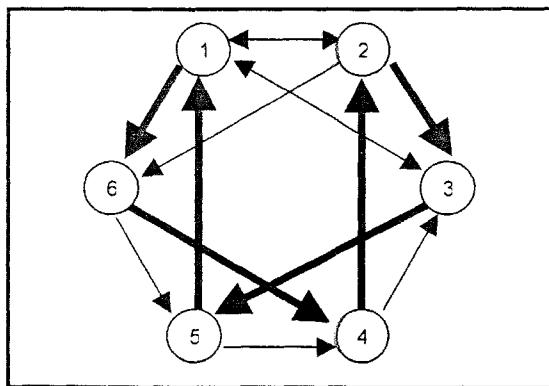


4.2.1 Ciclos e Circuitos Hamiltonianos

Um circuito hamiltoniano, em um grafo conexo G , é definido como um caminho simples fechado, isto é, passa-se em cada vértice de G exatamente uma vez, exceto naturalmente no vértice inicial que é considerado, também, vértice terminal. Portanto, um circuito hamiltoniano em um grafo de n vértices consiste em, exatamente, n arestas. Obviamente, nem todo grafo conexo possui um ciclo hamiltoniano.

Na Figura 4.4 tem-se uma representação geométrica de um Circuito Hamiltoniano.

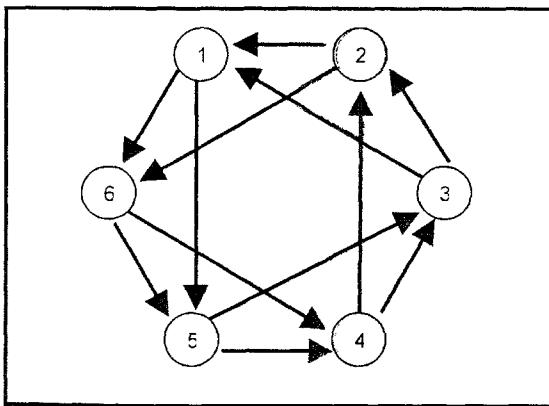
**Figura 4.4 – Representação Geométrica de um Circuito Hamiltoniano
(1 – 6 – 4 – 2 – 3 – 5 - 1)**



4.2.2 Ciclos Eulerianos

Um grafo não orientado conexo G é um grafo de Euler se, e somente se, todos os seus vértices são de grau par. Se um grafo G for euleriano, então é possível achar um caminho fechado, passando em cada aresta/arco uma única vez: este caminho/ciclo é denominado caminho/ciclo euleriano, representado na Figura 4.5.

**Figura 4.5 – Representação de um Ciclo Euleriano
(1 – 6 – 5 – 4 – 3 – 2 – 6 – 4 – 2 – 1 – 5 – 3 - 1)**



4.2.2.1 Definição de um Caminho Euleriano

Segundo BOAVENTURA, 1979, existe uma regra bastante fácil para traçar um caminho euleriano, utilizando o Algoritmo de Fleury, apresentado a seguir.

Algoritmo de Fleury

Seja G um grafo euleriano, então as seguintes construções são sempre possíveis e produzem um caminho euleriano em G .

Inicie em qualquer vértice v e atravesse as arestas de uma maneira arbitrária, segundo as seguintes regras:

R1) Apague a aresta que foi visitada e, se algum vértice ficar isolado, apague-o também;

R2) Em cada estágio, use um istmo (faixa de terra que liga uma península a um continente) somente se não houver alternativa, isto é, nunca atravesse uma aresta se, naquele particular momento, a remoção daquela aresta divide um grafo em duas ou mais componentes (excluindo os vértices isolados).

4.3 Representação de Grafos

A representação mais familiar de um grafo é através do desenho de pontos e linhas. Em computadores, o grafo pode ser representado através de matrizes de diversas maneiras. A eficiência do algoritmo vai depender da escolha certa de como representar um grafo.

4.3.1 Matriz de Adjacência

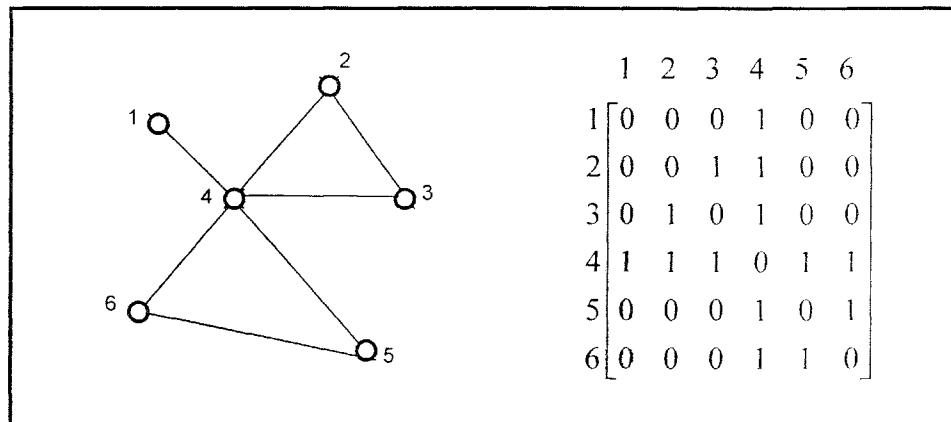
Dado um grafo $G(V, E)$, a matriz de adjacência $A = [a_{ij}]$ é uma matriz $n \times n$ tal que

$$a_{ij} = \begin{cases} 1, & \text{se e somente se existe } (v_i, v_j) \in E \\ 0, & \text{caso contrário.} \end{cases}$$

A matriz de adjacência requer n^2 bits. Para um grafo não dirigido, a matriz de adjacência é simétrica, portanto é suficiente armazenar somente a parte triangular superior. Obtém-se, assim, uma economia de 50% no armazenamento, mas o tempo computacional pode aumentar bastante devido à referência a_{ij} , que deve ser substituída

por “Se $i > j$ então a_{ij} senão a_{ji} ”. Na Figura 4.6 são apresentadas a representação geométrica de um Grafo G não dirigido e a Matriz de Adjacência correspondente.

Figura 4.6 – Representação Geométrica do Grafo G e sua Matriz de Adjacência.



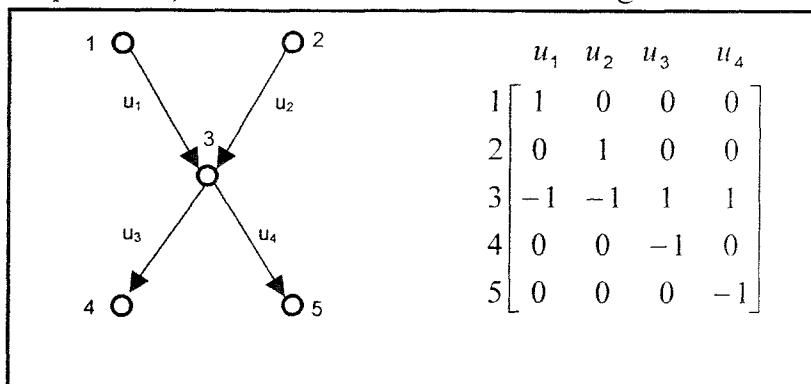
4.3.2 Matriz de Incidência

Uma matriz $A = [a_{ij}]$ de ordem $n \times m$ é denominada como de incidência do grafo $G(V, E)$ se, para todo arco j que liga o nó k ao nó l tem-se:

$$a_{ij} = \begin{cases} +1, & \text{se e somente se } i = k \\ -1, & \text{se e somente se } i = l \text{ (para grafo direcionado, senão } a_{ij} = 1) \\ 0, & \text{nos outros casos.} \end{cases}$$

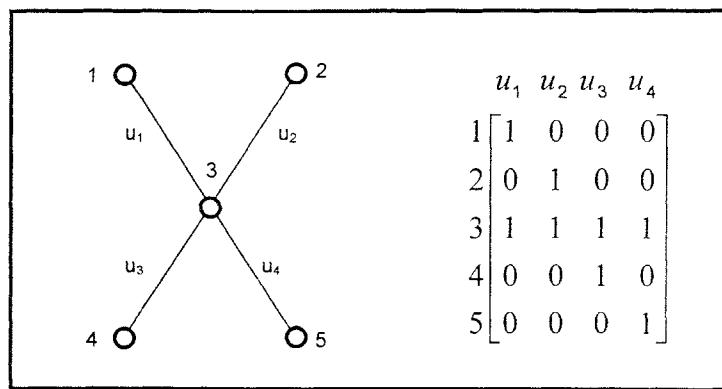
Na figura 4.7 a representação geométrica de um Grafo G dirigido e sua matriz de incidência .

Figura 4.7 – Representação Geométrica do Grafo G Dirigido e sua Matriz de Incidência



Na figura 4.8 a representação geométrica de um Grafo G não-dirigido e sua matriz de incidência .

Figura 4.8 – Representação Geométrica do Grafo G Não-Dirigido e sua Matriz de Incidência



4.3.3 Matriz de Custos

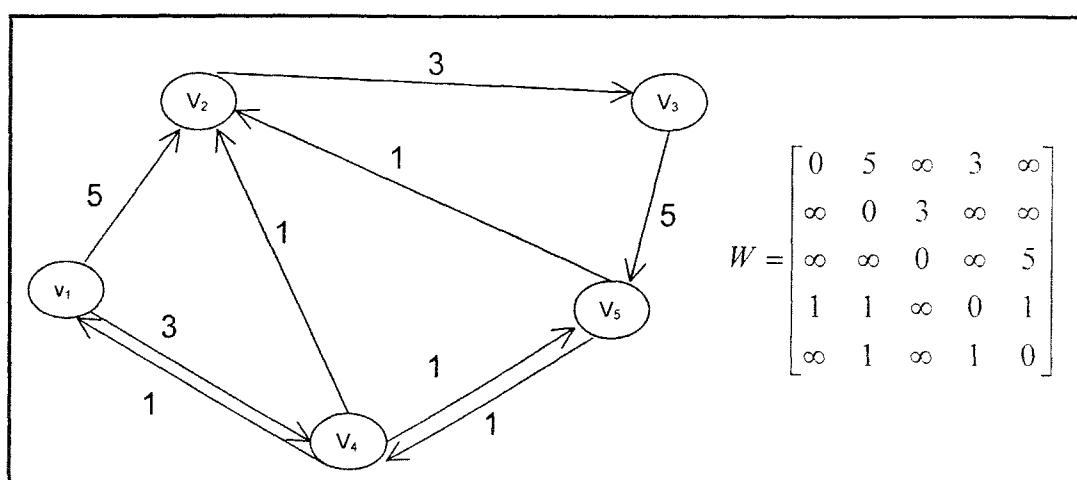
Um grafo, no qual um número w_{ij} está associado a cada aresta/arco, é denominado de grafo valorado e o número w_{ij} é chamado o custo da aresta/arco. Em redes ou transporte estes custos representam alguma quantidade física, tal como distância, eficiência, capacidade da aresta/arco correspondente, etc.

Um grafo simples valorado pode ser representado por sua matriz de custo $W = [w_{ij}]$, onde

$$w_{ij} = \begin{cases} \text{custo da aresta, se } (v_i, v_j) \in E \\ 0 \text{ (se } i = j) \text{ ou } \infty, \text{ caso contrário.} \end{cases}$$

Na figura 4.9 pode-se visualizar um grafo G simples valorado e sua matriz de custos.

Figura 4.9 – Representação Geométrica do Grafo G valorado e sua Matriz de Custos



CAPITULO V

5. Modelos de Roteamento de Veículos

5.1 Introdução

Um problema de roteamento pode ser considerado como um conjunto organizado de meios que objetiva o atendimento de demandas localizadas nos arcos/arestas ou nos vértices de alguma rede de transportes. O sistema de roteamento, como qualquer sistema operacional, pode ser decomposto, sob a ótica da operação, em três partes: estratégica, tática e logística.

Uma vez iniciado o processo de construção do sistema de roteamento, obviamente, outras decisões deverão ser tomadas. As decisões sobre número de rotas, sobre a forma de contratação de mão-de-obra ou regimes de trabalhos, são freqüentemente tomadas de uma forma mais localizada e específica.

Assim, define-se um problema de característica combinatória e de grande dificuldade de solução que denomina-se Problema de Roteamento de Veículos (PRV). Como elementos de entrada mais prováveis desse problema pode-se destacar [GOLDBARG,2000]:

- a área servida pelos depósitos;
- o tamanho da frota alocada a cada depósito;
- as regras que definirão as alocações de transporte a elementos da própria frota ou a unidades eventualmente contratadas.

De posse dessas informações, o objetivo do planejamento será estabelecer um roteamento e um seqüenciamento (*scheduling*) ou emprego de veículos que conduzam à

minimização do custo da atividade. Nesse ponto é pertinente esclarecer que a definição de uma função objetivo apropriada pode ser, por si só, uma tarefa não muito simples. O objetivo de reduzir custos poderá ser perseguido através da redução de:

- prazos de entrega (serviços de emergência, produtos perecíveis, etc.);
- caminhos a percorrer (combustível, manutenção, tempo de operação, etc.);
- emprego de mão-de-obra;
- número de veículos.

5.2 Problema de Roteamento de Veículos

A idéia básica do problema de roteamento é designar pontos de parada a veículos, bem como a determinação da seqüência em que estes pontos de parada serão visitados, estabelecendo as rotas para os veículos. Portanto o PRV é um problema, basicamente, euclidiano ou espacial.

O objetivo geral destes problemas é minimizar os custos associados ao sistema de transporte considerado. Os objetivos específicos estão ligados às particularidades de cada problema: determinar rotas de comprimento mínimo, minimizar a quantidade de veículos utilizados, diminuir o tempo total das rotas. Em qualquer caso deve-se levar em consideração diversas variáveis como: características dos veículos (capacidades, autonomias, quantidades), aspecto geométrico da rede (disposição dos nós e arcos/arestas do grafo), natureza das operações, entre outros.

5.3 Formulação para o Problema Geral de Roteamento de Veículos

Uma das formulações mais utilizadas como base a diversos métodos de solução é a de FISHER e JAIKUMAR [1981].

Considerando:

x_{ijk} ≡ variável binária que assume valor 1 quando o veículo k visita o cliente j imediatamente após o cliente i ; 0 em caso contrário;

y_{ik} é variável binária que assume valor 1 se o cliente i é visitado pelo veículo k ; 0 em caso contrário;

q_i é demanda do cliente i ;

Q_k é capacidade do veículo k ;

c_{ij} é custo de percorrer o trecho que vai do cliente i ao j . É o modelo matemático de Programação Linear Inteiro Binário para este problema definido como segue:

$$\text{Minimizar } z = \sum_{i,j} \left(c_{ij} \sum_k x_{ijk} \right)$$

sujeito a:

$$\sum_k y_{ik} = 1 \quad i = 2, \dots, n \quad (5.1)$$

$$\sum_k y_{ik} = m \quad i=1 \quad (5.2)$$

$$\sum_i q_i y_{ik} \leq Q_k \quad k=1, \dots, m \quad (5.3)$$

$$\sum_j x_{ijk} \leq \sum_j x_{jik} = y_{ik} \quad i=1, \dots, n \quad k=1, \dots, m \quad (5.4)$$

$$\sum_{i,j \in S} x_{ijk} \leq |S| - 1 \quad \forall S \subseteq \{2, \dots, n\}, k=1, \dots, m \quad (5.5)$$

$$y_{ik} \in \{0,1\} \quad i=1, \dots, n \quad k=1, \dots, m \quad (5.6)$$

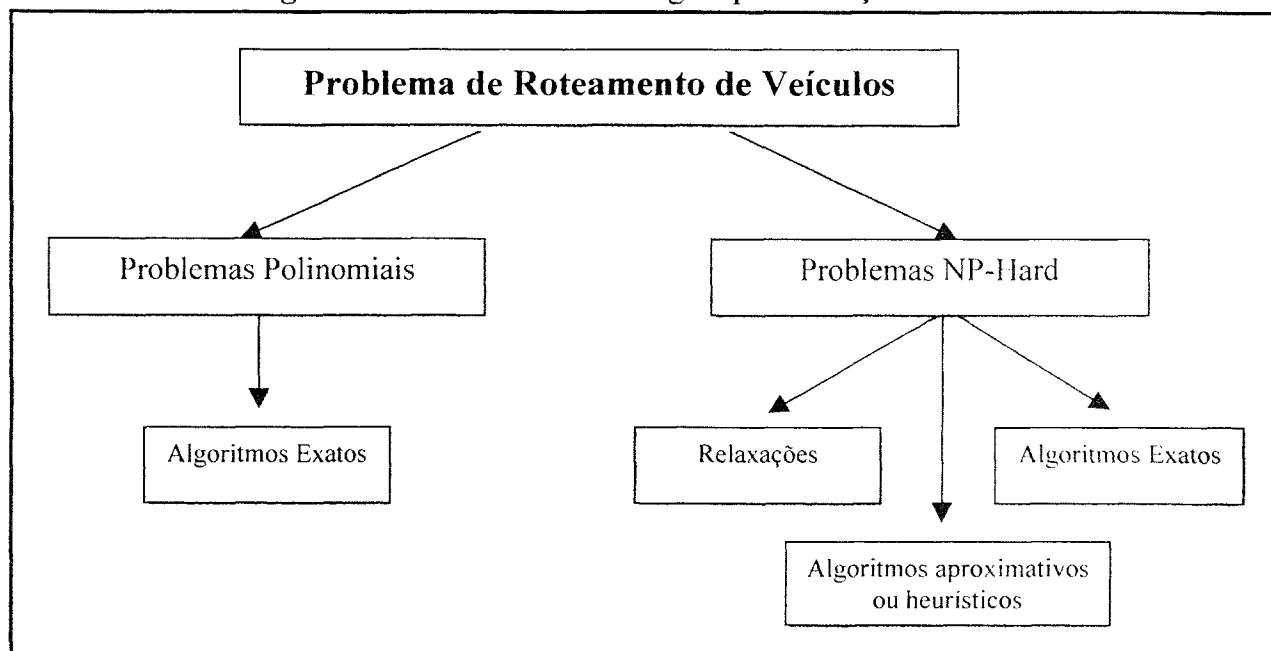
$$x_{ijk} \in \{0,1\} \quad i,j=1, \dots, n \quad k=1, \dots, m \quad (5.7)$$

A função objetivo minimiza os custos associados ao sistema de transporte considerado. A restrição (5.1) assegura que um cliente seja visitado uma e só uma vez. A restrição (5.2) garante que o depósito receba uma visita de todos os veículos. A restrição (5.3) obriga que as capacidades dos veículos não sejam ultrapassadas. A restrição (5.4) garante a continuidade das rotas. A restrição (5.5) garante as tradicionais restrições de eliminação de *subtours* (sub-rotas) e as restrições (5.6) e (5.7) definem as variáveis do problema como binárias.

5.4 Estratégias de Solução Para os Problemas de Roteamento

De acordo com GOLDBARG [2000], os problemas em que as variáveis assumem valores inteiros ou que possuem funções objetivo com descontinuidades não podem, geralmente, ser solucionados diretamente através de um modelo matemático de Programação Linear Inteiro (PLI) utilizando o algoritmo simplex. Esse é o caso de grande parte dos problemas de roteamento. Para tais problemas, tem-se na Pesquisa Operacional diversas formas e estratégias, sintetizadas na figura 5.1.

Figura 5.1 – Possíveis Estratégias para solução de PRV



5.5 Problemas Clássicos Básicos de Roteamento

Os principais problemas de roteamento de veículos, segundo BODIN [1983], são:

- i. O Problema do Caixeiro Viajante (PCV)
- ii. O Problema do Carteiro Chinês (PCC)
- iii. O Problema dos Múltiplos Caixeiros Viajantes (PCVM)
- iv. O Problema de Roteamento de Nós com um Único Depósito e Múltiplos Veículos (PRDMV)
- v. O Problema de Roteamento de Nós com Múltiplos Depósitos e Múltiplos Veículos (PRMDMV)
- vi. O Problema de Roteamento de Nós com Depósito Único, Múltiplos Veículos e Demanda Estocástica nos Vértices (PRDMVE)

vii. O Problema do Carteiro Chinês Capacitado (PCCC)

O estudo dos Problemas de Roteamento de Veículos tornou-se um desafio para a área de Otimização Combinatória da Pesquisa Operacional, após as divulgações dos artigos de DANTZIG e RAMSER [1959], os quais inspiraram o desenvolvimento de pesquisas neste assunto.

Segundo DASKIN [1985], duas abordagens básicas para roteamento de veículos têm sido adotadas, supondo que os veículos serão roteirizados numa rede composta por nós e arcos: problemas de cobertura de nós, que determinam a rota de comprimento mínimo que visita cada nó uma única vez; problemas de cobertura de arcos, que determinam uma rota de comprimento mínimo que atravessa cada arco uma única vez.

5.5.1 Os Problemas de Cobertura de Nós

Os problemas de cobertura de nós devem determinar uma rota de comprimento mínimo que visite cada nó uma única vez. Neste grupo de problemas encontra-se o problema do caixeiro viajante, o problema dos múltiplos caixeiros viajantes e os problemas de roteamento de veículos com um ou vários depósitos. A importância do problema do caixeiro viajante está diretamente associada com a proposta de solução de vários problemas do cotidiano.

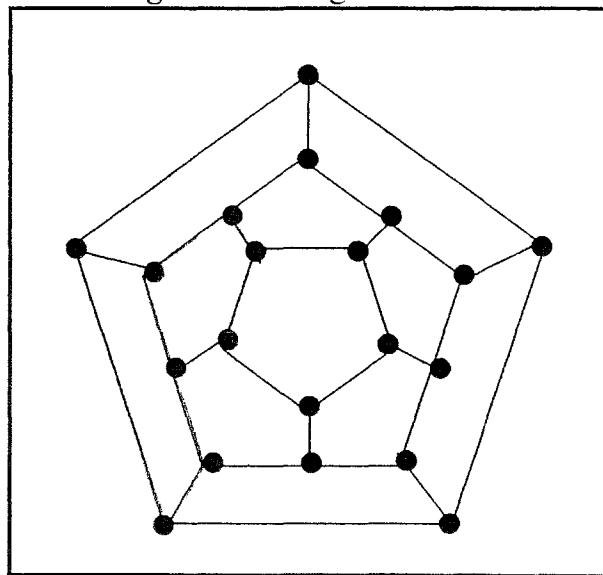
5.5.1.1 O Problema do Caixeiro Viajante (PCV)

5.5.1.1.1 As origens do Problema

O Problema do caixeiro viajante (PCV) é um dos mais tradicionais e conhecidos problemas de programação matemática. Os problemas de roteamento lidam, em sua maior parte, com passeios ou *tours* sobre pontos de demanda ou oferta. Esses pontos podem ser representados por cidades, postos de trabalho ou atendimento, depósitos, dentre outros. De acordo com GOLDBARG [2000], quanto aos tipos de passeios, destaca-se o hamiltoniano, um dos mais importantes. Seu nome é devido a Willian

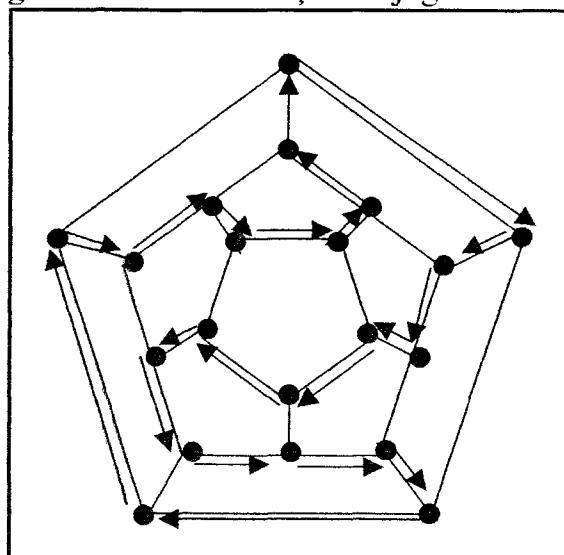
Rowan Hamilton que, em 1957, propôs um jogo que denominou *Around the World*. O jogo era feito sobre um dodecaedro em que cada vértice estava associado a uma cidade importante na época. O desafio consistia em encontrar uma rota através dos vértices do dodecaedro que iniciasse e terminasse em uma mesma cidade sem nunca repetir uma visita. O grafo do problema é mostrado na figura 5.2.

Figura 5.2 – Jogo de Hamilton



Uma solução do jogo de Hamilton, muitas vezes assim chamado em sua homenagem, passou a se denominar um ciclo hamiltoniano. Hamilton não foi o primeiro a propor esse problema, mas seu jogo o divulgou. Uma das soluções do jogo está apresentada na figura 5.3.

Figura 5.3 – Uma solução do jogo de Hamilton.



De acordo com GOLDBARG [2000], modernamente a primeira menção conhecida do problema é devida a Hassler Whitney, em 1934, em um trabalho na Princeton University. Independente desse trabalho de Hamilton, o problema do caixeiro viajante (PCV) é um problema de otimização associado ao da determinação dos caminhos hamiltonianos em um grafo qualquer. O objetivo do PCV é encontrar, em um grafo $G(N,A)$, o caminho hamiltoniano de menor custo.

5.5.1.1.2 Importância Atual do PCV

A importância do PCV é devida a pelo menos três de suas características:

- grande aplicação prática (REINELT [1994]);
- uma enorme relação com outros modelos (LAPORTE [1996]);
- grande dificuldade de solução exata (PAPADIMITRIOU e STEIGLITZ [1982]).

Dessa forma, a importância do modelo é indiscutível, tanto sob o aspecto prático, como o teórico. Considerado intratável por GAREY e JONHSON [1979] é classificado por KARP [1975] como NP-Hard.

5.5.1.1.3 Formulação matemática para o PCV

Assumindo que os custos são simétricos, isto é, $c_{ij} = c_{ji}$ e estabelecendo que $c_{ii} = +\infty$ para $i = 1,2,\dots,n$, o PCV consiste em formar uma rota passando por todos os nós, começando e terminando na origem, nó 1, que fornece a distância ou custo total mínimo. A formulação fornecida a seguir é devida a [GOLDEN, 1977] e [BODIN, 1983].

Sejam n o número de nós da rede, a_i a quantidade de arcos que possuem o nó i como vértice inicial, b_j a quantidade de arcos que possuem o nó j como vértice final, e x_{ij} a variável de decisão do problema, estabelecida como segue:

$$x_{ij} = \begin{cases} 1, & \text{se o arco } i - j \text{ está na rota} \\ 0, & \text{caso contrário,} \end{cases}$$

então o modelo matemático, onde a função objetivo (5.8) visa minimizar a distância ou custo total, fica definido do seguinte modo:

$$\text{Minimizar } \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (5.8)$$

s.a.:

$$\sum_{i=1}^n x_{ij} = b_j = 1 \quad (j = 1, 2, \dots, n) \quad (5.9)$$

$$\sum_{j=1}^n x_{ij} = a_i = 1 \quad (i = 1, 2, \dots, n) \quad (5.10)$$

$$X = (x_{ij}) \in S \quad (5.11)$$

$$x_{ij} = 0 \text{ ou } 1 \quad (i, j = 1, 2, \dots, n) \quad (5.12)$$

A formulação do problema seleciona uma matriz $X = [x_{ij}]$ de variáveis de decisão tal que exatamente um arco (i, j) , emana de cada nó e, exatamente um arco (i, j) , é dirigido para cada nó j . Isto implica numa designação de cada nó para seu nó sucessor na rota, que é estabelecido pela equações (5.9) e (5.10). A equação (5.12) define as variáveis do problema como binárias, caracterizando o PCV como um problema de Programação Linear Inteiro (zero ou um). Os requisitos para designação, entretanto, não garantem que a matriz X corresponda a uma rota. Para eliminar a possibilidade de formação de sub-rotas, restrições adicionais, fornecidas pela equação (5.11), são impostas nas escolhas para a seleção dos arcos na matriz X .

O conjunto S pode ser formado por quaisquer restrições que proíbam formações de sub-rotas que satisfaçam as restrições de designação (5.9), (5.10) e (5.12). Tais restrições são chamadas restrições de quebra de sub-rota. As possíveis escolhas para S incluem:

- $S = \{(x_{ij}): \sum_{i \in Q} \sum_{j \in Q} x_{ij} \geq 1 \text{ para todo subconjunto próprio não vazio } Q \text{ de } N, \text{ onde } N \text{ é o número de nós}\}$
- $S = \{(x_{ij}): \sum_{i \in R} \sum_{j \in R} x_{ij} \leq |R| - 1 \text{ para todo subconjunto não vazio } R \text{ de } \{1, 2, \dots, n\}\};$
- $S = \{(x_{ij}): y_i - y_j + nx_{ij} \leq (n-1) \text{ para } 2 \leq i \neq j \leq n \text{ para alguns números reais } y_i\}, \text{ onde } y_i = t, \text{ se o nó } i \text{ é visitado no passo } t \text{ da rota ou } y_i = 0, \text{ caso contrário.}$

Adicionando qualquer uma destas restrições ao problema, evita-se que ocorra formação de sub-rotas BODIN [1983].

Outras formulações são apresentadas por BODIN [1983], as quais são válidas por fornecerem idéias sobre a complexidade do PCV e sua relação com outros problemas de roteamento e distribuição, sugerindo algoritmos estratégicos e procedimentos heurísticos.

5.5.1.1.4 Procedimentos de Solução para os Problemas de Cobertura de Nós: Algoritmos Exatos, Relaxações e Algoritmos Aproximativos ou Heurísticos

A maioria dos procedimentos que fornecem a solução ótima para o PCV necessitam das formulações da Programação Inteira, embora a Programação Dinâmica também seja usada [CHRISTOFIDES, 1981]. Os algoritmos da Programação Inteira são baseadas em procedimentos do método *branch and bound*. Podem ser usados para solucionar uma série de problemas pequenos como, por exemplo, os que estão inseridos dentro dos problemas de designação.

Considerando o PCV, estes métodos são aplicáveis apenas para pequenos problemas, pois os algoritmos exatos disponíveis requerem um número de passos computacionais que crescem segundo uma função exponencial do número de pontos que precisam ser visitados. Assim, muitos procedimentos heurísticos foram e têm sido desenvolvidos.

Na solução do Problema do Caixeiro Viajante, as heurísticas existentes podem ser classificadas em [FISHER, JAIKUMAR, 1981]: procedimentos de construção de rota, algoritmos de melhoria de rota, técnica das duas fases (ou procedimentos compostos), ou seja, construção e melhoria, e métodos de otimização incompletos.

Outros métodos têm sido aplicados para solucionar o PCV, como por exemplo, a utilização de redes neurais [BARBOSA, 1989]; [BURKE, DAMANY, 1992], algoritmos genéticos [MAYERLE, 1994], *simulated annealing* [BARBOSA, 1989] e *guided local search* [VOUDOURIS, 1996].

5.5.2 Os Problemas de Cobertura de Arcos

Os problemas de cobertura de arcos determinam um caminho mínimo através de uma rede tal que todos os arcos sejam atravessados uma única vez. Este problema, conhecido na literatura como o Problema do Carteiro Chinês (PCC), tem muitas aplicações, como por exemplo, problemas do setor público incluindo varredura de ruas, coleta de lixo, roteamento de carteiros, inspeção de linhas de água, eletricidade ou gás, entre outros.

O problema do carteiro chinês é denominado dirigido ou não dirigido, dependendo se os arcos do grafo são ou não dirigidas. Ambos os casos podem ser solucionados por algoritmos com limitação polinomial. O problema do carteiro chinês misto, que possui alguns dos arcos dirigidos e outros não dirigidos, é NP-Hard.

5.5.2.1 O Problema do Carteiro Chinês (PCC)

5.5.2.1.1 Histórico

Os problemas de Percurso em Arcos são dos mais antigos relacionados a grafos. A primeira referência, que se conhece sobre eles, vem do famoso problema das sete pontes de Koningsberg, figura 5.4. Buscava-se saber se havia um caminho fechado que atravessasse, exatamente uma vez, sete pontes sobre o rio Pregel em Koningsberg, hoje Kaliningrad. O matemático suíço Leonhard Euler, em 1736, encontrou as condições para a existência de uma rota fechada (grafo euleriano) e mostrou que não havia solução que satisfizesse aquele caso particular, figura 5.5. A preocupação de Euler foi exclusivamente sobre a existência do caminho fechado; já a questão de determiná-lo foi resolvida 137 anos mais tarde por Heierholzer (percurso euleriano), [HEIERHOLZER, 1873].

Figura 5.4 – Visualização de Koningsberg, e as sete pontes sobre o rio Pregel.
[NEGREIRO, 1999]

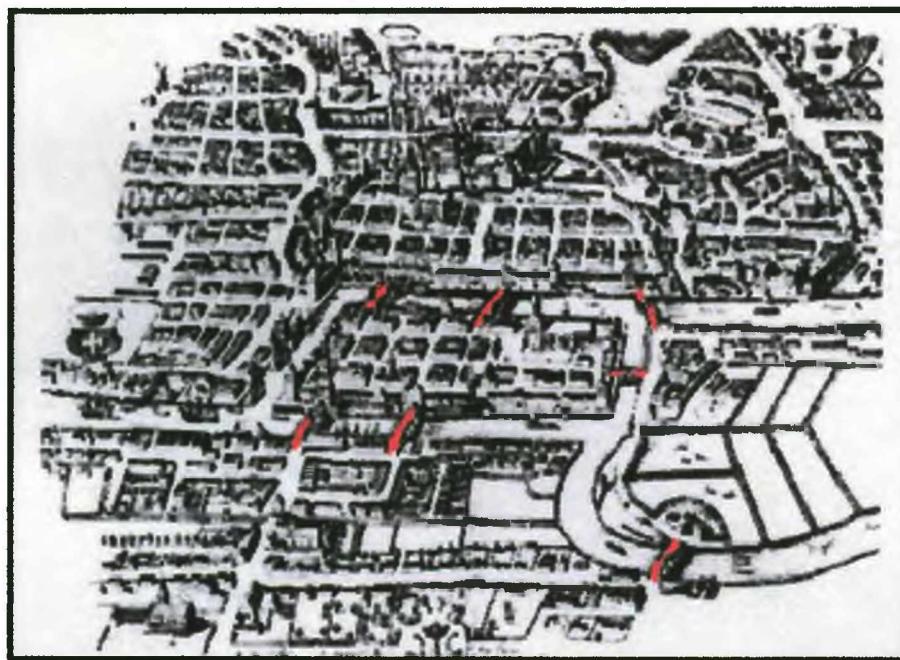
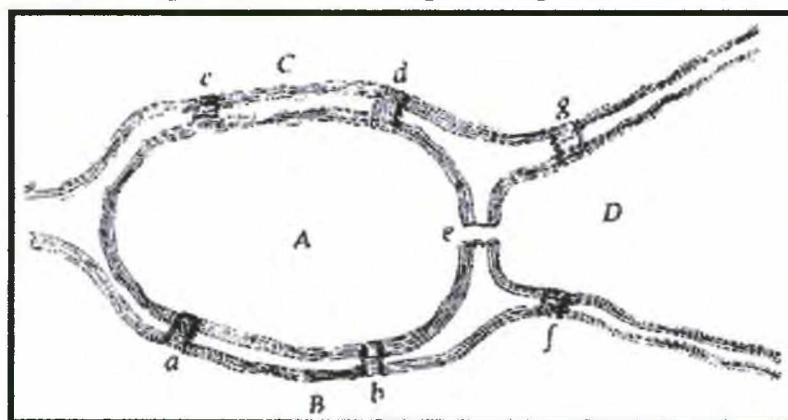


Figura 5.5 – Representação em Grafo do problema de Euler, 1736, representando a situação do problema das sete pontes. [NEGREIRO, 1999]



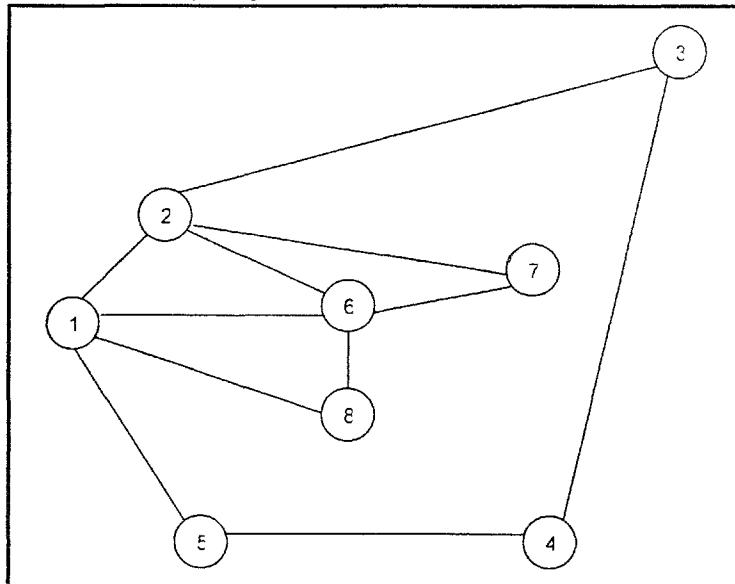
Muitos anos mais tarde, em 1962, um matemático da Universidade Normal de Shangtun, KWAN MEI-KO, quando de sua passagem como funcionário dos correios durante a revolução cultural chinesa, preocupou-se com uma situação semelhante à de Euler e Heierholzer, porém adequada ao percurso dos carteiros que atenderiam ruas de sua cidade. Neste caso, KWAN mostrou-se interessado em definir além da travessia, a forma mais fácil de fazê-la, percorrendo a menor distância possível. KWAN, definiu assim o problema: *Um carteiro tem que cobrir seu local de trabalho, antes de retornar ao posto. O problema é encontrar a menor distância de percurso para o carteiro*, [MEI-KO, 1962].

