

**UNIVERSIDADE FEDERAL DO PARANÁ**  
**SETOR DE TECNOLOGIA**  
**SETOR DE CIÊNCIAS EXATAS**

**SAMUEL BELLIDO RODRIGUES**

**A METAHEURÍSTICA COLÔNIA DE FORMIGAS APLICADA A UM PROBLEMA DE  
ROTEAMENTO DE VEÍCULOS: CASO DA ITAIPU BINACIONAL**

Dissertação apresentada como requisito parcial para  
obtenção do grau de Mestre em Ciências.

Programa de Pós-Graduação em Métodos  
Numéricos em Engenharia – Área de concentração  
em Programação Matemática.

Setor de Tecnologia – Departamento de Construção  
Civil e Setor de Ciências Exatas – Departamento de  
Matemática da Universidade Federal do Paraná.

Orientadora: Prof.<sup>a</sup> Maria Teresinha Arns Steiner,  
Dra.

Co-orientadora: Prof.<sup>a</sup> Rosangela Villwock, M.Sc.

CURITIBA

2007

**A METAHEURÍSTICA COLÔNIA DE FORMIGAS APLICADA A UM PROBLEMA DE  
ROTEAMENTO DE VEÍCULOS: CASO DA ITAIPU BINACIONAL**

**SAMUEL BELLIDO RODRIGUES**

Dissertação aprovada como requisito parcial para a obtenção do grau de Mestre em Ciências -  
Área de Concentração em Programação Matemática, do Programa de Pós-Graduação em  
Métodos Numéricos em Engenharia da Universidade Federal do Paraná, pela Comissão formada  
pelos professores:

Orientadora:

---

Prof.<sup>a</sup> Maria Teresinha Arns Steiner, D.Eng.  
Departamento de Matemática, UFPR

---

Prof. Marco César Goldberg, D.Eng.  
Departamento de Informática, UFRN

---

Prof. Paulo Henrique Siqueira, D.Sc.  
Departamento de Desenho, UFPR

---

Prof. Leandro dos Santos Coelho, D.Eng.  
Departamento de Engenharia de Produção, PUC-PR

Curitiba, 25 de junho de 2007

À minha família.

## **AGRADECIMENTOS**

A Deus.

A minha esposa Angelina e a minha filha Anna Júlia, pelo amor dispensando e pela compreensão nos momentos de ausência.

A meus pais José Edmundo e Madalena, por estarem ao meu lado em todos os momentos.

A meus irmãos Otoni e Oséias, pela amizade, carinho e confiança.

A professora Maria Teresinha Arns Steiner, pela sábia orientação deste trabalho.

E a professora Rosangela Villwock, pela co-orientação.

Ao professor Anselmo Chaves Neto pelo incentivo e ensinamentos.

Aos professores Celso Carnieri, Arinei Carlos Lindbeck da Silva, Liliana Madalena Gramani Cumin, Neida Maria Patias Volpi, Jair Mendes Marques e Volmir Eugênio Wilhem, pelos ensinamentos.

Aos meus amigos Jairo e Levi, pela ajuda em momentos decisivos.

A Hidrelétrica Itaipu Binacional, pelo fornecimento dos dados utilizados neste trabalho.

A todos os amigos e colegas que de uma maneira ou outra contribuíram para este momento.

## SUMÁRIO

<b>LISTAS DE FIGURAS .....</b>	<b>VII</b>
<b>LISTAS DE QUADROS .....</b>	<b>VIII</b>
<b>LISTAS DE GRÁFICOS .....</b>	<b>IX</b>
<b>LISTAS DE FLUXOGRAMAS.....</b>	<b>X</b>
<b>LISTAS DE SIGLAS.....</b>	<b>XI</b>
<b>RESUMO.....</b>	<b>XII</b>
<b>ABSTRACT .....</b>	<b>XIII</b>
<b>CAPÍTULO I .....</b>	<b>1</b>
<b>1. INTRODUÇÃO .....</b>	<b>1</b>
1.1 Objetivo do Trabalho.....	1
1.2 Importâncias do Trabalho .....	1
1.3 Estrutura do Trabalho .....	2
<b>CAPÍTULO II.....</b>	<b>3</b>
<b>2. DESCRIÇÃO DO PROBLEMA REAL.....</b>	<b>3</b>
2.1 Introdução.....	3
2.2 O Problema na Atualidade.....	3
2.2.1 Itinerários.....	4
<b>CAPÍTULO III .....</b>	<b>8</b>
<b>3. REVISÃO DE LITERATURA E MÉTODOS UTILIZADOS .....</b>	<b>8</b>
3.1 Revisão de Literatura.....	8
3.2 Métodos Utilizados.....	11
3.2.1 Problema de Localização de Facilidades.....	11
3.2.1.1 Problema das $p$ -Medianas.....	13
3.2.1.1.1 Formulação Matemática para o Problema das $P$ -Medianas .....	13
3.2.1.1.2 O Algoritmo das $p$ -Medianas de Teitz e Bart.....	14
3.2.2 Problemas de Designação.....	16
3.2.2.1 Procedimentos Básicos do Algoritmo de Designação de Gillet e Johnson .....	17
3.2.2.2 Procedimentos Básicos do Algoritmo Gillet e Johnson Modificado.....	17
3.2.3 O Problema de Roteamento de Veículos.....	18
3.2.3.1 Formulação Matemática para o Problema de Roteamento de Veículos (PRV) ....	19
3.2.3.2 Planejamento de Soluções Para o PRV .....	21
3.2.3.3 Principais Problemas de Roteamento .....	22
3.2.3.3.1 O Problema do Caixeiro Viajante (PCV).....	22
3.2.3.4 Classificação para o Problema de Roteamento.....	24
3.2.3.5 Variáveis Importantes para a Formulação do Problema de Roteamento.....	24
3.2.3.6 Desempenho Computacional.....	24
3.2.4 Otimização por Colônias de Formigas .....	25
3.2.4.1 Ant System ( $AS$ ) .....	29
3.2.4.1.1 Procedimentos Básicos do Algoritmo $AS$ .....	32
3.2.4.2 A Metaheurística $ACO$ .....	33
3.2.4.2.1 Algoritmo para a Metaheurística $ACO$ .....	35
<b>CAPÍTULO IV.....</b>	<b>38</b>
<b>4. IMPLEMENTAÇÃO DOS MÉTODOS UTILIZADOS E RESULTADOS.....</b>	<b>38</b>

4.1 Visão Geral da Metodologia Adotada .....	38
4.2 Primeira Etapa .....	39
4.3 Segunda Etapa .....	40
4.4 Terceira Etapa.....	46
4.4 Quarta Etapa .....	49
4.5 Análise dos Resultados.....	58
<b>CAPÍTULO V .....</b>	<b>61</b>
<b>5. CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS.....</b>	<b>61</b>
5.1 Conclusões.....	61
5.2 Sugestões para Trabalhos Futuros .....	63
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>64</b>
<b>APÊNDICE I.....</b>	<b>67</b>
<b>APÊNDICE II .....</b>	<b>82</b>

## LISTAS DE FIGURAS

Figura 1. Jogo de Hamilton. ....	23
Figura 2. Comportamento das formigas após o surgimento de um obstáculo entre o ninho e uma fonte de alimento [Dorigo e Gambardella, 1996]. ....	26
Figura 3. Experiência da ponte simples [Goss <i>et al.</i> (1990 <i>apud</i> DORIGO <i>et al.</i> 1999)]. ....	26
Figura 4. Mostra a porcentagem em relação a unidade de tempo, [Goss <i>et al.</i> , (1990 <i>apud</i> DORIGO <i>et al.</i> , 1999)]. ....	27
Figura 5. Experiência da ponte dupla [Goss <i>et al.</i> (1990 <i>apud</i> DORIGO <i>et al.</i> 1999)]. ....	28
Figura 6. Mediana selecionada da primeira região (Vila C). ....	41
Figura 7. Medianas selecionadas da segunda região (Vila B). ....	42
Figura 8. Medianas selecionadas da terceira região (Vila A), simulação 2. ....	44
Figura 9. Medianas selecionadas da quarta região (Centro). ....	45
Figura 10. Vértices designados para a terceira região (Vila A). ....	48
Figura 11. Vértices designados para a quarta região (Centro) ....	49
Figura 12. Rota obtida pelo <i>ACO</i> para a primeira região (Vila C). ....	50
Figura 13. Rota 1 obtida pelo <i>ACO</i> para a segunda região (Vila B). ....	52
Figura 14. Rota 2 obtida pelo <i>ACO</i> para a segunda região (Vila B). ....	52
Figura 15. Rota 3 obtida pelo <i>ACO</i> para a terceira região (Vila A). ....	54
Figura 16. Rota 4 obtida pelo <i>ACO</i> para a terceira região (Vila A). ....	54
Figura 17. Rota 8 obtida pelo <i>ACO</i> para a terceira região (Vila A). ....	55
Figura 18. Rota 12 obtida pelo algoritmo da formiga para a terceira região (Vila A). ....	55
Figura 19. Rota 14 obtida pelo <i>ACO</i> para a terceira região (Vila A). ....	56
Figura 20. Rota 3 obtida pelo <i>ACO</i> para a quarta região (Centro). ....	57
Figura 21. Pontos que o roteamento atual da região central não atende. ....	58

## LISTAS DE QUADROS

Quadro 1. Número da linha, itinerário e a quilometragem média, Vila “C”.....	4
Quadro 2. Números das linhas, itinerários e quilometragens médias, da Vila “B”.....	5
Quadro 3. Números das linhas, itinerários e quilometragens médias, da Vila “A”.....	6
Quadro 4. Números das linhas, itinerários e quilometragens médias, “Centro”.....	6
Quadro 5. Regiões da cidade com suas respectivas demandas.....	39
Quadro 6. Mediana selecionada da primeira região (Vila C).....	40
Quadro 7. Medianas selecionadas da segunda região (Vila B).....	41
Quadro 8. Medianas selecionadas da terceira região (Vila A), simulações 1 e 2.....	43
Quadro 9. Análise Estatística para a primeira simulação da terceira região (Vila A).....	44
Quadro 10. Análise Estatística para a segunda simulação da terceira região (Vila A).....	44
Quadro 11. Medianas selecionadas da quarta região (Centro).....	45
Quadro 12. Vértices designados para a primeira região (Vila C).....	46
Quadro 13. Vértices designados para a segunda região (Vila B).....	46
Quadro 14. Vértices designados para a terceira região (Vila A).....	47
Quadro 15. Vértices designados para a quarta região (Centro).....	48
Quadro 16. Parâmetros utilizados pelo algoritmo ACO.....	49
Quadro 17. Rota obtida pelo ACO para a primeira região (Vila C).....	50
Quadro 18. Simulações do algoritmo ACO para a primeira região (Vila C).....	51
Quadro 19. Rotas obtidas pelo ACO para a segunda região (Vila B).....	51
Quadro 20. Rotas obtidas pelo ACO para a terceira região (Vila A).....	53
Quadro 21. Rotas obtidas pelo ACO para a quarta região (Centro).....	57
Quadro 22. Número de veículos x Assentos disponíveis.....	58
Quadro 23. Gráfico 1 versus Gráfico 2.....	59
Quadro 24. Comparação entre a situação atual e a solução otimizada.....	60

## LISTAS DE GRÁFICOS

Gráfico 1. km por regiões da cidade.....	7
Gráfico 2. km por regiões da cidade após roteamento .....	59

## **LISTAS DE FLUXOGRAMAS**

Fluxograma 1. Solução dos problemas de localização de facilidades .....	12
Fluxograma 2. Estrategema para solução de PRV .....	21

## LISTAS DE SIGLAS

ACO	<i>Ant colony optimization</i>
AS	<i>Ant system</i>
CVRP	<i>Capacitated vehicle routing problem</i>
GA	<i>Genetic algorithm</i>
MDepCap	Problema de Múltiplos Depósitos e Capacitado
PCC	Problema do carteiro chinês
PCCC	Problema do carteiro chinês capacitado
PCV	Problema do caixeiro viajante
PCVM	Problema dos múltiplos caixeiros viajantes
PO	Pesquisa operacional
PRVC	Problema de roteamento de veículos capacitado
PRDMV	Problema de roteamento de nós com um único depósito e múltiplos veículos
PRDMVE	Problema de roteamento de nós com um único depósito e múltiplos veículos e demanda estocástica nos vértices
PPLIB	Problema programação linear inteira binária
PRMDMV	Problema de roteamento de nós com múltiplos depósitos e múltiplos veículos
PRV	Problema de roteamento de veículos
PRVJT	Problema roteamento de veículos com janela de tempo
RAM	<i>Random access memory</i>
TS	<i>Tabu search</i>
TSP	<i>Traveling salesman problem</i>

## RESUMO

O problema do roteamento de veículos é um dos problemas mais estudados da área de Pesquisa Operacional, tendo como objetivo designar pontos de paradas a veículos, bem como a determinação da seqüência em que estes pontos de parada serão visitados, estabelecendo assim as rotas para os veículos. Para determinação da solução existem três métodos: os métodos exatos, os métodos aproximativos e os métodos heurísticos ou metaheurísticos. Neste trabalho é proposta uma solução para o transporte de funcionários da empresa Hidrelétrica Itaipu Binacional, localizada no município de Foz do Iguaçu. Para resolver o problema, primeiramente, foi necessária a aplicação do algoritmo das  $p$ -Medianas de Teitz e Bart, medianas essas que representam o número de veículos a serem utilizados. Em seguida, aplicou-se o Algoritmo de Designação de Gillet e Johnson Modificado com o objetivo de designar os pontos de parada de cada veículo. Finalmente, o algoritmo de Otimização por Colônias de Formigas aplicado ao Problema do Caixeiro Viajante foi aplicado com a finalidade de determinar a ordem em que os pontos serão visitados, construindo desta maneira as rotas. Os resultados obtidos em comparação com os adotados pela empresa se mostraram melhores.

Palavras-chave:  $p$ -medianas, designação, roteamento de veículos, otimização por colônias de formigas.

## ABSTRACT

The vehicle routing problem has been largely studied in research operational area aiming to set vehicle stops, as well as the determination of the sequence where the location of stops will be visited, establishing thus the routes of vehicle. There are three methods for the solution: The metaheuristic or heuristic problem, the approaching method and the exact method. This paper presents a solution for the transportation of workers of Itaipu Power Plant, located in Foz do Iguaçu. To solve the problem, firstly, was necessary to apply the algorithm of the  $p$ -Median by Teitz and Bart, these medians represents the numbers of vehicle that will be used. Then, was applied the algorithm of designation by Gillet and Johnson modified with the goal of designate the locations of the stop of each vehicle. Finally, the algorithm Ant Colony Optimization applied on the traveling salesman problem was developed to determine the visited points order, constructing this way, the routes. The results obtained in comparison with the adopted ones for the company appear to be better.

**Key words:**  $P$ -Median, Designate, Routing of Vehicles, Ant Colony Optimization.

# CAPÍTULO I

## 1. Introdução

A Pesquisa Operacional (PO) é uma ciência que objetiva fornecer ferramentas quantitativas ao processo de tomada de decisões. O Problema do Roteamento de Veículos (PRV) é um dos problemas mais estudados da PO, sendo um dos percussores do seu sucesso nas últimas décadas, tendo sido abordado em vários trabalhos com formas de resolução diferentes. O PRV tem como objetivos designar pontos de paradas a veículos, bem como a determinação da seqüência em que estes pontos de parada serão visitados, estabelecendo assim as rotas para os veículos.

### 1.1 Objetivo do Trabalho

Este trabalho tem como objetivo principal o estudo e aplicação de heurísticas e metaheurísticas da área de Pesquisa Operacional para resolver um problema de otimização combinatória para a construção de rotas quase ótimas ou, até mesmo, ótimas, para os veículos que fazem o transporte de funcionários da empresa Itaipu Binacional localizada na cidade de Foz do Iguaçu, PR, comparando os resultados obtidos com a solução adotada atualmente pela empresa. Serão abordados os Problemas de Localização de Facilidades através do algoritmo das  $p$ -medianas de Teitz e Bart, os Problemas de Designação através do algoritmo de designação de Gillet e Johnson Modificado e a metaheurística *Ant Colony Optimization (ACO)* aplicado ao Problema do Caixeiro Viajante (PCV).

### 1.2 Importâncias do Trabalho

A relevância da aplicação deste trabalho se dá pelo fator qualidade, pois a Itaipu Binacional é uma empresa que se preocupa com o bem estar de seus funcionários, e, como este trabalho propicia a redução de custos oriundos da otimização, justifica-se sua aplicação.

### **1.3 Estrutura do Trabalho**

O trabalho está estruturado em cinco capítulos, sendo que neste primeiro é apresentada esta introdução.

O segundo capítulo consta da descrição detalhada do problema, de como ele é resolvido atualmente, bem como do levantamento dos dados necessários para o desenvolvimento do trabalho.

No terceiro capítulo se encontra a revisão de literatura, que consta de exemplos de trabalhos realizados envolvendo as técnicas matemáticas de roteamento.

No quarto capítulo é feita a implementação computacional das técnicas utilizadas, assim como a análise dos resultados.

E, por fim, no quinto capítulo são apresentadas as considerações finais e sugestões para trabalhos futuros.

## **CAPÍTULO II**

### **2. Descrição do Problema Real**

#### **2.1 Introdução**

A cidade de Foz do Iguaçu tem hoje cerca de 280.000 habitantes. A Usina Hidrelétrica de Itaipu está localizada no Rio Paraná, no trecho de fronteira entre o Brasil e o Paraguai, a 14 km ao norte da Ponte da Amizade. A área da usina se estende desde Foz do Iguaçu, no Brasil, e Ciudad del Este, no Paraguai, ao sul, até Guaira (Brasil) e Salto del Guairá (Paraguai), ao norte.

Comporta hoje 1.227 funcionários sendo que desses, 860 utilizam com frequência o transporte oferecido pela Itaipu o qual é terceirizado. A frota é constituída de ônibus e micro ônibus, sendo que os funcionários são apanhados em pontos de ônibus pré-determinados, e são entregues nos horários estabelecidos nos mesmos pontos em que foram pegos.

Como uma empresa Binacional é de grande porte, os funcionários são apanhados às 06h30min da manhã e entregues para o almoço, às 12h00min e, novamente apanhados às 13h30min e entregues às 17h30min.

Sendo a Itaipu uma empresa preocupada em atingir ao máximo a qualidade em seus serviços, é visível que um sistema de transporte de funcionários que seja seguro, e que utilize um menor espaço de tempo a um custo menor seja desejável.

#### **2.2 O Problema na Atualidade**

Dos 1.227 funcionários e estagiários cadastrados para utilizarem o transporte oferecido pela Itaipu, apenas 860 o usam com frequência. Isto é possível, pois a empresa utiliza uma política que permite que os seus funcionários cadastrem seus veículos e os utilizem para o trabalho.

Os funcionários que utilizam o transporte estão distribuídos em 3 bairros, centro da cidade de Foz do Iguaçu e suas intermediações.

Atualmente são disponibilizadas 926 poltronas para os 860 usuários do transporte. Sendo 19 ônibus com 46 poltronas cada e 2 micro ônibus com 26 lugares cada.

Os funcionários se concentram na sua maioria, em uma vila da cidade denominada de Vila A. Dos 860 usuários, 624 moram na Vila A, sendo que são disponibilizados 13 ônibus e 2 micro ônibus, ou seja, 650 lugares para dar atendimento à esta Vila.