5.5.2.1.2 A Importância do Modelo do Carteiro Chinês para o Problema de Roteamento de Veículos

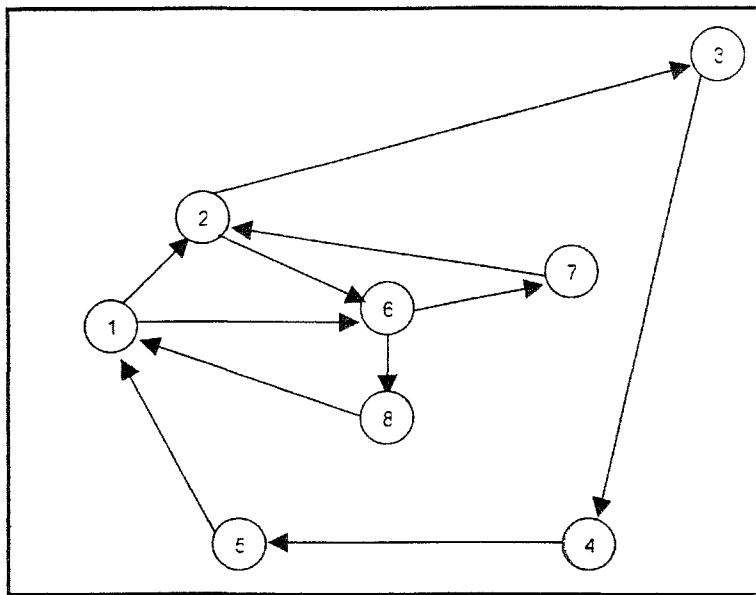
O problema da determinação de uma rota ótima para um veículo sobre um grafo pode transformar-se, na maioria dos casos práticos, em uma tarefa muito difícil. Existem, contudo, exceções a essa regra geral. O problema do carteiro chinês destaca-se exatamente pelo fato de admitir, na maioria das situações, uma solução em tempo polinomial.

O PCC é um problema de otimização que objetiva cobrir com um passeio (ou *tour*) por todos os arcos do grafo, minimizando a distância total percorrida. O passeio do carteiro distingue-se do circuito (ou ciclo) euleriano por nele ser permitida a repetição de arestas. Obviamente, o circuito (ou ciclo) euleriano, quando existente no grafo, é a solução do carteiro chinês. O carteiro chinês pode ser considerado sobre um grafo orientado ou não. Em ambos os casos existem algoritmos polinomiais para a resolução do problema [EDMONDS e JOHNSON, 1973]. A figura 5.6 (b) representa a solução para um ciclo euleriano no grafo representado na figura 5.6 (a). O caminho fechado 1-2-6-7-2-3-4-5-1-6-8-1 constitui uma solução para o caso.

Figura 5.6 – Grafo qualquer em (a) e seu Caminho Euleriano em (b).



(a) Grafo



(b) Caminho Euleriano

Segundo GOLDBARG [2000], as principais variantes do PCC são:

- o carteiro chinês em grafos orientados (possui solução polinomial);
- o carteiro chinês em grafos não-orientados (possui solução polinomial);
- o carteiro chinês em grafos mistos (caso geral não possui solução polinomial);
- o carteiro chinês capacitado (caso geral não possui solução polinomial).

5.5.2.1.3 O Problema do Carteiro Chinês em Grafos Não-Orientados

A solução do problema do carteiro chinês em grafos não-orientados quando o grafo é dito euleriano, reduz-se a determinação de tal ciclo. Um grafo conexo G é euleriano quando possui um número par de arestas incidentes em cada nó. Caso o grafo não seja euleriano, ou seja, se existirem nós de grau ímpar, será necessário percorrer algumas arestas mais de uma vez para que o passeio seja possível.

KWAN MEI-KO, como já foi mencionado, foi o primeiro a relatar esse problema em uma publicação datada de 1962 na *Chinese Mathematics* e, por esse motivo, o problema foi denominado de *carteiro chinês*. Para um grafo $G(N, A)$, pode-se formular o problema como segue. Considerando-se:

x_{ij} é número de vezes que a aresta (i,j) é percorrida de i para j .

c_{ij} é comprimento ou o custo da aresta (i,j) ,

o Modelo Matemático é:

$$(PCC1) \text{ Minimizar } z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$$

sujeito a:

$$\sum_{j=1}^n x_{ji} - \sum_{j=1}^n x_{ij} = 0 \quad i = 1, \dots, n \quad (5.13)$$

$$x_{ij} + x_{ji} \geq 1 \quad \forall (i,j) \in A \quad (5.14)$$

$$x_{ij} \geq 0 \text{ e inteiros} \quad (5.15)$$

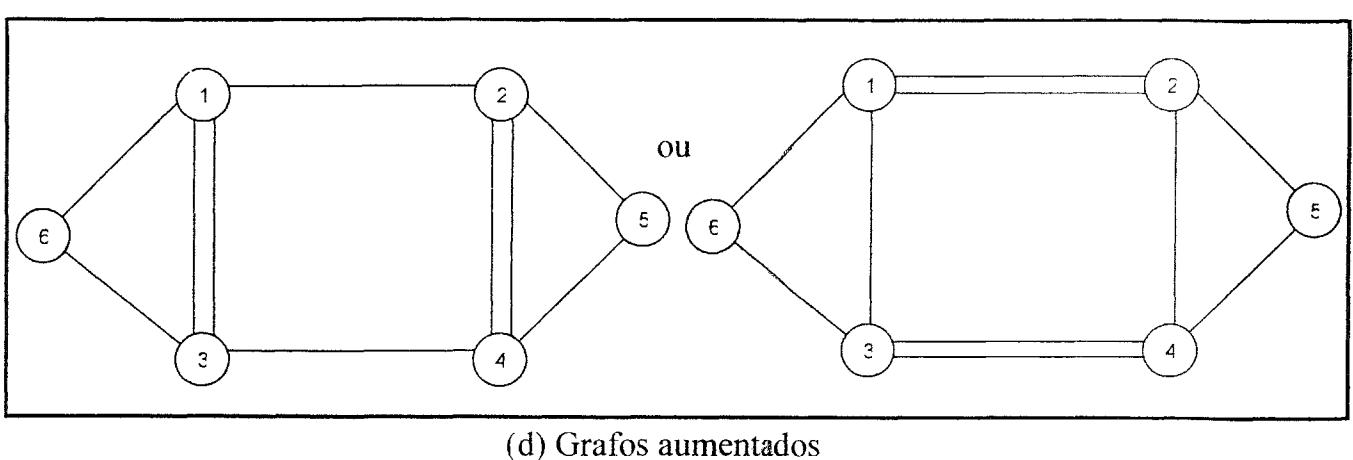
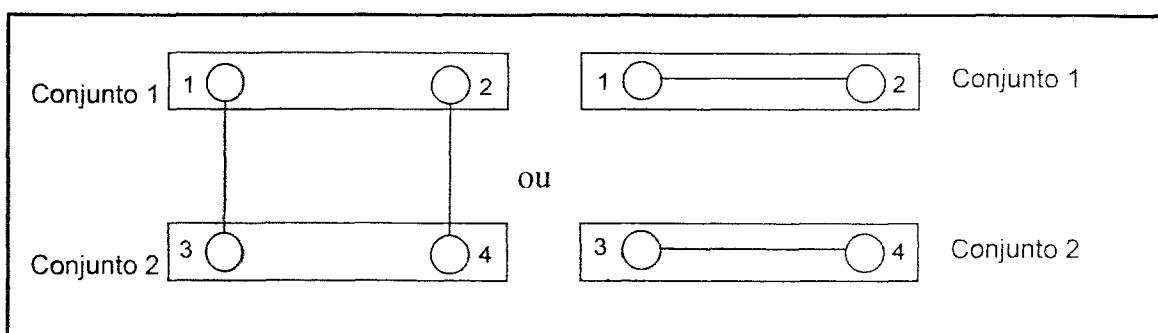
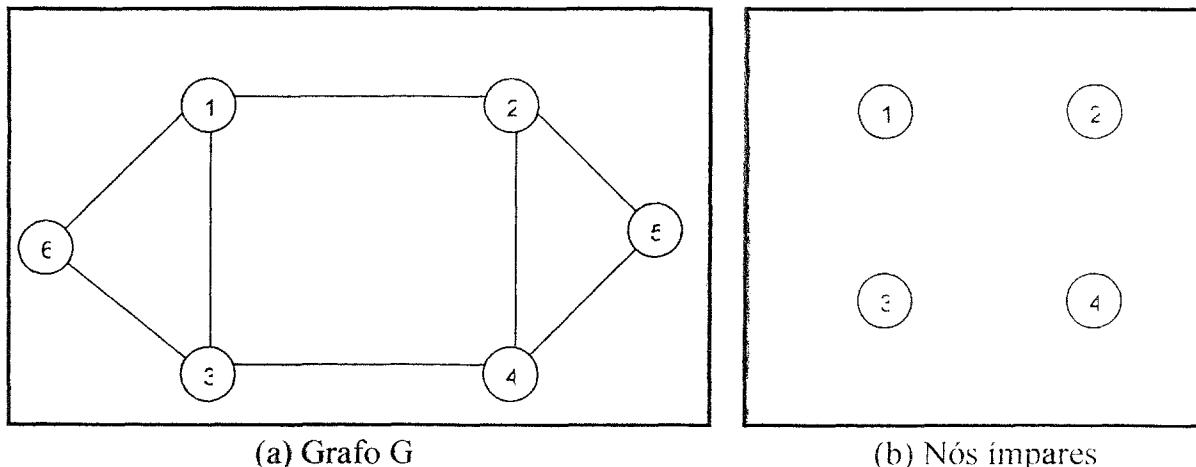
Observa-se que, no modelo matemático proposto, a função objetivo minimiza o custo total, ou seja, a distância total a ser percorrida. A restrição (5.13) garante a continuidade do passeio e a restrição (5.14) que nenhuma aresta deixará de ser considerada, e em (5.15) tem-se que as variáveis do problema são inteiros.

De acordo com GOLDBARG, 200, Como em todo grafo existe um número par de nós de grau ímpar, se denominarmos por d_i o grau do nó i e por $|E| = m$ o número de arestas, então $\sum_i d_i = \sum_{i \in \text{ímpar}} d_i + \sum_{i \in \text{par}} d_i = 2m$, pois cada aresta possui dois nós extremos.

Como a primeira parcela da soma é par, a segunda também o será. Considere N_i o conjunto de nós de grau ímpar em G e por N_p o conjunto de nós de grau par e, ainda, N o conjunto de todos os nós. Desde que o número de nós de grau ímpar seja par, $|N_i|/2$ é par. Utilizando esse fato, pode-se partitionar N_i em dois conjuntos e formar $k = \frac{1}{2} \frac{|N_i|}{2}$ caminhos entre pares de nós distintos. As arestas (aqui denominadas por E^*) contidas nesses caminhos são acrescentadas ao grafo original G como arcos artificiais, obtendo-se um grafo $G_i(E^*)$. Uma solução viável do PCC origina-se dos k caminhos que ligam os k pares de nós ímpares. De acordo com CHRISTOFIDES [1976], o problema então se reduz a determinar os melhores k caminhos que ligam os k pares de nó ímpares. Sua idéia de solução é a de transformar o PCC em um problema de determinação de um ciclo euleriano em um grafo convenientemente expandido.

Na figura 5.7 o grafo G é transformado em um grafo euleriano pelo acréscimo de arestas. As arestas 1-3 e 2-4, bem como as 1-2 e 3-4 representam os caminhos mais curtos entre esses pares de vértices. A solução do problema seria, dentre as duas alternativas possíveis, o percurso total mais barato.

Figura 5.7 – Processo de obtenção de $G_i(E^*)$



A solução exata desse problema pode ser obtida em $O(n^3)$ como mostra PAPADIMITRIOU [1982]. EDMONDS e JOHNSON [1973] apresentaram um

interessante algoritmo para a solução do PCC via *matching* (emparelhamento). Pode-se resumir o algoritmo da seguinte forma [GOLDBARG, 2000]:

Algoritmo Carteiro Chinês

INÍCIO

Ler o grafo $G(N, A)$;

Se todos os nós em G , o grafo original, possuem grau par **então determine** um ciclo euleriano em G e Fim.

Organize um grafo K_n da seguinte forma:

Reúna todos os vértices de grau ímpar no grafo K_n e

Associe a cada par de vértices i e j no grafo, uma aresta (i, j) com peso igual ao caminho mais curto que liga i a j no grafo G .

Determine o 1-matching mínimo em K_n , M^* .

Para cada aresta pertencente a M^* **associe** uma nova aresta em G no caminho mínimo que ela representa, obtendo um grafo G_i .

Determine a solução do carteiro chinês que é representada por um ciclo euleriano em G_i .

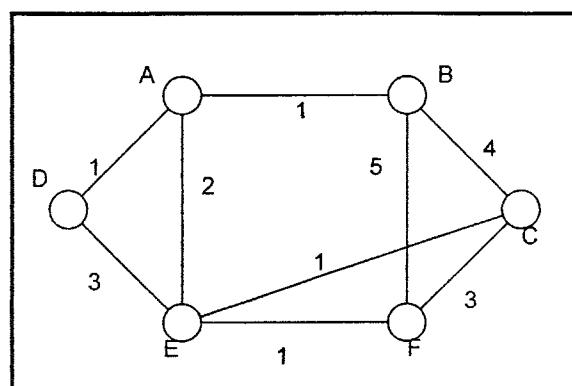
FIM

As figuras 5.8 e 5.9 exemplificam o processo descrito no algoritmo carteiro chinês.

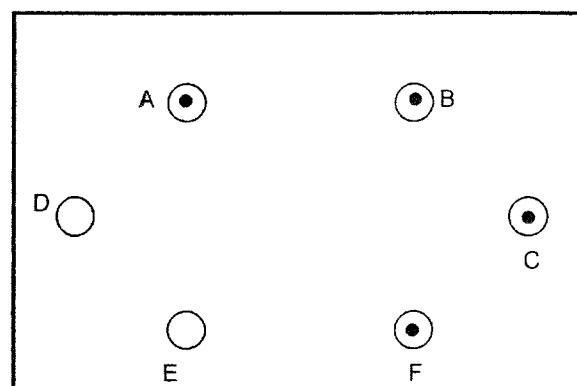
A solução para o problema sobre o grafo da figura é o caminho:

$A \Rightarrow B \Rightarrow F \Rightarrow E \Rightarrow A \Rightarrow D \Rightarrow E \Rightarrow F \Rightarrow C \Rightarrow E \Rightarrow C \Rightarrow B \Rightarrow A$ com $C = 24$.

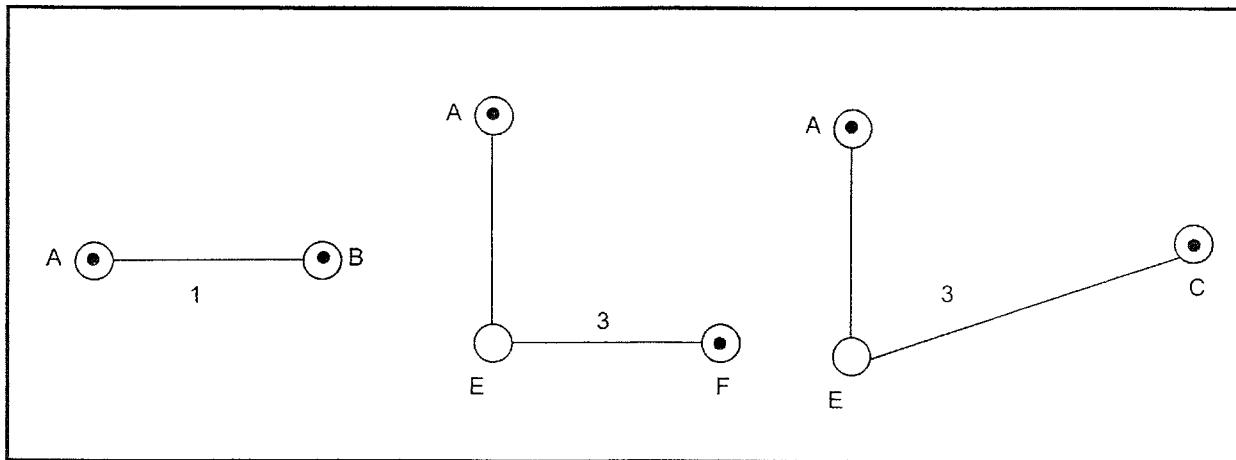
Figura 5.8 – Obtenção de K_4



(a) Grafo Original



(b) Vértices de Grau ímpar



(c) Caminhos mais Curtos de A

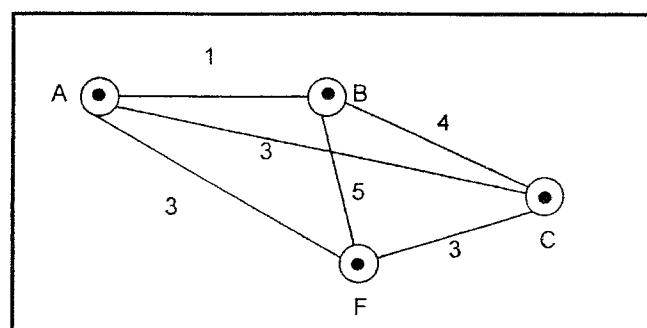
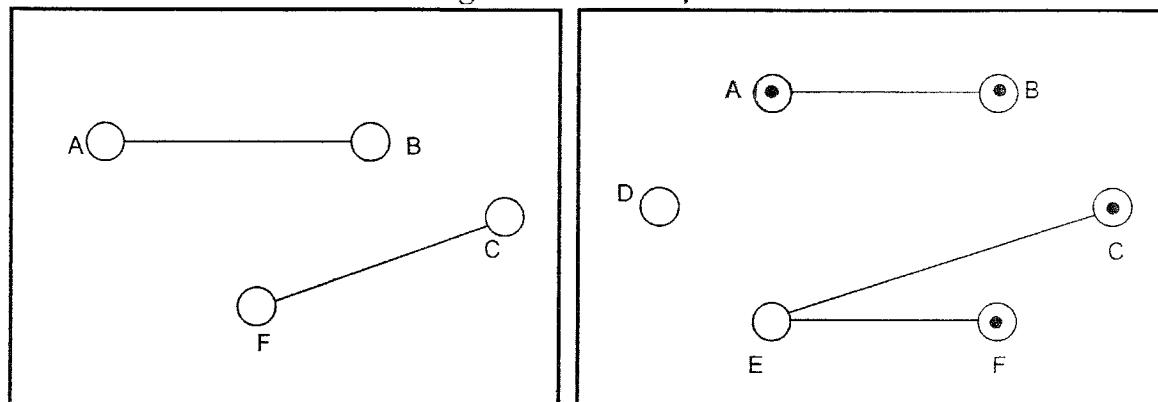
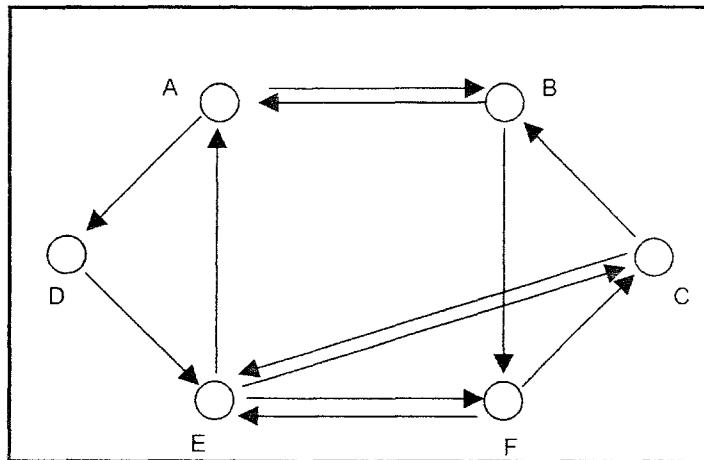
(d) K_4

Figura 5.9 – Solução Final

(a) Matching M em K_4

(b) Caminhos Associados

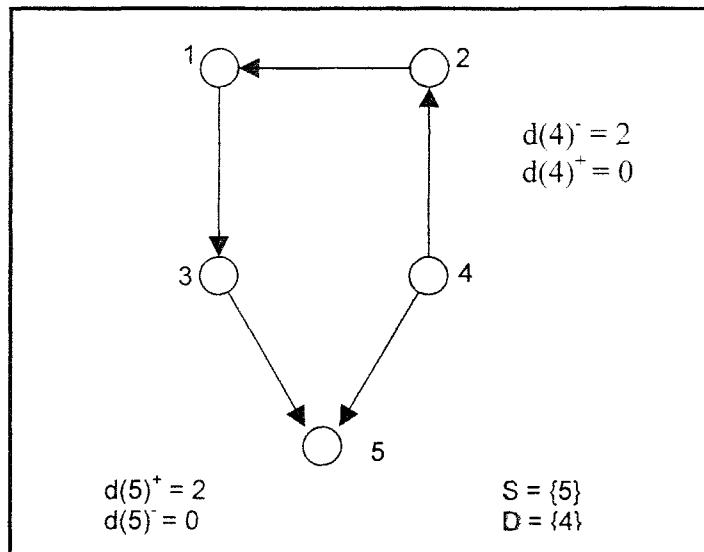


(c) Solução Final

5.5.2.1.4 O Problema Carteiro Chinês em Grafos Orientados

No caso do grafo orientado, uma condição suficiente para a existência do circuito euleriano é que o semigrau interior (grau de entrada) de cada nó seja igual ao semigrau exterior (grau de saída). Para solucionar o problema, determina-se dois conjuntos de nós S e D . O primeiro corresponde aos nós determinados de forma que o semigrau interior $d(i)^+$ supere o semigrau exterior $d(i)^-$ e o segundo, o caso contrário. Pode-se construir um grafo bipartido com os nós $s_j \in S$ e $d_k \in D$. Cada distância c_{jk} , entre dois nós desse grafo, representará o menor caminho entre o nó s_j e o nó s_k no grafo original. Tais distâncias podem ser determinadas pela aplicação de algum algoritmo de caminho mais curto sobre G , como por exemplo o algoritmo de Floyd, que será descrito à frente. Se existe um nó s_j que possui caminho que o ligue a todos os nós s_k , então o PCC não possui solução viável. A figura 5.10 demonstra esse fato para um grafo dirigido.

Figura 5.10 – Exemplo de um PCC inviável



Na verdade a solução do PCC exige a solução de um problema de 1-matching para a formação do grafo $G_i(E^*)$ e, posteriormente, a solução de um problema de circuito euleriano.

Para problemas de cobertura de arcos com mais de um veículo, dois procedimentos são propostos por BELTRANI [1974]

- rotear primeiro e agrupar depois;
- agrupar primeiro e rotear depois.

No primeiro procedimento, uma grande rota é determinada contendo todos os arcos que devem ser passados e, depois, a rota é particionada de acordo com a capacidade dos veículos considerados e em relação a demanda de cada arco. No segundo procedimento, a rede é dividida em tantas partes quanto o número de veículos existentes e, então, é solucionado o problema de roteamento de arcos em cada uma destas regiões.

5.5.2.1.5 O Carteiro Chinês Capacitado

De acordo com GOLDBARG, 2000, esse problema consiste em definir um conjunto de rotas para um conjunto $S = \{1, \dots, s\}$ de carteiros que devem atender à demanda originada no grafo $G(V,E)$. GOLDEN e WONG, [1981] formulam esse problema de roteamento como um problema de fluxo da seguinte forma:

Definindo:

$C = [c_{ij}]$ = matriz do comprimento dos arcos;

$Q = [q_{ij}]$ = matriz das demandas que são despertadas nos arcos e devem ser atendidas pelos carteiros;

w = capacidade dos carteiros, sendo $w \geq \max q_{ij}, \forall (i,j) \in E$;

l_{ij}^p = variável binária que assume valor 1 quando o carteiro p servir a demanda do arco (i,j) e 0 em caso contrário;

x_{ij}^p = variável binária que assume o valor igual a 1 se o arco (i,j) é percorrido pelo carteiro p e 0 caso contrário;

f_{ij}^p = variável de fluxo que assume valor positivo se $x_{ij}^p = 1, f_{ij}^p \in \mathbb{R}^+$.

Um depósito central que receberá o índice 1.

$$\text{Minimize } z = \sum_{i \in V} \sum_{j \in V} \sum_{p \in S} c_{ij}^p x_{ij}^p$$

Sujeito a:

$$\sum_{j \in V} x_{ji}^p - \sum_{j \in V} x_{ij}^p = 0 \quad \forall i \in V, p \in S \quad (5.16)$$

$$\sum_{p=1}^s (l_{ij}^p + l_{ji}^p) = \left\lceil \frac{q_{ij}}{w} \right\rceil \quad \forall (i,j) \in E \quad (5.17)$$

$$x_{ij}^p \geq l_{ij}^p \quad \forall (i,j) \in E, p \in S \quad (5.18)$$

$$\sum_{i \in V} \sum_{j \in V} l_{ij}^p q_{ij}^p \leq w \quad p \in S \quad (5.19)$$

$$\sum_{j \in V} f_{ji}^p - \sum_{j \in V} f_{ij}^p = \sum_{j \in V} l_{ij}^p \quad \forall i \in E \setminus \{1\}, p \in S \quad (5.20)$$

$$f_{ij}^p \leq (n^2) x_{ij}^p \quad \forall (i,j) \in E, p \in S \quad (5.21)$$

$$f_{ij}^p \geq 0 \quad p \in S \quad (5.22)$$

$$q_{ij} > 0 \quad \forall (i,j) \in E$$

$$f_{ij}^p, x_{ij}^p \in \{0,1\} \quad \forall (i,j) \in E, p \in S$$

A função objetivo é o custo total de percorrida do arcos pelos p carteiros. A restrição (5.16) garante a continuidade das rotas dos carteiros. A restrição (5.17) assegura que o atendimento dos carteiros é considerado em apenas uma das suas passadas pelo arco. A restrição (5.18) obriga que o carteiro percorra os arcos que lhe forem designados atender. A restrição (5.19) garante que os atendimentos dos diversos carteiros não ultrapassem sua capacidade. A restrição (5.20) garante que o fluxo de atendimento é igual ao computado pela designação aos carteiros. A restrição (5.21) garante que o fluxo não será maior que o valor de percorrida no arco (i,j) . A restrição (5.22) garante que a variável de fluxo vai assumir valor positivo se $x_y^p = 1$.

Segundo GOLDEN e WONG [1981], esse é um problema NP-Hard. Sugerem o Algoritmo de Christofides, [GOLDBARG, 2000] para a solução do problema.

Algoritmo de Christofides

INÍCIO

Ler o grafo $G(N,A)$;

Faça todos os arcos serem servidos em um circuito individual.

Iniciando com o maior circuito disponível, verifique se um arco de um circuito menor pode ser servido por um circuito maior.

Sujeito às restrições do problema, procure compor dois circuitos de forma a obter a maior economia possível.

Repita a composição até não existir composição que traga economia à solução;

FIM

O algoritmo é descrito de uma maneira um pouco vaga, exatamente para permitir a utilização de outros procedimentos de otimização já existentes. A etapa **repita** pode ser implementada com auxílio das heurísticas de CLARKE e WRIGHT [1964].

5.6 Uma Técnica para Determinação de Caminhos Mínimos entre Nós de um Grafo

O caminho mais curto entre os pares de nós pode ser obtido a partir do algoritmo de Floyd [CHRISTOFIDES, 1975]. Este algoritmo é devido a FLOYD, 1962. baseia-se numa sequência de n iterações sobre uma matriz inicial de custos $C = [c_{ij}]$, de dimensão $n \times n$, até que na k -ésima iteração, a matriz representa o caminho de distância mínima entre todos os pares de nós com a restrição de que o caminho entre x_i e x_j (para quaisquer x_i e x_j) contém apenas nós do conjunto restrito $\{x_1, x_2, \dots, x_k\}$ como intermediários.

Algoritmo de Floyd

Passo 1. Inicializar a matriz de custos $C = [c_{ij}]$, da seguinte maneira:

$$c_{ij} = \begin{cases} 0, & \forall (x_i, x_j) \in A \\ \infty, & \text{se } (x_i, x_j) \notin A \\ \text{custo associado ao arco } (x_i, x_j) \in A \end{cases}$$

Passo 2. Fazer $k = 0$.

Passo 3. $k := k + 1$.

Passo 4. Para todo $i \neq k$ tal que $c_{ik} \neq \infty$ e todo $j \neq k$ tal que $c_{kj} \neq \infty$ realizar a operação $c_{ij} = \min \{c_{ij}, (c_{ik} + c_{kj})\}$.

Passo 5. a) Se $k = n$, parar. A solução foi alcançada e $[c_{ij}]$ fornece os custos mínimos para cada par de nós.

Passo 5. b) Se $k < n$, voltar ao Passo 3.

Os trajetos mínimos, entre dois nós x_i e x_j , podem ser obtidos dos caminhos de mínimo custo usando uma relação recursiva similar a equação dada pelo passo 4. Para tanto, HU, 1969 [CHRISTOFIDES, 1975], forneceu um procedimento que pode ser utilizado para armazenar informações sobre os trajetos, envolvendo o armazenamento e a atualização de uma segunda matriz $\Theta = [\Theta_{ij}]$, de dimensão $n \times n$, em complemento a matriz de custos C . A entrada Θ_{ij} é o nó predecessor do nó x_j no trajeto mínimo entre os nós x_i e x_j . A matriz Θ é inicializada tal que $\Theta_{ij} = x_i$ para todo x_i e x_j .

Após o cálculo da equação utilizada pelo passo 4 do algoritmo de Floyd é introduzida a atualização da matriz Θ como segue:

$$\Theta_{ij} = \begin{cases} \Theta_{kj}, & \text{se } (c_{ik} + c_{kj}) < c_{ij} \\ \Theta_{ij}, & \text{se } (c_{ik} + c_{kj}) \geq c_{ij} \end{cases}$$

Ao final do algoritmo o trajeto mínimo pode ser obtido diretamente da matriz final Θ . Assim, para obter o caminho mínimo entre quaisquer dois nós x_i e x_j , este caminho é dado pela sequência de nós: $x_i, x_z, \dots, x_c, x_b, x_a, x_j$, onde $x_a = \Theta_{ij}$, $x_b = \Theta_{ia}$, $x_c = \Theta_{ib}$, ..., $x_j = \Theta_{iz}$.

CAPÍTULO VI

6. Descrição dos Métodos Propostos na Resolução do Problema

6.1 Introdução

Este capítulo descreve, em detalhes, a estrutura dos métodos propostos para a solução do problema. A primeira parte aborda o problema de localização de facilidades, objetivando a locação de medianas.

Depois, é introduzido o Algoritmo Genético, traçando-se sua estrutura básica, suas definições e a sua aplicação no problema das p -medianas.

E, na última parte, faz-se uma menção ao algoritmo para formação de *clusters*, um método proposto por Gillett e Johnson.

6.2 O Problema de Localização de Facilidades

6.2.1 Introdução

Os algoritmos de otimização relacionados a Problemas de Localização de Facilidades tratam do problema de selecionar a melhor localização, em uma região específica, para a facilidade de serviços.

Na literatura, problemas de localização são tratados pela teoria dos grafos. Tais problemas são conhecidos como problemas de centros e problemas das medianas. Em ambos os casos, o objetivo é a localização de facilidades, ao longo de uma rede viária, definida por um grafo.

No problema de centros, o objetivo é minimizar a distância até o ponto mais crítico a ser atendido, ou seja, busca minimizar a maior distância a ser percorrida e, por isso, são chamados “problemas de localização minimax”. Normalmente, são aplicados a serviços emergenciais como, por exemplo, a localização de postos de saúde, postos de bombeiros ou a localização de um hospital. Como esses problemas são, em geral, associados a um grafo, suas soluções são então chamadas “centros” de um grafo.

Os problemas de localização de medianas têm por objetivo minimizar a distância média, isto é, minimizar a soma de todas as menores distâncias da facilidade aos vértices de um grafo. A localização ótima da facilidade é chamada “mediana do grafo” e, em virtude da natureza da função objetivo, esses problemas são denominados “problemas de localização de soma mínima (minisum)”. Na prática, esses problemas aparecem sob várias formas como, por exemplo, a localização de centros de comutação em rede telefônicas, subestações em redes de energia elétrica, depósitos de fornecimento em uma rede de distribuição rodoviária e a localização de correspondência. CHRISTOFIDES [1975].

Segundo MINIEKA [1977], as facilidades podem ser localizadas nos vértices e/ou nas arestas, e essas localizações são chamadas de *medianas*. Utilizando p para denotar o número de facilidades a serem localizadas tem-se, assim o *problema da determinação das p-medianas*.

De acordo com ROSA [1996], o problema de locação de medianas pode ser aplicado a muitas situações práticas, onde se deseja minimizar uma soma ou uma média, dentre os quais pode-se citar:

- a) localizar um depósito de distribuição de mercadorias numa rede de rodovias para abastecer diversos clientes com localizações fixas e conhecidas de maneira a minimizar a soma das distâncias, aos clientes. Nesse caso, os vértices do grafo representam os clientes;
- b) localizar “P” cabines de telefones públicos de maneira a minimizar o deslocamento médio dos usuários. Nesse caso, os vértices são subconjuntos de usuários;
- c) localizar, numa cidade, uma escola de maneira que os alunos de cada bairro desloquem-se, em média (média ponderada pelo número de alunos de cada bairro), o mínimo possível. Nesse caso, os vértices são estes bairros;

d) localizar unidades operacionais do correio: localizar, por exemplo, os Centros de Distribuição Domiciliar (CDD) – as unidades de entrega da Empresa de Correios e Telégrafos (ECT) – de maneira que, em média, o deslocamento dos carteiros dos seus respectivos CDDs até os seus distritos seja o menor possível.

O objetivo do problema clássico das p-medianas é encontrar a localização para um certo número p de facilidades e minimizar a distância média que separa os consumidores da facilidade mais próxima. Na medida em que a distância afeta o acesso, a solução das p-medianas maximiza a acessibilidade ao serviço [DREZNER, 1995].

Para resolução do problema das medianas, acima mencionado, encontra-se na literatura alguns métodos de solução. Para problemas de pequeno porte, a solução pode ser encontrada por formas exatas, tais como: modelo de programação inteira e enumeração exaustiva (ou busca em árvore). Este último método foi usado por HAKIMI [1985] para encontrar as 3-medianas de um grafo com 10 vértices. Existem, também, os métodos aproximados, usados por vários autores, como MARANZANA [1964], TEITZ e BART [1968], os quais se aplicam a problemas de maior porte.

6.2.2 O Problema das P-Medianas

O problema abordado neste trabalho, descrito no capítulo III, no que tange à determinação dos clientes que devem ser atendidos por cada leiturista, será tratado como um problema de localização de p-medianas e, por este motivo, far-se-á, na sequência, um estudo mais detalhado deste tópico.

6.2.3 Formulação Matemática para o Problema das P-Medianas

Segundo CHRISTOFIDES [1975], este problema pode ser formulado como um Problema de Programação Linear Inteira Binário (PPLIB), da seguinte forma: sendo $[\xi_{ij}]$ uma matriz de locações, onde

$$\begin{aligned} \xi_{ij} &= 1, \text{ se o vértice } x_j \text{ é locado ao vértice } x_i \\ \text{e } \xi_{ij} &= 0, \text{ caso contrário} \end{aligned}$$

e se além disso, $\xi_{ij} = 1$, se o vértice x_i é um vértice-mediana e $\xi_{ij} = 0$, caso contrário, então tem-se o seguinte modelo matemático:

$$\text{Minimizar } Z = \sum_{i=1}^n \sum_{j=1}^n d_{ij} \xi_{ij} \quad (6.1)$$

s.a.

$$\sum_{i=1}^n \xi_{ij} = 1 \text{ para } j = 1, \dots, n \quad (6.2)$$

$$\sum_{i=1}^n \xi_{ii} = p \quad (6.3)$$

$$\xi_{ij} \leq \xi_{ii} \text{ para todo } i, j = 1, \dots, n \quad (6.4)$$

e

$$\xi_{ij} = 0 \text{ ou } 1 \quad (6.5)$$

onde $[d_{ij}]$ é a matriz de distâncias ponderadas, isto é, a matriz de distâncias com cada coluna j multiplicada pelo peso v_j . A função objetivo (6.1) minimiza a soma das distâncias dos vértices de demanda até o conjunto de medianas. A equação (6.2) garante que todo vértice x_j é locado a um e somente um vértice-mediana x_i ; a equação (6.3) garante que existem exatamente p vértices medianas; a restrição (6.4) garante que as locações só podem ser feitas a vértices medianas; a restrição (6.5) impõe a integralidade, isto é, ξ_{ij} é variável binária, podendo assumir o valor 0 ou 1.