Apenas 66 funcionários que utilizam o transporte moram na Vila B, sendo então disponibilizados 2 ônibus, ou seja, 92 lugares para a mesma.

Dos funcionários que moram na Vila C, 41 deles utilizam o transporte, sendo disponibilizados exatamente um ônibus para o trajeto até a usina.

Os demais 129 funcionários estão espalhados pelo centro e intermediações da cidade, onde se disponibiliza três ônibus, ou seja, 138 lugares.

Vale a pena frisar que com a venda das casas situadas nas Vilas A e B ocorrida no ano de 2005 e, também, com a renovação natural dos funcionários, tem havido uma tendência de espalhamento destes por toda a cidade.

O sistema de transporte da Itaipu foi definido por um roteamento baseado especificamente na experiência pessoal dos funcionários, sem nenhum critério de otimização matemático.

A seguir tem-se uma análise de cada um dos ônibus com suas respectivas rotas.

### 2.2.1 Itinerários

As rotas que fazem parte do percurso na Vila “C” são atendidas, conforme comentado, por um ônibus. A seguir, no Quadro 1, é mostrado o número da linha, o itinerário e a quilometragem média do veículo; observe-se que a distância total percorrida por este ônibus é de cerca de 58 quilômetros dia.

<b>Linha</b>	<b>Itinerários</b>	<b>Quilometragem Média (km)</b>
123	Vila “C”- R. Assunção n.º 78/ R. “F”/ R. Belo Horizonte/R. Campo Grande/ R. “Q”/ R. Recife/ R. “O”/ R. Rio Branco/ R. ”N” /R. Guararapes / R. “A”/ R. Florianópolis/ R. “D”/ R. Sapucaí/ R. “C”/ Barreira;	58

**Quadro 1.** Número da linha, itinerário e a quilometragem média, Vila “C”.

As rotas que fazem parte do percurso na Vila “B”, que são atendidas por dois ônibus, são identificadas por linhas. O Quadro 2, a seguir, fornece os números das linhas, os itinerários e as distâncias médias percorridas por dia por cada veículo. A distância total percorrida nesta região da cidade é de cerca de 151 quilômetros diários.

<b>Linha</b>	<b>Itinerários</b>	<b>Quilometragem Média (km)</b>
119	Vila ”B” – R. Pôr do Sol/Barreira;	93
121	Vila ”B” – R. Das Palmeiras/ R. Campanário/ R. Pôr do sol/ R. do Contorno/ R. Do Cedro/ R. Angico/ Barreira;	58
Total		151

**Quadro 2.** Números das linhas, itinerários e quilometragens médias, da Vila “B”.

Já as rotas que fazem parte do percurso na Vila “A”, atendidas por quinze ônibus, também são identificadas por linhas. No Quadro 3, a seguir, são apresentados os números das linhas, os itinerários e as quilometragens médias de cada veículo. Estes ônibus, perfazem um total de cerca de 1026 quilômetros diários, para coleta e entrega de funcionários nesta região da cidade.

<b>Linha</b>	<b>Itinerários</b>	<b>Quilometragem Média (km)</b>
102	Vila “A” – AV 1, Residencial Porto Seguro/Barreira;	63
103	Vila “A” – AV 5, 1, 4/Barreira;	63
105	Vila “A” - AV 6, 5, 3/Barreira;	68
106	Vila “A” - AV 3, 9, 15, 8, 3 /Barreira;	63
107	Vila “A” – AV 15 esq. c/ 7, 1, 6, 3 /Barreira;	73
108	Vila “A” - AV 5, 3, 6, 15, 8, 3, 9, 16, 4/Barreira;	108
109	Vila “A” - AV 7, 16, 9, 1, 7, 3 /Barreira;	73
111	Vila “A” – AV 8, 15, 7/Barreira;	73
112	Vila “A” – R. Trinta Reis /AV 1, 2, 3/Igreja São Jose OP./Htl. Salvatin /Porto Belo/Barreira;	53
113	Vila “A” – AV 16, 5, 1, 4, 3 /Barreira;	63

116	Vila “A” – AV 9, 1, 8, 3 /Barreira;	63
117	Vila “A” - AV 5, 15, 8, 3, 9, 16, 4, 1, 2 /Barreira;	68
124	Vila “A” - AV 3, 8, 15, 7, 3, Porto belo /Barreira;	63
114 (micro)	Vila “A” – R. Trinta Reis /AV 1, 2, 3/ Av. Tancredo Neves./ Frente vila ‘B’ / Frente Ecomuseu / Barreira;	49
128 (micro)	Vila “A” - AV 5, 15, 9, 16, 4, 3/Barreira;	83
Total		1026

**Quadro 3.** Números das linhas, itinerários e quilometragens médias, da Vila “A”.

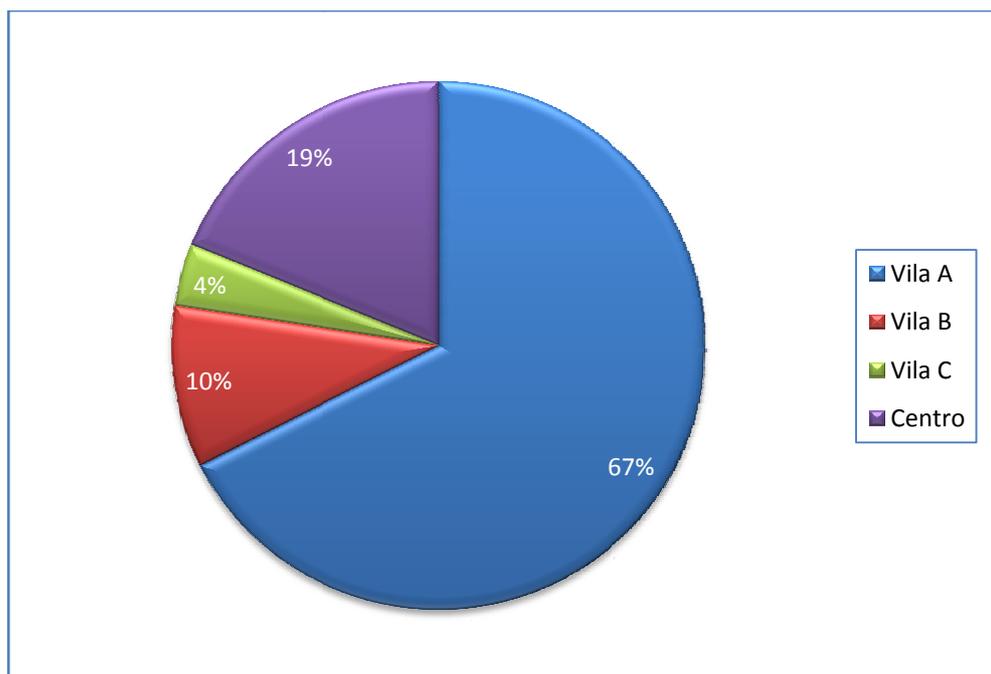
As rotas que fazem parte do percurso do centro da cidade são compostas por três ônibus, igualmente identificados pelas linhas, itinerários e quilometragens médias de cada veículo, conforme Quadro 4. Os ônibus perfazem um total de cerca de 285 quilômetros diários para coleta e entrega de funcionários nesta região da cidade.

<b>Linha</b>	<b>Itinerários</b>	<b>Quilometragem Média (km)</b>
110	Marechal Floriano / Mercosul / Igreja da Graça / Oeste Paraná Clube / Mitre / Correio / Hot. Ambassador /Rm Moveis /Posto /Batalhão /Formula 1 /Golden Eng. /Viaduto /Furnas /Ouro Verde /Barreira;	100
115	R. Santos Dumont (Pizza Hurt) /Jorge Schmelfeng (Lanville)/Av. Paraná /Esq. Barros/ Esq. Jorge Sanwais/ Hot. Continental /Eng. Rebolças /1º e 2º Apto após Venezuela/Inss /José M. <sup>a</sup> de Brito /Barreira;	89
129	Rep. Argentina (Queiroz Galvão)/ Mezomo/ Praça da bíblia/ Por do sol/ Paranapanema/ Tibagi/ Xingu/ Amazonas/ 1ª linha/ Rodoviária/ Av. José M. <sup>a</sup> de Brito/ JK/ Barreira;	96
Total		285

**Quadro 4.** Números das linhas, itinerários e quilometragens médias, “Centro”.

Deste modo, a distância percorrida considerando todos os ônibus é de cerca de 1.520 km diários para coleta e entrega de funcionários.

O Gráfico 1, a seguir, mostra mais detalhadamente a relação em porcentagem de quilometragem entre as diversas regiões da cidade.



**Gráfico 1.** km por regiões da cidade.

## CAPÍTULO III

### 3. Revisão de Literatura e Métodos Utilizados

Neste capítulo são apresentados alguns trabalhos desenvolvidos na área da Pesquisa Operacional, que utilizam métodos similares aos métodos abordados neste trabalho e, a seguir, se tem a apresentação dos métodos aqui utilizados.

#### 3.1 Revisão de Literatura

Vários trabalhos foram desenvolvidos na área de roteamento de veículos com diversas técnicas de resolução. A seguir são citados alguns trabalhos e suas abordagens adotadas.

ZAMBONI, 1997, objetiva em seu trabalho a apresentação de uma solução para o problema de roteamento do transporte escolar, que tem como objetivo minimizar a distância total percorrida pela frota, fazendo assim a redução de permanência dos alunos nos ônibus escolares. Foi utilizado um modelo de Programação Linear Inteira com o objetivo de obter quantidades ótimas de veículos juntamente com suas respectivas capacidades; já para obtenção dos *clusters* dos pontos de demanda, foi utilizada a heurística de Clarke e Wright e, finalmente, para as construções das rotas, utilizou-se o algoritmo de Inserção mais econômica.

COSTA, 1997, propôs uma solução para o problema de entregas de correspondências realizada pelos serviços postais que foi dividido em duas fases. Na primeira fase foi definida uma divisão de uma região da cidade em sub-regiões e, para isso, foram aplicados alguns algoritmos heurísticos para o agrupamento dos pontos de entrega, considerando vários “depósitos” (carteiros). Na segunda fase foi determinado para cada carteiro o roteiro de entrega das correspondências dentro de cada sub-região, sendo utilizados nesta fase algoritmos heurísticos mais precisamente os de inserção econômica de Clarke e Wright, *sweep* (varredura) e o de cobertura de arcos o algoritmo do carteiro chinês.

NUNES, 1998, apresenta uma proposta de solução para o problema de roteamento de veículos no transporte de funcionários de uma empresa, cujos objetivos eram: diminuir a quantidade de veículos utilizada no processo, bem como minimizar o percurso total percorrido. Através do algoritmo genético foram estabelecidos pontos de paradas dentre os vários possíveis

e, também, a seqüência para coleta dos funcionários em cada um dos *clusters* que foram determinados através do algoritmo MDepCap (Problema de Múltiplos Depósitos e Capacitado), assim denominado pelo autor que é uma adaptação do algoritmo de designação de Gillet e Johnson.

DORIGO, 1999, apresenta um apanhado geral sobre a metaheurística *ant colony optimization* (ACO). Na primeira parte do artigo, o autor compara as formigas reais com o comportamento das formigas artificiais, definindo assim a metaheurística ACO. Na segunda parte do artigo são apresentadas algumas aplicações do algoritmo ACO.

CORRÊA, 2000, apresenta uma aplicação do problema das  $p$ -medianas capacitado a um problema real. É proposto um algoritmo que otimiza a designação de candidatos ao vestibular para os locais de provas mais próximos de suas residências. Duas heurísticas são propostas, comparativamente: a primeira é baseada em um algoritmo genético simples que utiliza os operadores genéticos usuais e um operador heurístico chamado “hipermutação direcionada”; a segunda heurística é baseada em busca tabu que usa memória de curto e de longo prazo para controlar a busca.

COLOMBO, 2001, apresenta uma solução de um problema de roteamento de veículos de uma empresa de segurança eletrônica. Antes, no entanto, se preocupando em determinar a localização espacial dos centros de serviços, que foi obtida através da solução de um problema de  $p$ -medianas que, sobretudo visava a minimizar a distância média de deslocamento dos agentes de segurança, foram utilizados nessa etapa os algoritmos de Teitz e Bart e o Genético. Para a formação dos *clusters* utilizou-se o algoritmo de Gillet e Johnson. O algoritmo usado para o roteamento dos veículos foi o de Clarke e Wright.

SMIDERLE, 2001, propôs uma solução para o trabalho de leitura dos medidores das contas de água dos clientes de uma empresa de saneamento básico. O trabalho é dividido em duas fases distintas: a primeira fase é aquela onde são formados os *clusters* de atendimento para cada leiturista onde foi feito o melhoramento da solução obtida através do algoritmo genético com a heurística de Teitz e Bart. Já a segunda fase, consiste na criação de rotas de modo a minimizar a distância percorrida para cada trabalhador, onde foi utilizada a formulação matemática do Problema do Carteiro Chinês.

SILVA e RAMALHO, 2003, avaliam o desempenho e funcionamento da metaheurística *Ant Colony Optimization* (ACO, Otimização por Colônia de Formigas), através de análises experimentais rigorosas. O problema se refere às instâncias de grande porte do problema de cobertura de conjuntos (*Set Covering Problem* (SCP)). O estudo envolve adaptações de

heurísticas canônicas da metaheurística *ACO*. Incluindo versões híbridas baseadas no método de busca local, chega-se a conclusão de que não se justifica a utilização desta técnica para estes problemas, uma vez que existem métodos mais simples e eficientes.

ANDRADE, BATISTA e TOSO, 2004, apresentam um problema de roteamento de veículos para coletas e entregas de cargas em um ambiente reduzido, tais como armazéns e minas de extração de minério. Para a resolução do modelo apresentado utilizou-se um algoritmo baseado na heurística de Colônias de Formigas.

BELL e MCMULLEN, 2004, aplicam o método metaheurístico de Otimização por Colônias de Formigas (*ACO*) a um problema de roteamento de veículos (PRV), aplicado ao problema tradicional do caixeiro viajante. Algumas modificações são feitas no algoritmo *ACO* a fim de permitir a busca de rotas múltiplas. Os experimentos mostraram que o algoritmo é bem sucedido para encontrar boas soluções.

LOISEAU e MAZZEO, 2004, tratam do Problema de Roteamento de Veículos Capacitado (PRVC) - *Capacitated Vehicle Routing Problem (CVRP)* - onde desenvolveram um algoritmo baseado na metaheurística de Colônias de Formigas para o PRVC. Apresentam os resultados preliminares que mostram que os algoritmos da formiga podem competir com outras metaheurísticas para resolver o PRVC.

FORMIGONI, 2005, apresenta uma proposta de solução para o problema de localização de facilidades seguido do roteamento dos veículos que realizam entregas em uma empresa do setor avícola. Foi utilizado para a localização de facilidades o problema de  $p$ -medianas cujo objetivo é minimizar a distância média de deslocamento dos caminhões. Foi utilizado, ainda, um algoritmo de Programação Linear Inteira e outro heurístico utilizando como modelo o algoritmo de Teitz e Bart. Para a formação dos grupos foi utilizado o algoritmo de Gillet e Johnson. Já o roteamento em cada agrupamento, foi obtido como solução de um problema dos Múltiplos Caixeiros Viajantes através do algoritmo de Clarke e Wright.

OLIVEIRA, 2005, apresenta um trabalho que se refere ao Roteamento de Veículos com Janela de Tempo (PRVJT), que tem a particularidade de considerar o tempo de disponibilidade dos consumidores em suas restrições, tratando assim de uma particularidade bastante comum nos problemas de coleta ou entrega no mundo real. Utilizou-se para a resolução deste trabalho uma adaptação do Algoritmo Genético, que apresentou bom desempenho.

BASIR, CALAMAI e ELLABIB, 2007, aplicam o conceito de processamento em paralelo para melhorar o desempenho do Sistema de Colônia de Formigas. As novas estratégias de troca são baseadas em um esquema onde se torna mais pesado e são introduzidas sob três

tipos diferentes de interações. O desempenho do algoritmo do sistema múltiplo da Colônia de Formigas é aplicado ao Problema de Roteamento de Veículos com janelas de Tempo no Problema do Caixeiro Viajante, e é investigado e avaliado com respeito à qualidade da solução e esforço computacional. Os estudos indicam também que o esquema se torna mais pesado e com melhor desempenho, particularmente nas estratégias que compartilham a informação do feromônio entre todas as colônias.

### 3.2 Métodos Utilizados

A seguir são apresentados os métodos propostos para a solução do problema. Primeiramente é abordado o problema de localização de facilidades, mais especificamente, o problema das  $p$ -medianas. Em seguida é apresentado o algoritmo de Gillet e Johnson para formação dos *clusters*. Depois, o Problema de Roteamento de Veículos é exposto e, finalmente, é apresentado o algoritmo que faz o roteamento através do algoritmo *Ant Colony Optimization* (ACO, Otimização por Colônias de Formigas).

#### 3.2.1 Problema de Localização de Facilidades

Um estudo importante da área da Pesquisa Operacional envolve aplicações de problemas de localização de facilidades (instalações).

O problema de localização de facilidades trata de problemas cujo objetivo é selecionar a melhor localização em uma determinada região para a facilidade dos serviços. Podem ser classificados como problemas de localização de centros e problemas de localização de medianas, sendo este último o objetivo neste trabalho.

No problema de centros, o objetivo é minimizar a distância até o ponto mais crítico a ser atendido. Geralmente são aplicados a serviços de localização de hospital, postos de saúde, postos de bombeiros, escolas, etc.

Os problemas de localização de medianas têm por objetivo minimizar a distância a ser percorrida, isto é, minimizar a soma de todas as menores distâncias da facilidade aos vértices de um grafo. Um dos problemas mais conhecidos é o chamado problemas das  $p$ -medianas HAKIMI (1965) e REVELLE e SWAIN (1970).

Os problemas das  $p$ -medianas são ainda classificados em: problema das  $p$ -medianas não-capacitado e capacitado.

O caso do problema das  $p$ -medianas não-capacitado, considera que cada ponto candidato a mediana pode atender a um número infinito de pontos de demanda. Já no caso do problema das  $p$ -medianas capacitado, cada ponto candidato a mediana tem a capacidade limitada não podendo de forma alguma ter esta capacidade extrapolada.

O problema das  $p$ -medianas pertence à classe de problemas conhecidos como *NP-hard* (Não Polinomiais difíceis) como mostra o trabalho de Kariv e Hakimi (1979).

De acordo com Minieka (1977), as facilidades podem ser localizadas nos vértices e/ou nas arestas, e essas localizações são chamadas de medianas.

Rosa (1996) cita algumas aplicações práticas para o problema de locação de medianas. Dentre elas, pode-se citar a localização de 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. Outro problema é o da localização de uma escola numa determinada cidade de modo que os alunos de cada bairro desloquem-se, em média, o mínimo possível.

Corrêa (2000) apresenta uma aplicação do problema das  $p$ -medianas capacitado. Nele é proposto um algoritmo que otimiza a designação de candidatos ao vestibular para os locais de provas mais próximos de suas residências, conforme já comentado.

As  $p$  instalações que pertencem a uma solução qualquer para o problema são chamadas medianas e, por este motivo, tem-se  $p$ -medianas.

A seguir é apresentado o Fluxograma 1 para melhor visualização da solução dos problemas de localização de facilidades, conforme Pires (2002).