Se a restrição (6.4) do problema acima for escrita como $\xi_{ij} \geq 0$, o problema passa a ser um Problema de Programação Linear.

Como o problema é formulado como um PPLIB, por si só já bastante complexo, dependendo da cardinalidade de nós do grafo, o modelo apresentado torna-se inviável devido ao tempo computacional. Por este motivo, a seguir, faz-se a abordagem do Algoritmo de Teitz e Bart.

6.2.4 Algoritmo das P-Medianas de Teitz e Bart

Uma das heurísticas mais conhecidas para o problema das p-medianas é a desenvolvida por TEITZ e BART [1968] e é conhecida como Algoritmo das p-medianas de Teitz e Bart. Esta heurística é baseada na substituição de vértices e seu objetivo é, a partir de uma solução inicial, melhorar o valor da função objetivo a cada iteração. Esta heurística é fácil de ser implementada e produz boas soluções para problemas pequenos,

principalmente quando aplicada várias vezes ao mesmo problema com diferentes soluções iniciais.

Considerando-se todos os vértices do grafo como potenciais medianas, o algoritmo de Teitz e Bart para o problema das p -medianas pode ser definido como segue: dado $G(V, A)$ um grafo não direcionado onde V são os vértices e A as arestas. Seja v_i um vértice qualquer pertence a V . Chama-se **número de transmissão** à soma das menores distâncias existentes entre o vértice v_i e todos os outros vértices do grafo. Sendo n o número total de vértices do grafo, o número de transmissão é dado por:

$$\sigma(v_i) = \sum_{j=1}^n w_j d(v_i, v_j), \forall v_i, v_j \in V \quad (6.6)$$

onde, $d(v_i, v_j)$ é a menor distância entre v_i e v_j e w_j é um peso associado ao vértice v_j .

Assim, v_m é uma mediana se, entre todos os vértices do grafo, é aquele que produz a menor soma total das distâncias desde si próprio até cada um dos outros vértices do grafo.

Assim sendo, tem-se:

$$\sigma(v_m) = \min[\sigma(v_i)], \forall v_i \in V. \quad (6.7)$$

Para o problema de encontrar p -medianas ($p > 1$), seja $V_p \subset V$ e $|V_p| = p$, calcula-se:

$$D(V_p, v_j) = \min[d(v_i, v_j)], \forall v_i \in V_p, v_j \in V \quad (6.8)$$

e

$$\sigma(V_p) = \sum w_j d(V_p, v_j), \forall v_j \in V. \quad (6.9)$$

Desse modo, um conjunto de p vértices é a solução ótima para o problema das p -medianas se, entre todos os outros conjuntos de p vértices do grafo, é aquele que produz a menor distância total desde si próprio até todos os outros vértices do grafo. Portanto, deve-se ter:

$$\sigma(V_{p_{\text{solução ótima}}}) = \min[\sigma(V_p)], \forall V_p \subset V. \quad (6.10)$$

O objetivo do algoritmo de Teitz e Bart é, portanto, encontrar um conjunto V_p em V , para o qual o número de transmissão seja mínimo.

6.2.5 Procedimentos Básicos do Algoritmo das P-Medianas de Teitz e Bart

São descritos, a seguir, os procedimentos básicos executados pelo algoritmo das p-medianas de Teitz e Bart [TEITZ e BART, 1968].

Algoritmo de Teitz e Bart

Passo 0

Selecione, aleatoriamente, um conjunto $V_p \subset V$, com $|V_p| = p$ para formar uma solução inicial para o problema.

Passo 1

Rotule todos os vértices $v_i \in \{V - V_p\}$ como “não analisados”.

Passo 2

Enquanto existirem vértices não analisados em $\{V - V_p\}$ faça o seguinte:

Selecione um vértice não analisado $v_i \in \{V - V_p\}$, e calcule a redução Δ_{ij} do número de transmissão, para todos os vértices v_j pertencentes a V_p , ou seja:

$$\Delta_{ij} = \sigma(V_p) - \sigma(V_p \cup \{v_i\} - \{v_j\}), \forall v_j \in V_p \quad (6.11)$$

Faça $\Delta_{ij_máximo} = \text{máximo}[\Delta_{ij}]$, para todo Δ_{ij} calculado anteriormente.

Se $\Delta_{ij_máximo} > 0$ então:

Faça $V_p = (V_p \cup \{v_i\} - \{v_j\})$ e insira v_j em $\{V - V_p\}$.

Rotule v_j como “analisado”.

Caso contrário continue.

Passo 3

Se durante a execução do Passo 3, houver alguma modificação no conjunto V_p , então:

Volte ao Passo 2 e continue a execução do algoritmo.

Caso contrário, PARE e apresente o conjunto V_p como uma solução aproximada para o problema das p-medianas.

Fim.

O algoritmo de Teitz e Bart encontra a solução através do método exaustivo. Os problemas de pequeno porte têm solução ótima através deste método exaustivo, que examina todos os casos possíveis.

O método exaustivo, quando aplicado a problemas de maior porte, provoca a explosão combinatorial. Por esta razão, a estes casos são aplicadas as heurísticas, métodos aproximados cuja solução se aproxima da solução ótima.

6.3 Algoritmos Genéticos

6.3.1 Introdução

O Problema do Roteamento de Veículos busca a definição da melhor rota para a realização de uma tarefa de distribuição, de coleta ou de prestação de serviços. Procura-se minimizar a distância total percorrida, respeitando-se as restrições impostas.

Como já foi mencionado, o problema de roteamento de veículos pertence à categoria de problemas NP-hard, na qual o tempo computacional é uma função exponencial do tamanho do problema. Dessa forma, na grande maioria das aplicações, o uso de métodos heurísticos tem demonstrado ser mais adequado, resultando em soluções freqüentemente de boa qualidade e tempo reduzido. No entanto, OCHI [1994] salienta a dificuldade de criar heurísticas de caráter geral que sejam eficientes na solução de uma classe mais ampla de problemas práticos, como a classe de problemas de roteamento de veículos.

As meta-heurísticas, como o *simulated annealing*, a busca tabu, *guided local search* e o algoritmo genético, são heurísticas genéricas que se adaptam facilmente a uma classe de problemas e são direcionadas à otimização global de um problema. Segundo HAMACHER [1998], dentre as meta-heurísticas, o algoritmo genético apresenta um tempo de execução geralmente mais curto e uma flexibilidade maior para tratar as restrições do modelo, conseguindo assim solucionar deficiências históricas dos algoritmos convencionais de busca heurística.

Os Algoritmos Genéticos (AG's) constituem um método de otimização inspirado no processo Darwiniano de seleção natural dos seres vivos. Na realidade, os AG's fazem parte de uma classe de paradigmas e técnicas computacionais inspiradas na evolução natural, denominada de Computação Evolucionista.

HANS-PAUL SCHWEFEL, um dos pioneiros da Computação Evolucionista (CE) – na tentativa de imitar a evolução dos seres vivos na natureza - considera difícil definir quem teria sido o primeiro a conceber um algoritmo evolucionista.

DIAS e BARRETO, 1998, destaca, dentre os principais fatores que têm feito do Algoritmo Genético uma técnica bem sucedida: a simplicidade de operação; facilidade de implementação; eficácia na busca da região onde, provavelmente, encontra-se o máximo

global; ser aplicável em situações onde não se conhece o modelo matemático ou se este for impreciso; em funções lineares e não-lineares.

Contudo, em virtude da lenta e, até mesmo crítica convergência dos AG's quando o erro torna-se pequeno, recomenda-se utilizá-lo de forma híbrida. Nesse caso, o AG seria encarregado da aproximação necessária do máximo global e outros métodos, como o do gradiente, ficariam encarregados do ajuste fino.

6.3.2 Histórico

De acordo com CARVALHO, 2001, até meados do século XIX, os naturalistas acreditavam que cada espécie havia sido criada separadamente por um ser supremo ou através de geração espontânea. O trabalho do naturalista Carolus Linnacus, 1735, sobre a classificação biológica de organismos, despertou o interesse pela similaridade entre certas espécies, levando a acreditar na existência de uma certa relação entre elas. Outros trabalhos influenciaram os naturalistas em direção à teoria da seleção natural, tais como os de Jean Baptiste Lamark, 1809, que sugeriu uma teoria evolucionária no "uso e desuso" de órgãos; e de Thomas Robert Malthus, 1798, que propôs que fatores ambientais tais como doenças e carência de alimentos, limitavam o crescimento de uma população.

Depois de mais de 20 anos de observações e experimentos, Charles Darwin apresentou em 1858 sua teoria de evolução através de seleção natural, simultaneamente com outro naturalista inglês Alfred Russel Wallace. No ano seguinte, Darwin publica o seu *On the Origin of Species by Means of Natural Selection* com a sua teoria completa, sustentada por muitas evidências colhidas durante suas viagens a bordo do Beagle.

Esse trabalho influenciou muito o futuro não apenas da Biologia, Botânica e Zoologia, mas também teve grande influência sobre o pensamento religioso, filosófico, político e econômico da época. A teoria da evolução e a computação nasceram praticamente na mesma época: Charles Babbage, um dos fundadores da computação moderna e amigo pessoal de Darwin, desenvolveu sua máquina analítica em 1833. Ambos, provavelmente, estariam surpresos e orgulhosos com a ligação entre estas duas áreas.

Por volta de 1900, o trabalho de Gregor Mendel, desenvolvido em 1865, sobre os princípios básicos de herança genética, foi redescoberto pelos cientistas e teve grande influência sobre os futuros trabalhos relacionados à evolução. A moderna teoria da evolução combina a genética e as idéias de Darwin e Wallace sobre a seleção natural, criando o princípio básico de Genética Populacional: a variabilidade entre indivíduos em uma população de organismos que se reproduzem sexualmente é produzida pela mutação e pela recombinação genética.

Esse princípio foi desenvolvido durante os anos 30 e 40, por biólogos e matemáticos de importantes centros de pesquisa. Nos anos 50 e 60, muitos biólogos começaram a desenvolver simulações computacionais de sistemas genéticos. Entretanto, foi John Holland quem começou, seriamente, a desenvolver as primeiras pesquisas no tema. Holland foi, gradualmente, refinando suas idéias e em 1975 publicou o seu livro *Adaptation in Natural and Artificial Systems*, hoje considerado a *Bíblia de Algoritmos Genéticos*. Nos anos 80 David Goldberg, aluno de Holland, consegue o primeiro sucesso em aplicação industrial de Algoritmos Genéticos (AG). Desde então, esses algoritmos vêm sendo aplicados com sucesso nos mais diversos problemas de otimização e aprendizado de máquinas.

6.3.3 Aplicações dos Algoritmos Genéticos

Um sistema com bom desempenho em um ambiente dinâmico, geralmente exige soluções adaptativas. Sistemas adaptativos tentam resolver problemas acumulando conhecimento sobre o problema e utilizando estas informações para gerar soluções aceitáveis. Entre outras aplicações, exemplifica-se [BARBOSA, 1997]:

- a) na construção automática de programas para realização de tarefas específicas [HARVEY, 1992];
- b) nas áreas de economia e finanças (gerenciamento de carteira de ações, desenvolvimento de estratégias para leilões, detecção de fraudes em movimentos na bolsa de valores) [MORAN, 1996];

6.3.4 Funcionamento dos Algoritmos Genéticos

Segundo GREFENSTETTE, 1986, um AG é um procedimento iterativo que mantém uma população de estruturas, chamadas de “indivíduos”, que representam as possíveis soluções para um determinado problema. A cada iteração (“geração”), os indivíduos da população passam por uma avaliação que verifica sua capacidade em oferecer uma solução satisfatória para o problema. Essa avaliação é feita conforme uma função que recebe o nome de função de aptidão, ou função de *fitness*.

Com essa avaliação, alguns indivíduos são selecionados, de acordo com uma regra probabilística, para passar por um processo de reprodução. Na verdade, aplica-se sobre os indivíduos selecionados os chamados operadores genéticos, gerando uma nova população de possíveis soluções. Pressupõe-se que a população, em média, vai ficando, incrementalmente, mais apta para solucionar o problema. Após um grande número de gerações, de acordo com um critério de término do algoritmo, o indivíduo mais apto até então é uma possível solução para o problema.

Embora os AG’s nem sempre possam encontrar a solução ótima para um determinado problema (ótimo global), na maioria das vezes, são capazes de encontrar uma solução quase ótima, o que é aceitável quando se considera problemas muito complexos, como os de otimização combinatória, onde os métodos convencionais, normalmente, são inviáveis em razão do esforço computacional que seria necessário para resolvê-los. Convém lembrar que muitos problemas apresentam tantas dificuldades, que fica-se satisfeito em encontrar uma solução que atenda a todas as restrições impostas.

Assim sendo, os AG’s constituem uma classe de ferramentas muito versátil e robusta, pois a busca da solução pode inclusive se dar em conjuntos não-convexos e mesmo disjuntos, com funções objetivo também não convexas e não-diferenciáveis, podendo trabalhar, simultaneamente, com variáveis reais, lógicas e inteiras. É importante ressaltar que, em virtude de suas características, os AG’s evitam atrações irremediáveis para ótimos locais, o que ocorre, freqüentemente, com alguns algoritmos usuais de programação matemática, permitindo uma melhor exploração do espaço de busca.

Algumas das principais características que diferenciam os AG’s de outras técnicas de programação matemática, são as seguintes:

- a) empregam uma população de indivíduos, ou soluções, que pode ter tamanho fixo ou variável, ao contrário da maioria das técnicas que efetuam uma busca “ponto-a-ponto”;
- b) não trabalham diretamente com as possíveis soluções do problema, chamadas de *fenótipos*, mas sobre uma codificação das mesmas chamadas de *genótipos*;
- c) empregam regras de transição probabilísticas ou estocásticas, sendo que a maioria dos algoritmos tradicionais usam regras determinísticas;
- d) não exigem maiores informações adicionais sobre a função a otimizar.

Os maiores atrativos do AG têm sido a sua simplicidade de implementação e a eficiência, sendo que grande parte desta eficiência deve-se a eficácia do AG em realizar um passo na direção de uma busca global [DIAS e BARRETO, 1998].

A forte inspiração dos algoritmos genéticos, na teoria da evolução das espécies e na genética natural, aproxima a terminologia utilizada. A nível biológico, um indivíduo é formado por um conjunto de cromossomos, no entanto pode-se fazer uma analogia entre indivíduo e cromossomo. Segundo DIAS e BARRETO [1998], no AG os dois termos são utilizados indistintamente. O cromossomo (*string*) é composto de genes (bit), sendo que cada gene possui um local fixo no cromossomo (*locus*). Cada gene pode assumir um certo valor pertencente a um conjunto de valores (alelo). Desta forma, o valor (alelo) de um bit depende da posição (*locus*) que ele ocupa no cromossomo (*string*) [PAVAN,1966].

Ao conjunto de cromossomo, genes e alelos denomina-se **genótipo** (é a variável independente x) e as características conferidas por este denomina-se **fenótipo** (é a variável dependente ou função, $f(x)$).

BARBOSA [1997], apresenta o seguinte pseudo-código genérico, capaz de englobar a maioria dos AG's, apresentado a seguir:

Algoritmo AG genérico

Inicie a população

Avalie indivíduos na população

Repete

 Seleciona indivíduos para reprodução

 Aplique operadores de recombinação e mutação

 Avalie indivíduos na população

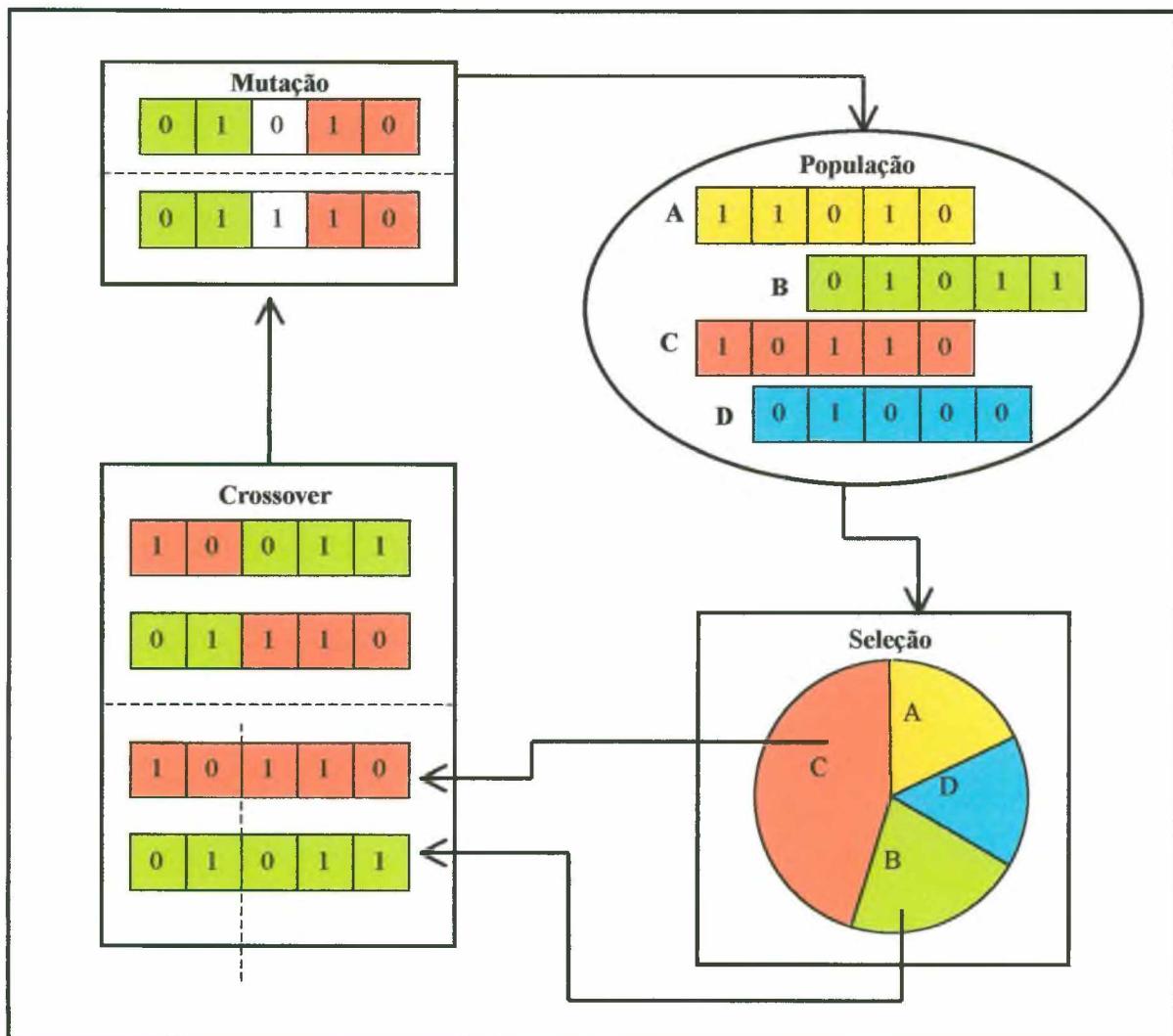
 Seleciona indivíduos para sobreviver

 Até critério de parada satisfeito

Fim

Na figura 6.1 pode-se visualizar a estrutura básica de um Algoritmo Genético, adaptada de HOFFMANN [1997].

Figura 6.1 – Estrutura básica de um Algoritmo Genético



Os procedimentos associados aos verbos iniciar, avaliar, selecionar e aplicar, utilizados nesse algoritmo serão comentados no decorrer deste capítulo. Considera-se inicialmente o processo de reprodução num AG.

6.3.5 O Processo de Reprodução

No que diz respeito à maneira pela qual os indivíduos criados são inseridos na população, considera-se dois tipos extremos de AG's [BARBOSA, 1997].

O primeiro tipo é chamado de AG *generacional*, onde toda a população é substituída pelos novos indivíduos gerados depois do processo de seleção e aplicação dos operadores genéticos. Pode-se representar este tipo de algoritmo da seguinte forma:

Algoritmo Genético *generacional*

Inicie a população P de alguma forma

Avalie os indivíduos da população P

Repete

Repete

 Seleciona indivíduos da população P

 Aplique os operadores genéticos

 Insira os novos indivíduos em P'

 Até que a população P' esteja completa

 Avalie os indivíduos da população P'

$P \leftarrow P'$

Até que um critério de parada esteja satisfeito

Fim

Considerando que neste processo toda a população é substituída pela nova, corre-se o risco de perder bons indivíduos. Para evitar isto, pode-se utilizar um procedimento conhecido como **elitismo**, que consiste em passar para a geração seguinte uma cópia de alguns dos melhores indivíduos.

O outro tipo de AG é conhecido como “*steady-state*”, o qual caracteriza-se por criar apenas um indivíduo de cada vez, sendo que o indivíduo gerado pode ou não ser passado para a geração seguinte. Normalmente, ele é transmitido para a próxima geração, se o seu valor de *fitness* for melhor do que o pior valor de *fitness* da população antiga. Pode-se representar estes AG's com o seguinte pseudo-código:

Algoritmo AG “*steady-state*”

Inicie a população P de alguma forma

Avalie os indivíduos da população P

Ordene a população de acordo com o seu *fitness*

Repete

Seleciona os indivíduos na população P

Aplique os operadores genéticos

Seleciona um indivíduo f para sobreviver

Se f é melhor que o pior elemento de P **Então**

 Remova um indivíduo da população

 Insira f em P de acordo com seu “*ranking*”

Até que um critério de parada esteja satisfeito

Fim

Neste trabalho, será implementado este algoritmo adaptado ao problema das p -medianas, e será descrito mais adiante.

6.3.6 O Sistema de Representação e Codificação

Considerando que os AG's não operam diretamente sobre os elementos do espaço de busca [GOLDBERG, 1989], [BARBOSA, 1997], a primeira etapa para se resolver um dado problema utilizando um AG consiste na codificação/representação destes elementos. Costuma-se chamar de **fenótipos** os elementos deste espaço de busca, enquanto que o código que os representa é denominado de **genótipo**, em analogia com a terminologia encontrada na Genética.

Matematicamente, a escolha da codificação para um dado problema é a função ou regra que associa os elementos do espaço de **genótipos** com aqueles do espaço de busca, os **fenótipos**. Convém lembrar que os espaços de busca podem ser formados por elementos das mais diversas naturezas. Pode-se considerar o espaço das matrizes, vetores, as combinações de variáveis lógicas, inteiras e reais, e até programas de computador numa dada linguagem.

Normalmente, esta forma de codificação corresponde a utilizar cadeias (“strings”) de comprimento l , formadas por caracteres de um determinado alfabeto. O caso mais comum é o binário, onde o alfabeto é composto pelos símbolos 0 e 1. Em alguns casos, utiliza-se também o código de Gray, onde a diferença entre duas cadeias consecutivas é

de apenas 1 *bit*. Tal construção é sempre possível, haja vista a bi-univocidade entre as cadeias de 0 e 1 e os números que elas representam.

Assim, no sistema binário a cadeia 10010101 poderia representar uma possível solução de um dado problema. Neste caso, tem-se $l = 8$ e o conjunto dos **genótipos** é formado por todos os números binários de 00000000 a 11111111 contendo portanto $2^l = 2^8 = 256$ elementos. A codificação é a regra que associa a cada uma destas cadeias de números binários uma solução. Um possível elemento do espaço de busca composto pelas variáveis v_i de diferentes tipos poderia ser codificado como uma cadeia da forma

$$\underbrace{100101}_{v_1} \underbrace{1101}_{v_2} \dots \underbrace{100111}_{v_n}$$

também denominada **cromossomo**.

Cada uma destas subcadeias é denominada de **gene** e representa uma das diversas variáveis que compõe o cromossomo.

Existem situações onde mais de um cromossomo é associado a um indivíduo. O *Homo sapiens* é **diploide**, isto é, possui 2 cromossomos. A grande maioria das aplicações de AG's utiliza, entretanto, indivíduos **haplóides** com um só cromossomo.

Assim, o **genótipo** é composto de um ou mais cromossomos que são compostos por subcadeias de símbolos pertencentes ao alfabeto utilizado. Esta representação/codificação é uma etapa que depende do problema a ser resolvido.

6.3.7 Geração da População Inicial

O processo de inicialização da população é quase sempre realizado aleatoriamente utilizando-se um gerador de números pseudo-aleatórios com distribuição uniforme dos indivíduos numa faixa previamente definida pelo usuário [BARBOSA, 1997]. Essa faixa é definida levando-se em consideração algum conhecimento prévio do problema a ser otimizado.

No caso de codificação binária, por exemplo, é fácil obter uma boa distribuição de zeros e uns na população inicial.

Pode-se também introduzir, na população inicial, um ou mais indivíduos “interessantes”, como, por exemplo, soluções aproximadas conhecidas ou contendo algum tipo de informação prévia.

O número de elementos que comporá a população ainda é questionável, ou seja, depende muito da experiência do usuário e do seu conhecimento prévio sobre a função a ser otimizada.

O número de elementos na população, a probabilidade de ocorrer cruzamento e a probabilidade de acontecer mutação são denominados de **parâmetros de controle** do AG.

6.3.8 Os Operadores Genéticos

O objetivo dos operadores genéticos é operar sobre os indivíduos que foram selecionados para reprodução, produzindo um ou mais “descendentes”. Os operadores são construídos após definida uma codificação para os elementos do espaço de busca. Dos diversos operadores genéticos propostos por GOLDBERG [1989] e outros autores, destaca-se, basicamente, dois tipos: os operadores de recombinação e os operadores de mutação, detalhados a seguir.

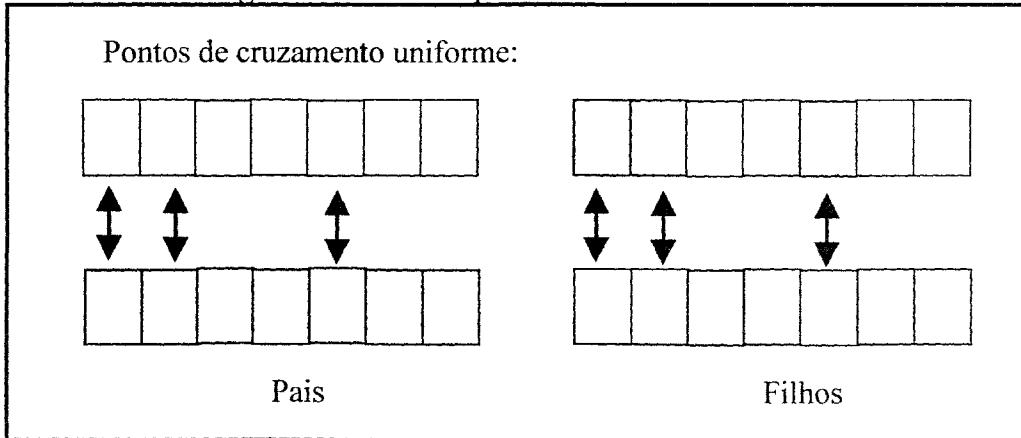
6.3.8.1 Operadores de Recombinação (Cruzamento)

O operador genético de cruzamento ou “crossover” corresponde a uma generalização do que ocorre na reprodução sexuada. Atuam sobre os **genótipos** dos indivíduos selecionados, promovendo uma recombinação do material genético dos elementos “pais”, gerando os elementos “filhos”. Este tipo de operador costuma se chamar na literatura de AG’s de operadores de **crossover**, em analogia com o termo da genética.

Segundo SPEARS [1991], a troca de partes do cromossomo pode ser efetuada de várias formas, como o cruzamento uniforme, cruzamento com 1-partição, cruzamento com 2-partições e cruzamento com n-partições.

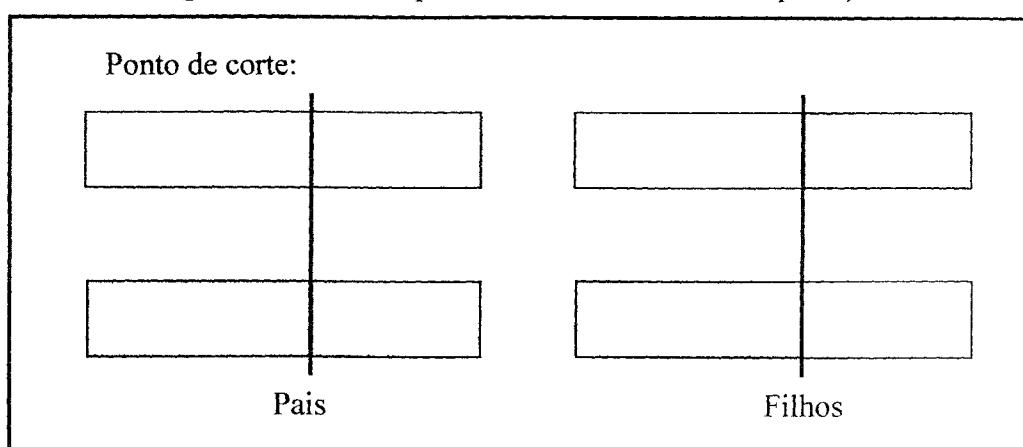
O cruzamento uniforme consiste no emparelhamento dos dois cromossomos pais e cada locus do cromossomo tem 50% de chance de ser trocado, conforme apresentado na figura 6.2.

Figura 6.2 – Exemplo de cruzamento uniforme



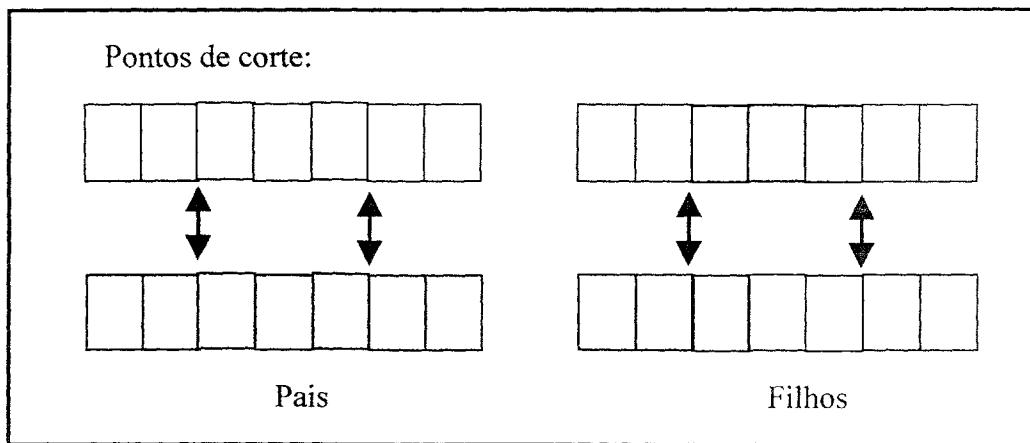
O cruzamento com 1-partição consiste na escolha aleatória de somente um ponto de corte. Todo o material genético à direita deste corte será intercambiado. Esta situação está ilustrada na figura 6.3.

Figura 6.3 – Exemplo de cruzamento com 1-partição



No caso do cruzamento com 2-partições, há a escolha aleatória de dois pontos de corte. Todo o material genético dos pais, existentes entre os dois pontos de corte, será trocado de acordo com a figura 6.4.

Figura 6.4 – Exemplo de cruzamento com 2-partições



O cruzamento com n -partições consiste em n cruzamentos com duas partições.

6.3.8.2 Operadores de Mutação

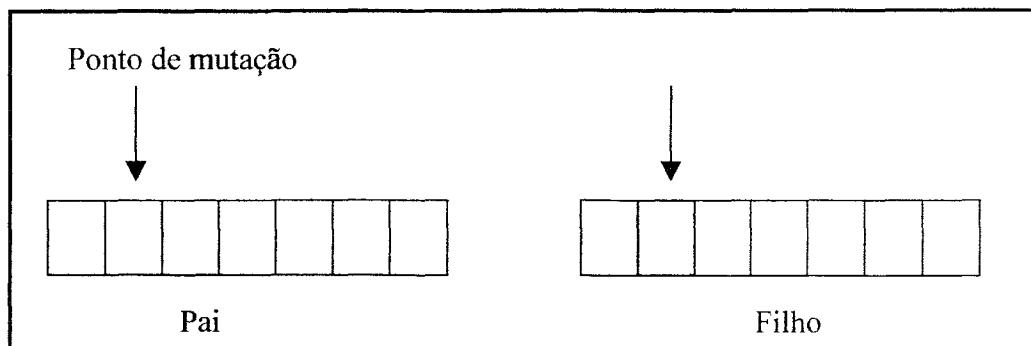
A mutação é, possivelmente, o operador genético mais simples de ser implementado. Ao se considerar o alfabeto binário [BARBOSA, 1997], uma posição do cromossomo é sorteada e o *bit* correspondente é invertido, isto é, se for 1 ele passa a ser 0 e vice-versa. A probabilidade de se efetuar uma mutação deve ser relativamente baixa, caso contrário o algoritmo se comportará fazendo uma busca aleatória, dificultando a convergência.

Esses operadores exploram, globalmente, o espaço de busca, possibilitando inclusive, recuperar algum bom material genético que possa ter sido perdido após sucessivas recombinações. Assim sendo, pode-se considerar os operadores de mutação como uma espécie de “apólice de seguro” contra perdas acidentais deste material genético de boa qualidade.

Dentre os principais mecanismos de alteração genética que recebem a denominação global de mutação destacam-se:

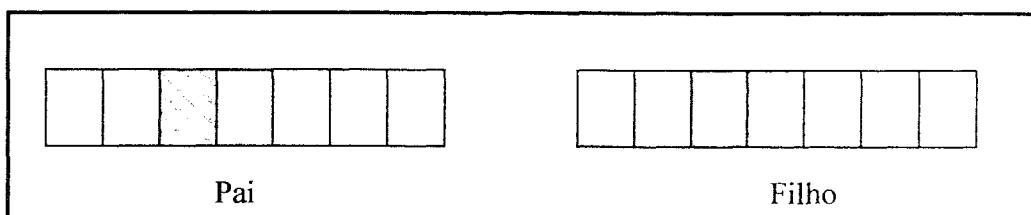
- a) Troca Simples: consiste de um erro de cópia de um ou mais genes da cadeia, conforme mostrado na figura 6.5.

Figura 6.5 – Exemplo de mutação por troca simples



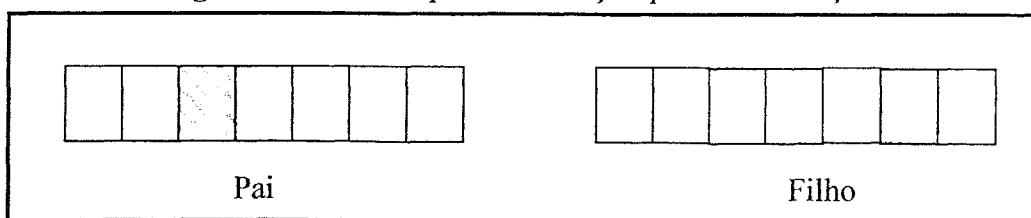
- b) Inversão: retirada e inserção de parte da cadeia, porém na ordem inversa em que foi retirada, conforme a figura 6.6.

Figura 6.6 – Exemplo de mutação por inversão



- c) Translocação: parte do cromossomo é retirada e colocada em outra posição, guardando a mesma ordem em que foi retirada, de acordo com a figura 6.7.

Figura 6.7 – Exemplo de mutação por translocação



6.3.9 Aplicação dos Operadores Genéticos

A aplicação dos operadores genéticos consiste na manipulação genética através da aplicação dos operadores de cruzamento e/ou mutação [DIAS e BARRETO, 1998]. Esses podem ser aplicados em toda ou em parte da população sorteada para compor o processo que irá gerar a nova população. Ao final desta etapa, terá sido criada uma nova população que deverá repetir os passos anteriores até que a adaptação da população seja aceitável.

Para aplicação do operador genético de cruzamento com 1-partição implica na geração de 2 números aleatórios:

- a) o primeiro, entre 0 e 1, indicará a probabilidade de ocorrer cruzamento;
- b) o segundo, o local da realização do cruzamento (ponto de corte), caso haja cruzamento. Nesse caso o número aleatório deverá estar entre 1 e $(g-1)$, onde g é o número de bits do indivíduo.

E assim para os demais operadores. A probabilidade de ocorrer mutação é sempre bem menor que a de ocorrer cruzamento, quase que na totalidade das aplicações.

Os indivíduos selecionados para serem manipulados pelos operadores genéticos são inicialmente divididos em casais e, a esses, é aplicado o operador genético de cruzamento. A seguir, a cada novo indivíduo gerado, aplica-se o operador genético da mutação. Ao final, tem-se uma nova geração e dessa, espera-se que, em média, a nova população tenha maior adaptação que a anterior e assim sucessivamente.

As populações seguintes à população inicial, geralmente, apresentam o mesmo número de indivíduos, porém, a composição pode ser totalmente diferente. A população inicial tem seus indivíduos gerados aleatoriamente dentro de uma faixa previamente definida; já a população seguinte é obtida, principalmente, através da manipulação genética da população inicial.

6.3.10 A Função Aptidão

O grau de adaptação de cada indivíduo é a aptidão, também denominada “*fitness*” e é obtido pela avaliação do indivíduo através da função a ser otimizada.