**Fluxograma 1.** Solução dos problemas de localização de facilidades

Para problemas de pequeno porte para a resolução do problema das medianas, a solução pode ser encontrada por formas exatas, tais como: modelo de Programação Inteira e enumeração exaustiva. Existem também métodos aproximativos ou heurísticos especializadas em resolver problemas que requerem um esforço computacional elevado. Uma dessas heurísticas é a de TEITZ E BART (1968), que é baseada na substituição de medianas na solução e tem como objetivo a partir de uma solução inicial melhorar a cada iteração o valor da função objetivo.

### 3.2.1.1 Problema das $p$ -Medianas

O problema das  $p$ -medianas consiste em determinar ( $p$ ) instalações em um conjunto  $n$  instalações candidatas com  $n > p$ , de forma que a soma total das distâncias percorridas de cada ponto de demanda até a instalação mais próxima seja a menor possível.

#### 3.2.1.1.1 Formulação Matemática para o Problema das $P$ -Medianas

Christofides (1975) formulou o Problema das  $p$ -medianas como um Problema de Programação Linear Inteira Binária (PPLIB), da seguinte forma:

Sendo:

$\xi_{ij}$  uma matriz de alocação, onde

$$\xi_{ij} = \begin{cases} 1, & \text{se o vértice } x_j \text{ é alocado ao vértice } x_i \\ 0, & \text{caso contrário;} \end{cases}$$

ou ainda:

$$\xi_{ii} = \begin{cases} 1, & \text{se o vértice } x_i \text{ é um vértice mediana} \\ 0, & \text{caso contrário.} \end{cases}$$

onde  $d_{ij}$  é uma matriz de distâncias ponderadas.

Tem-se a formulação:

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

Sujeito a:

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

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

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

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

A função objetivo (a) minimiza a soma das distâncias dos vértices de demanda até o conjunto de medianas. As restrições (b) garantem que todo vértice  $x_j$  será designado para exatamente um vértice – mediana  $x_i$ . A restrição (c) define o número de medianas igual a  $p$ . As restrições (d) garantem que as alocações só podem ser feitas nos vértices medianas. E, finalmente, as restrições (e) garantem que as variáveis de decisão sejam binárias.

Quando a restrição (e) for escrita como  $\xi_{ij} \geq 0$ , o problema passa a ser um problema de Programação Linear.

### 3.2.1.1.2 O Algoritmo das $p$ -Medianas de Teitz e Bart

Como o problema das  $p$ -medianas é descrito como um Problema de Programação Linear Inteira Binária (PPLIB) é notório que tem um grau de complexidade elevado dependendo do número de instalações, que apresentaria um elevado tempo computacional para sua resolução. Por este motivo, é abordada neste trabalho uma das heurísticas mais conhecidas para a resolução do problema das  $p$ -medianas desenvolvida por TEITZ e BART (1968) que é conhecido como o algoritmo das  $p$ -medianas de Teitz e Bart.

A heurística das  $p$ -medianas de Teitz e Bart 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. A implementação computacional desta heurística é fácil e produz soluções satisfatórias para problemas pequenos.

Considerando como potenciais medianas todos os vértices de um grafo, o algoritmo de Teitz e Bart tem a seguinte formulação: seja  $G(V, A)$  um grafo não direcionado, onde  $V$  é o conjunto de vértices e  $A$  é o conjunto das arestas.

A menor distância entre  $v_i$  e  $v_j$  é dada por  $d(v_i, v_j)$ , sendo  $w_j$  um peso associado ao vértice  $v_j$ .

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:

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

Tem-se que  $v_m$  é uma mediana se, dentre 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. Portanto tem-se:

$$\sigma(v_m) = \text{mínimo}[\sigma(v_i)], \forall v_i \in V \quad (\text{b})$$

Quando o objetivo é encontrar  $p$ -medianas ( $p > 1$ ), ou seja,  $V_p \subset V$  e  $|V_p| = p$ , calcula-se:

$$d(V_p, v_j) = \text{mínimo} \left[ d(v_i, v_j) \right], \forall v_i \in V_p, v_j \in V \quad (\text{c})$$

e

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

Desta maneira, tem-se que um conjunto de  $p$  vértices é a solução ótima para o problema das  $p$ -medianas se, dentre 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. Logo, se tem:

$$\sigma(V_{p\_solução\_ótima}) = \text{mínimo}[\sigma(V_p)], \forall V_p \subset V \quad (\text{e})$$

Portanto o objetivo do algoritmo das  $p$ -medianas de Teitz e Bart é encontrar um conjunto  $V_p$  em  $V$ , para o qual o número de transmissão seja mínimo.

A seguir são descritos 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:

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

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

Faça  $\Delta_{ij\_máximo} = \text{máximo}[\Delta_{ij}]$ ,  $\forall \Delta_{ij}$  calculado anteriormente.

Se  $\Delta_{ij\_máximo} > 0$  então:

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

Rotule  $v_j$  como “analisado”.

Caso contrário, continue.

#### Passo 3

Se durante a execução do Passo 2, houver alguma modificação no conjunto  $V_p$ , então:

Volte ao Passo 1 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.**

Um exemplo deste algoritmo se encontra no **Apêndice I**, exemplo 1.

### 3.2.2 Problemas de Designação

O problema de designação consiste em designar cada uma das origens a cada um dos destinos (medianas), de maneira ótima, formando desta maneira os *clusters* que serão atendidos por cada um dos ônibus do problema apresentado, ou seja, os pontos mais próximos entre si. Gillet e Johnson (1973, *apud* CORRÊA 2000, p. 26), propôs um algoritmo, que é conhecido como algoritmo de Gillet e Johnson.

### 3.2.2.1 Procedimentos Básicos do Algoritmo de Designação de Gillet e Johnson

Segue sua descrição. Sendo:

$t_i^1$ : Primeira mediana mais próxima;

$t_i^2$ : Segunda mediana mais próxima;

$c_i^1$ : Distância à primeira mediana mais próxima;

$c_i^2$ : Distância à segunda mediana mais próxima;

#### Passo 1

Calcular a distância entre cada nó ainda não designado até cada um dos depósitos fictícios “medianas” correspondentes que ainda possuem espaço.

#### Passo 2

Para cada nó  $i$ , obter  $t_i^1$  como sendo a mediana mais próxima de  $i$  e  $t_i^2$  como sendo a segunda mediana mais próxima de  $i$ , com distâncias iguais a  $c_i^1$  e  $c_i^2$ , respectivamente.

#### Passo 3

Para todos os nós  $i$ , 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, em uma lista.

#### Passo 4

Esta lista determina a ordem em que os nós serão designados às medianas, ou seja, os nós relativamente mais próximos a uma determinada mediana serão designados primeiro, até que sua capacidade esteja completa; quando isso ocorrer e houver ainda nós a serem designados, voltar ao passo 1.

O algoritmo prossegue até que todos os nós tenham sido designados.

#### Fim.

Para a aplicação neste trabalho foram feitas algumas modificações, mas especificamente no cálculo da razão, conforme aplicado em Corrêa, (2000).

### 3.2.2.2 Procedimentos Básicos do Algoritmo Gillet e Johnson Modificado

Segue sua formulação:

$t_i^1$ : Primeira mediana mais próxima;

$t_i^2$  : Segunda mediana mais próxima;

$c_i^1$  : Distância à primeira mediana mais próxima;

$c_i^2$  : Distância à segunda mediana mais próxima;

### **Passo 1**

Calcular a distância entre cada nó ainda não designado até cada um dos depósitos fictícios “medianas” correspondentes que ainda possuem espaço.

### **Passo 2**

Para cada nó  $i$ , obter  $t_i^1$  como sendo a mediana mais próxima de  $i$  e  $t_i^2$  como sendo a segunda mediana mais próxima de  $i$ , com distâncias iguais a  $c_i^1$  e  $c_i^2$ , respectivamente.

### **Passo 3**

Para todos os nós  $i$ , calcular a diferença  $d_i = c_i^2 - c_i^1$ .

Ordenar os nós  $i$  de acordo com os valores de  $d_i$ , em ordem decrescente, em uma lista.

### **Passo 4**

Esta lista determina a ordem em que os nós serão designados para as medianas, ou seja, os nós relativamente mais próximos a uma determinada mediana serão designados primeiros, até que sua capacidade esteja completa, quando isso ocorrer e houver ainda nós a serem designados voltar ao passo 1.

O algoritmo prossegue até que todos os nós tenham sido designados.

### **Fim.**

Um exemplo deste algoritmo se encontra no **Apêndice I**, exemplo 2.

## **3.2.3 O Problema de Roteamento de Veículos**

O interesse pelo roteamento decorre da combinação de dois fatores, conforme Cunha (2000): a importância que o mercado dá ao contexto logístico e a sua complexidade matemática que torna impossível a obtenção de soluções ótimas nos problemas reais impulsionando na busca de novas heurísticas e metaheurísticas que são de certa forma mais eficientes.

Um problema de natureza de características combinatória e de difícil solução denomina-se Problema de Roteamento de Veículos (PRV).

CUNHA (2000) se refere ao termo roteamento como um equivalente do inglês *routing* que designa o processo de determinação de um ou mais roteiros ou seqüências de paradas para

uma determinada frota de veículos, bem como na determinação da seqüência em que estes pontos de paradas serão visitados, estabelecendo assim rotas para os veículos.

O PRV tem como objetivos designar pontos de paradas a veículos, bem como a determinação da seqüência em que estes pontos de parada serão visitados, estabelecendo assim as rotas para os veículos. Assim, o objetivo geral destes problemas é minimizar os custos associados ao sistema de transporte considerado. O PRV consiste também em alguns objetivos específicos que, de forma geral, estão ligados às particularidades de cada problema, que podem ser: minimizar a quantidade de veículos; reduzir o tempo total do percurso das rotas; determinar as rotas de comprimento mínimo; entre outros. Portanto para a resolução do PRV várias características devem ser consideradas, tais como: quanto aos veículos (capacidades, autonomias, quantidades); quanto aos aspectos geométricos da rede (disposição dos nós e arcos/arestas do grafo); quanto a natureza das operações, entre outros.

### 3.2.3.1 Formulação Matemática para o Problema de Roteamento de Veículos (PRV)

Este problema, chamado de problema genérico de roteamento de veículos, tem a formulação exposta a seguir, devido a Golden *et al.* 1977, Bodin *et al.* 1983.

$$\text{Minimizar } \sum_{i=1}^n \sum_{j=1}^n \sum_{v=1}^V c_{ij} x_{ijv} \quad (\text{a})$$

Sujeito a:

$$\sum_{i=1}^n \sum_{v=1}^V x_{ijv} = 1 \text{ com } j = 2, \dots, n \quad (\text{b})$$

$$\sum_{j=1}^n \sum_{v=1}^V x_{ijv} = 1 \text{ com } i = 2, \dots, n \quad (\text{c})$$

$$\sum_{i=1}^n x_{ipv} - \sum_{j=1}^n x_{pvj} = 0 \text{ com } v = 1, \dots, V; p = 1, \dots, n \quad (\text{d})$$

$$\sum_{i=1}^n d_i \left( \sum_{j=1}^n x_{ijv} \right) \leq K_v \text{ com } v = 1, \dots, V \quad (\text{e})$$

$$\sum_{i=1}^n t_{iv} \sum_{j=1}^n x_{ijv} + \sum_{i=1}^n \sum_{j=1}^n t_{ijv} x_{ijv} \leq T_v \text{ com } v = 1, \dots, V \quad (\text{f})$$

$$\sum_{j=2}^n x_{1jv} \leq 1 \text{ com } v = 1, \dots, V \quad (\text{g})$$

$$\sum_{i=2}^n x_{i1v} \leq 1 \text{ com } v = 1, \dots, V \quad (\text{h})$$

$$X \in S$$

$$x_{ijv} = 0 \text{ ou } 1 \text{ para todos } i, j, v$$

onde:

$n$ : número de nós;

$V$ : número de veículos;

$K_v$ : capacidade do veículo  $v$ ;

$T_v$ : tempo máximo permitido para a rota do veículo  $v$ ;

$d_i$ : demanda do nó  $i$  ( $d_1=0$ , nó 1 = depósito);

$t_{iv}$ : tempo necessário do veículo  $v$  para entrega ou coleta no nó  $i$ ;

$t_{ijv}$ : tempo de viagem do veículo  $v$  a partir do nó  $i$  até o nó  $j$ ;

$c_{ij}$ : custo da viagem do nó  $i$  para o nó  $j$ ;

$x_{ijv}$ : 1 se o arco  $i-j$  é percorrido pelo veículo  $v$ ;

$x_{ijv}$ : 0 caso contrário.

Neste modelo matemático, tem-se que:

- A função objetivo (a) indica que a distância total deve ser minimizada;
- As restrições (b) e (c) asseguram que cada nó deve ser atendido por exatamente um veículo;
- A restrição (d) assegura a continuidade da rota;
- A restrição (e) é referente às capacidades dos veículos;
- A restrição (f) é quanto ao tempo da rota;
- As restrições (g) e (h) garantem que a disponibilidade dos veículos não é excedida;
- A restrição  $X \in S$  é a restrição para evitar a formação de sub-rotas.

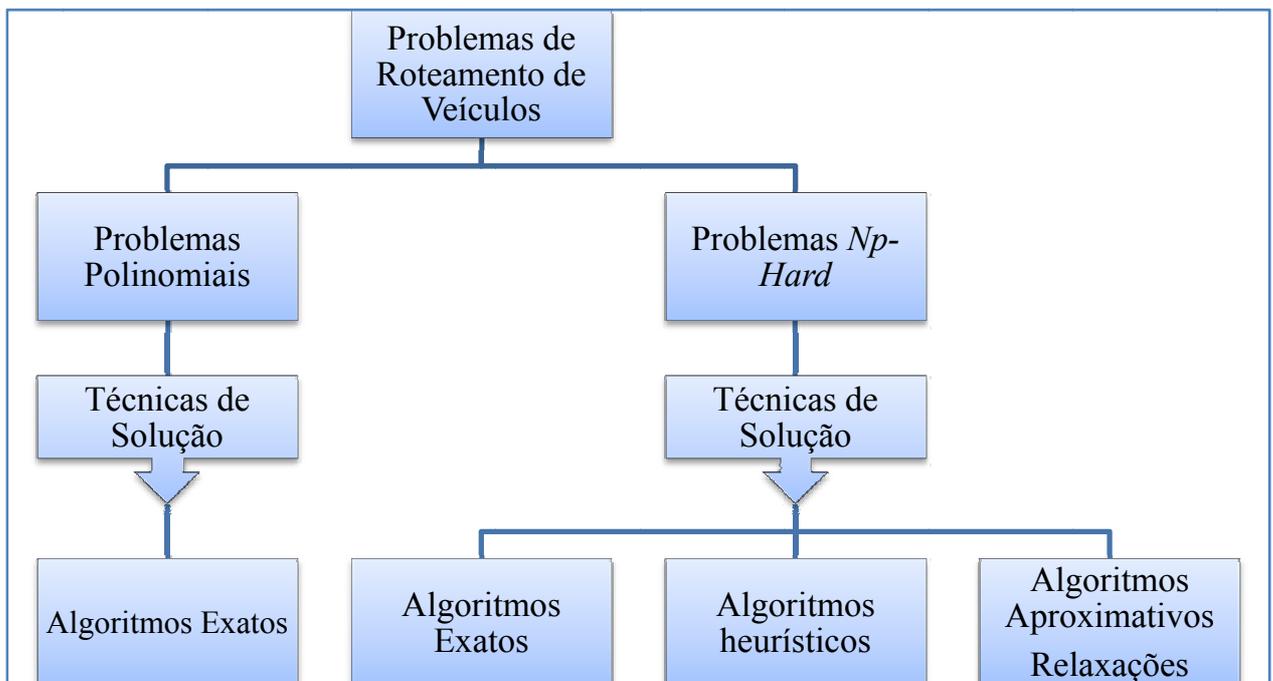
Como as restrições (b) e (d) implicam em (c), (d) e (g) implicam em (h), pode-se excluir do modelo as equações (c) e (g) que são redundantes. Assume-se que  $\max_{1 \leq i \leq n} d_i < \min_{1 \leq v \leq V} K_v$ , isto é, a demanda em cada nó não excede à capacidade total de qualquer veículo.

### 3.2.3.2 Planejamento de Soluções Para o PRV

De acordo com GOLDBARG (2000, *apud* SMIDERLE 2001), os problemas em que as variáveis assumem valores inteiros ou que possuem funções objetivo com descontinuidades, geralmente não podem ser solucionados através de um modelo matemático de Programação Linear Inteira (PLI) utilizando o algoritmo simplex.

Nestes casos se encaixam a maioria dos problemas de roteamento.

Para determinação da solução existem três métodos: os métodos exatos, os métodos aproximativos e os métodos heurísticos ou metaheurísticos, conforme esquematizado no Fluxograma 2.



**Fluxograma 2.** Estratagemas para solução de PRV

Os métodos exatos são aqueles que sempre encontram a melhor solução para o problema, ou seja, a solução ótima. De acordo com Cormen *et al.* (2002, *apud* TOSO *et al.* 2004, p. 17), um problema pode ou não possuir um algoritmo exato para sua solução. Se este algoritmo existir, o mesmo pode não encontrar um ótimo em tempo hábil, ou seja, o algoritmo pode levar décadas para encontrar a solução desejada.

Os métodos aproximativos encontram soluções para problemas exponenciais onde os métodos exatos não se aplicam computacionalmente. Os algoritmos de aproximação, por possuírem tempo polinomial, podem resolver problemas exponenciais com uma boa margem de

aproximação da solução ótima (CORMEN *et al.* 2002, *apud* TOSO *et al.* 2004, p. 37). Vários problemas clássicos da literatura possuem soluções aproximativas como alternativa a demanda computacional elevada que requer os métodos exatos, tais como: O Problema do Caixeiro Viajante (*Travelling Salesman Problem*), Problema de Cobertura de Vértices (*Vertex Cover Problem*), dentre outros.

Os métodos Heurísticos, conforme Toso *et al.* (2004), apresentam um conjunto novo de soluções para problemas de otimização combinatorial. Estes métodos têm origens distintas e sua utilização muitas das vezes é específico a um determinado problema combinatorial. Há na literatura o termo metaheurística, que possui grande abrangência e pode ser aplicada a maioria dos problemas de otimização combinatorial. Pode-se citar como exemplo de metaheurísticas a *Ant Colony Optimization (ACO – Otimização de Colônias de Formigas)*, *Genetic Algorithm (GA – Algoritmo Genético)*, *Simulated Annealing (AS)* e *Tabu Search (TS - Busca Tabu)*.

### 3.2.3.3 Principais Problemas de Roteamento

Tem-se na literatura que o primeiro problema de roteamento estudado foi o Problema do Caixeiro Viajante (PCV) (*Traveling Salesman Problem – TSP*), que está detalhado a seguir.