A função de aptidão deve refletir a qualidade de um elemento em solucionar o problema [BARBOSA, 1997], [GOLDBERG, 1989]. A regra que a determina depende do tipo de problema que está sendo considerado e, nos problemas de otimização, minimização ou maximização, está diretamente relacionada com a função objetivo. Ao se considerar o Problema do Caixeiro Viajante, por exemplo, a regra da função de aptidão pode ser a expressão que fornece a distância total percorrida pelo viajante.

Considerando que no decorrer das iterações os indivíduos vão se tornando cada vez mais semelhantes, pois a população tende a convergir, pode ser interessante aumentar a pressão de seleção, utilizando como função de aptidão uma composição da função objetivo com alguma função conveniente.

6.3.11 Os Esquemas de Seleção

São diversos os esquemas de seleção utilizados nos AG's. No esquema de seleção conhecido como seleção proporcional, a probabilidade de um indivíduo ser selecionado para participar do processo de reprodução é proporcional a medida relativa do grau de *fitness* (aptidão) do indivíduo relativamente à população. Nesse caso, a probabilidade p_i

do indivíduo a_i ser selecionado, poderia ser dada pela fórmula: $p_i = \frac{f(a_i)}{\sum_{j=1}^n f(a_j)}$ onde f é

a função de aptidão e n o tamanho da população.

Este tipo de esquema de seleção que tem o efeito de aumentar a aptidão média da população costuma ser chamada de seleção direcional [BARBOSA, 1997], [LOPES, 1996], [GOLDBERG, 1989]. Na prática, tudo se passa como se o sorteio dos elementos fosse feito através de um jogo de roleta, onde a probabilidade de cada indivíduo ser selecionado é proporcional ao seu *fitness*.

Em algumas situações, pode-se deixar de lado a magnitude do grau de *fitness* de um indivíduo, levando em consideração apenas o seu “*ranking*”, ou posição relativa da medida de aptidão [MAYERLE, 1994].

6.3.12 Convergência, Diversidade Populacional e Nichos

Um dos critérios para a convergência de uma população e de um gene em um cromossomo foi definido por DE JONG [1975]. Segundo ele, um gene converge quando 95% da população possui o mesmo gene, enquanto que uma população converge quando todos os seus genes convergem.

O termo diversidade diz respeito à falta de semelhança entre os indivíduos de uma população e sua perda está diretamente ligada à convergência da mesma. Em uma situação ideal, um AG deveria convergir sem perda de diversidade genética. Isso aumentaria as chances de se encontrar o ótimo global através de um equilíbrio entre uma exploração global e local.

Para diminuir a perda da diversidade, alguns AG's utilizam a chamada “redução de incesto” que reduz a operação de *crossover* entre elementos muito semelhantes, permitindo a recombinação apenas entre indivíduos cuja distância de *Hamming* seja grande. A distância de *Hamming* permite medir a diversidade entre dois cromossomos, sendo definida como o número de *alelos* (valores que os *genes* podem tomar) diferentes para as mesmas posições relativas [LOPES, 1996].

Muitas vezes, utiliza-se também o conceito de nicho ecológico, que consiste em manter subpopulações estáveis de indivíduos, de uma forma tal que cada subpopulação explore uma região do espaço de busca, sem que haja competição entre elas. Para aumentar a possibilidade de formação destas subpopulações pode-se diminuir a pressão seletiva, utilizando o fator de *crowding* que faz com que os “cromossomos-filhos” venham a substituir os elementos com os quais tenham a máxima semelhança.

Quando o espaço de busca se torna muito grande, torna-se necessário o uso de uma população também numerosa, fazendo, muitas vezes, com que os AG's clássicos (seqüenciais) se tornem menos eficientes. Ainda existe a questão dos parâmetros que

deve ser considerada. É muito difícil definir os parâmetros mais adequados para um determinado problema. O tamanho da população, por exemplo, vai depender, dentre outros fatores, do espaço de busca considerado, não sendo possível, *a priori*, determinar o tamanho ideal de uma população para uma determinada classe de problemas. GREFENSTETTE [1986], sugere que na grande maioria das aplicações uma população de 50 a 200 indivíduos possa parecer adequado. Assim, o método empírico de tentativas e erros, guiado por uma experiência prévia, parece ser o caminho mais indicado para o equacionamento de problemas.

6.3.13 Algoritmos Genéticos Paralelos

A idéia dos AG's paralelos surgiu do conceito de se evoluir diversas populações de forma paralela, muitas vezes, compartilhando informações entre elas [LOPES, 1996]. Obviamente, a implementação de um AG paralelo depende diretamente do *hardware* disponível pelo usuário. Existem diferentes abordagens para AG's paralelos. Na mais comum destas abordagens, diversas sub-populações evoluem paralelamente, sendo que o controle de cada uma delas é atribuído a um processador particular. Eventualmente, pode ocorrer uma migração de indivíduos, geralmente os melhores, entre os processadores, acarretando um aumento na diversidade genética da população que os recebe. Nos casos onde o custo computacional da avaliação de um indivíduo é significativamente grande, pode-se atribuir a outro processador esta tarefa.

Fica claro que na implementação de qualquer abordagem de AG paralelo, torna-se necessário a introdução de parâmetros de controle adicionais. Entre eles pode-se destacar: a quantidade de elementos que irão migrar, quais serão estes indivíduos, de onde e para onde será feita esta migração, etc.

6.4 O Algoritmo Genético Utilizado neste Trabalho

6.4.1 Algoritmo Genético para o Problema das P-Medianas

Inicialmente, será definida a estrutura do cromossomo, a função de aptidão, o processo de seleção e os operadores utilizados no algoritmo proposto neste trabalho para resolver o problema. A estrutura adotada é a mesma proposta por NUNES [1998] e por SAMPAIO [1999], para o problema das p-medianas.

a) Estrutura do Cromossomo:

Seja $G(V, E)$ um grafo não direcionado, onde V é o conjunto de vértices e E é o conjunto de arestas. Cada um dos m cromossomos (r_i) será um subconjunto de V contendo p elementos, ou seja, $r_i \subset V$ e $|r_i| = p$, para todo $i \in \{1, 2, \dots, m\}$.

b) A função de aptidão

Neste algoritmo, adotou-se como fitness o número de transmissão de cada cromossomo que representa a soma das distâncias percorridas pelos leituras, ao longo do grupo que deve ser percorrido, ou seja:

$$C_i = \text{fitness}(r_i) = \sum_{v_j \in V} \{\min_{v_k \in r_i} [d(v_k, v_j)]\}$$

Especificamente, neste exemplo, quanto menor é o número de transmissão, melhor é o fitness do indivíduo.

c) O processo de seleção

No processo de seleção foi utilizada a mesma função de seleção adotada por MAYERLE, 1994, para o problema do caixeiro viajante, o que torna provável a escolha para a reprodução de indivíduos com melhor fitness, isto é:

$$\text{Select}(R) = \left\{ r_j \in R / j = m + 1 - \left\lceil \frac{-1 + \sqrt{1 + 4 \cdot \text{rnd}(m^2 + m)}}{2} \right\rceil \right\}$$

onde R é uma lista $R = (r_1, r_2, \dots, r_m)$, com m cromossomos viáveis de p elementos cada, sorteados entre os v vértices do grafo. $rnd \in [0, 1]$ é um número aleatório uniformemente distribuído e $\lceil b \rceil$ é o menor inteiro maior do que b .

d) O operador de recombinação crossover

Sejam r_i e r_j dois cromossomos selecionados, com p elementos cada. no cromossomo r_i são sorteados c elementos, $c \in \{1, 2, \dots, p\}$, que serão transferidos para o cromossomo r_j . No cromossomo r_j são sorteados c elementos que serão transferidos para o cromossomo r_i . Desta maneira, foi adotado o cruzamento com n -partições.

Caso os dois cromossomos criados no processo de crossover não sejam viáveis, faz-se a mutação substituindo-se, num dos cromossomos, os elementos repetidos por outros sorteados entre os elementos que não pertencem ao cromossomo considerado.

6.4.1.1 Descrição do Algoritmo Proposto

A seguir, serão apresentados os passos do algoritmo genético proposto, o qual será chamado de AGPMed.

Passo 1. Construção da População Inicial

- Gere uma lista $R = (r_1, r_2, \dots, r_m)$, com m cromossomos viáveis de p elementos cada, sorteados entre os v vértices do grafo;
- Calcule $C_i = \text{fitness}(r_i)$, $\forall r_i \in R$;
- Ordene a lista R de modo que $C_1 \leq C_2 \leq \dots \leq C_m$;
- Faça $k = 0$ e defina o erro ε e o número máximo de iterações k_{\max} ;

Passo 2. Teste

- Se $C_m - C_1 \leq \varepsilon$ ou $k \geq k_{\max}$, então PARE e apresente o cromossomo r_1 ;

Passo 3. Seleção

- Selecione dois cromossomos, $r_i = \text{Select}(R)$ e $r_j = \text{Select}(R)$, com $r_i \neq r_j$;

Passo 4. Crossover

- Faça a reprodução. Crossover $(r_i, r_j) = \{r_x, r_y\}$;

Passo 5.

- Se r_x e r_y forem cromossomos viáveis, faça:

$$\begin{cases} r_t = r_x, & \text{se } \text{fitness}(r_x) \leq \text{fitness}(r_y) \\ r_t = r_y, & \text{caso contrário} \end{cases}, \text{ e vá ao Passo 7.}$$

- Ou então, se apenas um entre r_x e r_y é viável, faça:

$r_t = r_x$ ou r_y (viável), e vá ao Passo 7.

Passo 6. Mutação (no caso de r_x e r_y serem não viáveis)

- Escolha aleatoriamente r_x ou r_y ;
- Faça a mutação no cromossomo escolhido, produzindo r_t ;

Passo 7.

- Se $\text{fitness}(r_t) < \text{fitness}(r_m)$, faça:
 - Elimine r_m (o pior cromossomo) da lista R ;
 - Insira r_t na lista R , mantendo a ordem crescente dos fitness;
- Faça $k = k + 1$ e volte ao Passo 2.

6.5 Algoritmo Para Formação dos *Clusters* (Agrupamentos)

Obtidas as medianas que, para o problema aqui estudado, servirão como depósitos fictícios para cada um dos veículos (leitristas), deve-se, então, determinar qual será o grupo de atendimento (os pontos de parada) de cada um destes leitristas. Isto será obtido a partir de algoritmo de designação.

No método proposto por Gillett e Johnson [GOLDEN, 1977] [BODIN, 1983] os pontos de parada são designados aos depósitos de acordo com o algoritmo, considerando-se restrição de capacidade. Neste trabalho, o algoritmo será chamado de algoritmo M_Dep_Rz, e está descrito a seguir.

Passo 1. Calcule a distância entre cada nó ainda não designado, até cada um dos depósitos fictícios, cujos veículos correspondentes ainda possuam espaço;

Passo 2. Para cada nó i do passo anterior, obter t_i^1 como sendo o depósito mais próximo de i e t_i^2 como sendo o segundo depósito mais próximo de i , com distâncias iguais a c_i^1 e c_i^2 , respectivamente;

Passo 3. Para todos os nós i dos passos anteriores, calcular a razão $r_i = c_i^1/c_i^2$. Ordenar os nós i de acordo com os valores de r_i , em ordem crescente. Essa lista determina a ordem em que os nós serão designados a um dos depósitos: aqueles nó relativamente próximos a um depósito serão considerados primeiro;

Passo 4. Percorrer a lista do passo anterior, designando os nós i aos depósitos mais próximos, até que a capacidade do veículo correspondente a um depósito fictício, esteja esgotada. Nesse caso, volte ao passo 1.

A seguir, no capítulo VII, são mostradas as aplicações das técnicas ao problema abordado.

CAPITULO VII

7. Aplicação das Técnicas Apresentadas à Solução do Problema Proposto - Implementação Computacional

Para resolver o problema real descrito no Capítulo III, efetuou-se a implementação computacional de alguns dos algoritmos apresentados. Uma descrição detalhada, assim como os resultados obtidos, são apresentados a seguir.

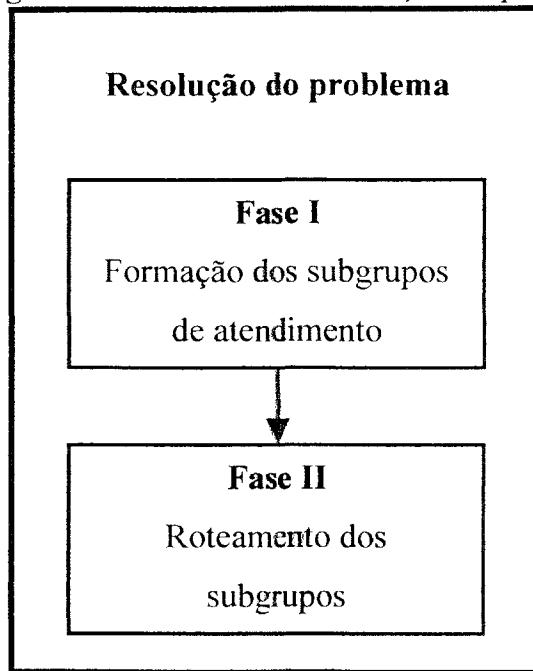
7.1 Introdução

Preliminarmente à implementação computacional dos algoritmos, definiu-se a área de atuação dos leiturstas na cidade de Pato Branco. Para fazer a leitura dos medidores de consumo de água, os leiturstas devem percorrer, mensalmente, uma rede viária, composta de ruas. Cada rua é dividida em trechos (quadras), onde cada trecho é um pedaço de rua compreendido entre duas ruas transversais, os quais podem ou não possuir medidores de água; os que possuem são os trechos produtivos, os outros são chamados improdutivos.

O problema é, então, determinar qual leiturstas deverá atender a quais pontos de leitura, de maneira a percorrer seu trajeto de forma otimizada, atravessando o mínimo possível trechos improdutivos.

Portanto, o problema a ser tratado divide-se em duas fases: divisão da parte da rede viária a ser estudada em subgrupos, sendo que cada subgrupo deverá ser percorrido por um leiturstas e o roteamento do percurso nos diferentes subgrupos, conforme apresentado no Fluxograma 7.1.

Fluxograma 7.1 – Fases de resolução do problema



7.2 Etapas da Implementação Computacional

A implementação computacional dos algoritmos para solução do problema foi dividida em 4 etapas distintas:

Na primeira etapa, efetuou-se o cadastramento de vários pontos no mapa digitalizado da cidade de Pato Branco. Esses pontos foram dispostos de maneira a mapear a parte da rede viária a ser estudada, que conforme mencionado no capítulo III, foi escolhida por apresentar características de centro e bairros e, também, por apresentar trechos com vários clientes e longos trechos com poucos clientes. Foram introduzidos pontos nas interseções das ruas e no meio das quadras, buscando a melhor caracterização dos trechos produtivos. No mapa, pequenos símbolos indicarão a presença destes pontos. Assim, um banco de dados é formado contendo as coordenadas geográficas de cada ponto.

Na segunda etapa, utilizando o Algoritmo Genético Aplicado ao Problema das P-Medianas (AGPMed), definiu-se 12 pontos que representam uma boa aproximação para o problema de localização de 12 facilidades (12-medianas que representarão depósitos fictícios) do grafo cujos vértices são os 774 pontos que demarcam a área de atendimento em estudo. Depois, utilizou-se o Algoritmo de Teitz e Bart, com o mesmo objetivo, cuja solução inicial foi o resultado obtido pelo AGPMed. A idéia foi encontrar 12 pontos de

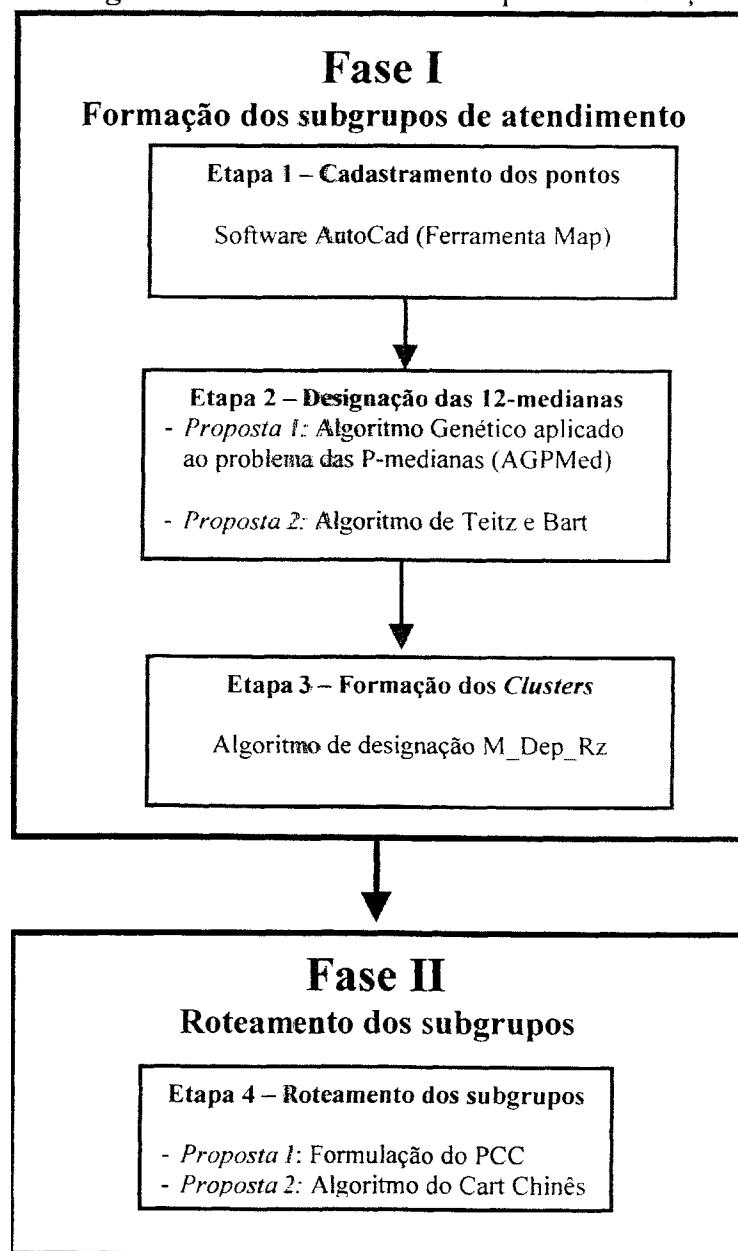
modo que a soma das distâncias de cada vértice do grafo à facilidade mais próxima fosse mínima, comparativamente ao Algoritmo Genético.

Em torno de cada uma das 12 medianas, formou-se *clusters* (agrupamentos), realizando-se a terceira etapa. Para isto, utilizou-se o algoritmo de designação M_Dep_Rz, que é uma adaptação feita a partir da heurística proposta por Gillet e Johnson.

Na quarta etapa, para cada um dos agrupamentos, construiu-se roteiros pela Formulação do Problema do Carteiro Chinês e pelo Algoritmo do Carteiro Chinês.

Estas etapas estão ilustradas no Fluxograma 7.2.

Fluxograma 7.2 – Resumo das etapas de resolução



É importante ressaltar que, em todas as fases deste trabalho, foram consideradas apenas as distâncias euclidianas entre os vértices do grafo.

7.3 A Implementação Computacional e Obtenção dos Resultados

A programação computacional dos algoritmos utilizados neste trabalho foi feita através da Linguagem Pascal e a utilização do Software LINGO ® 6.0, instalados em um microcomputador *Pentium II* – 400 MHz e 64 MB de memória RAM, sendo 4 MB compartilhada para vídeo.

7.3.1 O Cadastramento dos Dados – Definição do Grafo Inicial

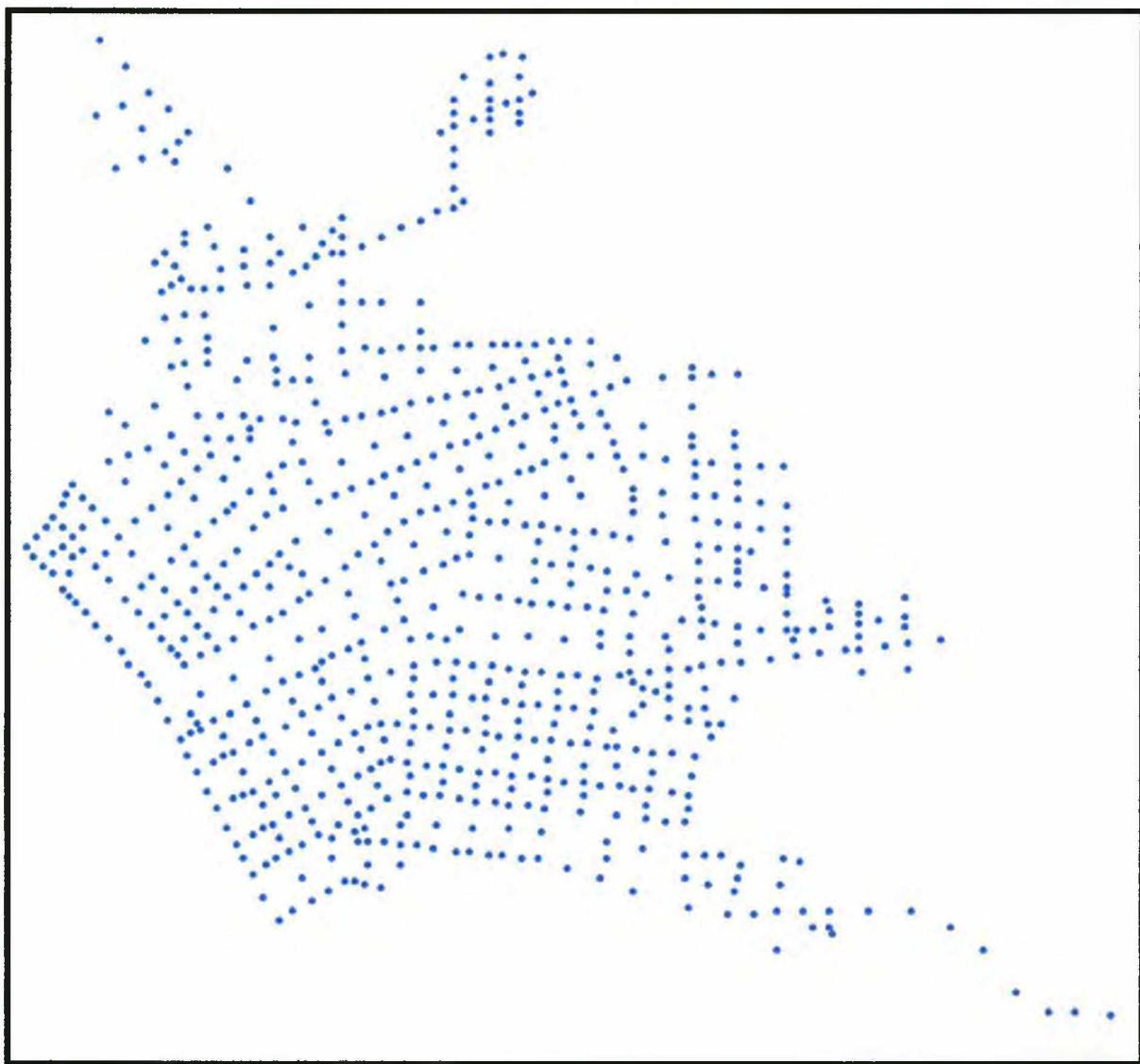
O cadastramento dos dados foi feito um a um. Em cada trecho de rua, existem vários clientes que devem ser atendidos pelos leitistas. Dessa forma, associou-se, a apenas um ponto, todos esses clientes que devem ser atendidos, que é estabelecido como o ponto médio do trecho produtivo.

Estes pontos médios formam, assim, o conjunto dos pontos de parada do problema. O problema é, então, redefinido da seguinte forma: determinar qual leitista deverá atender a quais pontos de parada, de modo a percorrer o menor caminho para atender a todos os seus pontos de parada.

Designar um ponto de parada a um leitista é o correspondente a designar um trecho produtivo ao leitista e, desta forma, estão definidos os seus pontos de leitura.

Os resultados da inserção dos pontos no mapa digitalizado da cidade de Pato Branco (PR), que indicarão os trechos produtivos, podem ser visualizados na Figura 7.1.

Figura 7.1 – Pontos que definem a área de atendimento estudada



7.3.2 A Definição das 12-Medianas

Para o cálculo das 12-medianas utilizou-se, primeiramente, o algoritmo genético descrito no capítulo VI. Fixou-se o número máximo de iteração $k_{\text{máx}} = 1000$ e uma população variando de 75 a 200 indivíduos. O algoritmo pára, quando se atinge o número máximo de iterações, ou quando a diferença dos valores de transmissão entre o melhor e o pior é menor de 2.000 metros.

Considerando-se o aspecto probabilístico dos algoritmos genéticos, efetuou-se 5 simulações. Em cada uma dessas simulações, a partir da solução obtida pelo algoritmo genético, iniciou-se o algoritmo de Teitz e Bart, buscando melhorar a solução através da

substituição de vértices. Os resultados são apresentados no Quadro 7.1, onde os valores de transmissão são dados em metros e os tempos computacionais em segundos.

Quadro 7.1 – Resultados numéricos das simulações para o problema das 12-medianas

Simulação	Algoritmo Genético (AGPMed)				Algoritmo de Teitz e Bart			
	Iterações	População	Valor Trans. (metros)	Tempo Comp. (segundos)	Iterações	Valor Trans. (metros)	Tempo Comp. (segundos)	
1	1000	200	201.569,1775	7.290	17	185.152.4898	7.444	
2	1000	150	202.306,3062	7.205	14	184.663,2083	6.192	
3	1000	75	203.176,7247	6.790	18	184.920.8433	7.811	
4	1000	200	202.946,4390	7.430	12	187.156.6691	6.057	
5	1000	100	213.958,3634	6.950	15	186.164.3975	6.369	

Verifica-se no Quadro 7.1 que, nas 5 simulações, o critério de parada foi o número de iterações previstas para o algoritmo. Em nenhuma dessas simulações o erro, ou seja, a diferença dos valores de transmissão entre o melhor e o pior cromossomo, foi menor que 2.000 metros.

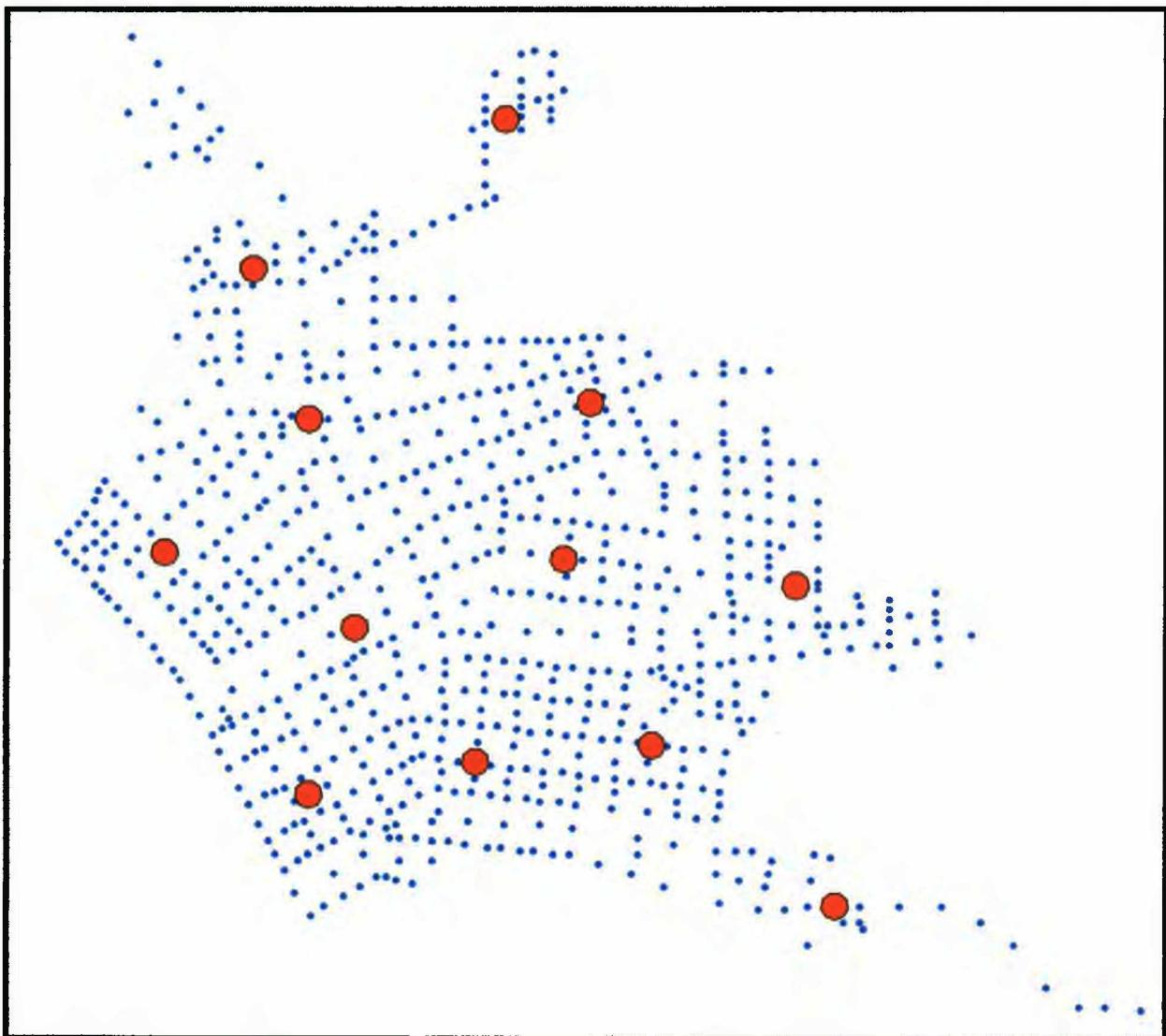
O Quadro 7.2 apresenta os vértices encontrados no AGPMed e no Algoritmo de Teitz e Bart, em cada simulação.

Quadro 7.2 – As 12-medianas encontradas em cada uma das 5 simulações.

Simulação	Vértices determinados pelo:							
	AGPMed				Algoritmo de Teitz e Bart			
1	639	636	477	428	750	625	474	325
	300	351	616	95	489	364	576,	95
	690	215	63	272	678	650	64	272
2	446	272	412	35	474	272	598	64
	357	551	323	95	364	541	325	111
	690	655	639	694	679	640	745	198
3	682	584	720	272	678	579	650	272
	357	624	323	95	364	618	326	95
	63	456	502	570	64	458	498	750
4	538	720	326	95	541	721	326	95
	357	412	495	272	364	746	474	272
	690	215	63	613	688	231	65	600
5	420	584	720	95	745	598	575	95
	357	329	675	56	364	325	679	64
	215	533	456	667	232	533	474	281

O melhor resultado foi obtido na simulação 2 e pode ser visualizado na Figura 7.2, onde foram destacados os 12 vértices que representam as 12 medianas: **64, 111, 198, 272, 325, 364, 474, 541, 598, 641, 679, 745** que serão utilizadas na etapa seguinte.

Figura 7.2 – Destaque das 12-medianas em relação aos demais pontos de parada.



Como pode-se observar na Figura 7.2, as medianas ficaram bem distribuídas, em relação aos demais pontos de parada, na área de atendimento estudada.

7.3.3 Formação dos *Clusters*

Após ter-se definido as 12-medianas, essas funcionarão como sementes ou depósitos fictícios. Associou-se a cada uma destas sementes um leiturista diferente, buscando-se que cada um atendesse, aproximadamente, a mesma quantidade de clientes. Para tanto utilizou-se o algoritmo M_Dep_Rz, que é uma adaptação do algoritmo de designação proposto por Gillet e Johnson [GOLDEN, 1977], [BODIN, 1983], descrito no capítulo VI. Foi proposto que cada leiturista atendesse em torno de 70 vértices.

A designação dos pontos a cada uma das 12 sementes originará a formação dos 12 *clusters* de atendimento, que são apresentados no quadro 7.3

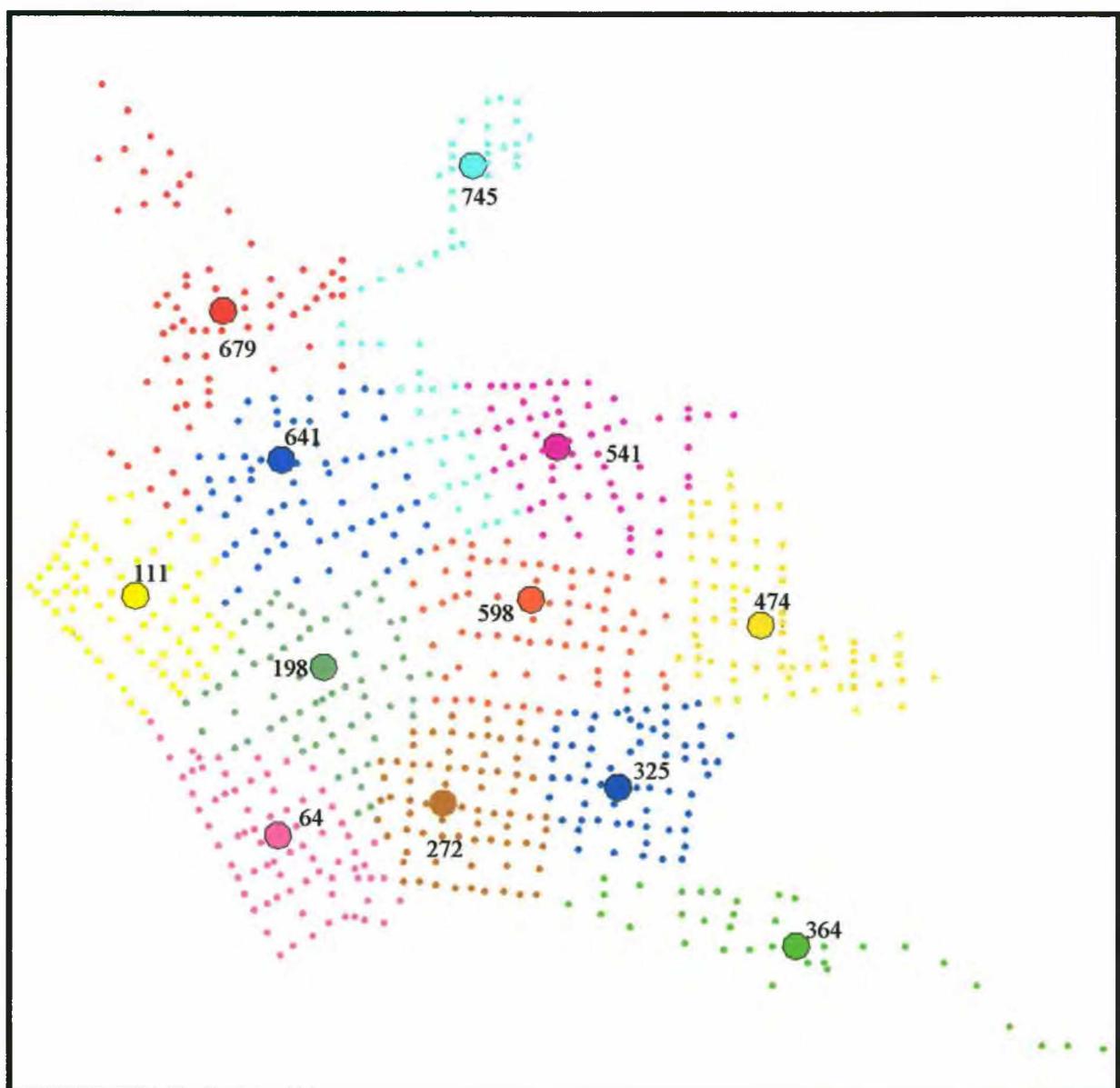
Quadro 7.3 – Formação dos 12 *clusters*, utilizando o algoritmo M_Dep_Rz

	Vértices designados a cada um dos <i>clusters</i>	Total de pontos atendidos
<i>Cluster 1</i>	64, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 65, 66, 67, 68, 69, 70, 71, 73, 75, 139, 141, 142, 144, 145, 147, 148, 149, 151, 152, 153, 154, 155, 156, 157, 158, 160, 161, 162, 163, 164, 166, 169	76
<i>Cluster 2</i>	111, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 127, 128, 208, 211, 212, 217	69
<i>Cluster 3</i>	198, 72, 74, 76, 77, 78, 79, 80, 126, 129, 130, 131, 132, 133, 135, 136, 137, 138, 140, 143, 146, 150, 171, 172, 173, 177, 184, 185, 186, 187, 188, 189, 191, 192, 193, 194, 195, 196, 197, 199, 200, 201, 202, 203, 204, 205, 206, 236, 237, 238, 288, 289, 292, 401, 402, 403, 405, 618, 620, 621, 622, 623, 624, 625, 626, 627	66
<i>Cluster 4</i>	272, 165, 167, 168, 170, 174, 175, 176, 178, 179, 180, 181, 182, 183, 190, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 267, 268, 269, 270, 271, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 293, 294, 295, 296, 297, 298, 299, 300, 305, 306, 307, 308, 309, 310, 313	72
<i>Cluster 5</i>	325, 228, 265, 266, 311, 312, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 396, 398, 399, 400, 435, 467, 468	62
<i>Cluster 6</i>	364, 261, 262, 263, 264, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376	35
<i>Cluster 7</i>	474, 428, 429, 430, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 469, 470, 471, 472, 473, 475, 476, 477, 478, 479, 480, 481, 482, 483, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 509, 511, 512, 513, 514, 515, 516, 517, 518, 519, 519, 774	71
<i>Cluster 8</i>	541, 505, 506, 507, 508, 510, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 558, 559, 560, 561, 562, 563, 564, 565, 566, 570, 576, 578, 579, 580, 582, 586, 587, 588, 590, 616	62
<i>Cluster 9</i>	598, 290, 291, 301, 302, 303, 304, 395, 404, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 431, 432, 433, 434, 466, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 591, 592, 593, 594, 595, 596, 597, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 611, 614, 617	67
<i>Cluster 10</i>	641, 125, 207, 209, 210, 213, 214, 215, 216, 225, 227, 230, 231, 232, 233, 234, 235, 610, 612, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 640, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 663, 718, 727, 728, 730, 731, 732, 733, 762, 763, 765, 766, 767, 768, 771, 772	66
<i>Cluster 11</i>	679, 218, 219, 220, 221, 222, 223, 224, 226, 229, 661, 662, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 719	66
<i>Cluster 12</i>	745, 134, 159, 567, 568, 569, 571, 572, 573, 574, 575, 577, 581, 583, 584, 585, 613, 615, 639, 720, 721, 722, 723, 724, 725, 726, 729, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 764, 769, 770, 773	58

No Quadro 7.3 tem-se que o primeiro vértice de cada *cluster* representa uma das 12-medianas determinadas na etapa anterior.