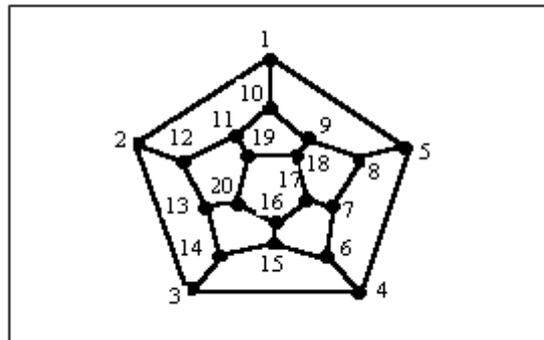
Além do Problema do Caixeiro Viajante, BODIN (1983) lista como principais problemas de roteamento de veículos, os seguintes:

- O Problema do Carteiro Chinês (PCC);
- O Problema do Carteiro Chinês Capacitado (PCCC);
- O Problema dos Múltiplos Caixeiros Viajantes (PCVM);
- O problema de Roteamento de Nós com um Único Depósito e Múltiplos Veículos (PRDMV);
- O Problema de Roteamento de Nós com Múltiplos Depósitos e Múltiplos Veículos (PRMDMV);
- O Problema de Roteamento de Nós com um Único Depósito e Múltiplos Veículos e Demanda Estocástica nos Vértices (PRDMVE).

#### 3.2.3.3.1 O Problema do Caixeiro Viajante (PCV)

De acordo com CUNHA (2002), o PCV tem sua origem creditada a William Rowan Hamilton (1805-1865), que inventou um jogo cujo objetivo era o de traçar um roteiro dos

vértices de um dodecaedro (vértices equivalem a cidades) que iniciasse e terminasse no mesmo vértice sem, contudo, repetir uma visita. O grafo do problema é mostrado na Figura 1.



**Figura 1.** Jogo de Hamilton.

O PCV consiste em estabelecer uma única rota que passe por cada nó de um grafo, uma e apenas uma vez, retornando ao nó inicial no final do percurso. Este roteiro Hamiltoniano deve ser feito de modo que a distância total percorrida seja mínima. A demanda de cada nó é determinística e o veículo tem a capacidade conhecida.

O PCV pode ser definido da seguinte forma:

Seja uma rede  $G(N, A, C)$  onde  $N$ = número de nós,  $A$ = número de arcos e  $C = [c_{ij}]$  a matriz de custos que é simétrica, isto é, considera-se  $c_{ij} = c_{ji}$ , assumindo ainda que  $c_{ii} = +\infty$ ,  $\forall i \in N$  e considerando a matriz  $X = [x_{ij}]$  das variáveis de decisão do problema, onde

$x_{ij} = \begin{cases} 1, & \text{se o arco } a_{ij} \in \text{rota} \\ 0, & \text{se o arco } a_{ij} \notin \text{rota} \end{cases}$ , tem-se a seguinte formulação de Programação Linear Inteira para

o problema, proposto por Golden *et al.* (1977) e Bodin *et al.* (1983):

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

Sujeito a:

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

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

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

$$x_{ij} = \begin{cases} 1, & \text{se o trajeto } (i, j) \text{ está no itinerário} \\ 0, & \text{c.c.} \end{cases} \quad (i, j = 1, 2, \dots, n)$$

As restrições (b) e (c) garantem que exatamente um arco  $(i, j)$  emana de cada nó  $i$  da rota e exatamente um arco  $(i, j)$  é direcionado para cada nó  $j$  da rota. A restrição (d) contém um conjunto  $S$  que pode ser qualquer conjunto de restrições que evita a formação de sub-rotas.

### 3.2.3.4 Classificação para o Problema de Roteamento

Segundo BODIN *et al.* (1983), os PRV são classificados em três grandes grupos: roteamento, escalonamento e roteamento e escalonamento.

Conforme NUNES (1998), um PRV consiste, basicamente, na designação de pontos de parada à veículos, bem como na determinação da seqüência em que estes pontos de parada serão visitados, estabelecendo as rotas para os veículos, levando em consideração aspectos espaciais. Será um problema de escalonamento se existir interesse em considerar o horário de visitação dos pontos de parada, isto é, se forem feitas considerações temporais. Mas como na maioria dos problemas, tanto os aspectos espaciais como temporais, são considerados, eles são considerados problemas de roteamento e escalonamento.

### 3.2.3.5 Variáveis Importantes para a Formulação do Problema de Roteamento

No processo de construção do sistema de roteamento, algumas decisões são extremamente relevantes, tais como:

- Número de rotas;
- Localização de centros de operações, garagens e centros de manutenção;
- Entre outras.

### 3.2.3.6 Desempenho Computacional

O PRV pertence à categoria NP-difícil (*NP-hard*), assim, o esforço computacional para a sua solução cresce exponencialmente com o tamanho do problema que é dado pelo número de pontos a serem atendidos. Deste modo, é praticamente impossível resolver problemas reais de grande porte, que pertencem a essa classe até a otimalidade.

Algoritmos heurísticos são métodos que não garantem a produção de soluções ótimas, mas tentam encontrar boas soluções (sub-ótimas) com um tempo de computacional razoável. Estes algoritmos não conseguem, necessariamente, definir se esta é a solução ótima, nem quão próxima ela está da solução ótima.

Recentemente, as metaheurísticas têm sido amplamente usadas, com resultados satisfatórios, na busca de soluções próximas do ótimo para soluções NP-*hard*. De acordo Reeves (1993), a principal vantagem das metaheurísticas é que elas são geralmente muito eficientes em procurar boas soluções através de espaços de soluções muito grandes, e cada classe de metaheurística tem uma estrutura estratégica e metódica que é independente do domínio do problema.

### 3.2.4 Otimização por Colônias de Formigas

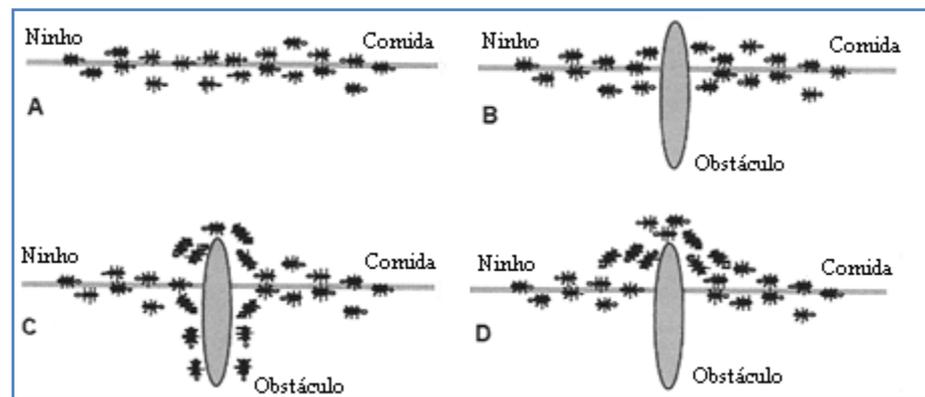
O primeiro trabalho publicado que utiliza a heurística de Otimização por Colônias de Formigas, conhecido por *Ant System*, foi aplicado por Dorigo (DORIGO e COLORNI, 1991) ao PCV.

DORIGO *et al.* (1999), fazem o *Ant System* tomar a forma de uma metaheurística, denominada *Ant Colony Optimization (ACO)*, ou Otimização por Colônias de Formigas.

Na metaheurística *ACO*, existe uma colônia de formigas artificiais que cooperam entre si com o objetivo de encontrar soluções satisfatórias para problemas de otimização discreta.

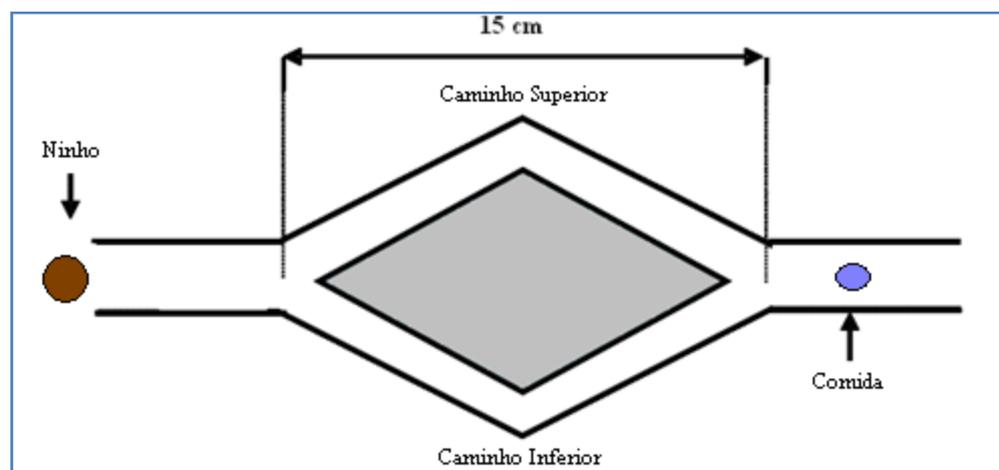
As idéias utilizadas na metaheurística *ACO* provêm do mundo real, ou seja, ocorrem realmente nas colônias de formigas reais. Os algoritmos das formigas foram inspirados pela observação de colônias reais, cujo comportamento é mais dirigido à sobrevivência da colônia do que à sobrevivência de um único componente da colônia.

Na Figura 2, a seguir, vê-se uma demonstração do que ocorre na heurística *ACO*. No momento A tem-se o ninho das formigas e o caminho livre até seu alimento; num momento B tem-se um obstáculo entre o ninho e a sua comida; no momento C observa-se como as formigas ultrapassam a barreira que as impede de chegar ao ninho, sendo que um maior número de formigas opta pelo caminho menor e, finalmente, num momento D, pode-se observar que todas as formigas fazem o mesmo trajeto e esse trajeto é o de menor distância entre a comida e seu ninho.

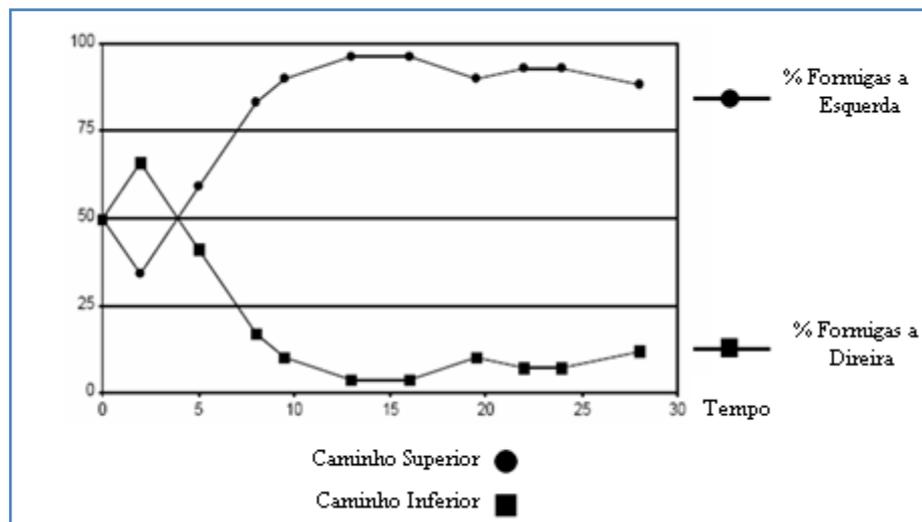


**Figura 2.** Comportamento das formigas após o surgimento de um obstáculo entre o ninho e uma fonte de alimento [Dorigo e Gambardella, 1996].

GOSS, 1990, estudou em circunstâncias controladas o comportamento das formigas da espécie *Linepithema humile*. O ninho dessa espécie de formiga e uma fonte de alimento foram separados por uma ponte simples com dois caminhos de mesmo comprimento conforme Figura 3. As formigas são deixadas livres para mover-se entre o ninho e a fonte de alimento, sendo observada a porcentagem das formigas que escolhem um dos caminhos com o decorrer do tempo. O resultado, Figura 4, mostra que após uma fase inicial de oscilações, as formigas tendem a convergir para um mesmo trajeto.



**Figura 3.** Experiência da ponte simples [Goss *et al.* (1990 *apud* DORIGO *et al.* 1999)].

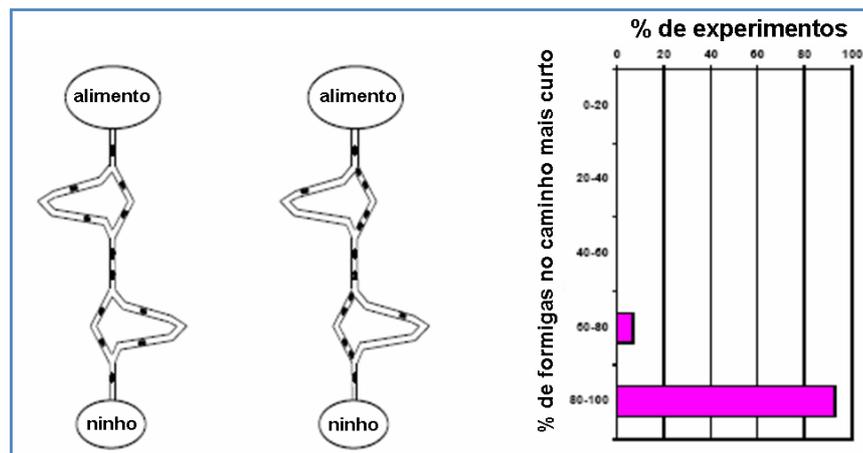


**Figura 4.** Mostra a porcentagem em relação a unidade de tempo, [Goss *et al.*, (1990 *apud* DORIGO *et al.*, 1999)].

Inicialmente no experimento da ponte simples não há feromônio em nenhum dos dois caminhos, que são selecionados pelas formigas com a mesma probabilidade, mas, após a fase inicial de transição ocorre que algumas formigas selecionam aleatoriamente um caminho; no exemplo anterior, o caminho superior. Isso ocorre porque as formigas depositam uma quantidade maior de feromônio no caminho superior, que estimula por sua vez mais formigas a escolherem este caminho, e assim por diante. No modelo para a ponte simples a probabilidade de escolher um caminho em algum cruzamento depende da quantidade total de feromônio ali depositado, o que, por sua vez, é proporcional ao número de formigas que usaram este caminho até o momento. Apesar dos caminhos serem iguais, há uma escolha probabilística por um dos caminhos.

Num segundo experimento conhecido como experimento da ponte dupla em que um dos caminhos é mais curto, acontece que as formigas escolhem o caminho mais curto, pois as primeiras formigas a chegarem ao alimento são aquelas que percorreram o caminho mais curto, de modo que, quando estas formigas começam o seu retorno, uma maior quantidade de feromônio é depositado no caminho estimulando novas formigas a escolherem o caminho mais curto.

A Figura 5 mostra o resultado do experimento da ponte dupla com comprimentos diferentes.



**Figura 5.** Experiência da ponte dupla [Goss *et al.* (1990 *apud* DORIGO *et al.* 1999)].

DORIGO *et al.* 1999, listam algumas similaridades e diferenças entre as formigas reais e as formigas artificiais.

As similaridades são:

- Tanto as formigas reais como as artificiais são compostas de uma população (colônia) que cooperam entre si de forma global para encontrar uma adequada solução para a tarefa em execução. Uma solução satisfatória só é possível graças à cooperação entre todos os agentes da colônia.
- Ambos os sistemas utilizam um fator de cooperação, sendo que, uma comunicação indireta, ou ainda, uma sinergia na busca de boas soluções. Essa sinergia ocorre através de uma substância chamada feromônio, que representa a informação coletiva. Em ambos os casos, ocorrem a evaporação do feromônio, o que causa o esquecimento do passado recente e traz as formigas a possibilidade de explorar novas soluções.
- As formigas artificiais e reais compartilham uma tarefa comum para encontrar um trajeto de caminho mais curto, ou seja, de custo mínimo, no caso da metaheurística. E do ninho ao alimento no caso das formigas reais.
- Tanto as formigas artificiais, como as reais, constróem as soluções que aplicam uma política probabilística de decisão para o movimento através dos estados adjacentes. Em ambos os sistemas não há como os agentes terem uma visão supra adjacente, ou seja, um agente não consegue visualizar um nó que se encontra a frente daqueles que estão em seu ambiente de visão atual.

As diferenças são:

- As formigas artificiais vivem em um mundo discreto e seus movimentos consistem em transições de estados discretos para estados discretos.
- As formigas artificiais têm uma memória ou estado interno, para que em suas ações não haja sobreposição dos movimentos.
- As formigas artificiais depositam uma quantidade de feromônio que é uma função da qualidade da solução encontrada.
- As formigas artificiais depositam o feromônio de forma temporal, enquanto as formigas reais depositam feromônio sob demanda.
- Para melhorar a eficiência do sistema total, os algoritmos de *ACO* podem ser enriquecidos com potencialidades extras, tais como:
  - ✓ *Backtracking* que são otimizações em caminhos previamente escolhidos com finalidade de se melhorar as rotas até então já encontradas;
  - ✓ *Lookahead* que consiste em analisar não somente o movimento para a próxima vizinhança como também a vizinhanças subseqüentes;
  - ✓ Otimizações locais, onde um agente ataca o problema localmente antes de decidir por qual caminho seguir. Estas otimizações locais recuperam uma seqüência de passos a seguir. Também existem implementações com otimizações locais por *backtracking*, onde o subcaminho já encontrado é melhorado.
  - ✓ entre outras;

#### 3.2.4.1 *Ant System (AS)*

*Ant System (AS)* foi o primeiro algoritmo *ACO*, antes mesmo de *ACO* atingir o estado de metaheurística em Dorigo *et al.* (1999). *Ant System* foi o protótipo de um número de algoritmos da formiga que encontram aplicações interessantes e bem sucedidas. A primeira aplicação do algoritmo da formiga foi feita usando o PCV.

Em *AS*, as soluções são geradas por formigas (agentes) que se movem de uma cidade para outra conforme o PCV. O algoritmo executa  $t_{\text{máx}}$  iterações onde  $t$  é o número da iteração vigente. Em cada iteração do algoritmo,  $m$  formigas constróem uma solução em  $n$  passos ( $n$  é o número de cidades do problema). Estando na cidade  $i$ , a formiga faz uma escolha probabilística da próxima cidade  $j$  e o arco  $(i, j)$  é adicionado à solução.

Três algoritmos *Ant System* foram desenvolvidos para o PCV (Colomi *et al.* 1992; Dorigo, 1992; Dorigo *et al.* 1996) diferenciando-se na forma de atualizar as trilhas de feromônios. Estes algoritmos são conhecidos como *ant-density* (formiga densidade), *ant-quantity* (formiga quantidade), e *ant-cycle* (formiga ciclo). Nos dois primeiros, o feromônio é depositado durante a construção da solução, já no terceiro, o feromônio é depositado após uma solução ter sido construída. O *ant-cycle* obteve o melhor resultado e passou a ser chamado de *AS*, enquanto os demais foram abandonados.

No algoritmo *AS* não existem ações extras (*daemon*) e a evaporação do feromônio acontece imediatamente antes que as formigas comecem a depositar o feromônio, e é intercalado com as atividades das formigas. O principal papel da evaporação do feromônio é evitar a estagnação, isto é, a situação em que todas as formigas terminam por fazer o mesmo caminho.

Em *AS* a quantidade de feromônio  $\tau_{ij}(t)$  depositado no arco  $(i, j)$  na iteração  $t$  representa a atratividade de escolher a cidade  $j$  após a cidade  $i$  (a atratividade de inserir o arco  $(i, j)$  na solução). A quantidade de feromônio depositada é função da qualidade da solução construída.

As formigas depositam uma quantidade de feromônio proporcional à qualidade das soluções que produzirem. Assim, quanto menor o caminho percorrido por uma formiga, maior será a quantidade de feromônio depositada nos arcos que usou durante o percurso do caminho. Isto ajuda dirigir a busca para soluções satisfatórias.

A memória ou estado interno de cada formiga guarda as cidades já visitadas e é chamada de lista tabu. A memória é usada para definir, para cada formiga  $k$ , o número de cidades que uma formiga que se encontra na cidade  $i$  ainda tem que visitar.

Explorando a memória, conseqüentemente, uma formiga  $k$  pode construir boas soluções (no PCV corresponde a visitar uma cidade exatamente uma vez). Também, a memória permite que a formiga cubra o mesmo trajeto para depositar o feromônio (*online delayed*) nos arcos visitados.

A tabela de decisão  $A_i = [a_{ij}(t)]_{|N_i|}$  do nó  $i$  é obtida pela composição dos valores locais de feromônio com os valores heurísticos locais:

$$a_{ij}(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta} \quad \forall j \in N_i \quad (a)$$

onde:

$\tau_{ij}(t)$  é a quantidade de feromônio no arco  $(i, j)$  na iteração  $t$ ;

$\eta_{ij} = \frac{1}{d_{ij}}$  é o valor heurístico,  $d_{ij}$  é o peso do arco  $(i, j)$ ;

$N_i$  é o conjunto de nós vizinhos ao nó  $i$ ;

$\alpha$  e  $\beta$  são os parâmetros que controlam o peso relativo entre o feromônio e a distância entre as cidades. Assim, um alto valor de  $\alpha$  significa que o valor do feromônio é muito importante e, deste modo, as formigas tendem a escolher caminhos pelos quais outras formigas já passaram. E se o valor de  $\beta$  for muito alto, a formiga tende a eleger a cidade mais próxima.

A probabilidade com que uma formiga  $k$  escolhe ir da cidade  $i$  para a cidade  $j$  na  $t$ -ésima iteração do algoritmo é:

$$p_{ij}^k(t) = \frac{a_{ij}(t)}{\sum_{l \in N_i^k} a_{il}(t)} \quad (b)$$

onde:

$N_i^k \subset N_i$  é o conjunto de nós vizinhos a  $i$  que ainda não foram visitados pela formiga  $k$ .

Após todas as formigas terminarem sua excursão, cada formiga  $k$  deposita um quantidade  $\Delta\tau_{ij}^k(t)$  de feromônio em cada arco que passou.

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{1}{L^k(t)} & , \text{ se } (i, j) \in T^k(t) \\ 0 & , \text{ se } (i, j) \notin T^k(t) \end{cases} \quad (c)$$

onde:

$T^k(t)$  é o *tour* feito pela formiga  $k$  na  $t$ -ésima iteração com comprimento  $L^k(t)$ .

Fica claro pela equação (c) que quanto mais curto foi o caminho percorrido pela formiga, maior é a quantidade de feromônio depositada nele.

A atualização do feromônio bem como sua evaporação é dada pela seguinte fórmula:

$$\tau_{ij}(t) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (d)$$

onde:

$$\Delta\tau_{ij}^k(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t), m \text{ é o número de formigas (mantidas constantes) em cada iteração}$$

e  $\rho \in (0, 1]$  é o coeficiente de evaporação.

Dorigo (1992) inicia  $\tau_{ij}(0)$  feromônio com um mesmo valor pequeno, constante e positivo  $\tau_0$ ; o número de formigas é igual ao número de nós  $m=n$ ;  $\alpha, \beta$  e  $\rho$  recebem respectivamente, os valores 1, 5 e  $\frac{1}{2}$  sendo que estes valores são encontrados experimentalmente.

Em Dorigo e Colormi (1991) e em Dorigo e Gambardella (1996), o AS foi comparado com outras heurísticas de uso geral em alguns problemas relativamente pequenos do PCV, problemas estes que variam de 30 e 75 cidades.

### 3.4.2.1.1 Procedimentos Básicos do Algoritmo AS

Os passos do algoritmo são os seguintes:

#### Passo 1

Iniciar

$t=0$

$NC=0$  {número de ciclos}

Para cada aresta  $(i, j)$

atribua um valor inicial  $\tau_{ij}(t)$  para a intensidade da trilha

$\Delta(\tau_{ij})=0$ .

Colocar  $m$  formigas em  $n$  nós.

#### Passo 2

$s=1$  {índice da lista tabu}

para  $k=1$  até  $m$  faça

Coloque a cidade inicial da  $k$ -ésima formiga em  $tabu_k(s)$ .

#### Passo 3

Repita até que a lista tabu esteja completa

$s = s + 1$

para  $k=1$  até  $m$  faça

Escolha a cidade  $j$  para mover com probabilidade  $p_{ij}^k(t)$

Mova a  $k$ -ésima formiga para a cidade  $j$ .

Coloque a cidade  $j$  em  $tabu_k(s)$ .

#### **Passo 4**

para  $k=1$  até  $m$  faça

Calcule o comprimento  $L^k$  do *tour* descrito por  $tabu_k(s)$

Atualize o menor *tour* encontrado

Para cada aresta  $(i, j)$

Para  $k=1$  até  $m$  faça

$$\Delta(\tau_{ij}) = \Delta(\tau_{ij}) + \Delta(\tau_{ij}^k)$$

#### **Passo 5**

Para cada aresta  $(i, j)$ , calcule

$$\tau_{ij}(t+n) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}$$

$$t=t+n$$

$$NC=NC+1$$

Para toda aresta  $(i, j)$ , faça  $\Delta(\tau_{ij})=0$

#### **Passo 6**

Se  $NC < N_{cmax}$

Esvazie as listas *tabu* e vá para o passo 2

Senão

imprima o menor *tour*, pare.

Um exemplo deste algoritmo se encontra no **Apêndice I**, exemplo 3.

### **3.2.4.2 A Metaheurística ACO**

A metaheurística *ACO* foi proposta por Marco Dorigo [Dorigo *et al.*, 1999], conforme já mencionado. Esta metaheurística é constituída por uma colônia de tamanho finito de formigas que procuram coletivamente por soluções ótimas. Cada formiga constrói uma solução ou uma

componente da solução, a partir de um estado inicial que é selecionado de acordo com critérios eminentes de cada problema.

Ao construir uma solução, cada formiga coleta informações nas características do problema e em seu próprio desempenho. Não existe uma comunicação direta entre as formigas – a troca de informações ocorre pela sinergia do meio onde as mesmas atuam. As formigas podem agir simultaneamente e independentemente, mostrando um comportamento cooperativo.

A solução para o problema abordado toma forma de caminho de custo mínimo entre os estados do problema, de acordo com suas restrições. A complexidade de cada formiga é tal que uma única formiga poderia encontrar a solução para o problema, mas, provavelmente, de qualidade ruim. As soluções de alta qualidade são encontradas somente como o resultado da cooperação global entre todos os agentes da colônia que constroem simultaneamente soluções diferentes.

Para a construção da solução, cada formiga constrói uma solução movendo-se sequencialmente de forma finita e probabilística a um melhor vizinho, noção de vizinhança que depende do problema. Os movimentos são selecionados de forma estocástica de busca dirigida por dois fatores: (a) memória interna de cada formiga e (b) pelo feromônio, que estão disponíveis através do sistema como um todo, mais informações locais conhecidas *a priori*, geralmente constituídas por diretivas relacionadas ao problema.

Cada formiga (agente) tem uma memória específica que está relacionada com o seu passado e é utilizada para a construção da solução final. Em alguns problemas de otimização combinatorial, os agentes podem usar suas memórias para detectar uma escolha mal feita, que estaria levando à solução atual a um estado inviável ou extremamente ruim.

As decisões sobre quando as formigas devem liberar o feromônio no ambiente e quanto feromônio deve ser depositado dependem das características do problema e do projeto de execução. As formigas podem liberar o feromônio ao construir a solução em (a) *step-by-step* (passo a passo, sob demanda) o feromônio é atualizado a cada passo na montagem de uma solução, (b) *online delayed* (atraso em linha, sob solução) onde o feromônio é atualizado a cada solução construída ou (c) ambos.

O fator autocatalítico denota um fator importante no funcionamento do algoritmo *ACO*, pois quanto mais agentes escolherem um determinado arco (trecho do caminho), mais feromônio será depositado no arco e mais interessante ele fica para as formigas seguintes. Em geral, a quantidade de feromônio depositada é feita proporcionalmente às boas soluções que uma formiga construiu. Desta maneira, se um arco contribuir para gerar uma solução satisfatória, sua

participação no sistema aumenta de forma proporcional à sua contribuição. Para o sistema não viciar em pontos de ótimos locais existe a evaporação eminente no ambiente e as variáveis específicas do problema influenciam o sistema.

Uma vez que a formiga realizou sua missão, ou seja, construiu uma solução e depositou feromônio, ela morre, ou seja, a formiga contribui para o sistema deixando sua memória para o grupo (feromônio), bem como a solução encontrada, caso esta seja a melhor solução encontrada até então.

A metaheurística *ACO*, além dos elementos principais (geração e atividade da formiga e evaporação do feromônio) compreende também alguns componentes extras que tem uma visão mais ampla, ou seja, no contexto global, em relação à visão local das formigas.

Segundo Dorigo *et al.* (1999), esses elementos ou ações recebem o nome de *daemon\_actions* (ações extras). Por exemplo, um *daemon* permite observar o comportamento das formigas e coletar as informações globais que sejam úteis, e, depositar as informações adicionais do feromônio, ou seja, um *daemon* consiste num agente global com função de otimizar o sistema a partir de sua visão ampla da atual configuração do mesmo.

Para a coordenação deste sistema, Dorigo *et al.* (1999) propuseram um sincronizador, denominado *schedule\_actives* (sincronizador de atividades), cuja função é ajustar as atividades de geração de novas formigas, depósito de feromônios e atuação de *daemon*. Tanto que para não haver distorções no sistema, o depósito do feromônio, tanto *online step-by-step* quanto *delayed* é uma atividade que necessita de exclusividade para o funcionamento do sistema.

A metaheurística *ACO* torna-se útil para problemas estocásticos com um número relativamente baixo de vizinhança. A metaheurística *ACO*, segundo Dorigo *et al.* (1999), tem um comportamento viável para problemas relacionados a redes de comunicação ou transportes. Mas, em contrapartida, para problemas com uma vizinhança muito grande não há uma eficiência nas soluções por dois motivos: (a) tempo elevado de busca por soluções e/ou (b) soluções não competitivas.

#### **3.2.4.2.1 Algoritmo para a Metaheurística *ACO***

A seguir é descrito o algoritmo *ACO* como proposto por Dorigo *et al.*, (1999):

1. procedimento *ACO\_Metaheurística* ()
2. enquanto (critério\_de\_parada\_não\_atingido)

```

3.   schedule_activities
4.   ants_generation_and_activity();
5.   pheromone_evaporation();
6.   daemon_actions ();      {opcional}
7.   fim_schedule_activities
8.   fim_enquanto
9. fim_procedimento

10. procedimento ants_generation_and_activity()
11. enquanto (houverem_recurso_disponiveis)
12.   agendar_a_criacao_de_uma_nova_formiga ();
13.   nova_formiga_ativa();
14.   fim_enquanto
15. fim_procedimento

16. procedimento nova_formiga_ativa ()
17.   inicializar_formiga ();
18.   M = atualizar_memoria_da_formiga ();
19.   enquanto (estado_atual ≠ estado_final)
20.     A = ler_tabela_rotas_locais();
21.     P = computar_probabilidades (A, M, inf_do_problema);
22.     próximo_estado = politica_decisao (P, inf_do_problema);
23.     mover_para (próximo_estado);
24.     se (atualizar_feromônio = online_step_by_step)
25.       depositar_feromônio_no_arco_visitado ()
26.       atualizar_tabela_de_rotas ();
27.     fim_se
28.     M = atualizar_estado_interno ();
29.   fim_enquanto
30.   se (atualizar_feromônio = online_delayed)
31.     avaliar_solucao ();
32.     depositar_feromônio_nos_arcos_da_solucao ()
33.   fim_se

```

- 34. morrer ();
- 35. fim\_procedimento

Como já descrito, alguns componentes e comportamentos são opcionais, como as atividades *daemon*, ou estritamente dependente da execução de como e quando o feromônio é depositado.

As ações *daemon* (linha 6) são opcionais e consultam às ações centralizadas e são executadas por um *daemon* que possui o conhecimento global.

As ações da linha 19 consultam a solução completa, ou uma componente de uma solução completa, construído pela formiga.

Em geral o feromônio *online\_step\_by\_step* e o feromônio *online\_dalayed* atualizam os componentes (respectivamente, as linhas 24-27 e 30-33 procedimento *nova\_formiga\_ativa*) são mutuamente exclusivos e somente em alguns casos, ambos estão presentes, quando ambos os componentes estão ausentes, o feromônio é depositado pelo *daemon*.

E, enfim, na linha 34 ocorre a morte da formiga.

## CAPÍTULO IV

### 4. Implementação dos Métodos Utilizados e Resultados

Este capítulo apresenta uma descrição da implementação da proposta do presente trabalho bem como de cada um dos métodos descritos no Capítulo III, começando pelo algoritmo das  $p$ -medianas de Teitz e Bart, seguido do algoritmo de Gillet e Johnson que faz as designações e, por fim, o algoritmo responsável pela criação das rotas o algoritmo *ACO*, algoritmo Otimização por Colônia de Formigas.

#### 4.1 Visão Geral da Metodologia Adotada

Conforme já mencionado no capítulo 2, a Usina Hidrelétrica de Itaipu tem hoje 1.227 funcionários, sendo que desses 860 utilizam com frequência o transporte oferecido. Os funcionários que utilizam o transporte estão distribuídos em 3 bairros, centro da cidade e suas intermediações. Portanto para a implementação dividiu-se a cidade em quatro partes, Vila C - parte I, Vila B - parte II, Vila A - parte III e Centro e suas intermediações – parte IV.

Para cada uma destas partes, primeiramente é feito o cadastramento das residências dos funcionários. Em seguida, é aplicado o algoritmo para determinar as medianas que representam os ônibus. Com a definição das medianas, é feita a designação dos pontos para cada mediana formando grupos de pontos. E, por fim, o roteamento para cada mediana com seus respectivos pontos a ela designados.

O problema de roteamento é definir a ordem em que os mesmos serão atendidos por cada um dos ônibus, de forma que o percurso total executado por todos os ônibus seja mínimo.

Portanto, o problema é dividido em duas partes:

- A formação dos grupos de pontos;
- E o roteamento (seqüenciamento) dos pontos dentro de cada um desses grupos.

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

A primeira etapa consistiu no cadastramento dos pontos (cada funcionário representa um ponto), fazendo uso do endereço de cada funcionário. Foi utilizado para esta tarefa o *software Google Earth*.

Na segunda etapa, utilizando o algoritmo das  $p$ -medianas de Teitz e Bart definem-se as medianas para as quatro partes da cidade anteriormente definidas.

Após a determinação das medianas foram feitas as designações e, para isso, foi implementado o algoritmo de Gillet e Johnson Modificado (seção 3.2.2.2).

Na quarta etapa foram construídos os roteiros e, para isto, foi implementada metaheurística *ACO* aplicado ao PCV.

A implementação computacional ocorreu através do ambiente computacional *MatLab* 6.5, em um microcomputador Pentium II – 2.26 GHz com 512MB de memória RAM (*Random Access Memory*).

## 4.2 Primeira Etapa

A primeira etapa é a do cadastramento dos pontos. Para as quatro regiões da cidade, previamente divididas, fez-se o cadastramento dos pontos.

Na primeira região (Vila C), o total de funcionários cadastrados é de 41. Na segunda região (Vila B), o total de funcionários cadastrados é de 66. Na terceira região (Vila A), o total de funcionários cadastrados é de 624. E na quarta região (Centro e intermediações), o total de funcionários cadastrados é de 129, conforme Quadro 5.

Vila	Região	Número de funcionários	Número de veículos
C	1 <sup>a</sup>	41	1
B	2 <sup>a</sup>	66	2
A	3 <sup>a</sup>	624	14
Centro	4 <sup>a</sup>	129	3
Total		860	20

**Quadro 5.** Regiões da cidade com suas respectivas demandas.

Como dispunha-se do endereço de cada funcionário utilizou-se um site localizador de endereços e também o *software Google Earth (freeware)* que fornece as coordenadas geográficas (x, y) para cada residência. Quando, por exemplo, as residências ficavam em uma rua sem saída, optou-se por colocar apenas um ponto, com as demandas acumuladas neste ponto. Portanto, cada ponto possui uma demanda, que deve ser considerada. Assim é formado o banco de dados

(**Apêndice II**), que contém as coordenadas geográficas, onde cada ponto indica um respectivo local de parada.

Para cada região ainda foi criada uma matriz de distâncias Euclidianas (a). O ideal seria utilizar a distância real, levando em conta o verdadeiro caminho percorrido, mas estes dados são difíceis de serem obtidos.

Equação para calcular a distância Euclidiana entre dois pontos:

$$d_{AB} = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2} \quad (a)$$

### 4.3 Segunda Etapa

Nesta etapa, para cada região determinaram-se as medianas de acordo com o número de ônibus a ser utilizado, através do algoritmo das  $p$ -medianas de Teitz e Bart, programado em ambiente computacional *Matlab*.

Na primeira região (Vila C), como o total de funcionários é de 41 localizados em 23 pontos distintos, é necessário determinar apenas uma mediana, pois será utilizado apenas um veículo para a coleta destes funcionários. Os resultados estão apresentados no Quadro 6 e na Figura 6. As coordenadas geográficas para cada ponto encontram-se no Apêndice II, Tabela 1.

Pontos	Mediana selecionada pelo algoritmo de Teitz e Bart
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23	2

**Quadro 6.** Mediana selecionada da primeira região (Vila C).



**Figura 6.** Mediana selecionada da primeira região (Vila C).

O ponto definido como mediana para a primeira região (Vila C) é o ponto 2; como pode - se observar, a mediana está bem distribuída em relação aos demais pontos de parada.

Na segunda região (Vila B), com o total de funcionários igual a 66, localizados em 18 pontos distintos, é necessário selecionar duas medianas, pois poderão ser utilizados dois veículos, sendo um ônibus com capacidade de 46 lugares e um micro-ônibus com capacidade de 26 lugares, ficando apenas seis lugares desocupados. Os resultados estão apresentados no Quadro 7 e na Figura 7. As coordenadas geográficas para cada ponto encontram-se no Apêndice II, Tabela 2.