Observa-se nesse quadro, que o número vértices atendidos por cada leitorista varia de 58 a 76 vértices, com exceção do leitorista do *cluster* 6, que atenderá 35 pontos. Isso justifica-se pelo fato de sua área possuir quadras mais extensas e, como os pontos demarcam essa área, tem-se um número menor de vértices; porém, a área a ser percorrida é praticamente a mesma dos demais leitoristas. Isso poderá ser verificado na figura 7.3, onde contempla-se, também, a determinação dos 12 clusters definidos pelo algoritmo M_Dep_Rz.

Figura 7.3 – Determinação dos 12 *clusters* definidos pelo algoritmo M_Dep_Rz



Na próxima etapa definiu-se a sequência de atendimento dos *clusters* formados acima, ou seja, o roteamento dos subgrupos.

7.3.4 Roteamento dos subgrupos

De acordo com o fluxograma 7.2, na segunda fase da resolução do problema, busca-se o roteamento dos subgrupos obtidos pelo algoritmo M_Dep_Rz. Para tanto, surgem duas propostas: Formulação do Problema do Carteiro Chinês e o Algoritmo do Carteiro Chinês.

7.3.4.1 Formulação do Problema do Carteiro Chinês.

A Formulação do Problema do Carteiro Chinês representa a proposta 1 para solução de um problema de cobertura de arcos. A formulação apresentada no item 5.5.2.1.3 foi desenvolvida para cada subgrupo separadamente.

Nesta formulação onde havia $x_{ij} + x_{ji} \geq 1$ (restrição (5.14)), adotou-se $x_{ij} + x_{ji} \geq 2$ em função da leitura em ambos lados da rua, ou seja, considerou-se arcos duplos em todos os trechos produtivos.

No anexo 1 apresenta-se, como exemplo, a formulação para a rota número 272, de acordo com o grafo (Figura A.2) obtido a partir do mapa do *cluster* 272 (Figura A.1). Nesse anexo, também são fornecidos os resultados obtidos pelo Software LINGO, para esta rota.

Aplicou-se a Formulação do Problema do Carteiro Chinês para todos *clusters* e, utilizando o Software LINGO, obteve-se os resultados que são apresentados no quadro 7.4.

Quadro 7.4 – Resultados obtidos pelo Software LINGO

Rota	Distância percorrida (em metros)
Rota 64	9.073
Rota 111	8.040
Rota 198	8.112
Rota 272	9.204
Rota 325	7.076
Rota 364	6.575
Rota 474	6.579
Rota 541	8.445
Rota 598	8.988
Rota 641	8.507
Rota 679	9.185
Rota 745	8.055
TOTAL	97.839

O Quadro 7.4 fornece a distância percorrida por cada leiturista e a distância total, 97.839 metros, que ao ser comparada com a situação atual, Quadro 3.4, num total de 105.797 metros, apresenta uma melhoria em torno de 8,13%. Nestas informações, podem-se avaliar as maiores rotas. Nesta proposta, é a rota 272 com 9.204,6410 metros; na situação atual, a maior rota é a 11040, com 11.180,29 metros, o que representa uma diferença de 1.975,65 metros.

7.3.4.2 Algoritmo do Carteiro Chinês

Na proposta 2 da segunda fase, sugere-se o uso do Algoritmo do Carteiro Chinês. Esse algoritmo foi implementado fornecendo a sequência a ser percorrida pelos leituras e, também, o comprimento, em metros, que cada uma dessas rotas terá. Esses valores podem ser visualizados no quadro 7.5.

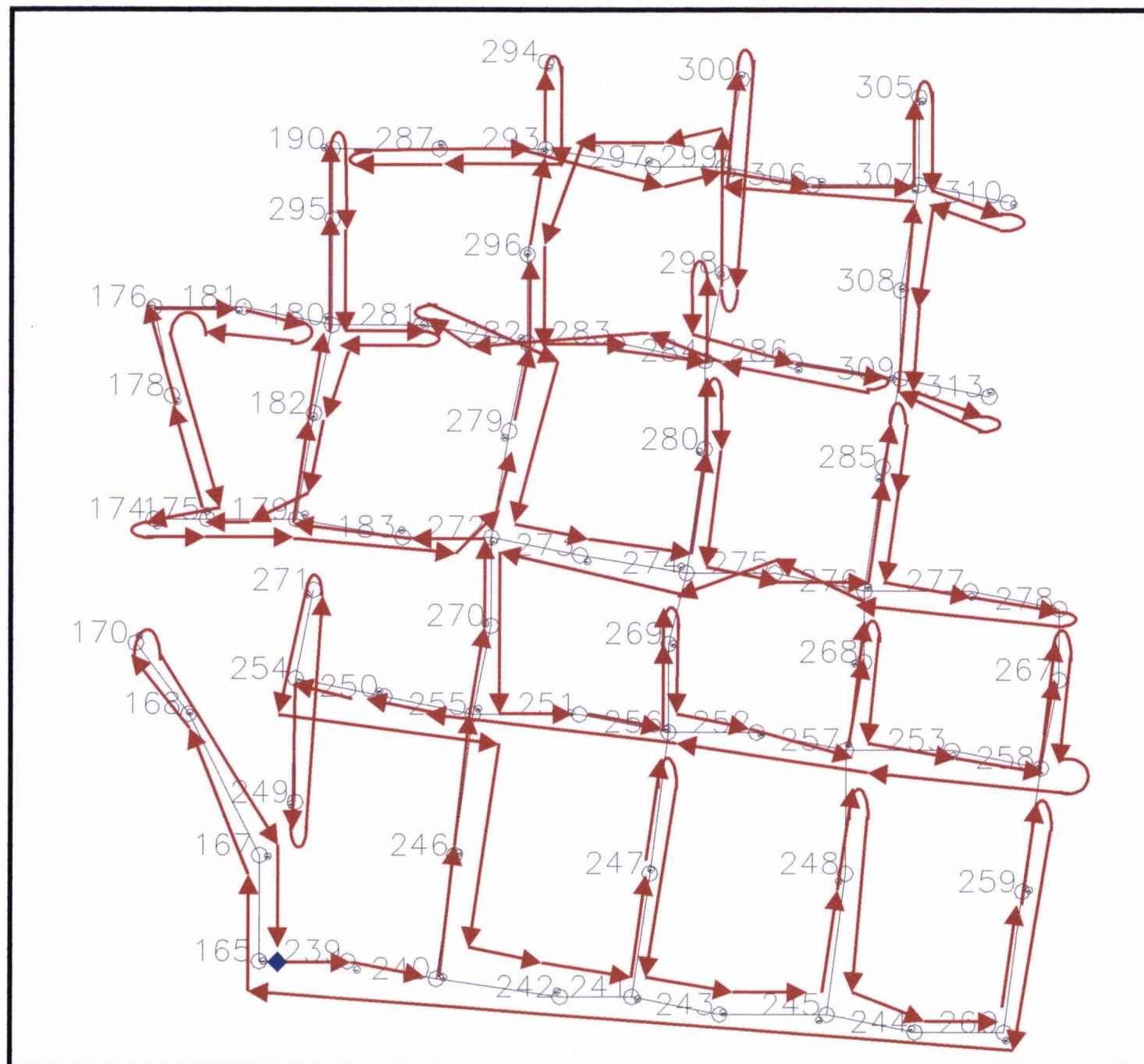
Quadro 7.5 - Distância percorrida, em metros, pelos leituras, nas rotas obtidas pelo Algoritmo do Carteiro Chinês

Rota	Distância percorrida (em metros)
Rota 64	9.175
Rota 111	8.231
Rota 198	8.264
Rota 272	9.369,
Rota 325	7.314
Rota 364	6.617
Rota 474	6.584
Rota 541	8.687
Rota 598	9.196
Rota 641	8.532
Rota 679	9.240
Rota 745	8.081
TOTAL	99.290

Comparando-se as distâncias encontradas pelo Algoritmo do Carteiro Chinês, no Quadro 7.5, e as distâncias encontradas pela Formulação do PCC, no Quadro 7.4, observamos que a diferença fica em torno de 1%. Essa diferença é devida ao modo de execução de cada uma das propostas. A Formulação matemática do PCC foi elaborada manualmente, considerando-se um arredondamento de valores. e o algoritmo do Carteiro Chinês foi implementado considerando-se todas as casas decimais das coordenadas geográficas dos 774 pontos utilizados.

Na figura 7.4 é apresentada a recuperação da rota 272, onde a sequência de pontos obtida pelo Algoritmo do Carteiro Chinês a ser seguida é a seguinte: 165, 239, 240, 246, 255, 270, 272, 183, 179, 182, 180, 295, 190, 295, 180, 281, 180, 182, 179, 175, 178, 176, 181, 180, 181, 176, 178, 175, 174, 175, 179, 183, 272, 279, 282, 296, 293, 294, 293, 287, 190, 287, 293, 297, 299, 306, 307, 305, 307, 310, 307, 308, 309, 313, 309, 308, 307, 306, 299, 300, 299, 298, 299, 297, 293, 296, 282, 283, 284, 298, 284, 286, 309, 286, 284, 283, 282, 281, 282, 279, 272, 273, 274, 280, 284, 280, 274, 275, 276, 285, 309, 285, 276, 277, 278, 277, 276, 275, 274, 273, 272, 270, 255, 251, 256, 269, 274, 269, 256, 252, 257, 268, 276, 268, 257, 253, 258, 267, 278, 267, 258, 259, 260, 259, 258, 253, 257, 252, 256, 251, 255, 250, 254, 249, 254, 271, 254, 250, 255, 246, 240, 242, 241, 247, 256, 247, 241, 243, 245, 248, 257, 248, 245, 244, 260, 244, 245, 243, 241, 242, 240, 239, 165, 167, 168, 170, 168, 167, 165

Figura 7.4 – Recuperação da rota 272 – Algoritmo do Carteiro Chinês



CAPITULO VIII

8. ANÁLISE DOS RESULTADOS NUMÉRICOS, CONCLUSÕES FINAIS E SUGESTÕES PARA TRABALHOS FUTUROS

O objetivo deste trabalho foi o de propor um modelo informatizado para o planejamento e otimização das rotas dos leituristas a ser utilizado pela Empresa de Saneamento Básico do Paraná, no município de Pato Branco.

8.1 Análise dos Resultados

Inicialmente, são analisados os resultados de cada fase em cada uma de suas etapas.

8.1.1 Fase I – Formação dos Subgrupos de Atendimento

A Fase I foi dividida em três etapas. Na primeira etapa, utilizando-se o Software AutoCad, obteve-se as coordenadas geográficas dos 774 pontos que demarcavam os trechos produtivos do grupo a ser trabalhado, formando-se assim, o grafo inicial.

Na segunda etapa foram designadas as 12 sementes para a subsequente formação dos grupos de atendimento para cada um dos leituristas. Nessa etapa, trabalhou-se com duas propostas de solução: a primeira com algoritmo genético aplicado ao problema das P-medianas; a outra, com o Algoritmo de Teitz e Bart.

Analizando-se o quadro 7.1, pode-se notar que o tempo computacional, nas cinco simulações, é praticamente o mesmo nas duas propostas. Porém, o que precisa ser analisado é que o Algoritmo Genético parte de uma solução inicial aleatória, faz 1000

iterações e apresenta uma solução aproximada da ótima. Na proposta deste trabalho, usou-se, como solução inicial para o Algoritmo de Teitz e Bart, a solução proposta pelo Algoritmo Genético e buscou-se melhorá-la. Assim, o número de iterações variou de 12 a 18, que não é um número tão elevado, em vista que busca-se 12 medianas em 774 pontos do grafo.

Na solução escolhida como ótima para continuação do trabalho, o algoritmo de Teitz e Bart melhorou em 8,72% a solução fornecida pela proposta 1 – aplicação do Algoritmo Genético. Esse resultado reflete a boa combinação desses dois algoritmos.

No Algoritmo Genético, o número de indivíduos para formar a população variou de 75 a 200 e apresentou a melhor solução com 150 indivíduos, o que mostra não ser necessário uma população numerosa, haja visto que esse algoritmo é probabilístico e busca os indivíduos mais aptos para o *crossover*.

Na terceira etapa, utilizou-se o algoritmo de designação proposto por Gillett e Johnson, formando-se os agrupamentos em torno das medianas encontradas na fase anterior. Esse algoritmo foi implementado com restrição de capacidade para cada um dos depósitos. Assim, pode-se verificar no quadro 7.3 e na figura 7.3 que a área de atendimento de cada leitorista é aproximadamente a mesma.

8.1.2 Fase II – Roteamento dos Subgrupos

Na segunda fase, tem-se duas propostas para o roteamento dos *clusters*, os grupos de atendimento de cada leitorista, obtidos na fase anterior.

A primeira proposta foi a formulação do Problema do Carteiro Chinês (PCC), e isso foi exemplificado com a formulação completa da rota 272, apresentada no Anexo 1. Apresentou-se a estrutura da Função Objetivo e suas respectivas restrições. O modelo foi resolvido utilizando o Software LINGO e a solução é mostrada logo em seguida. Percebe-se que o tempo de execução foi muito pequeno, apenas 1 segundo para um total de 108 iterações. A formulação considerou apenas os arcos válidos, ou seja, os vértices que podem ser realmente ligados.

Na segunda proposta, utilizou-se o Algoritmo do Carteiro Chinês que foi implementado de modo que se insere os pontos a serem roteirizados, confirmando-se a ligação entre os pontos.

Em seguida, o programa gera o ciclo a ser percorrido e a distância correspondente a esse percurso. O tempo computacional foi muito pequeno. Depois de implementado, pode-se gerar a rota de qualquer conjunto de pontos. No quadro 7.5 pode-se visualizar o comprimento das rotas obtidas pelo Algoritmo do Carteiro Chinês.

Comparando-se as soluções das duas propostas para roteamento, quadro 7.4 e 7.5, percebe-se que a solução do Algoritmo do Carteiro Chinês aproxima-se muito da solução da Formulação do Carteiro Chinês.

8.2 Conclusões

A utilização do Algoritmo Genético melhorado com a utilização do Algoritmo de Teitz e Bart fornece uma boa solução para o problema das p-medianas. O algoritmo M_Dep_Rz é fácil de ser implementado e apresenta bons resultados para a designação dos pontos aos respectivos depósitos.

Para o roteamento dos subgrupos, o Algoritmo do Carteiro Chinês mostrou-se ser bem eficiente comparativamente com o modelo do Problema do Carteiro Chinês.

Comparando-se a solução do quadro 7.5 e a solução atual do problema apresentado no quadro 3.4, percebe-se no total uma melhoria de 6,15% (Algoritmo do Carteiro Chinês) e 8,13% (Formulação do Problema do Carteiro Chinês). Isso representa uma redução média de 7.200 metros no circuito total deste grupo e, considerando-se que existem outros 6 grupos semelhantes nesse município, torna-se viável sua implementação. Outro aspecto importante a ser considerado é a automatização na obtenção das rotas, de maneira rápida, eficaz e sem erros.

8.3 Sugestões para Trabalhos Futuros

Com o objetivo de melhorar ainda mais os resultados apresentados neste trabalho, deixa-se as seguintes sugestões para trabalhos futuros:

- a) Efetuar, primeiramente, a divisão do município em 7 grupos, para depois subdividir os em grupos de atendimento para cada leiturista.
- b) Sugerir um número menor de leituras para cobrir o grupo. Fazer testes para verificar até que área que um leiturista poderia cobrir durante o tempo hábil existente.
- c) Atribuir pesos aos pontos médios dos trechos, definidos na etapa 1 da primeira fase, de acordo com o número de clientes que devem ser atendidos nesses trechos.
- d) Pesquisar a construção de outros algoritmos utilizando, além do algoritmo genético, as técnicas *ant system*, *simulated annealing* e outras.
- e) Aplicar variações do Problema do Carteiro Chinês com outras técnicas de cobertura de arcos.

REFERÊNCIAS BIBLIOGRÁFICAS

BARBOSA, Hélio José Correia. **Introdução aos Algoritmos Genéticos**. Mini curso, XX Congresso Nacional de Matemática Aplicada e Computacional – CNMAC, Gramado, RS. 1997.

BARBOSA, Valmir C. **Redes neurais e simulated annealing como ferramentas para otimização combinatória**. Investigación Operativa, v. 1,n. 2, p. 125-142. 1989.

BELTRAMI, E. J.; BODIN, L.D. **Networks and vehicle routing for municipal waste collection**. Networks, v.4, p.65-94. 1974.

BEZERRA, O.B. **Localização de postos de coleta para apoio ao escoamento de produtos extrativistas – Um estudo de caso aplicado ao babaçu**. Dissertação de Mestrado, UFSC, 1995.

BOAVENTURA, P.O.N. **Teoria e Modelos de Grafos**. Editora Edgard Blucher Ltda. 1979.

BODIN, L. D.; KURSH, Samuel J. **A computer-assisted system for the routing and scheduling of street sweepers**. Operations Research, v. 26, n.4, p. 525-537. 1978.

BODIN, L.; GOLDEN, B. **Classification in Vehicle Routing and Scheduling**. Networks, 11:97-108. 1981

BODIN, L.D., ASSAD, B.L. e BALL, A. **Routing and Scheduling of Vehicles and Crew, The State of the Art**. Computers & Ops. Res. 10, 69-211. 1983

BURKE, Laura I.; DAMANY, Poulo. **The guilty net for the traveling salesman problem**. Journal of Computers and Operations Research, v. 19,n.3/4, p. 255-265. 1992.

CARVALHO, Andre P. L. F. **Algoritmos Genéticos** [on line]
<www.icmsc.usp.br/~andre/gene1.htm#hist>

CHRISTOFIDES, N. **Graph theory – An algorithmic approach**. New York: Academic Press Inc, London, 1975.

CHRISTOFIDES, N. e BROOKER, P. **The Optimal Partitioning of Graphs**. SAIM J. Appl. Math. 30(1), 55-69. 1976.

CHRISTOFIDES, N.; MINGOZZI, A. e TOTH, P. **Exact Algorithms for the Vehicle Routing Problem basead on Spanning Tree and Shortest Path Relaxations**. Math. Prog. 20, 255-282. 1981.

- CLARKE, G.; WRIGHT, J.W. **Scheduling of vehicles from a central depot to a number of delivery points.** Operations Research, v.12, n.4, p.568-581. 1964.
- CORRÊA, E.S. **Algoritmos Genéticos e Busca Tabu aplicados ao problema das P-medianas.** Dissertação de Mestrado, UFPR, 2000.
- COSTA, D.M.B. **Aplicação de Algumas Técnicas da Pesquisa Operacional na Otimização dos Serviços Postais.** Dissertação de Mestrado, UFPR, 1997.
- DANTZIG, G. B. e RAMSER, J. **The Truck Dispatching Problem.** Man. Sci. 6, p. 81-94. 1959.
- DASKIN, Mark S. **Logistics: an overview of the state of the art and perspectives on future research.** Transportation Research, v. 19A, n.5/6, p. 383-398. 1985.
- DE JONG, K.A. **Analysis of Behavior of a Class of Genetic Adaptive Systems.** Ann Arbor, USA, Ph.D Thesis – Department of Computer and Communication Sciences, University of Michigan. 1975.
- DIAS, J. S.; BARRETO, J. M. **Algoritmo Genético: Inspiração Biológica na Solução de Problemas – Uma Introdução.** Pesquisa Naval, no. 11, 1998.
- DILLMANN, R., BECKER, B. e BECKEFELD, V. **Practical Aspects of Route Planning for Magazin and Newspaper Wholesalers.** EJOR 90, p. 1-12. 1996.
- DREZNER, Zvi. **Facility location – A survey of applications and methods.** USA: Ed. Springer: Stanford, 1995.
- EDMONDS, J.; JOHNSON, E. L. **Matching, Euler tour and the chinese postman.** Mathematical Programming, v. 5, p. 88-124.1973.
- EGLESE, R. W.; MURDOCK, H. **Routing Road Sweepers in a Rural Area.** JORS 42 (4), p. 281-288. 1991.
- EULER, L. **Solution problematics ad geometrian situs pertinentis.** Comentarii Academae Scientiarum, Petropolinae 8, p.128-140. 1976.
- FISHER, M.; JAIKUMAR, R. **A Generalized Assignment Heuristics for Vehicle Routing.** Networks, 11. p. 109-124. 1981.
- FLOYD, Robert W. **Algorithm 97: shortest path.** Communications of the Association for Computing Machinery, v. 5, p.345. 1962
- GAREY, M.; JOHNSON, D. **Computer and intractability: A guide to the Theory of NP Completeness.** Freeman, San Francisco. 1979.

GHIANI, G.; IMPROTA, G. **An algorithm for the hierarchical Chinese postman problem.** JORS 26 , p. 27-32. 2000.

GOLDBARG, M.C. **Otimização Combinatória e programação linear: modelos e algoritmos.** Rio de Janeiro: Campus, 2000.

GOLDBERG, David E. **Genetic Algorithms in Search, Optimization, and Machine Learning.** Addison-Wesley, Menlo Park, CA, 1986.

GOLDEN, B.L.; MAGNANTI, T.L.; NGUYEN, H. Q. **Implementing Vehicle Routing Algorithms.** Networks 7, p. 694-711. 1977.

GOLDEN, B.L.; WONG, R.T. **Capacited Arc Routing Problems.** Networks 11. p. 305-312. 1981.

GRACIOLLI, O.D. **Planejamento de Roteiros de Veículos Coletores de Resíduos Sólidos de Serviços de Saúde.** Dissertação de Mestrado, UFSC, 1994.

GREFENSTETTE, J.J. **Optimization of control parameters for genetic algorithms.** IEEE Transactions on Systems, Man and Cybernetics, v. 16, n.1, p. 122-128, 1986.

HAKIMI, S.L. **Optimum Distribuition of Swtching Centers in a Communication Network and Some Reladed Graph.** Theore Problems, Operations Research, 13, p. 462. 1985.

HAMACHER, S.; TANURE, C. Z. **Aplicação de Algoritmo Genético para um Problema de Distribuição dos Correios.** Anais do XII ANPET (Congresso de Pesquisa e Ensino em Transportes). Fortaleza. 1998.

HARVEY, I., HUSBANDS, P., CLIFF, D. **Issues in a evolutionary robotics.** Technical Report Cognitive Science Research Paper SRP219, School of Cognitive and Computing Sciences, University of Sussex, Brighton BN1 9QH, England, UK, 1992.

HEIERHOLZER, C. **Über die möglichkeit, einen linienzung ohne wiederholing und ohne unterbrechung zu umfahren.** Mathematische Annalen IV, p. 30-32. 1873.

HOFFMANN, H. **Genetic Algorithm.** [on line]
 <<http://www.cs.berkeley.edu/~hoffman/#gaclass>>
 <<http://HTTP.CS.Berkeley.EDU/~hoffman/ga.htm>> hoffman@cs.berkeley.edu, May 1997.

KARP, R.M. **On the Computacional Complexity of Combinatorial Problems.** Networks 5, p. 45-68. 1975.

- KLOTS, B.; GAL, S.; HARPAZ, A. **Multi-Depot and Multi-Product Delivery Optimization Problem with Time and Services Constraints.** IBM Israel Report, p. 88-315, Science and Tecnology, Haifa, Israel. 1992.
- LAPORTE, G.; ASEF-VAZIR, A.; SRISKANDARAJAH, C. **Some Applications of the Generalized Travelling Salesman Problem.** JORS 47, P. 1461-1467. 1996.
- LARSON, R.C. **Urban Police Patrol Analysis.** The MIT Press, Cambridge. 1972.
- LOBO, Débora da Silva. **Localização de unidades de educação infantil: uma aplicação para creches municipais em Florianópolis.** UFSC, Florianópolis, SC, 1998.
- LOPES, H.S. **Algoritmos Genéticos.** Trabalho de circulação interna do CEFET – Centro Federal de Educação Tecnológica do Paraná, 1996.
- MARANZANA, F. E. **On the of Supplu Points to Minimize Transport Costs.** Opl. Res. Quart., 15, p. 261. 1964.
- MAYERLE, S.F. **Um Algoritmo Genético para Solução do Problema do Caixeiro Viajante.** Trabalho Interno. Florianópolis: UFSC, 1994.
- MEI-KO, K. **Graphic Programming Using Odd and Even Points.** Chinese Mathematics, vol 1. p. 273-277. 1962.
- MINIEKA, Edward. **The Centers and Medians of a Graph.** Ops. Res. Vol 25, nº 4, p. 641-650, July-August, 1977.
- MORAN, N. **Stock exchange gets fuzzy logic.** Financial Times, September 4th. 1996.
- NEGREIRO GOMES, F.J.; NEGREIRO GOMES, M.J. **Xnês: Um Ambiente Visual para Geração de Soluções Ótimas de Instâncias do Problema do Carteiro Chinês.** Universidade Federal do Espírito Santo e Universidade Estadual do Ceará. 1999.
- NUNES, L. F. **Algoritmos Genéticos Aplicados na Abordagem de um Problema Real de Roteirização de Veículos.** Dissertação de Mestrado, UFPR, 1998.
- OCHI, L. S. **Conhecimento Heurístico: Aplicações em Problemas de Otimização.** XIV Congresso da Sociedade Brasileira de Computação, XIII Jornada de Atualização em Informática, Caxambú, MG. 1994
- PAPADIMITRIOU, C.H.; STEIGLITZ, K. **Combinatorial Optimization Algorithms and Complexity.** Prentice-Hall, Nova York. 1992.
- PAVAN, C.; CUNHA, A. B. **Elementos de genética.** 2. e. São Paulo, ed. Nacional: 1966.

PEARL, J.; DASKIN, M.S. **A Warehouse Location-Routing Problem.** *Transportation Research*, 19B, p. 381-396. 1985.

RABUSKE, M. A. **Introdução à teoria dos grafos.** Florianópolis Ed. UFSC, 184p. 1992.

REINELT, G. **The Traveling Salesman: Computational Solutions for the Applications.** Lectures Notes in Computer Science, 840, Spring-Verlag, Berlim. 1994.

ROSA, A. A. **Dimensionamento e Localização de Centro de Distribuição de Correios Numa Cidade de Médio Porte.** Dissertação de Mestrado, UFSC, 1996.

SAMPAIO, M. E. C. S. **Aplicação de Meta-heurísticas ao Problema de Localização de Escolas de Ensino Fundamental.** Dissertação de Mestrado, UFPR, 1999.

SPEARS, W.M. De JONG, K. **On the Virtues of Uniform Crossover.** Proceedings of the Fourth International Conference On Genetic Algorithms, eds. R. Belew and L. Booker, San Mateo, CA: Morgan Kaufmann, p. 230-236, July, 1991.

STERN, H.I.; DROR, M. **Routing Eletric Meter Readers.** *Computers & Opns. Res*, 6, p. 209-223. 1978.

TEITZ, M.B. e BART, P. **Heuristics methods for estimating the generalized vertex median of a weighted graph.** *Operations Research*, 16, p. 955-961m 1968.

VOUDOURIS, C. **Partial constraint satisfaction problems and guided local search.** Proceedings of 2nd International Conference on Practical Application of Constraint Technology, London, 1996.

ZAMBONI, L. V. S, **Técnicas de Roteirização de Veículos Aplicadas ao Transporte Escolar.** Dissertação de Mestrado, UFPR, 1997.

ANEXO 1

FORMULAÇÃO DO PROBLEMA DO CARTEIRO CHINÊS PARA A ROTA 272

Apresenta-se a formulação para a rota número 272, de acordo com o grafo (Figura A.2) obtido a partir do mapa do *cluster* 272 (Figura A.1).

MIN(Rota 272) =

$$\begin{aligned} & 59.4087 * x_{165167} + 53.3799 * x_{165239} + 59.4087 * x_{167165} + \\ & 92.5317 * x_{167168} + 92.5317 * x_{168167} + \\ & 51.7863 * x_{168170} + 51.7863 * x_{170168} + 29.3234 * x_{174175} + \\ & 29.3234 * x_{175174} + 64.3671 * x_{175178} + 55.1628 * x_{175179} + \\ & 56.4053 * x_{176178} + 49.3563 * x_{176181} + 64.3671 * x_{178175} + \\ & 56.4053 * x_{178176} + 55.1628 * x_{179175} + 55.8345 * x_{179182} + \\ & 55.2511 * x_{179183} + 49.9795 * x_{180181} + 56.7377 * x_{180182} + \\ & 54.3037 * x_{180281} + 51.7179 * x_{180295} + 49.3563 * x_{181176} + \\ & 49.9795 * x_{181180} + 55.8345 * x_{182179} + 56.7377 * x_{182180} + \\ & 55.2511 * x_{183179} + 51.8161 * x_{183272} + 65.7341 * x_{190287} + \\ & 44.6710 * x_{190295} + 53.3799 * x_{239165} + 48.9008 * x_{239240} + \\ & 48.9008 * x_{240239} + 64.7169 * x_{240242} + 71.0053 * x_{240246} + \\ & 45.3849 * x_{241242} + 49.3309 * x_{241243} + 73.6572 * x_{241247} + \\ & 64.7169 * x_{242240} + 45.3849 * x_{242241} + 49.3309 * x_{243241} + \\ & 53.8263 * x_{243245} + 52.4802 * x_{244245} + 53.5837 * x_{244260} + \\ & 53.8263 * x_{245243} + 52.4802 * x_{245244} + 79.6651 * x_{245248} + \\ & 71.0053 * x_{246240} + 82.4609 * x_{246255} + 73.6572 * x_{247241} + \\ & 80.1123 * x_{247256} + 79.6651 * x_{248245} + 77.7053 * x_{248257} + \\ & 32.3870 * x_{249167} + 68.5141 * x_{249254} + 45.3848 * x_{250254} + \\ & 56.6444 * x_{250255} + 52.2905 * x_{251255} + 54.4369 * x_{251256} + \\ & 53.7339 * x_{252256} + 49.0428 * x_{252257} + 59.6467 * x_{253257} + \\ & 48.5263 * x_{253258} + 68.5141 * x_{254249} + 45.3848 * x_{254250} + \\ & 50.5318 * x_{254271} + 56.6444 * x_{255250} + 52.2905 * x_{255251} + \\ & 48.3653 * x_{255270} + 80.1123 * x_{256247} + 54.4369 * x_{256251} + \\ & 53.7339 * x_{256252} + 48.3653 * x_{256269} + 77.7053 * x_{257248} + \\ & 49.0428 * x_{257252} + 59.6467 * x_{257253} + 45.9215 * x_{257268} + \\ & 48.5263 * x_{258253} + 69.2646 * x_{258259} + 47.9398 * x_{258267} + \\ & 69.2646 * x_{259258} + 86.2066 * x_{259260} + 53.5837 * x_{260244} + \\ & 86.2066 * x_{260259} + 47.9398 * x_{267258} + 43.3259 * x_{267278} + \\ & 45.9215 * x_{268257} + 43.9424 * x_{268276} + 48.3653 * x_{269256} + \\ & 43.9138 * x_{269274} + 48.3653 * x_{270255} + 46.7214 * x_{270272} + \\ & 50.5318 * x_{271254} + 51.8161 * x_{272183} + 46.7214 * x_{272270} + \\ & 54.1419 * x_{272273} + 58.8570 * x_{272279} + 54.1419 * x_{273272} + \\ & 53.0289 * x_{273274} + 43.9138 * x_{274269} + 53.0289 * x_{274273} + \\ & 52.2313 * x_{274275} + 65.6069 * x_{274280} + 52.2313 * x_{275274} + \\ & 53.7339 * x_{275276} + 43.9429 * x_{276268} + 53.7339 * x_{276275} + \\ & 58.8730 * x_{276277} + 61.5935 * x_{276285} + 58.8730 * x_{277276} + \\ & 45.6305 * x_{277278} + 43.3259 * x_{278267} + 45.6305 * x_{278277} + \\ & 58.8570 * x_{279272} + 57.1410 * x_{279282} + 65.6069 * x_{280274} + \\ & 52.4026 * x_{280284} + 54.3037 * x_{281180} + 55.6152 * x_{281282} + \\ & 57.1410 * x_{282279} + 55.6152 * x_{282281} + 53.9554 * x_{282283} + \\ & 48.0445 * x_{282296} + 53.9554 * x_{283282} + 51.9308 * x_{283284} + \\ & 52.4026 * x_{284280} + 51.9308 * x_{284283} + 51.1475 * x_{284286} + \\ & 48.7600 * x_{284298} + 61.5935 * x_{285276} + 56.6379 * x_{285309} + \\ & 51.1475 * x_{286284} + 53.00928 * x_{286309} + 65.7341 * x_{287190} + \\ & 58.3850 * x_{287293} + 58.3850 * x_{293287} + 47.4014 * x_{293294} + \\ & 59.3096 * x_{293296} + 58.5636 * x_{293297} + 47.4014 * x_{294293} + \\ & 51.7179 * x_{295180} + 44.6710 * x_{295190} + 48.0445 * x_{296282} + \\ & 59.3096 * x_{296293} + 58.5636 * x_{297293} + 48.0329 * x_{297299} + \end{aligned}$$

$$\begin{aligned}
& 48.7600 * x298284 + 57.7697 * x298299 + 48.0329 * x299297 + \\
& 57.7697 * x299298 + 56.1000 * x299300 + 50.1243 * x299306 + \\
& 56.1000 * x300299 + 50.5934 * x305307 + 50.1243 * x306299 + \\
& 53.1148 * x306307 + 50.5934 * x307305 + 53.1148 * x307306 + \\
& 56.4172 * x307308 + 55.8791 * x307310 + 56.4172 * x308307 + \\
& 51.6333 * x308309 + 56.6379 * x309285 + 53.0928 * x309286 + \\
& 51.6333 * x309308 + 55.7166 * x309313 + 55.8791 * x310307 + \\
& 55.7166 * x313309;
\end{aligned}$$
ST

$$\begin{aligned}
& + x167165 + x239165 - x165167 - x165239 = 0; \\
& + x165167 + x168167 - x167165 - x167168 = 0; \\
& + x167168 + x170168 - x168167 - x168170 = 0; \\
& + x168170 - x170168 = 0; \\
& + x175174 - x174175 = 0; \\
& + x174175 + x178175 + x179175 - x175174 - x175178 - x175179 = 0; \\
& + x178176 + x181176 - x176178 - x176181 = 0; \\
& + x175178 + x176178 - x178175 - x178176 = 0; \\
& + x175179 + x182179 + x183179 - x179175 - x179182 - x179183 = 0; \\
& + x181180 + x182180 + x281180 - x180181 - x180182 - x180281 = 0; \\
& + x176181 + x180181 - x181176 - x181180 = 0; \\
& + x179182 + x180182 - x182179 - x182180 = 0; \\
& + x179183 + x272183 - x183179 - x183272 = 0; \\
& + x287190 + x295190 - x190287 - x190295 = 0; \\
& + x165239 + x240239 - x239165 - x239240 = 0; \\
& + x239240 + x242240 + x246240 - x240239 - x240242 - x240246 = 0; \\
& + x242241 + x243241 + x247241 - x241242 - x241243 - x241247 = 0; \\
& + x240242 + x241242 - x242240 - x242241 = 0; \\
& + x241243 + x245243 - x243241 - x243245 = 0; \\
& + x245244 + x260244 - x244245 - x244260 = 0; \\
& + x243245 + x244245 + x248245 - x245243 - x245244 - x245248 = 0; \\
& + x255246 + x240246 - x246240 - x246255 = 0; \\
& + x241247 + x256247 - x247241 - x247256 = 0; \\
& + x257248 + x245248 - x248245 - x248257 = 0; \\
& + x254249 - x249254 = 0; \\
& + x255250 + x254250 - x250254 - x250255 = 0; \\
& + x255251 + x256251 - x251255 - x251256 = 0; \\
& + x257252 + x256252 - x252256 - x252257 = 0; \\
& + x257253 + x258253 - x253257 - x253258 = 0; \\
& + x249254 + x250254 + x271254 - x254249 - x254250 - x254271 = 0; \\
& + x250255 + x251255 + x270255 - x255250 - x255251 - x255270 = 0; \\
& + x252256 + x251256 + x247256 + x269256 - x256247 - x256251 - x256252 - \\
& x256269 = 0; \\
& + x248257 + x252257 + x253257 + x268257 - x257248 - x257252 - x257253 - \\
& x257268 = 0; \\
& + x253258 + x259258 + x267258 - x258253 - x258259 - x258267 = 0; \\
& + x258259 + x260259 - x259260 - x259258 = 0; \\
& + x244260 + x259260 - x260244 - x260259 = 0; \\
& + x258267 + x278267 - x267258 - x267278 = 0; \\
& + x257268 + x276268 - x268257 - x268276 = 0; \\
& + x256269 + x274269 - x269256 - x269274 = 0; \\
& + x255270 + x272270 - x270255 - x270272 = 0; \\
& + x254271 - x271254 = 0; \\
& + x183272 + x270272 + x273272 + x279272 - x272183 - x272270 - x272273 - \\
& x272279 = 0; \\
& + x272273 + x274273 - x273272 - x273274 = 0; \\
& + x269274 + x273274 + x275274 + x280274 - x274269 - x274273 - x274275 - \\
& x274280 = 0; \\
& + x274275 + x276275 - x275274 - x275276 = 0;
\end{aligned}$$

```

+ x268276 + x275276 + x277276 + x285276 -x276268 - x276275 - x276277 -
x276285 = 0;
+ x276277 + x278277 -x277276 - x277278 = 0;
+ x277278 + x267278 -x278267 - x278277 = 0;
+ x272279 + x282279 -x279272 - x279282 = 0;
+ x274280 + x284280 -x280274 - x280284 = 0;
+ x282281 + x180281 -x281180 - x281282 = 0;
+ x279282 + x281282 + x283282 + x296282 -x282279 - x282281 - x282283 -
x282296 = 0;
+ x282283 + x284283 -x283282 - x283284 = 0;
+ x298284 + x286284 + x283284 + x280284 -x284280 - x284283 - x284286 -
x284298 = 0;
+ x276285 + x309285 -x285276 - x285309= 0;
+ x284286 + x309286 -x286284 - x286309 = 0;
+ x190287 + x293287 -x287190 - x287293 = 0;
+ x297293 + x296293 + x294293 + x287293 -x293287 - x293294 - x293296 -
x293297 = 0;
+ x293294 -x294293 = 0;
+ x180295 + x190295 -x295180 - x295190 = 0;
+ x282296 + x293296 -x296282 - x296293 = 0;
+ x293297 + x299297 -x297293 - x297299 = 0;
+ x299298 + x284298 -x298284 - x298299 = 0;
+ x306299 + x300299 + x298299 + x297299 -x299297 - x299298 - x299300 -
x299306 = 0;
+ x299300 -x300299= 0;
+ x307305 -x305307 = 0;
+ x299306 + x307306 -x306299 - x306307 = 0;
+ x310307 + x308307 + x306307 + x305307 -x307305 - x307306 - x307308 -
x307310 = 0;
+ x307308 + x309308 -x308307 - x308309 = 0;
+ x313309 + x308309 + x286309 + x285309 -x309285 - x309286 - x309308 -
x309313 = 0;
+ x307310 -x310307 = 0;
+ x309313 -x313309 = 0;

```

```

x165167 + x167165 >= 2;
x165239 + x239165 >= 2;
x167165 + x165167 >= 2;
x167168 + x168167 >= 2;
x168167 + x167168 >= 2;
x168170 + x170168 >= 2;
x170168 + x168170 >= 2;
x174175 + x175174 >= 2;
x175174 + x174175 >= 2;
x175178 + x178175 >= 2;
x175179 + x179175 >= 2;
x176178 + x178176 >= 2;
x176181 + x181176 >= 2;
x178175 + x175178 >= 2;
x178176 + x176178 >= 2;
x179175 + x175179 >= 2;
x179182 + x182179 >= 2;
x179183 + x183179 >= 2;
x180181 + x181180 >= 2;
x180182 + x182180 >= 2;
x180281 + x281180 >= 2;
x180295 + x295180 >= 2;
x181176 + x176181 >= 2;
x181180 + x180181 >= 2;
x182179 + x179182 >= 2;

```

```
x182180 + x180182 >= 2;
x183179 + x179183 >= 2;
x183272 + x272183 >= 2;
x190287 + x287190 >= 2;
x190295 + x295190 >= 2;
x239165 + x165239 >= 2;
x239240 + x240239 >= 2;
x240239 + x239240 >= 2;
x240242 + x242240 >= 2;
x240246 + x246240 >= 2;
x241242 + x242241 >= 2;
x241243 + x243241 >= 2;
x241247 + x247241 >= 2;
x242240 + x240242 >= 2;
x242241 + x241242 >= 2;
x243241 + x241243 >= 2;
x243245 + x245243 >= 2;
x244245 + x245244 >= 2;
x244260 + x260244 >= 2;
x245243 + x243245 >= 2;
x245244 + x244245 >= 2;
x245248 + x248245 >= 2;
x246240 + x240246 >= 2;
x246255 + x255246 >= 2;
x247241 + x241247 >= 2;
x247256 + x256247 >= 2;
x248245 + x245248 >= 2;
x248257 + x257248 >= 2;
x249254 + x254249 >= 2;
x250254 + x254250 >= 2;
x250255 + x255250 >= 2;
x251255 + x255251 >= 2;
x251256 + x256251 >= 2;
x252256 + x256252 >= 2;
x252257 + x257252 >= 2;
x253257 + x257253 >= 2;
x253258 + x258253 >= 2;
x254249 + x249254 >= 2;
x254250 + x250254 >= 2;
x254271 + x271254 >= 2;
x255250 + x250255 >= 2;
x255251 + x251255 >= 2;
x255270 + x270255 >= 2;
x256247 + x247256 >= 2;
x256251 + x251256 >= 2;
x256252 + x252256 >= 2;
x256269 + x269256 >= 2;
x257248 + x248257 >= 2;
x257252 + x252257 >= 2;
x257253 + x253257 >= 2;
x257268 + x268257 >= 2;
x258253 + x253258 >= 2;
x258259 + x259258 >= 2;
x258267 + x267258 >= 2;
x259258 + x258259 >= 2;
x259260 + x260259 >= 2;
x260244 + x244260 >= 2;
x260259 + x259260 >= 2;
x267258 + x258267 >= 2;
x267278 + x278267 >= 2;
x268257 + x257268 >= 2;
```

```
x268276 + x276268 >= 2;
x269256 + x256269 >= 2;
x269274 + x274269 >= 2;
x270255 + x255270 >= 2;
x270272 + x272270 >= 2;
x271254 + x254271 >= 2;
x272183 + x183272 >= 2;
x272270 + x270272 >= 2;
x272273 + x273272 >= 2;
x272279 + x279272 >= 2;
x273272 + x272273 >= 2;
x273274 + x274273 >= 2;
x274269 + x269274 >= 2;
x274273 + x273274 >= 2;
x274275 + x275274 >= 2;
x274280 + x280274 >= 2;
x275274 + x274275 >= 2;
x275276 + x276275 >= 2;
x276268 + x268276 >= 2;
x276275 + x275276 >= 2;
x276277 + x277276 >= 2;
x276285 + x285276 >= 2;
x277276 + x276277 >= 2;
x277278 + x278277 >= 2;
x278267 + x267278 >= 2;
x278277 + x277278 >= 2;
x279272 + x272279 >= 2;
x279282 + x282279 >= 2;
x280274 + x274280 >= 2;
x280284 + x284280 >= 2;
x281180 + x180281 >= 2;
x281282 + x282281 >= 2;
x282279 + x279282 >= 2;
x282281 + x281282 >= 2;
x282283 + x283282 >= 2;
x282296 + x296282 >= 2;
x283282 + x282283 >= 2;
x283284 + x284283 >= 2;
x284280 + x280284 >= 2;
x284283 + x283284 >= 2;
x284286 + x286284 >= 2;
x284298 + x298284 >= 2;
x285276 + x276285 >= 2;
x285309 + x309285 >= 2;
x286284 + x284286 >= 2;
x286309 + x309286 >= 2;
x287190 + x190287 >= 2;
x287293 + x293287 >= 2;
x293287 + x287293 >= 2;
x293294 + x294293 >= 2;
x293296 + x296293 >= 2;
x293297 + x297293 >= 2;
x294293 + x293294 >= 2;
x295180 + x180295 >= 2;
x295190 + x190295 >= 2;
x296282 + x282296 >= 2;
x296293 + x293296 >= 2;
x297293 + x293297 >= 2;
x297299 + x299297 >= 2;
x298284 + x284298 >= 2;
x298299 + x299298 >= 2;
```

```
x299297 + x297299 >= 2;  
x299298 + x298299 >= 2;  
x299300 + x300299 >= 2;  
x299306 + x306299 >= 2;  
x300299 + x299300 >= 2;  
x305307 + x307305 >= 2;  
x306299 + x299306 >= 2;  
x306307 + x307306 >= 2;  
x307305 + x305307 >= 2;  
x307306 + x306307 >= 2;  
x307308 + x308307 >= 2;  
x307310 + x310307 >= 2;  
x308307 + x307308 >= 2;  
x308309 + x309308 >= 2;  
x309285 + x285309 >= 2;  
x309286 + x286309 >= 2;  
x309308 + x308309 >= 2;  
x309313 + x313309 >= 2;  
x310307 + x307310 >= 2;  
x313309 + x309313 >= 2;
```

$x_{ij} \geq 0$ e inteiros

Figura A.1 – Mapa do *Cluster 272*

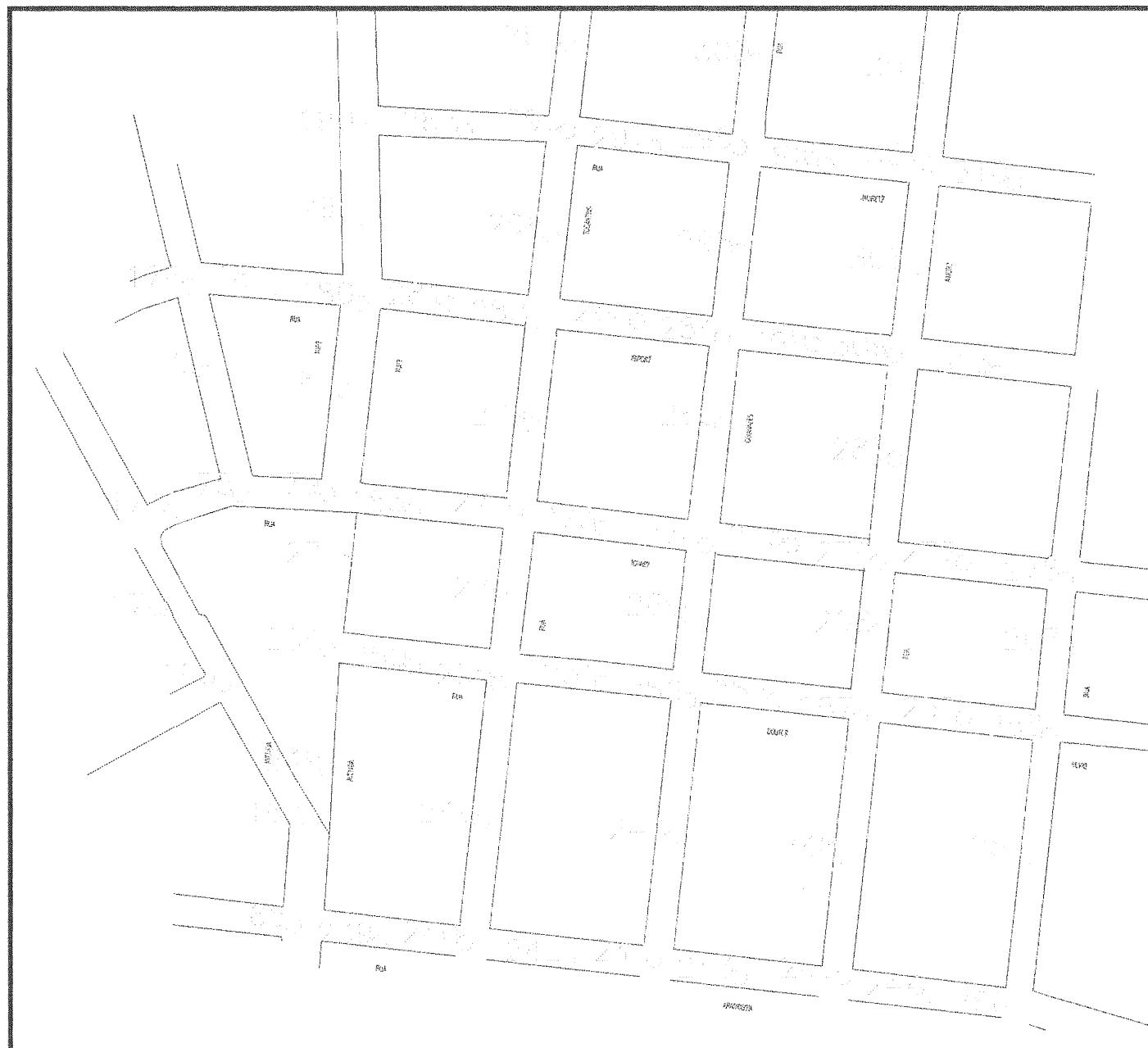
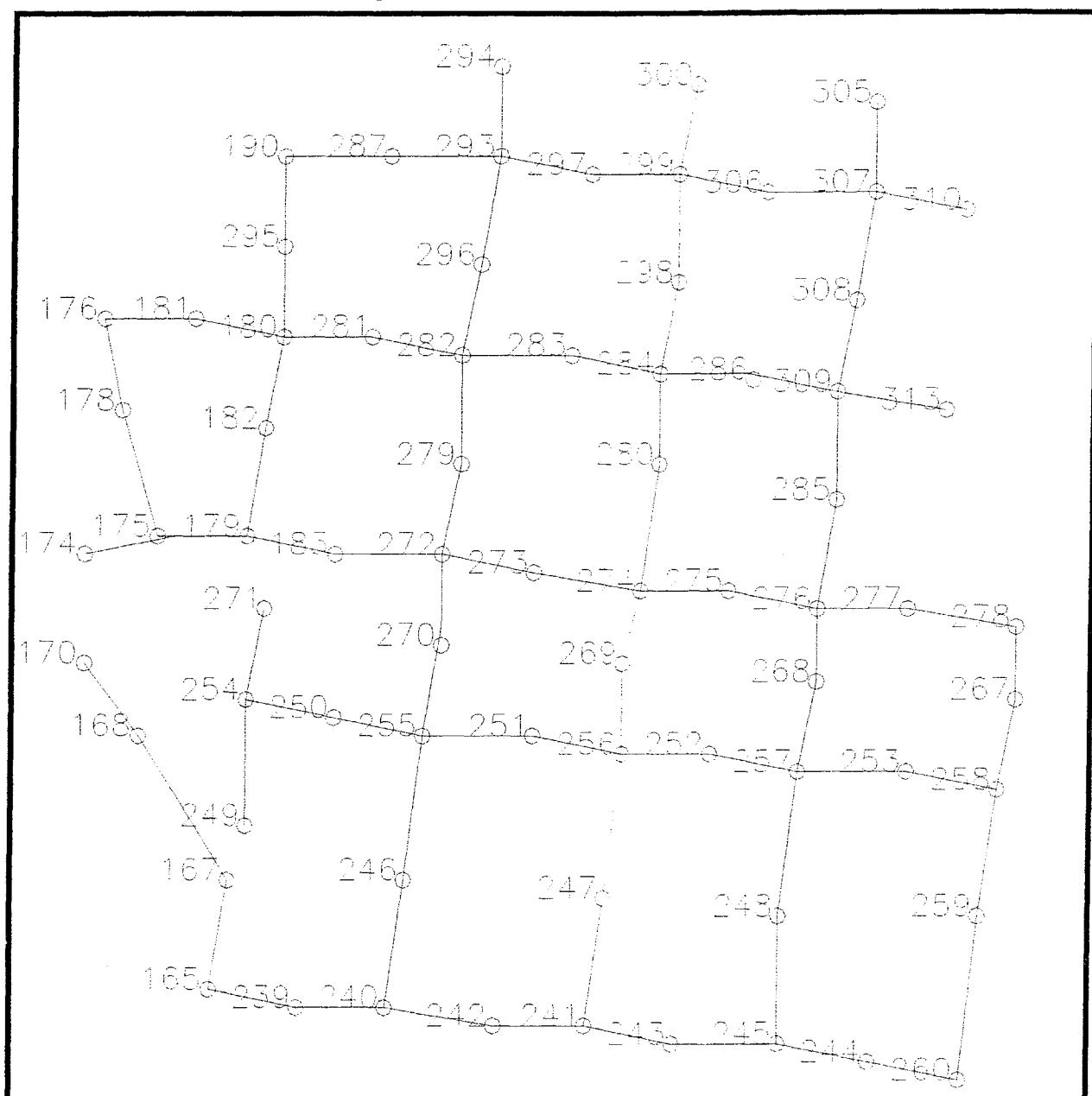
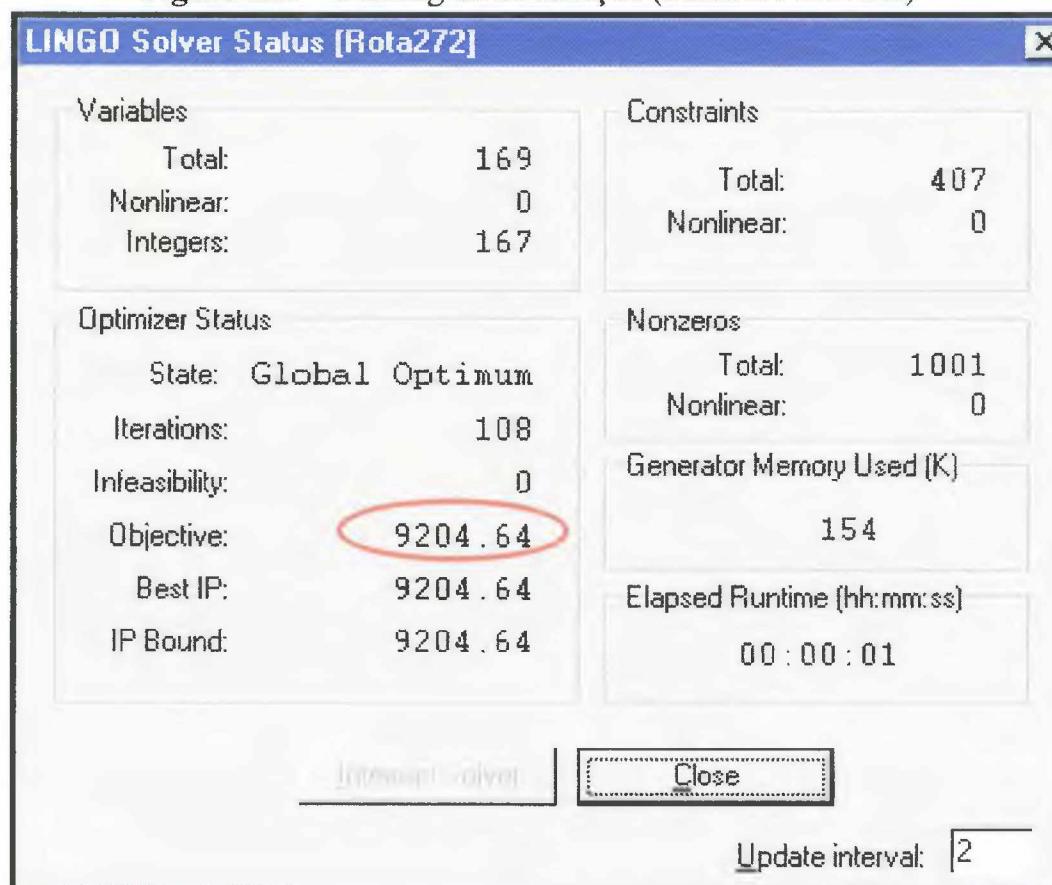


Figura A.2 – Grafo do Cluster 272

São fornecidos os resultados obtidos pelo Software LINGO, para esta rota.

Figura A.3 – Mensagem de solução (Software LINGO)



Global optimal solution found at step: 108

Objective value: 9204.641

Branch count: 0

Variable	Value	Reduced Cost
X165167	1.000000	0.0000000
X165239	1.000000	0.0000000
X167165	1.000000	0.0000000
X167168	1.000000	0.0000000
X168167	1.000000	0.0000000
X168170	1.000000	0.0000000
X170168	1.000000	0.0000000
X174175	1.000000	0.0000000
X175174	1.000000	0.0000000
X175178	2.000000	0.0000000
X175179	0.0000000	0.1647949E-06
X176178	0.0000000	0.0000000
X176181	2.000000	0.0000000
X178175	0.0000000	0.0000000
X178176	2.000000	0.0000000
X179175	2.000000	0.0000000
X179182	1.000000	0.0000000
X179183	0.0000000	0.0000000
X180181	0.0000000	0.0000000

X180182	1.000000	0.0000000
X180281	2.000000	0.0000000
X180295	0.0000000	0.0000000
X181176	0.0000000	0.0000000
X181180	2.000000	0.0000000
X182179	1.000000	0.0000000
X182180	1.000000	0.0000000
X183179	2.000000	0.0000000
X183272	0.0000000	0.0000000
X190287	0.0000000	0.0000000
X190295	2.000000	0.0000000
X239165	1.000000	0.0000000
X239240	1.000000	0.0000000
X240239	1.000000	0.0000000
X240242	2.000000	0.0000000
X240246	0.0000000	0.0000000
X241242	0.0000000	0.0000000
X241243	2.000000	0.0000000
X241247	1.000000	0.0000000
X242240	0.0000000	82.46090
X242241	2.000000	0.0000000
X243241	0.0000000	0.1715088E-05
X243245	2.000000	0.0000000
X244245	0.0000000	-0.1721191E-05
X244260	2.000000	0.0000000
X245243	0.0000000	0.0000000
X245244	2.000000	0.0000000
X245248	1.000000	0.0000000
X246240	2.000000	0.0000000
X246255	0.0000000	0.0000000
X247241	1.000000	0.0000000
X247256	1.000000	0.0000000
X248245	1.000000	0.0000000
X248257	1.000000	0.0000000
X249167	0.0000000	32.38700
X249254	1.000000	0.0000000
X250254	1.000000	0.0000000
X250255	1.000000	0.0000000
X251255	0.0000000	0.0000000
X251256	2.000000	0.0000000
X252256	1.000000	0.0000000
X252257	1.000000	0.0000000
X253257	2.000000	0.0000000
X253258	0.0000000	0.0000000
X254249	1.000000	0.0000000
X254250	1.000000	0.0000000
X254271	1.000000	0.0000000
X255250	1.000000	0.0000000
X255251	2.000000	0.0000000
X255270	0.0000000	0.0000000
X256247	1.000000	0.0000000
X256251	0.0000000	0.0000000
X256252	1.000000	0.0000000
X256269	2.000000	0.0000000
X257248	1.000000	0.0000000
X257252	1.000000	0.0000000
X257253	0.0000000	0.0000000
X257268	2.000000	0.0000000
X258253	2.000000	0.0000000
X258259	0.0000000	0.0000000
X258267	1.000000	0.0000000

X259258	2.000000	0.0000000
X259260	0.0000000	0.0000000
X260244	0.0000000	0.0000000
X260259	2.000000	0.0000000
X267258	1.000000	0.0000000
X267278	1.000000	0.0000000
X268257	0.0000000	0.0000000
X268276	2.000000	0.0000000
X269256	0.0000000	0.7751465E-06
X269274	2.000000	0.0000000
X270255	2.000000	0.0000000
X270272	0.0000000	-0.1214600E-05
X271254	1.000000	0.0000000
X272183	2.000000	0.0000000
X272270	2.000000	0.0000000
X272273	0.0000000	0.8911133E-06
X272279	0.0000000	0.0000000
X273272	2.000000	0.0000000
X273274	0.0000000	0.0000000
X274269	0.0000000	0.0000000
X274273	2.000000	0.0000000
X274275	1.000000	0.0000000
X274280	1.000000	0.0000000
X275274	1.000000	0.0000000
X275276	1.000000	0.0000000
X276268	0.0000000	0.4981140E-03
X276275	1.000000	0.0000000
X276277	1.000000	0.0000000
X276285	2.000000	0.0000000
X277276	1.000000	0.0000000
X277278	1.000000	0.0000000
X278267	1.000000	0.0000000
X278277	1.000000	0.0000000
X279272	2.000000	0.0000000
X279282	0.0000000	0.1159668E-05
X280274	1.000000	0.0000000
X280284	1.000000	0.0000000
X281180	0.0000000	0.0000000
X281282	2.000000	0.0000000
X282279	2.000000	0.0000000
X282281	0.0000000	0.0000000
X282283	1.000000	0.0000000
X282296	1.000000	0.0000000
X283282	1.000000	0.0000000
X283284	1.000000	0.0000000
X284280	1.000000	0.0000000
X284283	1.000000	0.0000000
X284286	0.0000000	0.0000000
X284298	2.000000	0.0000000
X285276	0.0000000	0.0000000
X285309	2.000000	0.0000000
X286284	2.000000	0.0000000
X286309	0.0000000	5300875.
X287190	2.000000	0.0000000
X287293	0.0000000	0.0000000
X293287	2.000000	0.0000000
X293294	1.000000	0.0000000
X293296	1.000000	0.0000000
X293297	0.0000000	0.0000000
X294293	1.000000	0.0000000
X295180	2.000000	0.0000000

X295190	0.0000000	0.0000000
X296282	1.000000	0.0000000
X296293	1.000000	0.0000000
X297293	2.000000	0.0000000
X297299	0.0000000	0.0000000
X298284	0.0000000	0.0000000
X298299	2.000000	0.0000000
X299297	2.000000	0.0000000
X299298	0.0000000	0.9033203E-06
X299300	1.000000	0.0000000
X299306	1.000000	0.0000000
X300299	1.000000	0.0000000
X305307	1.000000	0.0000000
X306299	1.000000	0.0000000
X306307	1.000000	0.0000000
X307305	1.000000	0.0000000
X307306	1.000000	0.0000000
X307308	1.000000	0.0000000
X307310	1.000000	0.0000000
X308307	1.000000	0.0000000
X308309	1.000000	0.0000000
X309285	0.0000000	-0.1306152E-05
X309286	2.000000	0.0000000
X309308	1.000000	0.0000000
X309313	1.000000	0.0000000
X310307	1.000000	0.0000000
X313309	1.000000	0.0000000
X255246	2.000000	0.0000000

APÊNDICE I

EXEMPLOS DE APLICAÇÃO DOS ALGORITMOS UTILIZADOS NO TRABALHO

Para um melhor entendimento do funcionamento dos algoritmos utilizados neste trabalho, os programas computacionais foram aplicados a um exemplo composto de um conjunto de 10 vértices, representando as interseções e os pontos médios das quadras a serem trabalhadas.

Sejam os 10 pontos e suas respectivas coordenadas geográficas, apresentadas no Quadro A.1

Quadro A.1 – Coordenadas Geográficas

	X	Y
1	334050	7097900
2	334110	7097870
3	334010	7097810
4	334100	7097830
5	334090	7097790
6	334290	7097880
7	334240	7097890
8	334230	7097810
9	334220	7097730
10	334070	7097720

A matriz de distâncias Euclidianas entre todos os pares de pontos, encontrada computacionalmente, é apresentada no Quadro A.2 abaixo. Trabalhou-se com três casas decimais.

Quadro A.2 – Distâncias Euclidianas

	1	2	3	4	5	6	7	8	9	10
1	0,000	67,082	98,489	86,023	117,047	240,832	190,263	201,246	240,416	181,108
2	67,082	0,000	116,619	41,231	82,462	180,278	131,529	134,164	178,045	155,242
3	98,489	116,619	0,000	92,195	82,462	288,617	243,516	220,000	224,722	108,167
4	86,023	41,231	92,195	0,000	41,231	196,469	152,315	131,529	156,205	114,018
5	117,047	82,462	82,462	41,231	0,000	219,317	180,278	141,421	143,178	72,801
6	240,832	180,278	288,617	196,469	219,317	0,000	50,990	92,195	165,529	272,029
7	190,263	131,529	243,516	152,315	180,278	50,990	0,000	80,623	161,245	240,416
8	201,246	134,164	220,000	131,529	141,421	92,195	80,623	0,000	80,623	183,576
9	240,416	178,045	224,722	156,205	143,178	165,529	161,245	80,623	0,000	150,333
10	181,108	155,242	108,167	114,018	72,801	272,029	240,416	183,576	150,333	0,000

A seguir são apresentados os exemplos numéricos de cada algoritmo utilizado neste trabalho.

- 1) Algoritmo das P-medianas de Teitz e Bart
- 2) Algoritmo Genético aplicado ao Problema das P-medianas
- 3) Algoritmo de Gillet e Johnson
- 4) Algoritmo do Carteiro Chinês
- 5) Problema do Carteiro Chinês

1) Algoritmo das P-medianas de Teitz e Bart

Encontrar 3 medianas para os 10 pontos acima.

Passo 0

$$V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}$$

$$V_p \subset V \Rightarrow V_p = \{v_1, v_2, v_3\} \Rightarrow \text{Solução inicial}$$

Passo 1

$v_i \in \{V - V_p\}$ serão os vértices não analisados:

$$V - V_p = \{v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}$$

Passo 2

Para todos os vértices não analisados, calcule:

$$\Delta_{ij} = \sigma(V_p) - \sigma(V_p \cup \{vi\} - \{vj\}), \forall v_j \in V_p$$

$$\sigma(V_p) = 855,876$$

-Análise do vértice 4

$$\sigma(v_1, v_2, v_4) = 846,985$$

$$\Delta_{43} = 8,891$$

$$\sigma(v_1, v_3, v_4) = 827,147$$

$$\Delta_{42} = 28,729$$

$$\sigma(v_2, v_3, v_4) = 816,021$$

$$\Delta_{41} = 39,855$$

-Análise do vértice 8

$$\sigma(v_1, v_2, v_8) = 630,865$$

$$\Delta_{83} = 225,011$$

$$\sigma(v_1, v_3, v_8) = 597,175$$

$$\Delta_{82} = 258,701$$

$$\sigma(v_2, v_3, v_8) = 552,383$$

$$\Delta_{81} = 303,493$$

-Análise do vértice 5

$$\sigma(v_1, v_2, v_5) = 785,643$$

$$\Delta_{53} = 70,233$$

$$\sigma(v_1, v_3, v_5) = 865,308$$

$$\Delta_{52} = -$$

$$\sigma(v_2, v_3, v_5) = 770,263$$

$$\Delta_{51} = 85,613$$

-Análise do vértice 9

$$\sigma(v_1, v_2, v_9) = 750,196$$

$$\Delta_{93} = 105,680$$

$$\sigma(v_1, v_3, v_9) = 751,131$$

$$\Delta_{92} = 104,745$$

$$\sigma(v_2, v_3, v_9) = 676,623$$

$$\Delta_{91} = 179,253$$

-Análise do vértice 6

$$\sigma(v_1, v_2, v_6) = 686,138$$

$$\Delta_{63} = 169,738$$

$$\sigma(v_1, v_3, v_6) = 652,448$$

$$\Delta_{62} = 203,428$$

$$\sigma(v_2, v_3, v_6) = 607,656$$

$$\Delta_{61} = 248,220$$

-Análise do vértice 10

$$\sigma(v_1, v_2, v_{10}) = 808,825$$

$$\Delta_{103} = 47,051$$

$$\sigma(v_1, v_3, v_{10}) = 990,910$$

$$\Delta_{102} = 135,034$$

$$\sigma(v_2, v_3, v_{10}) = 777,418$$

$$\Delta_{101} = 78,458$$

-Análise do vértice 7

$$\sigma(v_1, v_2, v_7) = 670,282$$

$$\Delta_{73} = 185,594$$

$$\sigma(v_1, v_3, v_7) = 636,592$$

$$\Delta_{72} = 219,284$$

$$\sigma(v_2, v_3, v_7) = 591,796$$

$$\Delta_{71} = 264,080$$

O máximo $\Delta_{ij} = \Delta_{81} = 303,493$ – Assim, retirar v_1 e inserir v_8 .

$V_p = (V_p \cup \{v_8\} - \{v_1\})$ inserir v_1 em $\{V - V_p\}$

$$V_p = \{v_2, v_3, v_8\}$$

$$V - V_p = \{v_1, v_4, v_5, v_6, v_7, v_9, v_{10}\}$$

Passo 2

$$\sigma(V_p) = 552,383$$

-Análise do vértice 1

$$\begin{aligned}\sigma(v_2, v_3, v_1) &= 855,876 & \Delta_{18} &= - \\ \sigma(v_2, v_8, v_1) &= 630,865 & \Delta_{13} &= - \\ \sigma(v_3, v_8, v_1) &= 597,175 & \Delta_{12} &= 22,290\end{aligned}$$

-Análise do vértice 4

$$\begin{aligned}\sigma(v_2, v_3, v_4) &= 816,021 & \Delta_{48} &= - \\ \sigma(v_2, v_8, v_4) &= 567,967 & \Delta_{43} &= - \\ \sigma(v_3, v_8, v_4) &= 530,093 & \Delta_{42} &= 22,290\end{aligned}$$

-Análise do vértice 5

$$\begin{aligned}\sigma(v_2, v_3, v_5) &= 770,263 & \Delta_{58} &= - \\ \sigma(v_2, v_8, v_5) &= 517,016 & \Delta_{53} &= 35,367 \\ \sigma(v_3, v_8, v_5) &= 548,424 & \Delta_{52} &= 3,959\end{aligned}$$

-Análise do vértice 6

$$\begin{aligned}\sigma(v_2, v_3, v_6) &= 607,656 & \Delta_{68} &= - \\ \sigma(v_2, v_8, v_6) &= 594,249 & \Delta_{63} &= - \\ \sigma(v_3, v_8, v_6) &= 629,545 & \Delta_{62} &= -\end{aligned}$$

O máximo $\Delta_{ij} = \Delta_{53} = 35,367$ – Assim, retirar v_3 e inserir v_5 .

$V_p = (V_p \cup \{v_5\} - \{v_3\})$ inserir v_3 em $\{V - V_p\}$

$V_p = \{v_2, v_5, v_8\}$

$V - V_p = \{v_1, v_3, v_4, v_6, v_7, v_9, v_{10}\}$

Passo 2

$$\sigma(V_p) = 517,016$$

-Análise do vértice 1

$$\begin{aligned}\sigma(v_2, v_8, v_1) &= 630,865 & \Delta_{15} &= - \\ \sigma(v_2, v_5, v_1) &= 785,643 & \Delta_{18} &= - \\ \sigma(v_5, v_8, v_1) &= 517,017 & \Delta_{12} &= -\end{aligned}$$

-Análise do vértice 3

$$\begin{aligned}\sigma(v_2, v_8, v_3) &= 552,383 & \Delta_{35} &= - \\ \sigma(v_2, v_5, v_3) &= 770,263 & \Delta_{38} &= - \\ \sigma(v_5, v_8, v_3) &= 548,424 & \Delta_{32} &= -\end{aligned}$$

-Análise do vértice 4

$$\begin{aligned}\sigma(v_2, v_8, v_4) &= 567,967 & \Delta_{45} &= - \\ \sigma(v_2, v_5, v_4) &= 808,859 & \Delta_{48} &= - \\ \sigma(v_5, v_8, v_4) &= 535,958 & \Delta_{42} &= -\end{aligned}$$

-Análise do vértice 6

$$\begin{aligned}\sigma(v_2, v_8, v_6) &= 594,249 & \Delta_{65} &= - \\ \sigma(v_2, v_5, v_6) &= 549,939 & \Delta_{68} &= -\end{aligned}$$

-Análise do vértice 7

$$\begin{aligned}\sigma(v_2, v_3, v_7) &= 591,800 & \Delta_{78} &= - \\ \sigma(v_2, v_8, v_7) &= 594,249 & \Delta_{73} &= - \\ \sigma(v_3, v_8, v_7) &= 629,545 & \Delta_{72} &= -\end{aligned}$$

-Análise do vértice 9

$$\begin{aligned}\sigma(v_2, v_3, v_9) &= 676,623 & \Delta_{98} &= - \\ \sigma(v_2, v_8, v_9) &= 630,545 & \Delta_{93} &= - \\ \sigma(v_3, v_8, v_9) &= 670,750 & \Delta_{92} &= -\end{aligned}$$

-Análise do vértice 10

$$\begin{aligned}\sigma(v_2, v_3, v_{10}) &= 777,418 & \Delta_{108} &= - \\ \sigma(v_2, v_8, v_{10}) &= 542,722 & \Delta_{103} &= 9,661 \\ \sigma(v_3, v_8, v_{10}) &= 633,545 & \Delta_{102} &= -\end{aligned}$$

$$\sigma(v_5, v_8, v_6) = 527,616$$

$$\Delta_{62} = -$$

-Análise do vértice 7

$$\begin{aligned}\sigma(v_2, v_8, v_7) &= 594,249 & \Delta_{75} &= - \\ \sigma(v_2, v_5, v_7) &= 538,367 & \Delta_{78} &= - \\ \sigma(v_5, v_8, v_7) &= 527,616 & \Delta_{72} &= -\end{aligned}$$

-Análise do vértice 9

$$\begin{aligned}\sigma(v_2, v_8, v_9) &= 630,545 & \Delta_{95} &= - \\ \sigma(v_2, v_5, v_9) &= 641,257 & \Delta_{98} &= - \\ \sigma(v_5, v_8, v_9) &= 568,821 & \Delta_{92} &= -\end{aligned}$$

-Análise do vértice 10

$$\begin{aligned}\sigma(v_2, v_8, v_{10}) &= 542,722 & \Delta_{105} &= - \\ \sigma(v_2, v_5, v_{10}) &= 779,924 & \Delta_{108} &= - \\ \sigma(v_5, v_8, v_{10}) &= 576,643 & \Delta_{102} &= -\end{aligned}$$

Como não houve alteração, V_p é solução.

$$V_p = \{v_2, v_8, v_5\} \text{ e } \sigma(V_p) = 517,016$$

2) Algoritmo Genético aplicado ao Problema das P-medianas

Encontrar 3 medianas ($p=3$) para os 10 pontos dados acima. Considere $V=\{v_1=1, v_2=2, v_3=3, \dots, v_{10}=10\}$, onde 1, 2, 3, ..., 10 são os pontos considerados e $v_1, v_2, v_3, \dots, v_{10}$, são os vértices de um grafo $G(V, E)$, para o qual E (conjunto de arestas) são as distâncias euclidianas entre os pontos. No caso deste exemplo, os pontos estão representando as esquinas e pontos médios das quadras e os vértices escolhidos pelas p-medianas vão ser as p-sementes para a formação dos clusters. Assim, o peso ou demanda associado a cada ponto será sempre igual a 1 ($w_j=1$).