Pontos	Medianas selecionadas pelo algoritmo de Teitz e Bart
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18	3 12

**Quadro 7.** Medianas selecionadas da segunda região (Vila B).



**Figura 7.** Medianas selecionadas da segunda região (Vila B).

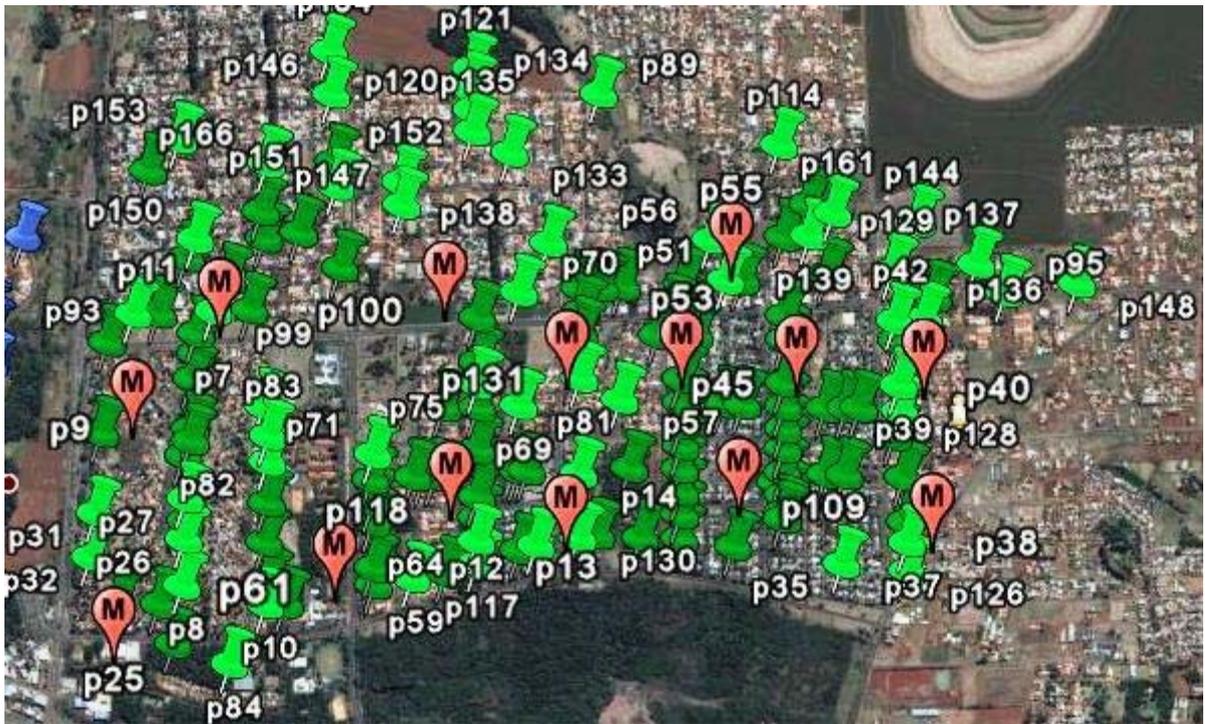
Os pontos selecionados como medianas para esta segunda região (Vila B), são os pontos 3 e 12; como pode-se observar, as medianas ficaram bem distribuídas em relação aos demais pontos de parada.

Na terceira região (Vila A), o total de funcionários é de 624 localizadas em 167 pontos; neste caso é necessário selecionar 14 medianas, pois serão utilizados 14 veículos sendo 13 ônibus com capacidade de 46 lugares e um micro-ônibus com capacidade de 26. Foram feitas quatro simulações, sendo que, a seguir, são apresentados os resultados de duas dessas simulações. Os resultados estão apresentados no Quadro 8. O melhor resultado foi obtido na segunda simulação, o qual é apresentado na Figura 8. As coordenadas geográficas para cada ponto encontram-se no Apêndice II, Tabela 3.

Pontos (simulação 1)	Mediana selecionada pelo algoritmo de Teitz e Bart
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	10 13 36 41 47 53 61 96 98 100 109 118 131 163
16 17 18 19 20 21 22 23 24 25 26	
27 28 29 30 31 32 33 34 35 36 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 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167	
Pontos (simulação 2)	Mediana selecionada pelo algoritmo de Teitz e Bart
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 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 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167	<b>9 11 13 25 38 40 45 53 55 61 100 109 118 131</b>

**Quadro 8.** Medianas selecionadas da terceira região (Vila A), simulações 1 e 2.



**Figura 8.** Medianas selecionadas da terceira região (Vila A), simulação 2.

Os pontos definidos como medianas para a terceira região (Vila A), são os pontos 9, 11, 13, 25, 38, 40, 45, 53, 55, 61, 100, 109, 118 e 131; como pode-se observar, as medianas ficaram bem distribuídas em relação aos demais pontos de parada.

O Quadro 9 apresenta a análise estatística para a primeira simulação e o Quadro 10 apresenta a análise estatística para a segunda simulação. Nesta análise estatística são mostradas a máxima distância, a mínima distância, a média das distâncias percorrida e o desvio padrão das distâncias, justificando assim a escolha pela segunda simulação.

Mínima Distância em metros	Máxima Distância em metros	Distância Média em metros	Desvio Padrão das Distâncias em metros
15,64	8774,88	1531,08	752,86

**Quadro 9.** Análise Estatística para a primeira simulação da terceira região (Vila A).

Mínima Distância em metros	Máxima Distância em metros	Distância Média em metros	Desvio Padrão das Distâncias em metros
15,64	7925,49	1165,60	734,40

**Quadro 10.** Análise Estatística para a segunda simulação da terceira região (Vila A).

Na quarta região (Centro), o total de funcionários é de 129 localizados em 73 pontos; neste caso, é necessário selecionar três medianas, pois serão utilizados três veículos com capacidade de 46 lugares. Os resultados estão apresentados no Quadro 11 e Figura 9. As coordenadas geográficas para cada ponto encontram-se no Apêndice II, Tabela 4.

Pontos	Medianas selecionadas pelo algoritmo de Teitz e Bart
1 2 3 4 5 6 7 8 9 10 11 12 13 14	27 31 46
15 16 17 18 19 20 21 22 23 24 25	
26 27 28 29 30 31 32 33 34 35 36	
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 64 65 66 67 68 69	
70 71 72 73	

**Quadro 11.** Medianas selecionadas da quarta região (Centro).



**Figura 9.** Medianas selecionadas da quarta região (Centro).

Os pontos selecionados como medianas para a quarta região (Centro) são os pontos 27, 31 e 46; como pode-se observar, as medianas ficaram bem distribuídas em relação aos demais pontos de parada.

Após a determinação das medianas é necessária a designação dos pontos para cada mediana, em cada uma das regiões.

#### 4.4 Terceira Etapa

Nesta etapa são feitas, para todas as regiões, as designações dos pontos para as medianas selecionadas anteriormente, respeitando a capacidade de cada mediana. Foi utilizado o algoritmo de GILLET e JOHNSON Modificado conforme apresentado no Capítulo III, que também foi programado no ambiente computacional *Matlab*.

Na primeira região (Vila C), como foi selecionada apenas uma mediana, não é necessário aplicar o algoritmo, pois todos os pontos serão designados para a mesma mediana. Os resultados estão apresentados no Quadro 12.

Mediana	Vértices designados para a mediana	Poltronas utilizadas
Vértice 2	1 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23	41

**Quadro 12.** Vértices designados para a primeira região (Vila C)

Na segunda região (Vila B), foram selecionadas duas medianas; assim, aplicando o algoritmo de designação de GILLET e JOHNSON Modificado. Foi determinada a designação de cada ponto a sua respectiva mediana, respeitando a capacidade de cada uma delas, pois uma das medianas tem capacidade 46 lugares e a outra de 26 lugares. Os resultados estão apresentados no Quadro 13.

Mediana	Vértices designados para a mediana	Poltronas utilizadas
Vértice 12	11 8 9 7 10	26
Vértice 3	2 1 17 13 4 5 16 15 18 14 6	40

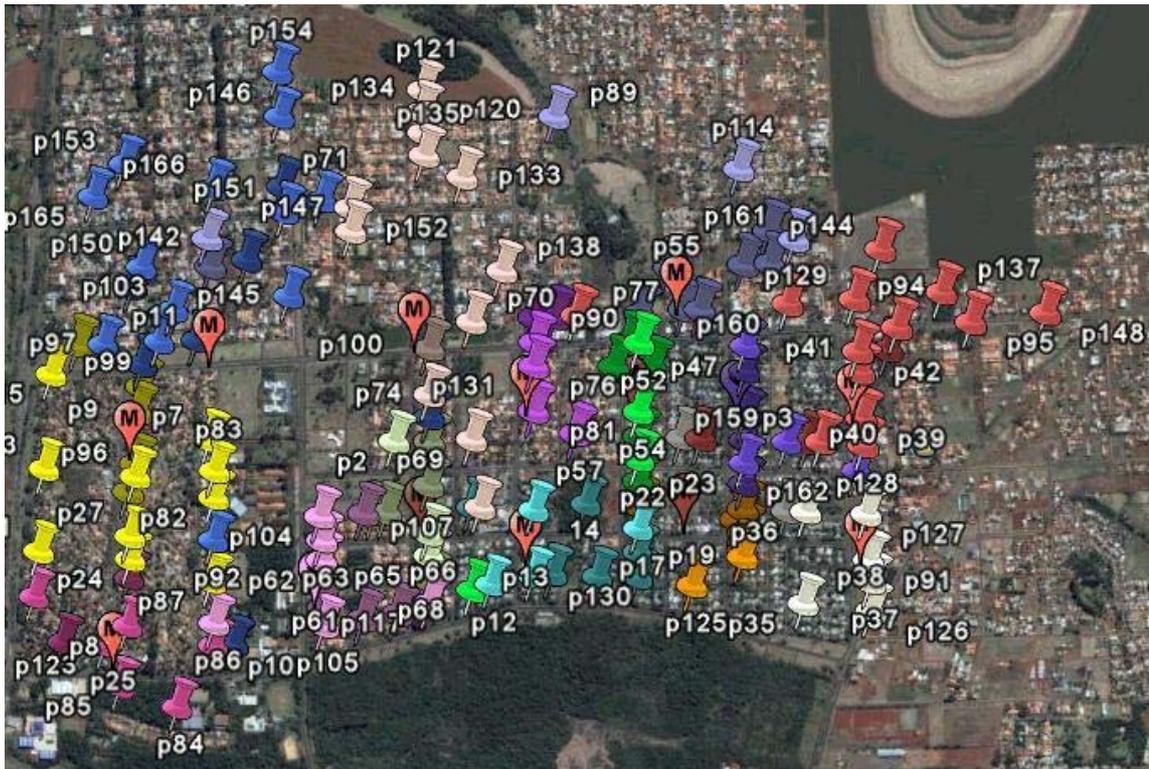
**Quadro 13.** Vértices designados para a segunda região (Vila B)

Na terceira região (Vila A), foram selecionadas 14 medianas, aplicando o algoritmo de designação de GILLET e JOHNSON Modificado. Foi determinada a designação de cada ponto para sua respectiva mediana, respeitando a capacidade de cada uma delas, pois treze das

mediana têm capacidade de 46 lugares e uma de 26 lugares. Os resultados estão apresentados no Quadro 14 e Figura 10.

Mediana	Vértices designados para a mediana	Poltronas utilizadas
Vértice 9	116 96 1 33 29 115 83 82 6 24 93 7 31 27 87	46
Vértice 11	149 166 98 103 167 153 150 165 143 145 99 151 141 146 154 28 92 97 30 71 75	46
Vértice 13	124 130 14 17 15 108 12 16 67 155	46
Vértice 25	85 123 8 122 84 32 26 10	26
Vértice 38	37 91 126 127 35 19 36 34 50 75	46
Vértice 40	95 39 136 41 137 94 148 42 164 144 163 129 128 3 77 160 43 49	46
Vértice 45	44 46 162 47 112 21 159 23 49	46
Vértice 53	54 5 4 157 12 158 52 22 68 69 117	46
Vértice 55	56 161 114 51 48 113 140 142 139 89 102 88 77	46
Vértice 61	60 105 62 59 63 104 10 79 86 58 117	46
Vértice 100	135 73 120 134 121 133 147 152 70 74 138 107 69	46
Vértice 109	125 78 18 111 110 156 19	46
Vértice 118	66 72 65 80 119 2 64 106 67	46
Vértice 131	81 76 90 132 101 57 107	46

**Quadro 14.** Vértices designados para a terceira região (Vila A)

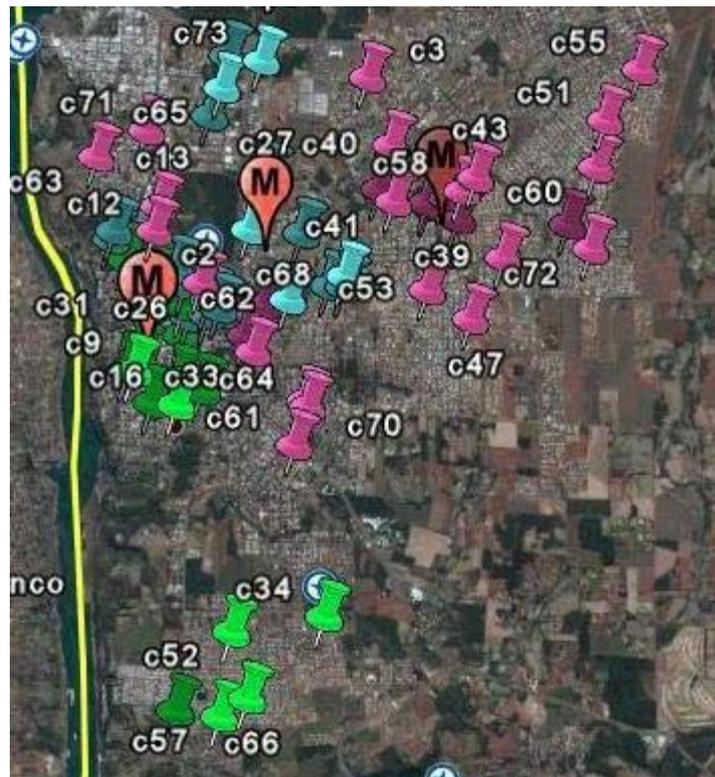


**Figura 10.** Vértices designados para a terceira região (Vila A)

Na quarta região (Centro), foram selecionadas três medianas, aplicando o algoritmo de designação de GILLET e JOHNSON Modificado. Foi determinada a designação de cada ponto para sua respectiva mediana, respeitando a capacidade de cada uma delas, ou seja, 46 lugares para cada mediana. Os resultados estão apresentados no Quadro 15 e Figura 11.

Mediana	Vértices designados para a mediana	Poltronas utilizadas
Vértice 27	69 48 2 65 36 67 73 50 24 8 22 28 20 38 62 10 49 23 68 61 5	46
Vértice 31	14 15 4 9 30 6 17 16 18 32 21 57 35 11 52 66 25 19 37 29 34	46
Vértice 46	60 58 43 51 56 55 44 72 45 39 54 42 47 53 40 41 5 71 7 26	37

**Quadro 15.** Vértices designados para a quarta região (Centro)



**Figura 11.** Vértices designados para a quarta região (Centro)

#### 4.4 Quarta Etapa

Nesta etapa são feitos para todas as quatro regiões e seus respectivos 20 grupos de pontos (1; 2; 14; 3), os roteamentos, ou seja, a definição da ordem em que os veículos farão a coleta dos funcionários de modo que as distâncias percorridas sejam as menores possíveis. Vamos considerar para isto que todos os veículos partam do mesmo local, a própria usina, sendo este local o “ponto zero”. Foi utilizado o algoritmo *ACO* proposto por Dorigo *et al.* (1999), programado na linguagem *Matlab*.

Para todos os 20 grupos foram utilizadas os parâmetros apresentados no Quadro 16.

Parâmetros	
$\alpha$	0,1
$\beta$	2
$\rho$	0,8
Número de formigas ( $m$ )	Igual ao número de nós ( $n$ )
Número de iterações para critério de parada	100

**Quadro 16.** Parâmetros utilizados pelo algoritmo ACO.

Onde:

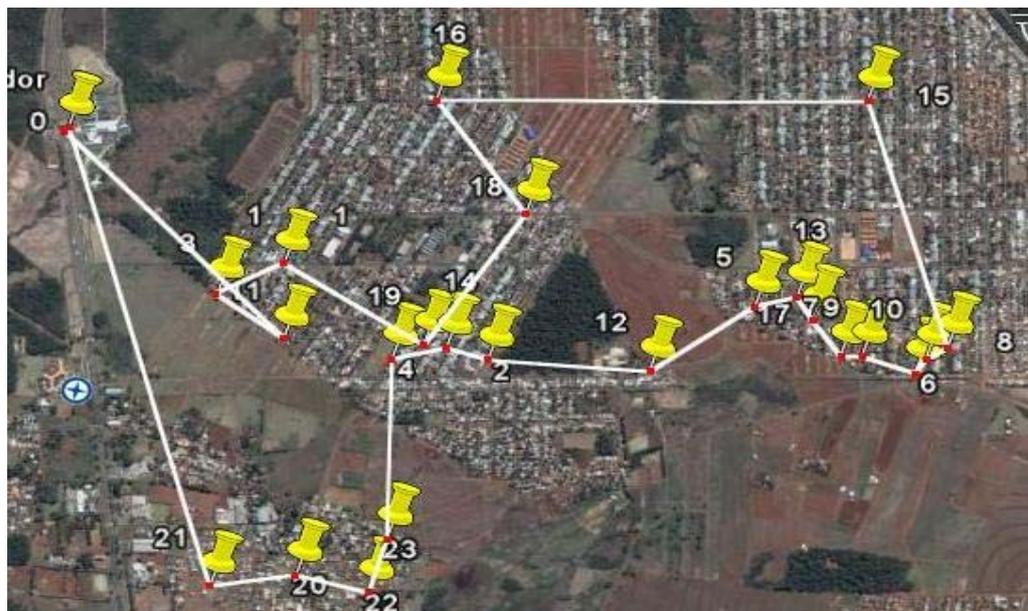
- $\alpha$  e  $\beta$  são os parâmetros que controlam o peso relativo entre o feromônio e a distância entre as cidades. Assim, um alto valor de  $\alpha$  significa que o valor do feromônio é muito importante e, deste modo, as formigas tendem a escolher caminhos pelos quais outras formigas já passaram. E se o valor de  $\beta$  for muito alto, a formiga tende a eleger a cidade mais próxima.
- $\rho \in (0, 1]$  é o coeficiente de evaporação.
- $m$  é o número de formigas e  $n$  é o número de nós.

O feromônio para todos os grupos inicia-se com valor igual a um e os valores para estes parâmetros são encontrados experimentalmente.

Para a primeira região (Vila C), com apenas uma mediana, o roteamento com a menor distância está apresentado no Quadro 17 e Figura 12, a seguir.

	Seqüência obtida pelo ACO	Comprimento da rota em metros
Rota 1	0 21 20 22 23 4 14 2 12 5 13 17 7 9 6 10 8 15 16 18 19 1 3 11 0	9.156,13

**Quadro 17.** Rota obtida pelo ACO para a primeira região (Vila C).



**Figura 12.** Rota obtida pelo ACO para a primeira região (Vila C).

A menor rota encontrada para a primeira região (Vila C) perfaz um total de 9.156,13 metros; considerando que tanto para coleta dos funcionários como para a entrega é feito o mesmo trajeto, e no total são realizadas quatro viagens, o ônibus deverá percorrer por dia 36.624,52 metros.

No Quadro 18, a seguir, é apresentada uma análise para a primeira região (Vila C) considerando as simulações realizadas até a obtenção do melhor resultado. A simulação 4 apresenta a mesma distância para o roteamento dos pontos que a simulação 3, mas com um tempo computacional maior. Logo, a simulação três foi a escolhida.

Dados	Simulação 1	Simulação 2	Simulação 3	Simulação 4
$\alpha$	0,1	0,1	0,1	0,1
$\beta$	2	2	2	2
$\rho$	0,8	0,8	0,8	0,8
Número de pontos	24	24	24	24
Número de formigas	10	10	24	24
Número de iterações	100	150	100	150
Tempo computacional em segundos	379	593	896	1172
Distância em metros	9606,93	9561,94	9156,13	9156,13

**Quadro 18.** Simulações do algoritmo *ACO* para a primeira região (Vila C)

Os resultados apresentados a seguir são os melhores resultados encontrados pelo algoritmo *ACO*.

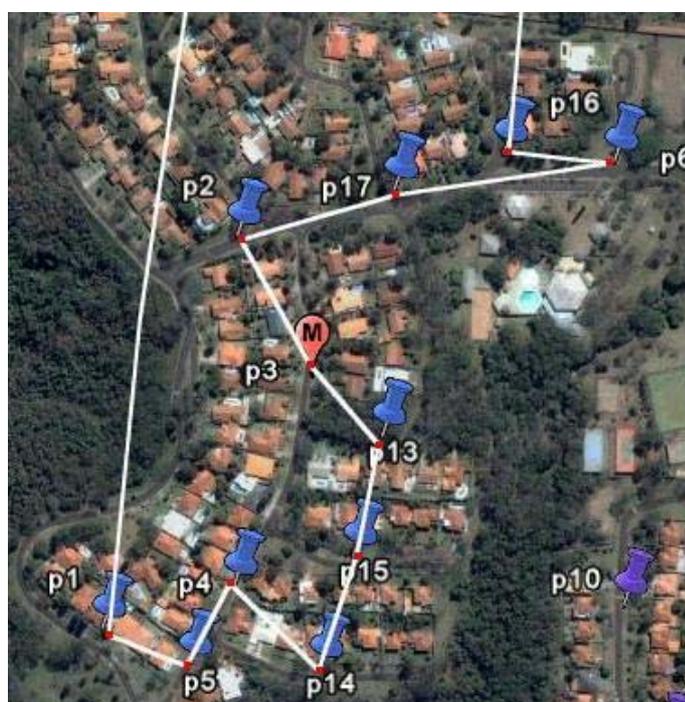
Para a segunda região (Vila B), com duas medianas (capacidades 46 e 26), o roteamento com a menor distância para cada uma das rotas está apresentado no Quadro 19 e Figuras 13 e 14.

	Seqüência obtida pelo <i>ACO</i>	Comprimento da rota em metros
Rota 1	0 7 12 11 8 9 10 0	11.124,51
Rota 2	0 18 16 6 17 2 3 13 15 14 4 5 1 0	11.399,61

**Quadro 19.** Rotas obtidas pelo *ACO* para a segunda região (Vila B).



**Figura 13.** Rota 1 obtida pelo ACO para a segunda região (Vila B).



**Figura 14.** Rota 2 obtida pelo ACO para a segunda região (Vila B).

As menores rotas encontradas para a segunda região (Vila B) é de 11.124,51 metros e 11.399,61 metros respectivamente; considerando que tanto para coleta dos funcionários como para a entrega é feito o mesmo trajeto e, no total são realizadas quatro viagens, os ônibus deverão

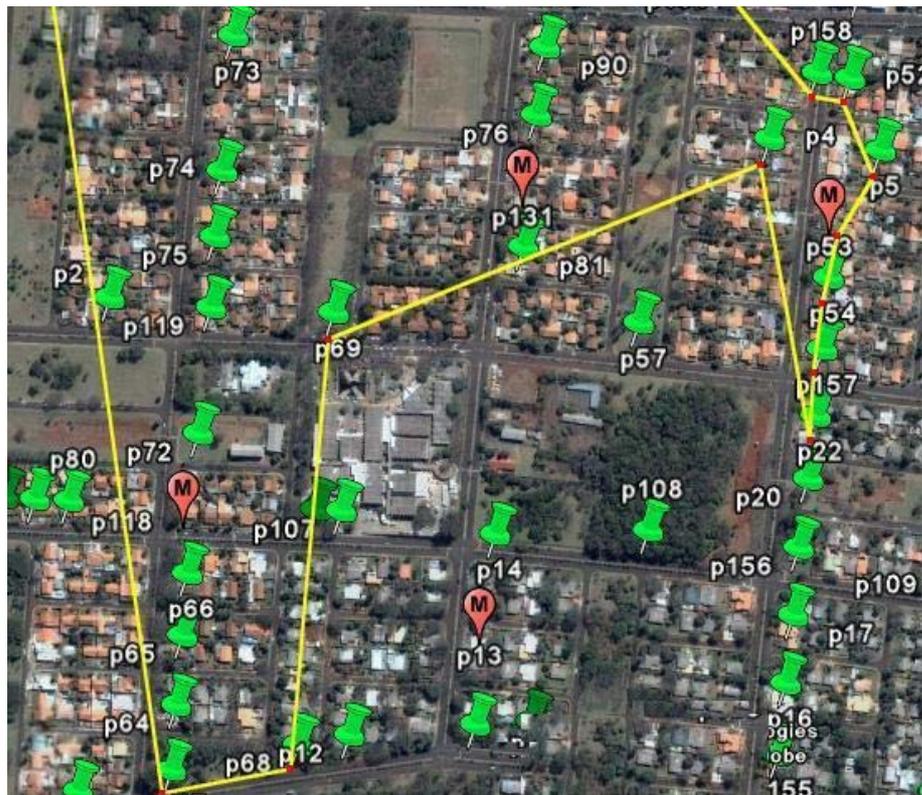
percorrer por dia 44.498,04 metros e 45.598,44 metros respectivamente, perfazendo um total de 90.096,48 metros dia.

Para a terceira região (Vila A), com 14 medianas (treze com capacidade de 46 lugares e uma de 26 lugares), o roteamento com a menor distância para cada uma das rotas está apresentado no Quadro 20, e a rotas 3, 4, 8, 12 e 14 nas Figuras 15, 16, 17, 18 e 19, respectivamente.

Rotas	Seqüência obtida pelo <i>ACO</i>	Comprimento da rota em metros
Rota 1	0 93 115 6 29 9 116 96 1 27 24 87 82 83 7 33 31 0	13908,28
Rota 2	0 28 92 71 75 145 149 141 166 151 165 150 143 103 99 30 97 98 11 146 154 167 0	16136,74
Rota 3	0 13 14 108 17 155 15 124 130 12 63 0	13650,61
Rota 4	0 10 84 85 25 123 8 26 122 32 0	14645,05
Rota 5	0 35 126 37 91 38 127 36 34 19 50 75 0	14712,04
Rota 6	0 163 164 148 95 137 94 136 42 41 40 39 128 3 43 49 160 129 144 77 0	22478,43
Rota 7	0 49 112 23 21 159 44 45 46 47 162 0	13014,39
Rota 8	0 158 52 5 53 54 157 22 20 4 69 68 117 0	14044,97
Rota 9	0 89 161 114 140 142 139 113 48 51 55 56 102 77 88 0	12374,27
Rota 10	0 104 63 62 61 60 105 59 79 117 58 86 10 0	14089,69
Rota 11	0 107 69 74 73 100 70 138 133 135 120 134 121 152 147 0	13397,46
Rota 12	0 125 19 110 18 78 111 109 156 0	13455,8
Rota 13	0 2 119 72 118 66 65 64 80 106 67 0	13056,26
Rota 14	0 132 101 90 76 131 81 57 107 0	12569,98

**Quadro 20.** Rotas obtidas pelo *ACO* para a terceira região (Vila A).





**Figura 17.** Rota 8 obtida pelo ACO para a terceira região (Vila A).



**Figura 18.** Rota 12 obtida pelo algoritmo da formiga para a terceira região (Vila A).



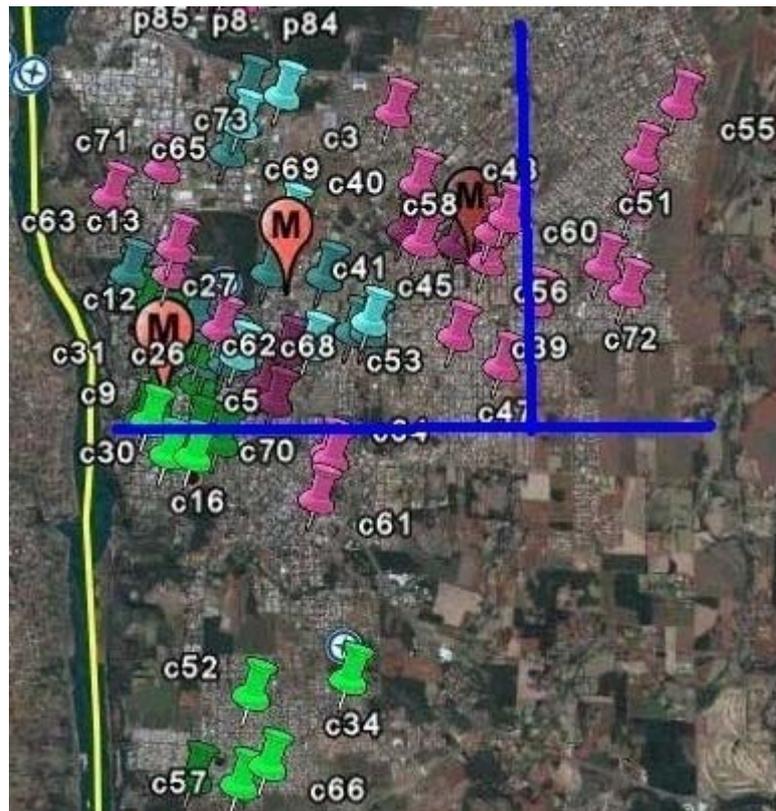
**Figura 19.** Rota 14 obtida pelo *ACO* para a terceira região (Vila A).

As menores rotas encontradas para a terceira região (Vila A), perfazem um total 201.533,97 metros; considerando que tanto para coleta dos funcionários como para a entrega é feito o mesmo trajeto e, no total são realizadas quatro viagens, os ônibus deverão percorrer por dia 806.135,88 metros.

Para a quarta região (Centro), com três medianas (cada uma delas com 46 lugares), o roteamento com a menor distância para cada uma das rotas está apresentado no Quadro 21 e a rota 3 na Figura 20.

	Seqüência obtida pelo <i>ACO</i>	Comprimento da rota em metros
Rota 1	0 67 65 36 69 27 2 48 24 62 50 49 68 28 5 23 20 10 8 38 61 0	33586,16
Rota 2	0 32 57 66 52 16 17 21 30 9 4 15 14 31 6 1 25 35 37 29 19 18 24 0	35374,41





**Figura 21.** Pontos que o roteamento atual da região central não atende.

Os pontos que estão abaixo e a direita da linha azul não são atendidos pela roteamento utilizado atualmente.

#### 4.5 Análise dos Resultados

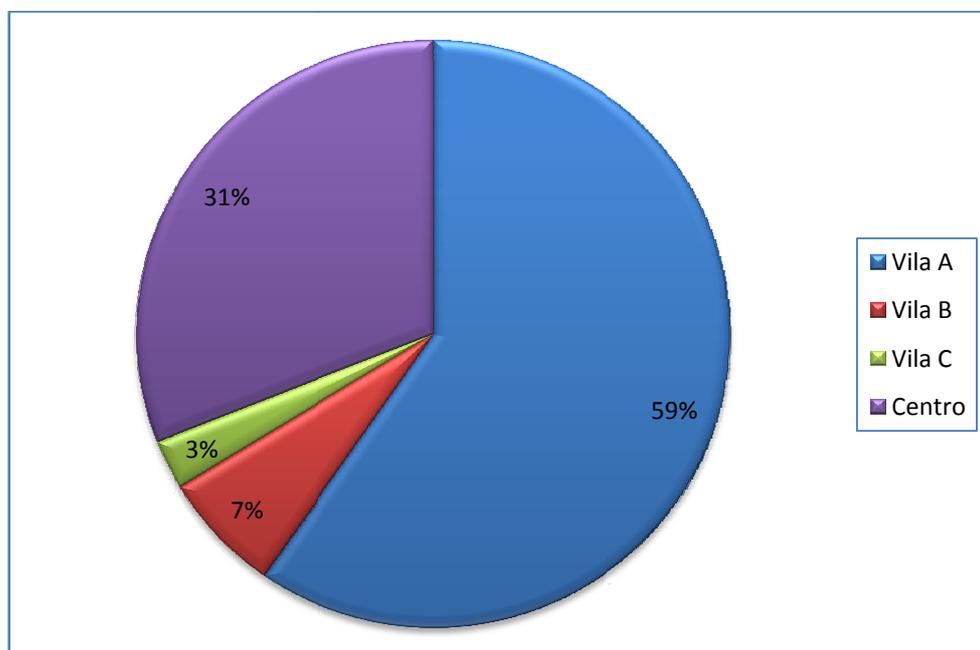
A frota é constituída de 19 ônibus, com 46 poltronas cada e 2 micro-ônibus com 26 lugares cada, sendo disponibilizadas 926 poltronas para os 860 usuários do transporte. A distância percorrida considerando todos os ônibus é de cerca de 1.520 km diários para coleta e entrega dos funcionários. Conforme Quadro 22.

Número Veículos	Assentos disponíveis	Usuários	Assentos remanescentes	Distância total Percorrida em km
21	926	860	66	1520

**Quadro 22.** Número de veículos x Assentos disponíveis.

O sistema atual de transporte da Itaipu foi definido por um roteamento baseado especificamente na experiência pessoal dos funcionários, sem critério de otimização matemático.

O Gráfico 2, apresentado a seguir, mostra a relação de quilometragem após a aplicação dos métodos abordados neste trabalho.



**Gráfico 2.** km por regiões da cidade após roteamento

O Gráfico 2, confrontado com o Gráfico 1, apresentado na seção 2.2.4, apresenta os dados apresentados no Quadro 23.

	Solução Atual (km)	Solução Otimizada (km)
Vila A	1026	806
Vila B	151	90
Vila C	58	37
Centro	285	419*
Total	1520	1352

\* atendendo 17 pontos adicionais

**Quadro 23.** Gráfico 1 *versus* Gráfico 2.

O Quadro 24 mostra a comparação entre a situação atual e os resultados obtidos neste trabalho.

	Pontos de parada	Quantidade de veículos	Quilometragem Percorrida
Situação atual da Empresa	Pontos de ônibus pré-determinados	19 ônibus e 2 micro ônibus	1520 km
Situação Otimizada	O próprio endereço dos funcionários	18 ônibus e 2 micro ônibus	1352 km

**Quadro 24.** Comparação entre a situação atual e a solução otimizada.

Fica evidente que há uma redução do número de veículos e da quilometragem para a situação otimizada. Deve-se atentar, no entanto, de que na situação otimizada não se leva em consideração a malha viária.

## CAPÍTULO V

### 5. Conclusões e Sugestões para Trabalhos Futuros

Este capítulo tem por finalidade fornecer as conclusões finais do presente trabalho e apresenta sugestões para futuros trabalhos que possam dar continuidade ao assunto aqui tratado.

#### 5.1 Conclusões

O objetivo deste trabalho foi o estudo, e a aplicação das heurísticas de  $p$ -medianas de Teitz e Bart (1968) e Gillet e Johnson Modificado, conforme Corrêa (2000) e da metaheurística *Ant Colony Optimization*, conforme Dorigo (1999), da área de Pesquisa Operacional, para a construção de rotas para os veículos que fazem o transporte de funcionários da empresa Itaipu Binacional comparando os resultados obtidos com a solução adotada atualmente pela empresa.

Foi aplicado o algoritmo de Teitz e Bart para definição das  $p$ -medianas; em seguida, o algoritmo de designação de Gillet e Johnson Modificado para a formação dos agrupamentos, e, por fim, o algoritmo *Ant Colony Optimization (ACO)* (Otimização por Colônias de Formigas) para a construção das rotas em cada agrupamento. Todos os algoritmos foram programados no ambiente computacional *Matlab*.

Como mencionado no Capítulo II, a Usina Hidrelétrica de Itaipu comporta, atualmente, 1.227 funcionários sendo que desses, 860 utilizam com frequência o transporte oferecido pela Itaipu, o qual é terceirizado.

A frota é constituída de 19 ônibus, com 46 poltronas cada e dois micro ônibus com 26 lugares cada, sendo disponibilizadas 926 poltronas para os 860 usuários do transporte. A distância percorrida considerando todos os ônibus é de cerca de 1.520 km diários para coleta e entrega de funcionário.

O sistema de transporte da Itaipu foi definido por um roteamento baseado especificamente na experiência pessoal dos funcionários, sem critério de otimização matemático.

Os funcionários que utilizam o transporte estão distribuídos em 3 bairros e no centro da cidade e suas intermediações. Para a implementação da metodologia proposta, dividiu-se a

cidade em quatro regiões, Vila C - parte I, Vila B - parte II, vila C - parte III e centro e suas intermediações – parte IV.

A metodologia, composta das técnicas apresentadas no Capítulo III foram aplicadas em quatro etapas, apresentados no Capítulo IV.

A Etapa I consistiu no cadastramento dos pontos (cada funcionário representa um ponto), através do endereço de cada funcionário. Foi utilizado para esta tarefa o *software Google Earth*.

Na segunda Etapa, utilizando o algoritmo das *p*-medianas de Teitz e Bart definiu-se as medianas para cada região.

Na Etapa III foi implementado o algoritmo de Gillet e Johnson Modificado, para a designação dos pontos de parada às medianas.

E, finalmente, na quarta Etapa, para cada um dos agrupamentos construiu-se os roteiros e para isto, foi implementada a metaheurística *ACO*, aplicado ao PCV.

A implementação computacional ocorreu através da linguagem *Matlab*, em um microcomputador Pentium II 2.26 GHZ com 512MB de memória RAM.

Para a parte I da cidade (Vila C), foi necessária a determinação de apenas uma mediana devido ao número de pontos cadastrados, ou seja, a utilização de um ônibus. A menor rota encontrada perfaz um total de 9.156,13 metros em cada viagem; como no total são quatro viagens dá um total de 36.624,52 metros dia.

Para a parte II da cidade (Vila B), foi necessária a determinação de duas medianas, o que significa a utilização de dois ônibus; neste caso como a demanda não completa dois ônibus foi possível a utilização de um ônibus e um micro ônibus. Neste caso foram encontradas duas rotas que perfazem um total de 22.524,12 metros em cada viagem; como no total são quatro viagens dá um total de 90.096,48 metros dia.

Na parte III da cidade (Vila A) foram encontradas 14 medianas, pois serão utilizados 14 veículos sendo 13 ônibus com capacidade de 46 lugares e um micro ônibus com capacidade de 26 lugares. Portanto, as 14 menores rotas perfazem um total de 201.533,97 metros, como no total são quatro viagens dia o total de quilometragem dia é de 806.135,88 metros.

Para a parte IV da cidade (Centro) foram encontradas 3 medianas, pois serão utilizados 3 ônibus de 46 lugares. Neste caso, as três menores rotas encontradas perfazem um total de 104.897,85 metros; como no total são quatro viagens dia o total de quilometragem dia é de 419.591,4 metros.

A aplicação dos referidos algoritmos mostra como é possível a minimização de custos, pois a uma redução na quilometragem e, também, na utilização de um veículo, apesar da

comparação não ser viável por se tratarem de problemas diferentes, já que na solução otimizada não se leva em conta a malha viária.

## 5.2 Sugestões para Trabalhos Futuros

Objetivando um aprimoramento dos resultados obtidos, são propostas algumas sugestões para trabalhos futuros:

- a) Utilizar as distâncias reais, considerando os sentidos da malha viária;
- b) Pesquisar e aplicar metaheurística *ACO* para o problema de localização de facilidades;
- c) Utilizar outros algoritmos para o roteamento com o objetivo de comparar com a metaheurística *ACO*.

## REFERÊNCIAS BIBLIOGRÁFICAS

ANDRADE C. E. A. TOSO, F.R., BATISTA, F.L.N. **Modelos de otimização para transportes de cargas em ambientes reduzidos.** Dissertação de Mestrado, UFLA, Lavras, Ciência da Computação, 2004.