Primeiramente é calculada a matriz de distâncias euclidianas entre os 10 pontos (dada acima).

Passo 1: Inicialização

Parâmetros $p=3$ e $m=5$ (5 cromossomos de 3 elementos) gerar aleatoriamente a lista R :

$r_1 = (v_1, v_9, v_{10})$
 $r_2 = (v_1, v_3, v_5)$
 $r_3 = (v_2, v_8, v_9)$
 $r_4 = (v_4, v_7, v_{10})$
 $r_5 = (v_3, v_1, v_6)$

$R = (r_1, r_2, r_3, r_4, r_5)$

Calcular $C_i = \text{fitness}(r_i) = \sum w_j \{ \min[d(v_k, v_j)] \}$; para todo $v_k \in r_i$, $i = \{1, 2, 3, 4, 5\}$

Como neste exemplo o peso w_i será igual a 1, o cálculo do C_i resume-se ao $\sum \{ \min[d(v_k, v_j)] \}$.

Assim, para calcular:

$C_1 = \text{fitness}(r_1)$

$d(r_1, v_1) = \min[d(v_1, v_1); d(v_{10}, v_1); d(v_9, v_1)] = \min [0; 181,108; 240,416] = 0$

$d(r_1, v_2) = \min[d(v_1, v_2); d(v_{10}, v_2); d(v_9, v_2)] = \min [67,082; 155,242; 178,045] = 67,08$

$d(r_1, v_3) = 98,489$

$d(r_1, v_4) = 86,023$

$d(r_1, v_5) = 72,801$

$d(r_1, v_6) = 165,529$

$d(r_1, v_7) = 161,245$

$d(r_1, v_8) = 80,623$

$d(r_1, v_9) = 0$

$d(r_1, v_{10}) = 0$

$C_1 = \text{fitness}(r_1) = 731,792$

$C_2 = \text{fitness}(r_2)$

$d(r_2, v_1) = 0$

$d(r_2, v_2) = 67,082$

$d(r_2, v_3) = 0$

$d(r_2, v_4) = 41,231$

$d(r_2, v_5) = 0$

$d(r_2, v_6) = 219,317$

$d(r_2, v_7) = 180,278$

$d(r_2, v_8) = 141,421$

$d(r_2, v_9) = 143,178$

$d(r_2, v_{10}) = 72,801$

$C_2 = \text{fitness}(r_2) = 865,308$

$C_3 = \text{fitness}(r_3)$

$d(r_3, v_1) = 67,082$

$d(r_3, v_2) = 0$

$d(r_3, v_3) = 116,619$

$d(r_3, v_4) = 41,231$

$d(r_3, v_5) = 82,462$

$d(r_3, v_6) = 92,195$

$d(r_3, v_7) = 80,623$

$d(r_3, v_8) = 0$

$d(r_3, v_9) = 0$

$d(r_3, v_{10}) = 150,333$

$C_3 = \text{fitness}(r_3) = 630,545$

$C_4 = \text{fitness}(r_4)$
 $d(r_4, v_1) = 86,023$
 $d(r_4, v_2) = 41,231$
 $d(r_4, v_3) = 92,195$
 $d(r_4, v_4) = 0$
 $d(r_4, v_5) = 41,231$
 $d(r_4, v_6) = 50,990$
 $d(r_4, v_7) = 0$
 $d(r_4, v_8) = 80,623$
 $d(r_4, v_9) = 150,333$
 $d(r_4, v_{10}) = 0$
 $C_4 = \text{fitness}(r_4) = 542,626$

$C_5 = \text{fitness}(r_5)$
 $d(r_5, v_1) = 0$
 $d(r_5, v_2) = 67,082$
 $d(r_5, v_3) = 0$
 $d(r_5, v_4) = 86,023$
 $d(r_5, v_5) = 82,462$
 $d(r_5, v_6) = 0$
 $d(r_5, v_7) = 50,990$
 $d(r_5, v_8) = 92,195$
 $d(r_5, v_9) = 165,529$
 $d(r_5, v_{10}) = 108,167$
 $C_5 = \text{fitness}(r_5) = 652,445$

Como $C_4 \leq C_3 \leq C_5 \leq C_1 \leq C_2 \Rightarrow C_i = (542,626; 630,545; 652,448; 731,792; 865,308)$ e portanto $R = (r_4, r_3, r_5, r_1, r_2)$, ou seja $R = [(v_4, v_7, v_{10}); (v_2, v_8, v_9); (v_1, v_3, v_6); (v_1, v_9, v_{10}); (v_1, v_3, v_5)]$. Reordenado $R = (r_1, r_2, r_3, r_4, r_5)$.

Contador $k \leftarrow 0$ e definimos o erro $\varepsilon = 25$ e o n° de iterações $k_{\max} = 10$.

Passo 2: Teste

$C_m - C_1 \leq \varepsilon$ ou $k \geq k_{\max}$?

$865,308 - 542,626 = 322,682 > \varepsilon = 25$ e $k=0 < k_{\max} \leq 10$, portanto continua.

Passo 3: Seleção

$$\text{Select}(R) = \left\{ r_j \in R / j = m + 1 - \left\lceil \frac{-1 + \sqrt{1 + 4 \cdot \text{rnd}(m^2 + m)}}{2} \right\rceil \right\}$$

$j_1 = 1$ e $j_2 = 3$, como $j_1 \neq j_2$, selecionamos r_1 e r_3 de R .

Pai₁ $r_1 = (v_4, v_7, v_{10})$

Pai₂ $r_3 = (v_2, v_8, v_9)$

Passo 4: Crossover

Sortear aleatoriamente $c \in \{1, 2, 3\}$, $c=2$.

Pai₁ $r_1 = (v_4, v_7, v_{10})$ $r_x = (v_4, v_8, v_9)$ Filho₁

Pai₂ $r_3 = (v_2, v_8, v_9)$ $r_y = (v_2, v_7, v_{10})$ Filho₂

Passo 5: Como r_x e r_y são ambos cromossomos viáveis, calcular $C_x = \text{fitness}(r_x)$ e $C_y = \text{fitness}(r_y)$

$C_x = \text{fitness}(r_x)$
 $d(r_x, v_1) = 86,023$
 $d(r_x, v_2) = 41,231$
 $d(r_x, v_3) = 92,195$
 $d(r_x, v_4) = 0$
 $d(r_x, v_5) = 41,231$
 $d(r_x, v_6) = 92,195$
 $d(r_x, v_7) = 80,623$
 $d(r_x, v_8) = 0$
 $d(r_x, v_9) = 0$
 $d(r_x, v_{10}) = 114,018$
 $C_x = \text{fitness}(r_x) = 547,516$

$C_y = \text{fitness}(r_y)$
 $d(r_y, v_1) = 67,082$
 $d(r_y, v_2) = 0$
 $d(r_y, v_3) = 108,167$
 $d(r_y, v_4) = 41,231$
 $d(r_y, v_5) = 72,801$
 $d(r_y, v_6) = 50,990$
 $d(r_y, v_7) = 0$
 $d(r_y, v_8) = 80,623$
 $d(r_y, v_9) = 150,333$
 $d(r_y, v_{10}) = 0$
 $C_y = \text{fitness}(r_y) = 571,227$

$\Rightarrow r_t \leftarrow r_x, r_t = (v_4, v_8, v_9)$

Vai ao passo 7.

Passo 7:

Como $\text{fitness}(r_t) = 547,516 < \text{fitness}(r_5) = 865,308$

Faça:

Elimine r_5 da lista R ,

$R = [(v_4, v_7, v_{10}); (v_2, v_8, v_9); (v_1, v_3, v_6); (v_1, v_9, v_{10})]$.

Insira r_t na lista, mantendo a ordem crescente dos fitness.

$C_i = (542,626; 547,516; 630,545; 652,448; 731,792)$

$R = (r_1, r_t, r_2, r_3, r_4)$

$R = [(v_4, v_7, v_{10}); (v_4, v_8, v_9); (v_2, v_8, v_9); (v_1, v_3, v_6); (v_1, v_9, v_{10})]$.

Faça:

$k \leftarrow k + 1 = 1$

Volte ao passo 2.

Passo 2: Teste

$C_m - C_1 \leq \varepsilon$ ou $k \geq k_{\max}$?

$731,792 - 542,626 = 189,166 > \varepsilon = 25$ e $k=1 < k_{\max} \leq 10$, portanto continua.

Reordenar $R = [r_1, r_2, r_3, r_4, r_5]$

Passo 3: Seleção

$j_1=5$ e $j_2=3$, como $j_1 \neq j_2$, selecionamos r_5 e r_3 de R .

Pai₁ $r_5 = (v_1, v_9, v_{10})$

Pai₂ $r_3 = (v_2, v_8, v_9)$

Passo 4: Crossover

Sortear aleatoriamente $c \in \{1,2,3\}$, $c=1$.

Pai₁ $r_5 = (v_1, v_9, v_{10})$ $r_x = (v_1, v_8, v_{10})$ Filho₁

Pai₂ $r_3 = (v_2, v_8, v_9)$ $r_y = (v_1, v_9, v_9)$ Filho₂

Passo 5: r_y é inviável e r_x viável, calcular $C_x = \text{fitness}(r_x)$

$C_x = \text{fitness}(r_x)$

$d(r_x, v_1) = 0$

$d(r_x, v_2) = 67,082$

$d(r_x, v_3) = 98,489$

$d(r_x, v_4) = 86,023$

$d(r_x, v_5) = 72,801$

$d(r_x, v_6) = 92,195$

$d(r_x, v_7) = 80,623$

$d(r_x, v_8) = 0$

$d(r_x, v_9) = 80,623$

$d(r_x, v_{10}) = 0$

$C_x = \text{fitness}(r_x) = 577,836$

$\Rightarrow r_t \leftarrow r_x, r_t = (v_1, v_8, v_{10})$

Vai ao passo 7.

Passo 7:

Como $\text{fitness}(r_t) = 577,836 < \text{fitness}(r_5) = 731,792$

Faça:

Elimine r_5 da lista R,

$R = [(v_4, v_7, v_{10}); (v_4, v_8, v_9); (v_2, v_8, v_9); (v_1, v_3, v_6)]$.

Insira r_t na lista, mantendo a ordem crescente dos fitness.

$C_i = (542,626; 547,516; 577,836; 630,545; 652,448)$

$R = (r_1, r_2, r_t, r_3, r_4)$

$R = [(v_4, v_7, v_{10}); (v_4, v_8, v_9); (v_1, v_8, v_{10}); (v_2, v_8, v_9); (v_1, v_3, v_6)]$.

Faça:

$k \leftarrow k + 1 = 2$

Volte ao passo 2.

Passo 2: Teste

$C_m - C_1 \leq \varepsilon$ ou $k \geq k_{\max}$?

$652,448 - 542,626 = 109,822 > \varepsilon = 25$ e $k=2 < k_{\max} \leq 10$, portanto continua.

Reordenar $R = [r_1, r_2, r_3, r_4, r_5]$

Passo 3: Seleção

$j_1=4$ e $j_2=2$, como $j_1 \neq j_2$, selecionamos r_4 e r_2 de R.

Pai₁ $r_4 = (v_2, v_8, v_9)$

Pai₂ $r_2 = (v_4, v_8, v_9)$

Passo 4: Crossover

Sortear aleatoriamente $c \in \{1,2,3\}$, $c=1$.

Pai₁ $r_4 = (v_2, v_8, v_9)$ $r_x = (v_2, v_8, v_9)$ Filho₁

Pai₂ $r_2 = (v_4, v_8, v_9)$ $r_y = (v_4, v_8, v_9)$ Filho₂

Passo 5: Como r_y e r_x não alteraram r_4 e r_2 , o passo 5 não modifica nada. Promover uma mutação.

Passo 6: Mutação

Escolher r_x ou r_y aleatoriamente: escolhido r_x . Faça a mutação no cromossomo r_x com os elementos de V que não estarão em r_x .

$V - r_x = \{v_1, v_3, v_4, v_5, v_6, v_7, v_{10}\}$ sortear um elemento de $V - r_x$. Sorteado v_5 .

Mutação (r_x) = (v_2, v_5, v_9) , r_x é viável e $r_t \leftarrow r_x \Rightarrow r_t = (v_2, v_5, v_9)$.

Calcular $C_t = \text{fitness}(r_t) = 641,257$

Passo 7:

Como $\text{fitness}(r_t) = 641,257 < \text{fitness}(r_5) = 652,448$

Faça:

Elimine r_5 da lista R,

$R = [(v_4, v_7, v_{10}); (v_4, v_8, v_9); (v_1, v_8, v_{10}); (v_2, v_8, v_9)]$.

Insira r_t na lista, mantendo a ordem crescente dos fitness.

$C_i = (542,626; 547,516; 577,836; 630,545; 641,257)$

$R = (r_1, r_2, r_3, r_4, r_5)$

$R = [(v_4, v_7, v_{10}); (v_4, v_8, v_9); (v_1, v_8, v_{10}); (v_2, v_8, v_9); (v_2, v_5, v_9)]$.

Faça:

$k \leftarrow k + 1 = 3$

Volte ao passo 2.

Passo 2: Teste

$C_m - C_1 \leq \varepsilon$ ou $k \geq k_{\max}$?

$641,257 - 542,626 = 98,631 > \varepsilon = 25$ e $k=3 < k_{\max} \leq 10$, portanto continua.

Passo 3: Seleção

$j_1=1$ e $j_2=2$, como $j_1 \neq j_2$, selecionamos r_1 e r_3 de R .

Pai₁ $r_1 = (v_4, v_7, v_{10})$

Pai₂ $r_3 = (v_4, v_8, v_9)$

Passo 4: Crossover

Sortear aleatoriamente $c \in \{1,2,3\}$, $c=1$.

Pai₁ $r_1 = (v_4, v_7, v_{10})$ $r_x = (v_4, v_7, v_9)$ Filho₁

Pai₂ $r_3 = (v_4, v_8, v_9)$ $r_y = (v_4, v_8, v_{10})$ Filho₂

Passo 5: Como r_x e r_y são ambos cromossomos viáveis, calcular $C_x = \text{fitness}(r_x)$ e $C_y = \text{fitness}(r_y)$

$C_x = \text{fitness}(r_x) = 506,311$

$C_y = \text{fitness}(r_y) = 514,121$

$\Rightarrow r_t \leftarrow r_x, r_t = (v_4, v_7, v_{10})$

Vai ao passo 7.

Passo 7:

Como $\text{fitness}(r_1) = 506,311 < \text{fitness}(r_5) = 641,257$

Faça:

Elimine r_5 da lista R ,

$R = [(v_1, v_7, v_{10}); (v_4, v_8, v_9); (v_1, v_8, v_{10}); (v_2, v_8, v_9)]$.

Insira r_t na lista, mantendo a ordem crescente dos fitness.

$C_i = (506,311; 542,626; 547,516; 577,836; 630,545)$

$R = (r_t, r_1, r_2, r_3, r_4)$

$R = [(v_4, v_7, v_9); (v_4, v_7, v_{10}); (v_4, v_8, v_9); (v_1, v_8, v_{10}); (v_2, v_8, v_9)]$.

Faça:

$k \leftarrow k + 1 = 4$

Volte ao passo 2.

Passo 2: Teste

$C_m - C_1 \leq \varepsilon$ ou $k \geq k_{\max}$?

$630,545 - 506,311 = 124,234 > \varepsilon = 25$ e $k=4 < k_{\max} \leq 10$, portanto continua.

Continuar até que $C_m - C_1 \leq 25$ ou $k \geq 10$.

3) Algoritmo de Gillet e Johnson

Passo 1: Calcular a distância de cada nó (ainda não designado) até cada um dos depósitos fictícios (sementes) que ainda possuam espaço

Neste exemplo, vamos partir da solução do Algoritmo de Teitz e Bart e designar os demais pontos a estas sementes.

Sementes $\{v_2, v_5, v_8\}$

$d_{1,2} = 67,082$	$d_{1,5} = 117,047$	$d_{1,8} = 201,246$
$d_{3,2} = 116,619$	$d_{3,5} = 82,462$	$d_{3,8} = 220,000$
$d_{4,2} = 41,231$	$d_{4,5} = 41,231$	$d_{4,8} = 131,529$
$d_{6,2} = 180,278$	$d_{6,5} = 219,317$	$d_{6,8} = 92,195$
$d_{7,2} = 131,529$	$d_{7,5} = 180,278$	$d_{7,8} = 80,623$
$d_{9,2} = 178,045$	$d_{9,5} = 143,178$	$d_{9,8} = 80,623$
$d_{10,2} = 155,242$	$d_{10,5} = 72,801$	$d_{10,8} = 183,570$

Passo 2: Para cada nó i do passo anterior, obter t_i^1 como sendo o depósito mais próximo de i e t_i^2 como o segundo depósito mais próximo de i , com distâncias iguais a c_i^1 e c_i^2 , respectivamente.

i	c_i^1	t_i^1	c_i^2	t_i^2
1	$c_1^1 = 67,082$	2	$c_1^2 = 117,047$	5
3	$c_3^1 = 82,462$	5	$c_3^2 = 116,619$	2
4	$c_4^1 = 41,231$	2	$c_4^2 = 41,231$	5
6	$c_6^1 = 92,195$	8	$c_6^2 = 180,278$	2
7	$c_7^1 = 80,623$	8	$c_7^2 = 131,529$	2
9	$c_9^1 = 80,623$	8	$c_9^2 = 143,178$	5
10	$c_{10}^1 = 72,801$	5	$c_{10}^2 = 155,242$	2

Passo 3: Calcular a razão $r_i = c_i^1 / c_i^2$, ordenar os nós i de acordo com r_i (crescente).

$$r_1 = 0,5731$$

$$r_3 = 0,7071$$

$$r_4 = 1$$

$$r_6 = 0,5114$$

$$r_7 = 0,6130$$

$$r_9 = 0,5631$$

$$r_{10} = 0,4690$$

Ordem crescente: $r_{10} / r_6 / r_9 / r_1 / r_7 / r_3 / r_4$

Passo 4: Designação

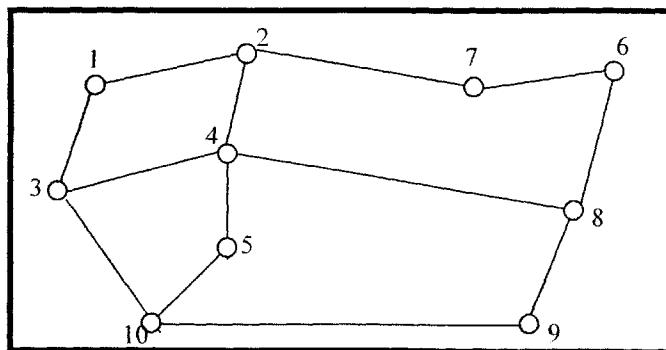
$$v_2 = \begin{cases} v_1 \\ v_4 \end{cases} \quad v_5 = \begin{cases} v_{10} \\ v_3 \end{cases} \quad v_8 = \begin{cases} v_6 \\ v_9 \\ v_7 \end{cases}$$

4) Algoritmo do Carteiro Chinês

4.1) Considerando-se apenas uma aresta ligando os nós.

Neste exemplo, são consideradas as coordenadas geográficas dos pontos dadas anteriormente, porém, as distâncias euclidianas foram arredondadas para facilitar o exemplo.

Figura A.4 – Grafo G



Atravessar todas as arestas do grafo G, contendo 10 nós e 13 arestas.

Passo 1. Obter a matriz de custos iniciais $C = [c_{ij}]$, entre todos os nós do grafo.

$$C = \begin{bmatrix} 0 & 67 & 98 & \infty \\ 67 & 0 & \infty & 41 & \infty & \infty & 131 & \infty & \infty & \infty \\ 98 & \infty & 0 & 92 & \infty & \infty & \infty & \infty & \infty & 108 \\ \infty & 41 & 92 & 0 & 41 & \infty & \infty & 131 & \infty & \infty \\ \infty & \infty & \infty & 41 & 0 & \infty & \infty & \infty & \infty & 73 \\ \infty & \infty & \infty & \infty & \infty & 0 & 51 & 92 & \infty & \infty \\ \infty & 131 & \infty & \infty & \infty & 51 & 0 & \infty & \infty & \infty \\ \infty & \infty & \infty & 131 & \infty & 92 & \infty & 0 & 81 & \infty \\ \infty & 81 & 0 & 150 \\ \infty & \infty & 108 & \infty & 73 & \infty & \infty & \infty & 150 & 0 \end{bmatrix}$$

Passo 2. Os vértices de grau ímpar e par são:

$$X^- = \{x_2, x_3, x_8, x_{10}\}$$

$$X^+ = \{x_1, x_4, x_5, x_6, x_7, x_9\}$$

Passo 3. Aplicar o algoritmo de Floyd obtendo a matriz $C' = [c'_{ij}]$, de custo mínimo entre todos os nós da rede. Sendo a matriz $\Theta = [\Theta_{ij}]$ a matriz de trajetos entre todos os nós i e j.

$$C' = \begin{bmatrix} 0 & 67 & 98 & 108 & 149 & 249 & 198 & 239 & 320 & 206 \\ 67 & 0 & 133 & 41 & 82 & 182 & 131 & 172 & 253 & 155 \\ 98 & 133 & 0 & 92 & 133 & 315 & 264 & 223 & 258 & 108 \\ 108 & 41 & 92 & 0 & 41 & 223 & 172 & 131 & 212 & 114 \\ 149 & 82 & 133 & 41 & 0 & 264 & 213 & 172 & 223 & 73 \\ 249 & 182 & 315 & 223 & 264 & 0 & 51 & 92 & 173 & 323 \\ 198 & 131 & 264 & 172 & 213 & 51 & 0 & 143 & 224 & 286 \\ 239 & 172 & 223 & 131 & 172 & 92 & 143 & 0 & 81 & 231 \\ 320 & 253 & 258 & 212 & 223 & 173 & 224 & 81 & 0 & 150 \\ 206 & 155 & 108 & 114 & 73 & 323 & 286 & 231 & 150 & 0 \end{bmatrix} \quad \Theta = \begin{bmatrix} 1 & 1 & 1 & 2 & 4 & 7 & 2 & 4 & 8 & 3 \\ 2 & 2 & 4 & 2 & 4 & 7 & 2 & 4 & 8 & 5 \\ 3 & 4 & 3 & 3 & 4 & 7 & 2 & 4 & 10 & 3 \\ 2 & 4 & 4 & 4 & 4 & 7 & 2 & 4 & 8 & 5 \\ 2 & 4 & 4 & 5 & 5 & 7 & 2 & 4 & 10 & 5 \\ 2 & 4 & 4 & 2 & 4 & 6 & 6 & 6 & 8 & 9 \\ 2 & 7 & 4 & 2 & 4 & 7 & 7 & 6 & 8 & 5 \\ 2 & 4 & 4 & 8 & 4 & 8 & 6 & 8 & 8 & 9 \\ 2 & 4 & 10 & 8 & 10 & 8 & 6 & 9 & 9 & 9 \\ 3 & 4 & 10 & 5 & 10 & 8 & 2 & 9 & 10 & 10 \end{bmatrix}$$

Obter a matriz $D = [d_{ij}]$, a partir da matriz C' , com as informações dos custos apenas entre os vértices de grau ímpar:

$$D = \begin{pmatrix} x_2 & x_3 & x_8 & x_{10} \\ x_2 & 0 & 133 & 172 & 155 \\ x_3 & 133 & 0 & 223 & 108 \\ x_8 & 172 & 223 & 0 & 231 \\ x_{10} & 155 & 108 & 231 & 0 \end{pmatrix}$$

Passo 4. Determinar o matching M^* entre todos os vértices de X' . Obtendo a todas as combinações entre dois vértices. As seguintes situações são possíveis:

Combinando x_2 - x_3 e x_8 - x_{10} , tem-se o custo de: $dx_2,x_3 + dx_8,x_{10} = 133 + 231 = 364$

Combinando x_2 - x_8 e x_3 - x_{10} , tem-se o custo de: $dx_2,x_8 + dx_3,x_{10} = 172 + 108 = 240$

Combinando x_2 - x_{10} e x_3 - x_8 , tem-se o custo de: $dx_2,x_{10} + dx_3,x_8 = 155 + 223 = 378$

A combinação que fornece o menor custo é x_2 com x_8 e x_3 com x_{10} .

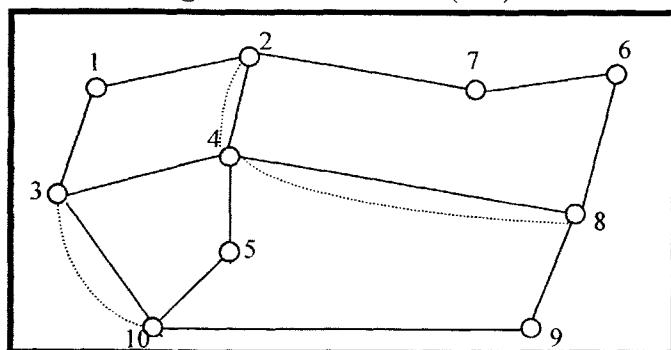
Passo 5. Aumentar o grafo com as arestas artificiais:

x_2 combinado com x_8 , caminho $x_2 - x_4 - x_8$

x_3 combinado com x_{10} , caminho $x_3 - x_{10}$

Incluir as arestas artificiais: $(x_2, x_4), (x_4, x_8), (x_3, x_{10})$, obtendo o grafo $G-(M^*)$

Figura A.5 – Grafo $G-(M^*)$



Passo 6. Obter a matriz de adjacência A do Grafo $G-(M^*)$

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 2 & 1 & 0 & 1 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 2 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Passo 7. Obter os circuitos

$$\phi_1 = \{x_1, x_2, x_4, x_2, x_7, x_6, x_8, x_9, x_{10}, x_3, x_1\}$$

$$\phi_2 = \{x_3, x_{10}, x_5, x_4, x_3\}$$

$$\phi_3 = \{x_4, x_8, x_4\}$$

Obter o circuito euleriano, de solução do problema:

Incluir ϕ_3 dentro de ϕ_2 :

$$\phi_2 = \{x_3, x_{10}, x_5, \underbrace{x_4,}_{Y} x_3\}$$

$$\phi_3 = \{x_4, x_8, x_4\}$$

tornando-se:

$$\phi_2 = \{x_3, x_{10}, x_5, x_4, x_8, x_4, x_3\}$$

Incluir ϕ_2 dentro de ϕ_1 :

$$\phi_1 = \{x_1, x_2, x_4, x_2, x_7, x_6, x_8, x_9, x_{10}, \underbrace{x_3,}_{\phi_2 = \{x_3, x_{10}, x_5, x_4, x_8, x_4, x_3\}} x_1\}$$

tornando-se:

$$\phi_1 = \{x_1, x_2, x_4, x_2, x_7, x_6, x_8, x_9, x_{10}, x_3, x_{10}, x_5, x_4, x_8, x_4, x_3, x_1\}$$

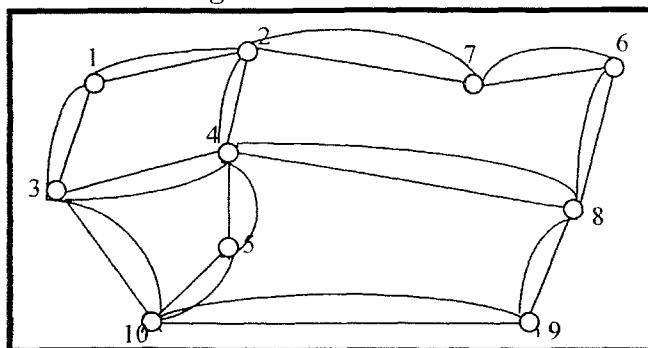
formando o circuito final de solução para o PCC.

Passo 8. Somando os custos do grafo $G-(M^*)$, obtém-se o custo total mínimo do PCC, que neste caso é 1436.

4.2) Considerando-se arestas duplas, exemplificando o problema real, onde devem ser percorridos os dois lados das ruas.

Neste exemplo, são consideradas as coordenadas geográficas dos pontos dadas anteriormente, porém, as distâncias euclidianas foram arredondadas para facilitar o exemplo.

Figura A.6 – Grafo G



Atravessar todas as arestas do grafo G, contendo 10 nós e 26 arestas.

Passo 1. Obter a matriz de custos iniciais $C = [c_{ij}]$, entre todos os nós do grafo.

$$C = \begin{bmatrix} 0 & 67 & 98 & \infty \\ 67 & 0 & \infty & 41 & \infty & \infty & 131 & \infty & \infty & \infty \\ 98 & \infty & 0 & 92 & \infty & \infty & \infty & \infty & \infty & 108 \\ \infty & 41 & 92 & 0 & 41 & \infty & \infty & 131 & \infty & \infty \\ \infty & \infty & \infty & 41 & 0 & \infty & \infty & \infty & \infty & 73 \\ \infty & \infty & \infty & \infty & \infty & 0 & 51 & 92 & \infty & \infty \\ \infty & 131 & \infty & \infty & \infty & 51 & 0 & \infty & \infty & \infty \\ \infty & \infty & \infty & 131 & \infty & 92 & \infty & 0 & 81 & \infty \\ \infty & 81 & 0 & 150 \\ \infty & \infty & 108 & \infty & 73 & \infty & \infty & \infty & 150 & 0 \end{bmatrix}$$

Passo 2. Todos os vértices de são de grau par. Portanto, a solução do Problema do Carteiro Chinês se reduz a encontrar um circuito euleriano.

Passo 3. Obter a matriz de adjacência A do Grafo G

$$A = \begin{bmatrix} 0 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 & 0 \\ 2 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 2 & 2 & 0 & 2 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 2 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 2 \\ 0 & 0 & 2 & 0 & 2 & 0 & 0 & 0 & 2 & 0 \end{bmatrix}$$

Passo 4. Obter os circuitos

$$\phi_1 = \{x_1, x_2, x_1, x_3, x_1\}$$

$$\phi_2 = \{x_2, x_4, x_2, x_7, x_6, x_7, x_2\}$$

$$\phi_3 = \{x_3, x_4, x_5, x_{10}, x_3, x_4, x_5, x_{10}, x_3\}$$

$$\phi_4 = \{x_4, x_8, x_6, x_8, x_9, x_{10}, x_9, x_8, x_4\}$$

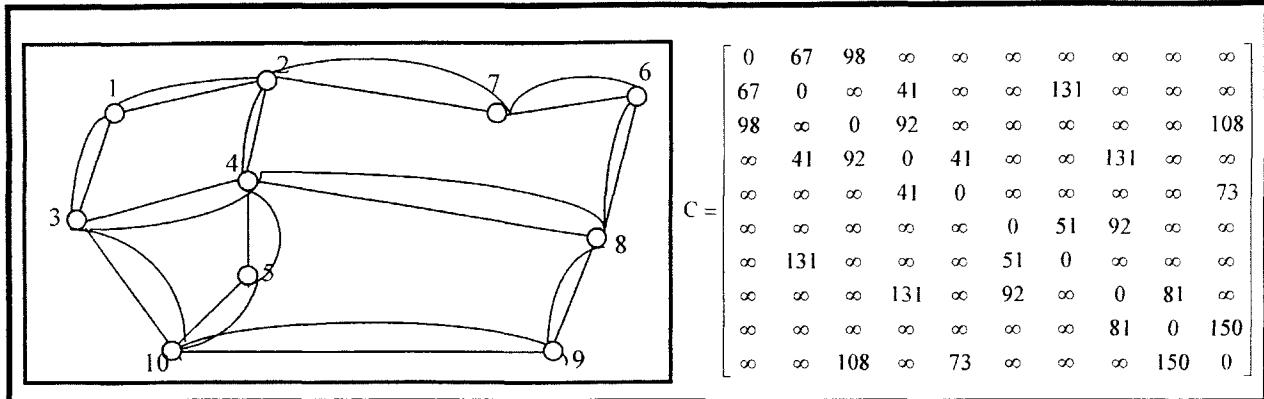
Incluir os circuitos, um dentro do outro, formando o circuito final de solução para o PCC.

$$\phi = \{x_1, x_2, x_4, x_2, x_7, x_6, x_7, x_2, x_1, x_3, x_4, x_8, x_6, x_8, x_9, x_{10}, x_9, x_8, x_4, x_5, x_{10}, x_3, x_4, x_5, x_{10}, x_3, x_1\}$$

Passo 5. Somando os custos do grafo G, obtém-se o custo total mínimo do PCC, que neste caso é 2312.

5) Problema do Carteiro Chinês (formulação)

Figura A.7 – Grafo G e Matriz de Distância C



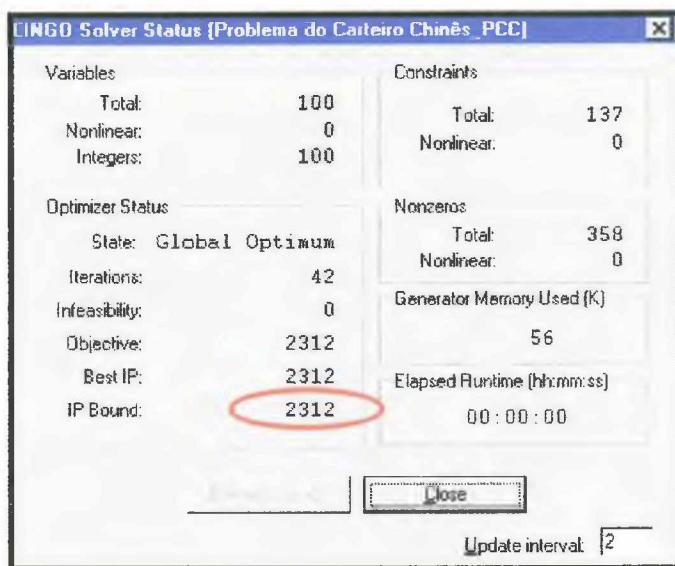
$$\begin{aligned}
 (\text{PCC})\text{MIN} = & 67 * x_{12} + 98 * x_{13} + 67 * x_{21} + 41 * x_{24} + 131 * x_{27} + 98 * x_{31} + 92 * x_{34} + 108 * x_{310} + 41 * \\
 & x_{42} + 92 * x_{43} + 41 * x_{45} + 131 * x_{48} + 41 * x_{54} + 73 * x_{510} + 51 * x_{67} + 92 * x_{68} + 131 * x_{72} + 51 * x_{76} + 131 * \\
 & x_{84} + 92 * x_{86} + 81 * x_{89} + 81 * x_{98} + 150 * x_{910} + 108 * x_{103} + 73 * x_{105} + 150 * x_{109};
 \end{aligned}$$

ST

$$\begin{aligned}
 & (x_{11} + x_{21} + x_{31} + x_{41} + x_{51} + x_{61} + x_{71} + x_{81} + x_{91} + x_{101}) - (x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} + x_{18} + x_{19} + x_{110}) = 0 \\
 & (x_{12} + x_{22} + x_{32} + x_{42} + x_{52} + x_{62} + x_{72} + x_{82} + x_{92} + x_{102}) - (x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26} + x_{27} + x_{28} + x_{29} + x_{210}) = 0 \\
 & (x_{13} + x_{23} + x_{33} + x_{43} + x_{53} + x_{63} + x_{73} + x_{83} + x_{93} + x_{103}) - (x_{31} + x_{32} + x_{33} + x_{34} + x_{35} + x_{36} + x_{37} + x_{38} + x_{39} + x_{310}) = 0 \\
 & (x_{14} + x_{24} + x_{34} + x_{44} + x_{54} + x_{64} + x_{74} + x_{84} + x_{94} + x_{104}) - (x_{41} + x_{42} + x_{43} + x_{44} + x_{45} + x_{46} + x_{47} + x_{48} + x_{49} + x_{410}) = 0 \\
 & (x_{15} + x_{25} + x_{35} + x_{45} + x_{55} + x_{65} + x_{75} + x_{85} + x_{95} + x_{105}) - (x_{51} + x_{52} + x_{53} + x_{54} + x_{55} + x_{56} + x_{57} + x_{58} + x_{59} + x_{510}) = 0 \\
 & (x_{16} + x_{26} + x_{36} + x_{46} + x_{56} + x_{66} + x_{76} + x_{86} + x_{96} + x_{106}) - (x_{61} + x_{62} + x_{63} + x_{64} + x_{65} + x_{66} + x_{67} + x_{68} + x_{69} + x_{610}) = 0 \\
 & (x_{17} + x_{27} + x_{37} + x_{47} + x_{57} + x_{67} + x_{77} + x_{87} + x_{97} + x_{107}) - (x_{71} + x_{72} + x_{73} + x_{74} + x_{75} + x_{76} + x_{77} + x_{78} + x_{79} + x_{710}) = 0 \\
 & (x_{18} + x_{28} + x_{38} + x_{48} + x_{58} + x_{68} + x_{78} + x_{88} + x_{98} + x_{108}) - (x_{81} + x_{82} + x_{83} + x_{84} + x_{85} + x_{86} + x_{87} + x_{88} + x_{89} + x_{810}) = 0 \\
 & (x_{19} + x_{29} + x_{39} + x_{49} + x_{59} + x_{69} + x_{79} + x_{89} + x_{99} + x_{109}) - (x_{91} + x_{92} + x_{93} + x_{94} + x_{95} + x_{96} + x_{97} + x_{98} + x_{99} + x_{910}) = 0 \\
 & (x_{110} + x_{210} + x_{310} + x_{410} + x_{510} + x_{610} + x_{710} + x_{810} + x_{910} + x_{1010}) - \\
 & (x_{101} + x_{102} + x_{103} + x_{104} + x_{105} + x_{106} + x_{107} + x_{108} + x_{109} + x_{1010}) = 0
 \end{aligned}$$

ST

$$\begin{aligned}
 & x_{12} + x_{21} \geq 2; & x_{43} + x_{34} \geq 2; & x_{84} + x_{48} \geq 2; \\
 & x_{13} + x_{31} \geq 2; & x_{45} + x_{54} \geq 2; & x_{86} + x_{68} \geq 2; \\
 & x_{21} + x_{12} \geq 2; & x_{48} + x_{84} \geq 2; & x_{89} + x_{98} \geq 2; \\
 & x_{24} + x_{42} \geq 2; & x_{54} + x_{45} \geq 2; & x_{98} + x_{89} \geq 2; \\
 & x_{27} + x_{72} \geq 2; & x_{510} + x_{105} \geq 2; & x_{910} + x_{109} \geq 2; \\
 & x_{31} + x_{13} \geq 2; & x_{67} + x_{76} \geq 2; & x_{103} + x_{310} \geq 2; \\
 & x_{34} + x_{43} \geq 2; & x_{68} + x_{86} \geq 2; & x_{105} + x_{510} \geq 2; \\
 & x_{310} + x_{103} \geq 2; & x_{72} + x_{27} \geq 2; & x_{109} + x_{910} \geq 2; \\
 & x_{42} + x_{24} \geq 2; & x_{76} + x_{67} \geq 2; & \\
 \\
 & x_{12} \geq 0; & x_{43} \geq 0; & x_{84} \geq 0; \\
 & x_{13} \geq 0; & x_{45} \geq 0; & x_{86} \geq 0; \\
 & x_{21} \geq 0; & x_{48} \geq 0; & x_{89} \geq 0; \\
 & x_{24} \geq 0; & x_{54} \geq 0; & x_{98} \geq 0; \\
 & x_{27} \geq 0; & x_{510} \geq 0; & x_{910} \geq 0; \\
 & x_{31} \geq 0; & x_{67} \geq 0; & x_{103} \geq 0; \\
 & x_{34} \geq 0; & x_{68} \geq 0; & x_{105} \geq 0; \\
 & x_{310} \geq 0; & x_{72} \geq 0; & x_{109} \geq 0; \\
 & x_{42} \geq 0; & x_{76} \geq 0; & x_{ij} \geq 0 \text{ e inteiro}
 \end{aligned}$$

Figura A.8 – Mensagem de solução (Software Lingo)

Global optimal solution found at step: 42
 Objective value: 2312.000
 Branch count: 0

Variable	Value	Reduced Cost
X12	0.0000000	0.0000000
X13	0.0000000	0.0000000
X21	2.000000	0.0000000
X24	1.000000	0.0000000
X27	0.0000000	0.0000000
X31	2.000000	0.0000000
X34	0.0000000	0.0000000
X310	1.000000	0.0000000
X42	1.000000	0.0000000
X43	2.000000	0.0000000
X45	0.0000000	0.0000000
X48	1.000000	0.0000000
X54	2.000000	0.0000000
X510	0.0000000	0.0000000
X67	2.000000	0.0000000
X68	0.0000000	0.0000000
X72	2.000000	0.0000000
X76	0.0000000	0.0000000
X84	1.000000	0.0000000
X86	2.000000	0.0000000
X89	0.0000000	0.0000000
X98	2.000000	0.0000000
X910	0.0000000	0.0000000
X103	1.000000	0.0000000
X105	2.000000	0.0000000
X109	2.000000	0.0000000
X11	0.0000000	0.0000000
X41	0.0000000	0.0000000
X51	0.0000000	0.0000000
X61	0.0000000	0.0000000
X71	0.0000000	0.0000000
X81	0.0000000	0.0000000
X91	0.0000000	0.0000000
X101	0.0000000	0.0000000
X14	0.0000000	0.0000000
X15	0.0000000	0.0000000
X16	0.0000000	0.0000000
X17	0.0000000	0.0000000

X18	0.0000000	0.0000000
X19	0.0000000	0.0000000
X110	4.000000	0.0000000
X22	0.0000000	0.0000000
X32	0.0000000	0.0000000
X52	0.0000000	0.0000000
X62	0.0000000	0.0000000
X82	0.0000000	0.0000000
X92	0.0000000	0.0000000
X102	0.0000000	0.0000000
X23	0.0000000	0.0000000
X25	0.0000000	0.0000000
X26	0.0000000	0.0000000
X28	0.0000000	0.0000000
X29	0.0000000	0.0000000
X210	0.0000000	0.0000000
X33	0.0000000	0.0000000
X53	0.0000000	0.0000000
X63	0.0000000	0.0000000
X73	0.0000000	0.0000000
X83	0.0000000	0.0000000
X93	0.0000000	0.0000000
X35	0.0000000	0.0000000
X36	0.0000000	0.0000000
X37	0.0000000	0.0000000
X38	0.0000000	0.0000000
X39	0.0000000	0.0000000
X44	0.0000000	0.0000000
X64	0.0000000	0.0000000
X74	0.0000000	0.0000000
X94	0.0000000	0.0000000
X104	0.0000000	0.0000000
X46	0.0000000	0.0000000
X47	0.0000000	0.0000000
X49	0.0000000	0.0000000
X410	0.0000000	0.0000000
X55	0.0000000	0.0000000
X65	0.0000000	0.0000000
X75	0.0000000	0.0000000
X85	0.0000000	0.0000000
X95	0.0000000	0.0000000
X56	0.0000000	0.0000000
X57	0.0000000	0.0000000
X58	0.0000000	0.0000000
X59	0.0000000	0.0000000
X66	0.0000000	0.0000000
X96	0.0000000	0.0000000
X106	0.0000000	0.0000000
X69	0.0000000	0.0000000
X610	0.0000000	0.0000000
X77	0.0000000	0.0000000
X87	0.0000000	0.0000000
X97	0.0000000	0.0000000
X107	0.0000000	0.0000000
X78	0.0000000	0.0000000
X79	0.0000000	0.0000000
X710	0.0000000	0.0000000
X88	0.0000000	0.0000000
X108	0.0000000	0.0000000
X810	0.0000000	0.0000000
X99	0.0000000	0.0000000
X1010	0.0000000	0.0000000

APÊNDICE II

AS COORDENADAS GEOGRÁFICAS DOS 774 PONTOS DE PARADA

Na relação abaixo são fornecidos os números dos vértices e suas coordenadas geográficas.

0001;331987.5180;7098901.069	0060;332700.3347;7097952.967	0119;332545.7895;7098688.092
0002;331958.2463;7098870.489	0061;332649.2471;7098051.473	0120;332440.3223;7098713.103
0003;331930.8718;7098840.179	0062;332621.4399;7098095.381	0121;332366.4952;7098799.982
0004;331951.2968;7098817.039	0063;332671.8809;7098124.437	0122;332357.2668;7098896.075
0005;331979.6904;7098787.812	0064;332690.6345;7098083.113	0123;332452.1873;7098877.646
0006;332009.8403;7098815.286	0065;332716.5016;7098148.974	0124;332442.9590;7098796.033
0007;332039.9901;7098841.006	0066;332595.8329;7098147.242	0125;332524.6962;7098790.557
0008;332010.1329;7098752.577	0067;332561.5590;7098209.229	0126;332589.2948;7098723.424
0009;332048.1861;7098709.450	0068;332610.7066;7098238.931	0127;332502.2844;7098645.760
0010;332059.1149;7098762.718	0069;332533.7519;7098257.011	0128;332511.5128;7098716.843
0011;332067.5218;7098816.311	0070;332654.6807;7098262.176	0129;332581.3848;7098637.862
0012;332063.6418;7098690.399	0071;332629.4602;7098314.478	0130;332562.9280;7098565.463
0013;332078.5153;7098674.052	0072;332554.4455;7098335.140	0131;332620.5635;7098597.636
0014;332108.9092;7098637.893	0073;332505.2980;7098309.312	0132;332668.0237;7098621.330
0015;332140.5965;7098603.025	0074;332602.2998;7098359.677	0133;332677.2521;7098498.910
0016;332178.7505;7098558.809	0075;332683.4333;7098213.105	0134;333340.0000;7100210.000
0017;332219.4912;7098516.839	0076;332459.0360;7098389.172	0135;332664.0688;7098395.921
0018;332245.0349;7098484.528	0077;332556.6845;7098446.639	0136;332714.1657;7098426.197
0019;332277.3688;7098448.369	0078;332411.8285;7098480.215	0137;332765.5810;7098452.524
0020;332298.0624;7098425.707	0079;332466.7961;7098509.917	0138;332744.4875;7098369.594
0021;332326.5163;7098375.691	0080;332508.1835;7098530.580	0139;332716.8023;7098295.879
0022;332362.0836;7098310.476	0081;332389.5183;7098508.882	0140;332768.2177;7098327.684
0023;332397.6509;7098249.780	0082;332409.5653;7098563.120	0141;332802.4945;7098272.398
0024;332428.0448;7098266.568	0083;332369.4712;7098529.544	0142;332773.4910;7098180.254
0025;332457.1453;7098282.065	0084;332336.4906;7098563.766	0143;332814.6078;7098352.695
0026;332433.2182;7098329.847	0085;332308.6835;7098599.925	0144;332833.0645;7098217.111
0027;332446.7985;7098306.601	0086;332346.1909;7098630.919	0145;332860.7497;7098160.508
0028;332424.1647;7098198.434	0087;332266.6493;7098645.124	0146;332881.8432;7098244.754
0029;332450.6786;7098151.944	0088;332234.3155;7098683.220	0147;332744.7358;7098101.273
0030;332500.7959;7098177.126	0089;332476.1732;7098563.636	0148;332885.7982;7098107.855
0031;332533.1298;7098197.788	0090;332446.4260;7098599.796	0149;332830.4279;7098080.211
0032;332478.1623;7098096.414	0091;332414.7388;7098637.892	0150;332939.8502;7098139.080
0033;332513.7296;7098038.947	0092;332386.9316;7098668.240	0151;332909.5283;7098065.365
0034;332566.1104;7098067.357	0093;332353.3043;7098707.652	0152;332934.5768;7098017.977
0035;332593.9176;7098082.854	0094;332313.2103;7098752.205	0153;332876.5698;7097987.701
0036;333230.0000;7100170.000	0095;332278.9363;7098722.503	0154;332930.6218;7097967.430
0037;332544.7700;7097982.360	0096;332181.9345;7098738.000	0155;332942.4869;7097935.838
0038;332572.5772;7097934.578	0097;332144.7261;7098779.556	0156;332864.7047;7097945.052
0039;332617.8448;7097962.343	0098;332175.1199;7098820.235	0157;332958.7840;7097979.115
0040;332654.0588;7097979.777	0099;332210.3640;7098872.046	0158;332976.5703;7097940.267
0041;332588.0976;7097895.190	0100;332102.6919;7098826.225	0159;333390.0000;7100200.000
0042;332623.0182;7097844.180	0101;332072.2980;7098864.322	0160;332968.7888;7097868.941
0043;332659.2322;7097863.551	0102;332040.2877;7098899.705	0161;332963.7864;7097810.668
0044;332692.2128;7097883.568	0103;332013.1272;7098930.699	0162;332930.4370;7097819.548
0045;332655.9988;7097774.368	0104;332038.3476;7098968.795	0163;333015.4779;7097805.496
0046;332696.0928;7097706.570	0105;332055.1613;7099003.895	0164;333068.8368;7097869.319
0047;332743.6235;7097734.335	0106;332131.4693;7098961.823	0165;333072.4496;7097930.034
0048;332803.7646;7097764.037	0107;332099.7820;7098907.686	0166;333022.4255;7097935.029
0049;332770.1373;7097830.544	0108;332069.3881;7099029.025	0167;333074.8558;7097989.394
0050;332730.0433;7097904.720	0109;332101.0754;7098992.453	0168;333030.8652;7098070.800
0051;332775.3107;7097927.320	0110;332168.3299;7098918.561	0169;332986.2881;7098044.445
0052;332854.2054;7097794.952	0111;332253.6915;709815.224	0170;333002.1247;7098113.879
0053;332898.2182;7097822.873	0112;332179.2909;7099104.057	0171;332981.5958;7098158.388
0054;332860.7109;7097890.127	0113;332233.3428;7099038.240	0172;332926.4611;7098265.936
0055;332824.4969;7097955.342	0114;332305.8516;7098955.310	0173;332950.5091;7098220.841
0056;332774.0560;7098044.835	0115;332408.6822;7098842.105	0174;333012.0960;7098176.331
0057;332799.9232;7098003.510	0116;332481.1909;7098753.910	0175;333040.2498;7098184.530
0058;332732.0220;7098021.590	0117;332358.5852;7099001.382	0176;333008.5767;7098301.075
0059;332678.3476;7097991.888	0118;332259.7097;7098918.453	0177;332968.6920;7098287.019

0178;333023.2402;7098246.609
 0179;333095.3846;7098182.773
 0180;333107.7019;7098294.633
 0181;333057.8461;7098298.147
 0182;333100.0769;7098238.410
 0183;333149.9328;7098173.988
 0184;332893.2991;7098325.484
 0185;332861.6259;7098376.436
 0186;332919.6934;7098406.890
 0187;332973.6551;7098437.344
 0188;332993.5975;7098370.861
 0189;332993.0109;7098370.275
 0190;333105.6266;7098390.773
 0191;332833.3969;7098426.450
 0192;332808.7623;7098474.473
 0193;332861.5508;7098503.756
 0194;332947.7723;7098552.247
 0195;332964.1954;7098497.196
 0196;332755.5877;7098565.552
 0197;332712.7703;7098646.958
 0198;332828.3188;7098593.921
 0199;332790.7801;7098703.249
 0200;332633.5874;7098762.400
 0201;332672.2991;7098716.134
 0202;332755.0013;7098677.832
 0203;332772.0109;7098757.362
 0204;332666.4336;7098791.916
 0205;332692.2415;7098816.513
 0206;332731.2464;7098778.446
 0207;332569.9479;7098831.247
 0208;332497.8035;7098918.063
 0209;332626.2558;7098899.276
 0210;332535.1666;7098944.628
 0211;332438.9741;7098985.038
 0212;332396.7431;7099036.575
 0213;332563.9071;7098972.153
 0214;332447.7721;7099079.795
 0215;332525.7819;7099054.027
 0216;332494.1088;7099124.305
 0217;332242.1899;7099121.071
 0218;332302.0170;7099140.398
 0219;332351.2863;7099087.103
 0220;332229.8726;7099205.310
 0221;332418.1519;7099330.100
 0222;332177.6705;7099244.549
 0223;332323.7191;7099265.679
 0224;332371.2287;7099186.616
 0225;332448.6521;7099242.838
 0226;332375.9211;7099390.304
 0227;332476.8059;7099173.029
 0228;334050.0000;7098300.000
 0229;332413.4597;7099130.862
 0230;332615.5226;7099012.492
 0231;332668.8978;7099065.201
 0232;332688.2536;7099006.050
 0233;332659.5131;7098945.916
 0234;332708.1960;7098963.486
 0235;332747.2008;7098860.411
 0236;332842.2203;7098742.931
 0237;332830.4895;7098491.970
 0238;332905.5667;7098532.966
 0239;333125.6055;7097925.149
 0240;333174.3230;7097920.919
 0241;333283.7609;7097908.934
 0242;333238.5736;7097913.164
 0243;333332.4784;7097901.179
 0244;333438.3859;7097892.015
 0245;333386.1382;7097896.949
 0246;333182.7956;7097991.417
 0247;333290.8213;7097982.252
 0248;333396.7289;7097975.907
 0249;333088.1848;7098018.911
 0250;333137.6084;7098083.064
 0251;333245.6341;7098071.080
 0252;333352.9538;7098059.095
 0253;333460.9795;7098047.110
 0254;333092.4212;7098087.294
 0255;333193.3864;7098073.195
 0256;333299.2939;7098061.915
 0257;333401.6712;7098053.455
 0258;333508.9910;7098040.060
 0259;333504.0487;7097970.972
 0260;333491.5926;7097885.670
 0261;333583.7213;7097858.092
 0262;333685.3926;7097825.663
 0263;333695.9833;7097890.522
 0264;333699.5137;7097934.935
 0265;333620.4360;7098032.981
 0266;333623.9662;7098083.740
 0267;333517.3525;7098087.265
 0268;333405.0905;7098099.249
 0269;333303.0663;7098110.133
 0270;333197.1588;7098121.413
 0271;333096.8996;7098137.627
 0272;333201.3950;7098167.942
 0273;333254.3488;7098156.662
 0274;333307.3027;7098153.842
 0275;333358.8443;7098145.382
 0276;333412.5042;7098142.562
 0277;333471.1064;7098136.922
 0278;333516.2937;7098130.578
 0279;333207.7495;7098226.455
 0280;333317.1874;7098218.700
 0281;333161.8562;7098290.608
 0282;333216.9282;7098282.854
 0283;333270.5880;7098277.214
 0284;333322.1297;7098270.869
 0285;333418.1526;7098203.896
 0286;333372.9653;7098265.229
 0287;333171.3278;7098392.854
 0288;333042.6786;7098463.610
 0289;333106.4179;7098484.565
 0290;333178.9010;7098490.986
 0291;333241.8631;7098490.210
 0292;333105.9408;7098436.628
 0293;333229.6076;7098389.351
 0294;333231.9393;7098436.695
 0295;333109.8800;7098346.305
 0296;333220.2579;7098330.783
 0297;333287.8842;7098383.560
 0298;333329.1842;7098319.116
 0299;333335.4028;7098376.550
 0300;333340.1945;7098432.445
 0301;333346.4128;7098479.789
 0302;333293.5969;7098484.534
 0303;333400.8663;7098469.792
 0304;333449.0594;7098464.359
 0305;333442.0635;7098417.791
 0306;333385.3201;7098371.999
 0307;333438.2304;7098367.343
 0308;333431.2128;7098311.364
 0309;333425.8015;7098260.015
 0310;333493.5026;7098359.130
 0311;333544.8050;7098352.921
 0312;333534.7002;7098249.696
 0313;333481.0565;7098252.858
 0314;333537.7999;7098304.082
 0315;333524.6197;7098188.428
 0316;333599.2413;7098345.982
 0317;333649.7665;7098289.325
 0318;333659.0945;7098339.773
 0319;333586.8028;7098244.344
 0320;333644.3234;7098236.583
 0321;333575.1862;7098123.942
 0322;333638.1482;7098172.838
 0323;333631.1524;7098120.061
 0324;333698.8158;7098229.447
 0325;333730.6854;7098228.671
 0326;333724.4668;7098284.553
 0328;333792.8702;7098220.134
 0327;333843.4870;7098215.473
 0329;333734.6635;7098166.576
 0330;333736.9956;7098109.142
 0331;333681.8067;7098115.352
 0332;333835.0148;7098180.158
 0333;333830.4067;7098098.142
 0334;333778.3329;7098102.684
 0335;333729.2702;7098018.691
 0336;333826.4340;7098046.632
 0337;333822.5474;7098008.601
 0338;333895.6795;7098091.559
 0339;333941.7143;7097996.154
 0340;333953.3740;7098085.410
 0341;333900.0000;7098210.000
 0342;333970.0000;7098200.000
 0343;333960.0000;7098140.000
 0344;333880.0000;7098000.000
 0345;333950.0000;7098040.000
 0346;333810.0000;7097920.000
 0347;333940.0000;7097900.000
 0348;333940.0000;7097830.000
 0349;334000.0000;7097900.000
 0350;334050.0000;7097900.000
 0351;334110.0000;7097870.000
 0352;334010.0000;7097810.000
 0353;334100.0000;7097830.000
 0354;334290.0000;7097880.000
 0355;334090.0000;7097790.000
 0356;334240.0000;7097890.000
 0357;334230.0000;7097810.000
 0358;334220.0000;7097730.000
 0359;334070.0000;7097720.000
 0360;334150.0000;7097720.000
 0361;333950.0000;7097740.000
 0362;333780.0000;7097790.000
 0363;334220.0000;7097610.000
 0364;334300.0000;7097730.000
 0365;334380.0000;7097730.000
 0366;334330.0000;7097680.000
 0367;334390.0000;7097660.000
 0368;334380.0000;7097680.000
 0369;334500.0000;7097730.000
 0370;334630.0000;7097730.000
 0371;334750.0000;7097680.000
 0372;334850.0000;7097610.000
 0373;334950.0000;7097480.000
 0374;335050.0000;7097420.000
 0375;335130.0000;7097420.000
 0376;335240.0000;7097410.000
 0377;333780.0000;7098330.000
 0378;333890.0000;7098320.000

0379;334010.0000;7098300.000	0446;334030.0000;7098610.000	0513;333970.0000;7099060.000
0380;334020.0000;7098260.000	0447;334350.0000;7098520.000	0514;333970.0000;7099100.000
0381;334000.0000;7098350.000	0448;334430.0000;7098530.000	0515;334020.0000;7099100.000
0382;334090.0000;7098380.000	0449;334470.0000;7098610.000	0516;334240.0000;7099090.000
0383;333990.0000;7098480.000	0450;334470.0000;7098570.000	0517;334170.0000;7099090.000
0384;334000.0000;7098410.000	0451;334470.0000;7098670.000	0518;334090.0000;7099150.000
0385;333950.0000;7098350.000	0452;334530.0000;7098620.000	0519;334090.0000;7099190.000
0386;333890.0000;7098380.000	0453;334470.0000;7098530.000	0520;333960.0000;7099150.000
0387;333950.0000;7098470.000	0454;334380.0000;7098600.000	0521;333960.0000;7099180.000
0388;333890.0000;7098470.000	0455;334370.0000;7098680.000	0522;333880.0000;7099110.000
0389;333890.0000;7098420.000	0456;334380.0000;7098650.000	0523;333730.0000;7099120.000
0390;333810.0000;7098360.000	0457;334470.0000;7098640.000	0524;333750.0000;7099080.000
0391;333810.0000;7098410.000	0458;334330.0000;7098590.000	0525;334100.0000;7099370.000
0392;333740.0000;7098450.000	0459;334480.0000;7098460.000	0526;333960.0000;7099360.000
0393;333550.0000;7098400.000	0460;334550.0000;7098540.000	0527;333960.0000;7099390.000
0394;333850.0000;7098400.000	0461;334620.0000;7098470.000	0528;333960.0000;7099270.000
0395;333560.0000;7098450.000	0462;334620.0000;7098550.000	0529;334020.0000;7099370.000
0396;333780.0000;7098430.000	0463;334720.0000;7098560.000	0530;333870.0000;7099360.000
0397;333500.0000;7098460.000	0464;334610.0000;7098690.000	0531;333760.0000;7099350.000
0398;333610.0000;7098450.000	0465;334610.0000;7098630.000	0532;333660.0000;7099310.000
0399;333670.0000;7098440.000	0466;333890.0000;7098530.000	0533;333680.0000;7099250.000
0400;333660.0000;7098400.000	0467;333830.0000;7098460.000	0534;333700.0000;7099210.000
0401;333020.0000;7098590.000	0468;333950.0000;7098310.000	0535;333810.0000;7099210.000
0402;333080.0000;7098620.000	0469;334250.0000;7098700.000	0536;333720.0000;7099160.000
0403;333100.0000;7098560.000	0470;334250.0000;7098640.000	0537;333590.0000;7099290.000
0404;333210.0000;7098560.000	0471;333990.0000;7098660.000	0538;333570.0000;7099340.000
0405;333160.0000;7098570.000	0472;334250.0000;7098760.000	0539;333620.0000;7099210.000
0406;333250.0000;7098590.000	0473;334250.0000;7098720.000	0540;333600.0000;7099250.000
0407;333360.0000;7098570.000	0474;334180.0000;7098720.000	0541;333550.0000;7099280.000
0408;333460.0000;7098570.000	0475;334100.0000;7098730.000	0542;333510.0000;7099260.000
0409;333570.0000;7098560.000	0476;334100.0000;7098800.000	0543;333540.0000;7099210.000
0410;333680.0000;7098540.000	0477;334100.0000;7098770.000	0544;333490.0000;7099310.000
0411;333260.0000;7098700.000	0478;334040.0000;7098740.000	0545;333570.0000;7099120.000
0412;333370.0000;7098690.000	0479;333980.0000;7098800.000	0546;333540.0000;7099170.000
0413;333310.0000;7098690.000	0480;334140.0000;7098830.000	0547;333710.0000;7099330.000
0414;333470.0000;7098670.000	0481;334100.0000;7098840.000	0548;333810.0000;7099120.000
0415;333430.0000;7098680.000	0482;334040.0000;7098840.000	0549;333730.0000;7099420.000
0416;333530.0000;7098670.000	0483;333980.0000;7098850.000	0550;333650.0000;7099400.000
0417;333530.0000;7098670.000	0484;333880.0000;7098860.000	0551;333560.0000;7099380.000
0418;333580.0000;7098660.000	0485;333770.0000;7098870.000	0552;333550.0000;7099420.000
0419;333480.0000;7098740.000	0486;333720.0000;7098880.000	0553;333530.0000;7099470.000
0420;333590.0000;7098730.000	0487;333600.0000;7098890.000	0554;333650.0000;7099470.000
0421;333820.0000;7098700.000	0488;333660.0000;7098890.000	0555;333580.0000;7099470.000
0422;333700.0000;7098710.000	0489;333590.0000;7098790.000	0556;333430.0000;7099460.000
0423;333700.0000;7098770.000	0490;333600.0000;7098840.000	0557;333480.0000;7099460.000
0424;333760.0000;7098710.000	0491;333650.0000;7098780.000	0558;333450.0000;7099410.000
0425;333640.0000;7098660.000	0492;333790.0000;7098760.000	0559;333470.0000;7099360.000
0426;333690.0000;7098650.000	0493;333890.0000;7098750.000	0560;333520.0000;7099370.000
0427;333780.0000;7098640.000	0494;334250.0000;7098860.000	0561;333350.0000;7099390.000
0428;333990.0000;7098620.000	0495;334250.0000;7098900.000	0562;333370.0000;7099330.000
0429;333980.0000;7098690.000	0496;334170.0000;7098900.000	0563;333410.0000;7099350.000
0430;333980.0000;7098740.000	0497;334100.0000;7098910.000	0564;333390.0000;7099460.000
0431;333770.0000;7098570.000	0498;333970.0000;7098930.000	0565;333350.0000;7099460.000
0432;333680.0000;7098580.000	0499;334250.0000;7098970.000	0566;333280.0000;7099460.000
0433;333880.0000;7098570.000	0500;334170.0000;7098980.000	0567;333240.0000;7099460.000
0434;333770.0000;7098520.000	0501;334040.0000;7098920.000	0568;333240.0000;7099380.000
0435;333770.0000;7098460.000	0502;334100.0000;7098990.000	0569;333260.0000;7099310.000
0436;333990.0000;7098570.000	0503;334030.0000;7099000.000	0570;333310.0000;7099320.000
0437;334100.0000;7098600.000	0504;333970.0000;7099000.000	0571;333210.0000;7099300.000
0438;334100.0000;7098550.000	0505;333870.0000;7099010.000	0572;333170.0000;7099450.000
0439;334110.0000;7098490.000	0506;333780.0000;7099020.000	0573;333130.0000;7099450.000
0440;334050.0000;7098490.000	0507;333780.0000;7098990.000	0574;333130.0000;7099380.000
0441;334200.0000;7098500.000	0508;333780.0000;7098950.000	0575;333160.0000;7099290.000
0442;334280.0000;7098510.000	0509;334610.0000;7098600.000	0576;333390.0000;7099270.000
0443;334280.0000;7098590.000	0510;333870.0000;7098940.000	0577;333290.0000;7099240.000
0444;334250.0000;7098590.000	0511;334100.0000;7099040.000	0578;333450.0000;7099240.000
0445;334170.0000;7098590.000	0512;334100.0000;7099090.000	0579;333410.0000;7099220.000

0580;333360.0000;7099200.000	0645;332590.0000;7099240.000	0710;332770.0000;7099820.000
0581;333310.0000;7099180.000	0646;332520.0000;7099240.000	0711;332830.0000;7099770.000
0582;333440.0000;7099150.000	0647;332550.0000;7099170.000	0712;332860.0000;7099810.000
0583;333340.0000;7099120.000	0648;332570.0000;7099110.000	0713;332810.0000;7099730.000
0584;333370.0000;7099040.000	0649;332610.0000;7099170.000	0714;332860.0000;7099740.000
0585;333430.0000;7099070.000	0650;332610.0000;7099200.000	0715;332890.0000;7099790.000
0586;333470.0000;7099080.000	0651;332810.0000;7099280.000	0716;332780.0000;7099700.000
0587;333650.0000;7099120.000	0652;332570.0000;7099350.000	0717;332790.0000;7099580.000
0588;333590.0000;7099050.000	0653;332690.0000;7099340.000	0718;332790.0000;7099420.000
0589;333510.0000;7099000.000	0654;332790.0000;7099350.000	0719;332890.0000;7099740.000
0590;333620.0000;7099000.000	0655;332900.0000;7099230.000	0720;332890.0000;7099650.000
0591;333400.0000;7098980.000	0656;332950.0000;7099240.000	0721;332890.0000;7099590.000
0592;333390.0000;7098920.000	0657;333010.0000;7099250.000	0722;332950.0000;7099590.000
0593;333450.0000;7098910.000	0658;332680.0000;7099420.000	0723;333010.0000;7099590.000
0594;333500.0000;7098910.000	0659;332600.0000;7099410.000	0724;333130.0000;7099590.000
0595;333550.0000;7098900.000	0660;332690.0000;7099380.000	0725;333130.0000;7099500.000
0596;333530.0000;7098790.000	0661;332480.0000;7099400.000	0726;332890.0000;7099520.000
0597;333490.0000;7098860.000	0662;332410.0000;7099400.000	0727;332960.0000;7099450.000
0598;333480.0000;7098800.000	0663;332740.0000;7099350.000	0728;333010.0000;7099440.000
0599;333370.0000;7098810.000	0664;332290.0000;7099470.000	0729;333220.0000;7099150.000
0600;333280.0000;7098820.000	0665;332390.0000;7099470.000	0730;332890.0000;7099440.000
0601;333280.0000;7098880.000	0666;332480.0000;7099480.000	0731;332900.0000;7099370.000
0602;333290.0000;7098930.000	0667;332480.0000;7099440.000	0732;333020.0000;7099360.000
0603;333340.0000;7098920.000	0668;332350.0000;7099540.000	0733;333050.0000;7099260.000
0604;333290.0000;7098970.000	0669;332410.0000;7099550.000	0734;333100.0000;7099280.000
0605;333240.0000;7098810.000	0670;332470.0000;7099550.000	0735;332950.0000;7099760.000
0606;333210.0000;7098890.000	0671;332680.0000;7099510.000	0736;333010.0000;7099790.000
0607;333180.0000;7098960.000	0672;332670.0000;7099640.000	0737;333070.0000;7099820.000
0608;333140.0000;7098770.000	0673;332600.0000;7099640.000	0738;333130.0000;7099840.000
0609;333200.0000;7098790.000	0674;332520.0000;7099630.000	0739;333180.0000;7099870.000
0610;333080.0000;7098920.000	0675;332430.0000;7099630.000	0740;333230.0000;7099890.000
0611;333110.0000;7098850.000	0676;332470.0000;7099630.000	0741;333260.0000;7099900.000
0612;333130.0000;7098950.000	0677;332500.0000;7099760.000	0742;333230.0000;7100010.000
0613;333280.0000;7099010.000	0678;332590.0000;7099750.000	0743;333190.0000;7100110.000
0614;333230.0000;7098980.000	0679;332520.0000;7099690.000	0744;333230.0000;7100130.000
0615;333330.0000;7099030.000	0680;332590.0000;7099700.000	0745;333290.0000;7100150.000
0616;333510.0000;7099100.000	0681;332340.0000;7099620.000	0746;333340.0000;7100180.000
0617;333170.0000;7098660.000	0682;332400.0000;7099660.000	0747;333340.0000;7100110.000
0618;333040.0000;7098730.000	0683;332370.0000;7099640.000	0748;333430.0000;7100210.000
0619;333930.0000;7098620.000	0684;332320.0000;7099710.000	0749;333430.0000;7100140.000
0620;333060.0000;7098680.000	0685;332350.0000;7099740.000	0750;333340.0000;7100150.000
0621;333010.0000;7098800.000	0686;332380.0000;7099700.000	0751;333230.0000;7100210.000
0622;332990.0000;7098850.000	0687;332410.0000;7099800.000	0752;333340.0000;7100340.000
0623;332940.0000;7098820.000	0688;332410.0000;7099770.000	0753;333430.0000;7100340.000
0624;332900.0000;7098780.000	0689;332480.0000;7099820.000	0764;333380.0000;7100350.000
0625;332910.0000;7098700.000	0690;332610.0000;7099900.000	0755;333260.0000;7100280.000
0626;332930.0000;7098630.000	0691;332670.0000;7099790.000	0756;333340.0000;7100260.000
0627;333080.0000;7098740.000	0692;332540.0000;7100000.000	0757;333430.0000;7100280.000
0628;333030.0000;7098890.000	0693;332420.0000;7100110.000	0758;333470.0000;7100230.000
0629;332850.0000;7098890.000	0694;332350.0000;7100050.000	0759;333230.0000;7100060.000
0630;332820.0000;7098980.000	0695;332380.0000;7100020.000	0760;333230.0000;7099940.000
0631;332710.0000;7099090.000	0696;332390.0000;7100080.000	0761;333070.0000;7099450.000
0632;332770.0000;7099100.000	0697;332200.0000;7100000.000	0762;332990.0000;7099150.000
0633;332790.0000;7099040.000	0698;332280.0000;7100020.000	0763;333090.0000;7099180.000
0634;332870.0000;7099000.000	0699;332140.0000;7100160.000	0764;333200.0000;7099220.000
0635;332920.0000;7099020.000	0700;332150.0000;7100390.000	0765;333120.0000;7099100.000
0636;332960.0000;7098940.000	0701;332230.0000;7100310.000	0766;333020.0000;7099060.000
0637;332890.0000;7099100.000	0702;332300.0000;7100230.000	0767;332980.0000;7099040.000
0638;332850.0000;7099190.000	0703;332220.0000;7100190.000	0768;333070.0000;7099080.000
0639;333430.0000;7100170.000	0704;332280.0000;7100120.000	0769;333170.0000;7099120.000
0640;332840.0000;7099220.000	0705;332360.0000;7100180.000	0770;333270.0000;7099160.000
0641;332710.0000;7099230.000	0706;332670.0000;7099710.000	0771;333050.0000;7099000.000
0642;332750.0000;7099220.000	0707;332740.0000;7099680.000	0772;333150.0000;7099030.000
0643;332740.0000;7099160.000	0708;332700.0000;7099740.000	0773;333250.0000;7099080.000
0644;332640.0000;7099230.000	0709;332890.0000;7099850.000	0774;334270.0000;7098560.000