BODIN, L.D.; GOLDEN B.; ASSAD A.; BALL M. **Routing and scheduling of vehicles and crews: the state of the art.** Computers and Operations Research, 1983.

BASIR, O.; CALAMAI, P.; ELLABIB, I. **Exchange strategies for multiple ant colony system.** Elsevier Science Inc. New York, NY, USA, 2007.

BELL, J. E.; MCMULLEN P. R. **Ant colony optimization techniques for the vehicle routing problem.** Elsevier Science Inc. New York, NY, USA, 2004.

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

COLOMBO, J. A. A. **Localização e roteamento para serviços de atendimento emergencial – O caso da segurança eletrônica.** Dissertação de Mestrado, UFPR, Curitiba, Programa de Pós-Graduação em Métodos Numéricos em Engenharia, 2001.

COLORNI, A. DORIGO, M. MANIEZZO, V. **Distributed optimization by ant colonies.** In F. J. Varela & P. Bourgina (Eds.), Proceedings of the First European Conference on Artificial Life (pp. 134–142). Cambridge, 1992.

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. **Algoritmos: teoria e prática.** Rio de Janeiro, 2002.

CORRÊA, E. S. **Algoritmos genéticos e busca tabu aplicados ao problema das p - medianas.** Dissertação de Mestrado, UFPR, Curitiba, Programa de Pós-Graduação em Métodos Numéricos em Engenharia, 2000.

COSTA, D. M. B. **Aplicação de algumas técnicas da pesquisa operacional na otimização de serviços postais.** Dissertação de Mestrado, UFPR, Curitiba, Programa de Pós-Graduação em Métodos Numéricos em Engenharia, 1997.

CUNHA, C. B. **Contribuição para o problema de roteirização de veículos com restrições operacionais.** Tese de Doutorado, EPUSP, São Paulo, Engenharia de Transportes, 1997.

CUNHA, C.B. **Aspectos práticos da aplicação de modelos de roteirização de veículos a problemas reais.** Rio de Janeiro: Transportes, v. 8, n. 2 p. 51-74, 2000.

CUNHA, C. B., BONASSER, U. O. ABRAHÃO F. T. M. **Experimentos computacionais com heurísticas de melhorias para o problema do caixeiro**

**viajante.** Anais do XVI Congresso de Pesquisa e Ensino em Transportes, ANPET, Natal-RN, v. 2, p. 105-117. 2002

DORIGO, M; COLORNI, M. A., **Positive feedback as a search strategy.** Italy: Dipartimento di Elettronica, Politecnico di Milano, Italy 1991.

DORIGO, M. C. **Optimization, learning and natural algorithms (in Italian).** Unpublished doctoral dissertation, Dipartimento di Elettronica. Politecnico di Milano, Italy, 1992.

DORIGO, M., GAMBARDELLA L. M. **Ant colony system: a cooperative learning approach to the traveling salesman problem.** Belgium: Université Libre de Bruxelles, 1996.

DORIGO, M., CARO, G. D., GAMBARDELLA L. M. **Ant algorithms for discrete optimization.** Université Libre de Bruxelles, Belgium, 1999.

FORMIGONI, E. E. **Resolução de problemas de roteamento de veículos na entrega de produtos da indústria avícola.** Dissertação de Mestrado, UFPR, Curitiba, Programa de Pós-Graduação em Métodos Numéricos em Engenharia, 2005.

GOLDBARG, M. C. **Colônia de Formigas.** Departamento de Informática, UFRN, Natal, RN, 2005.

GOLDBARG, M. C, LUNA, H. P. L. **Programação linear e otimização combinatória: Modelos e Algoritmos.** Rio de Janeiro: Campus, 2000.

GOLDEN, B. L; MAGNATI, T.L; NGUYEN, H.Q. **Implementing vehicle routing algorithms.** Networks, p. 113-148, 1977.

GOSS, S.; ARON, S., DENEUBOURG, J. L., PASTEELS, J. M. **The self-organizing exploratory pattern of the Argentine ant.** Journal of Insect Behavior, 3:159–168, 1990.

HAKIMI, S. L. **Optimum distribution of switching centers in a communication network and some related graph theoretic problems.** Operational Research, London, v.13, p. 462-475, 1965.

KARIV, O. HAKIMI, S. L. **The p-median problems. In: an algorithmic approach to network location problems.** SIAN Journal on Applied Mathematics, v.37, p. 539-560, 1979.

LOISEAU, I; MAZZEO, S.; **An ant colony algorithm for the capacitated vehicle routing.** Elsevier Science Inc. New York, NY, USA, 2004.

MINIEKA, E. **The center and medians of a graph,** Operations Research, v. 25, n. 4, p. 641-650, 1977.

NEVES, T. A. **Construção de um protótipo de *framework* para otimização e seu uso para a resolução do problema do roteamento de veículos com frota heterogênea e**

**janelas de tempo.** Dissertação de Mestrado, UFOP, Ouro Preto, Ciência da Computação, 2004.

NUNES, L. F. **Algoritmos genéticos aplicados na abordagem de um problema real de roteirização de veículos.** Dissertação de Mestrado, UFPR, Curitiba, Programa de Pós-Graduação em Métodos Numéricos em Engenharia, 1998.

OLIVEIRA, A. L. R. **Otimização de recebimento e distribuição em unidades armazenadoras de soja.** Dissertação de Mestrado, UFPR, Curitiba, Programa de Pós-Graduação em Métodos Numéricos em Engenharia, 2005.

PIRES, L. A. **Algoritmo genético aplicado na localização de escolas do município de Coronel Vivida – PR.** Dissertação de Mestrado, UFPR, Curitiba, Programa de Pós-Graduação em Métodos Numéricos em Engenharia, 2002.

REVELLE, C.; SWAIN, R. **Central facilities location.** *Geographical Analysis*, v. 2, p. 30-42, 1970.

ROSA, A. A. **Dimensionamento e localização de centro de distribuição de correios numa cidade de médio porte.** Dissertação de Mestrado, UFSC, Florianópolis, Engenharia de Produção, 1996.

SILVA, R. M. A & RAMALHO G. L. **Otimização baseada em colônias de formigas aplicada ao problema da cobertura de conjuntos.** Tese de Doutorado, UFPE, Recife, Ciência da Computação, 2003.

SMIDERLE, A. **Técnicas da pesquisa operacional aplicadas a um problema de cobertura de arcos.** Dissertação de Mestrado, UFPR, Curitiba, Programa de Pós-Graduação em Métodos Numéricos em Engenharia, 2001.

STEINER, M.T.A. **Notas de aula da Disciplina de “Programação Inteira e Otimização em Redes”.** UFPR, Programa de Pós-Graduação em Métodos Numéricos em Engenharia, Campo Mourão, 2005.

TEITZ, M. B.; BART, P. **Heuristics methods for estimating the generalized vertex median of a weighted graph.** London: Operations Research Society, 1968.

TEIXEIRA, R.G.; CUNHA, C.B. **Heurísticas para o problema de dimensionamento e roteirização de uma frota heterogênea utilizando o algoritmo Out-of-Kilter.** Rio de Janeiro: Transportes, v.10, n.2, p.9-30. 2002.

ZAMBONI, L. V. S. **Técnicas de roteirização de veículos aplicadas ao transporte escolar.** Dissertação de Mestrado, UFPR, Curitiba, Programa de Pós-Graduação em Métodos Numéricos em Engenharia, 1997.

## APÊNDICE I

### EXEMPLOS DE APLICAÇÕES DOS ALGORITMOS UTILIZADOS NO TRABALHO

Para um melhor entendimento seguem exemplos de aplicações dos algoritmos utilizados nesse trabalho. A matriz de distâncias Euclidianas ( $D$ ) é dada e metros e será utilizada nos três exemplos apresentados a seguir.

$$D = \begin{bmatrix} 0 & 564.03 & 1984.50 & 143.28 & 2121.48 & 2192.68 & 1716.07 & 759.91 & 993.17 & 631.11 \\ 564.03 & 0 & 1810.84 & 439.69 & 1902.16 & 2026.25 & 1687.08 & 378.97 & 760.54 & 618.57 \\ 1984.50 & 1810.84 & 0 & 1989.83 & 204.18 & 215.42 & 508.78 & 1431.86 & 1052.88 & 1353.99 \\ 143.28 & 439.69 & 1989.83 & 0 & 2115.91 & 2201.32 & 1755.09 & 662.27 & 967.42 & 642.25 \\ 2121.48 & 1902.16 & 204.18 & 2115.91 & 0 & 227.78 & 712.90 & 1525.51 & 1157.94 & 1491.00 \\ 2192.68 & 2026.25 & 215.42 & 2201.32 & 227.78 & 0 & 624.24 & 1647.28 & 1268.19 & 1563.32 \\ 1716.07 & 1687.08 & 508.78 & 1755.09 & 712.90 & 624.24 & 0 & 1327.34 & 959.48 & 1120.80 \\ 729.91 & 378.97 & 1431.86 & 662.27 & 1525.51 & 1647.28 & 1327.34 & 0 & 383.02 & 378.64 \\ 993.17 & 760.54 & 1052.88 & 967.42 & 1157.94 & 1268.19 & 959.48 & 383.02 & 0 & 400.62 \\ 631.11 & 618.57 & 1353.99 & 642.25 & 1491.00 & 1563.32 & 1120.80 & 378.64 & 400.62 & 0 \end{bmatrix}$$

Exemplo 1:  $P$ -medianas de Teitz e Bart

Exemplo 2: Designação de Gillet e Johnson Modificado

Exemplo 3: Roteamento  $ACO$  (Otimização por Colônias de Formigas)

### O EXEMPLO 1: P-MEDIANAS DE TEITZ E BART

Considere o conjunto  $V$  que são os vértices:

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

#### **PASSO 0**

Escolher número de medianas ( $p$ ), aleatoriamente. Neste exemplo  $p = 2$ .

$V_p = \{v_1, v_2\}$ , será nossa solução inicial.

#### **PASSO 1**

Destacar os vértices que não fazem parte da solução, ou seja, pertencentes a  $V - V_p$ , como não analisados.

$V - V_p \rightarrow$  Vértices não analisados

$$v_i \in \{V - V_p\} = \{v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}$$

#### **PASSO 2 (importante)**

Neste passo calcular a redução  $\Delta_{ij} = \sigma(V_p) - \sigma(V_p \cup \{v_i\} - \{v_j\}) \quad \forall v_j \in V_p$

Quando os  $\Delta_{ij}$  forem menores que zero para todos os  $v_j$ , temos a solução final.

Para calcularmos a redução, primeiro deve-se calcular  $\sigma(V_p)$  que é dado por:

$$\sigma(V_p) = \sum_{j=1}^n w_j d(V_p, v_j) \quad \forall j,$$

onde  $w_j$  é o peso do vértice  $v_j$ , ou seja, é o número de funcionários em cada vértice; aqui consideraremos  $w_j = 1$ .

Deve-se considerar a mínima distância, (mínima)  $d(V_p, v_i)$ , ou seja, a mínima distância da solução inicial a cada um dos vértices.

Fazendo os cálculos para:  $\sigma(V_p) = \sigma(v_1, v_2) = \sum_{j=1}^n w_j d(V_p, v_j) \quad \forall j$ , obtemos:

$$\sigma(V_p) = 9327.69.$$

Agora calcula-se  $\sigma(V_p \cup \{v_i\} - \{v_j\})$  para calcular a redução  $\Delta_{ij}$ , para isso deve-se analisar cada vértice não analisado ( $V - V_p$ ) separadamente.

$\rightarrow$  Vértice 3 ( $v_3$ ) (temos  $i=3$ )

$$\sigma(V_p \cup \{v_i\} - \{v_j\}) \rightarrow \sigma(v_1, v_3) \rightarrow \Delta_{32} = 5337.81.$$

$$\rightarrow \sigma(v_3, v_2) \rightarrow \Delta_{31} = 5637.51.$$

O processo se repete para todos os vértices não analisados  $V - V_p = \{v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}\}$ .

$\rightarrow$  Vértice 4 ( $v_4$ ) (tem-se  $i=4$ )

$$\rightarrow \sigma(v_1, v_4) \rightarrow \Delta_{42} = -1381.96.$$

$$\rightarrow \sigma(v_4, v_2) \rightarrow \Delta_{41} = 0$$

→ Vértice 5 ( $v_5$ ) (tem-se  $i=5$ )  
 →  $\sigma(v_1, v_5) \rightarrow \Delta_{52} = 5121.33$ .  
 →  $\sigma(v_5, v_2) \rightarrow \Delta_{51} = 5421.03$ .

→ Vértice 6 ( $v_6$ ) (tem-se  $i=6$ )  
 →  $\sigma(v_1, v_6) \rightarrow \Delta_{62} = 5198.74$ .  
 →  $\sigma(v_6, v_2) \rightarrow \Delta_{61} = 5498.44$

→ Vértice 7 ( $v_7$ ) (tem-se  $i=7$ )  
 →  $\sigma(v_1, v_7) \rightarrow \Delta_{72} = 4453.95$ .  
 →  $\sigma(v_7, v_2) \rightarrow \Delta_{71} = 4719.96$ .

→ Vértice 8 ( $v_8$ ) (tem-se  $i=8$ )  
 →  $\sigma(v_1, v_8) \rightarrow \Delta_{82} = 2111.79$ .  
 →  $\sigma(v_8, v_2) \rightarrow \Delta_{81} = 1630.31$ .

→ Vértice 9 ( $v_9$ ) (tem-se  $i=9$ )  
 →  $\sigma(v_1, v_9) \rightarrow \Delta_{92} = 3398.26$ .  
 →  $\sigma(v_9, v_2) \rightarrow \Delta_{91} = 3105.89$ .

→ Vértice 10 ( $v_{10}$ ) (tem-se  $i=10$ )  
 →  $\sigma(v_1, v_{10}) \rightarrow \Delta_{102} = 2312.01$ .  
 →  $\sigma(v_{10}, v_2) \rightarrow \Delta_{101} = 2015.59$ .

→ O maior  $\Delta_{ij}$  é:  $\Delta_{31} = 5637.51$

Retira-se o vértice 1 ( $v_1$ ) de  $V_p$  e inserir o vértice 3 ( $v_3$ ) em  $V_p$ .  
 Como a solução inicial era  $V_p = \{v_1, v_2\}$ , agora a solução será  $V_p = \{v_3, v_2\}$ .  
 Voltamos ao passo 2.

## **PASSO 2**

Neste passo deve-se calcular a redução  $\Delta_{ij} = \sigma(V_p) - \sigma(V_p \cup \{v_i\} - \{v_j\}) \quad \forall v_j \in V_p$

Por:  $\sigma(V_p) = \sigma(v_2, v_3) = \sum_{j=1}^n w_j d(V_p, v_i) \forall j$ ,  $\sigma(V_p) = \sum_{j=1}^n w_j d(V_p, v_i) \forall j$ , portanto:

$$\sigma(v_2, v_3) = 3690.20.$$

Agora se calcula  $\sigma(V_p \cup \{v_i\} - \{v_j\})$  para calcular a redução  $\Delta_{ij}$ . Para isso, deve-se analisar cada vértice não analisado em  $(V - V_p)$  separadamente.

→ Vértice 1 ( $v_1$ ) (tem-se  $i=1$ )  
 $\Delta_{12} = -299.70$ .  
 $\Delta_{13} = -5637.50$ .

→ Vértice 4 ( $v_4$ ) (tem-se  $i=4$ )

$$\Delta_{42} = -93.10.$$

$$\Delta_{43} = -5637.50.$$

→ Vértice 5 ( $v_5$ ) (tem-se  $i=5$ )

$$\Delta_{52} = -6657.91.$$

$$\Delta_{53} = -216.47.$$

→ Vértice 6 ( $v_6$ ) (tem-se  $i=6$ )

$$\Delta_{62} = -6646.67.$$

$$\Delta_{63} = -139.05.$$

→ Vértice 7 ( $v_7$ ) (tem-se  $i=7$ )

$$\Delta_{72} = -5295.27.$$

$$\Delta_{73} = -917.54.$$

→ Vértice 8 ( $v_8$ ) (tem-se  $i=8$ )

$$\Delta_{82} = 228.99.$$

$$\Delta_{83} = -4007.18.$$

→ Vértice 9 ( $v_9$ ) (tem-se  $i=9$ )

$$\Delta_{92} = -742.95.$$

$$\Delta_{93} = -2531.60.$$

→ Vértice 10 ( $v_{10}$ ) (tem-se  $i=10$ )

$$\Delta_{102} = 90.63.$$

$$\Delta_{103} = -3621.90.$$

→ O maior  $\Delta_{ij}$  para este caso é  $\Delta_{82} = 228.99$ .

Assim, retiramos o vértice 2 ( $v_2$ ) e inserimos o vértice 8 ( $v_8$ )

Tínhamos como solução  $V_p = \{v_3, v_2\}$ ; agora a solução é  $V_p = \{v_3, v_8\}$ .

Voltamos ao passo 2.

## **PASSO 2**

Calcula-se  $\sigma(V_p) = \sigma(v_3, v_8) = \sum_{j=1}^n w_j d(V_p, v_j), \forall j$ .

$$\sigma(v_3, v_8) = 3461.21.$$

Agora se determina  $\sigma(V_p \cup \{v_i\} - \{v_j\})$  para calcular a redução  $\Delta_{ij}$  e, para isso, deve-se analisar cada vértice não analisado em  $(V - V_p)$  separadamente.

→ Vértice 1 ( $v_1$ ) (tem-se  $i=1$ )

$$\Delta_{18} = -528.69.$$

$$\Delta_{13} = -3754.70.$$

→ Vértice 2 ( $v_2$ ) (tem-se  $i=2$ )

$$\Delta_{28} = -228.99.$$

$$\Delta_{23} = -4236.17.$$

→ Vértice 4 ( $v_4$ ) (tem-se  $i=4$ )

$$\Delta_{48} = -322.09.$$

$$\Delta_{43} = -3754.70.$$

→ Vértice 5 ( $v_5$ ) (tem-se  $i=5$ )

$$\Delta_{58} = -6886.90.$$

$$\Delta_{53} = -216.47.$$

→ Vértice 6 ( $v_6$ ) (tem-se  $i=6$ )

$$\Delta_{68} = -6875.66.$$

$$\Delta_{63} = -139.05.$$

→ Vértice 7 ( $v_7$ ) (tem-se  $i=7$ )

$$\Delta_{78} = -5524.26.$$

$$\Delta_{73} = -917.54.$$

→ Vértice 9 ( $v_9$ ) (tem-se  $i=9$ )

$$\Delta_{98} = -971.94.$$

$$\Delta_{93} = -3127.08.$$

→ Vértice 10 ( $v_{10}$ ) (tem-se  $i=10$ )

$$\Delta_{108} = -138.36.$$

$$\Delta_{103} = -4103.26.$$

→ O maior valor para  $\Delta_{ij}$  é  $\Delta_{108} = -138.36$ .

Neste caso, tem-se todos os  $\Delta_{ij}$  menores que zero, sendo que o maior também é negativo, portanto não há alteração na solução  $V_p = \{v_3, v_8\}$ .

O conjunto  $V_p = \{v_3, v_8\}$  são as 2 medianas

## O EXEMPLO 2: ALGORITMO DE GILLET E JOHNSON MODIFICADO

**Passo 1:** Calcular a distância de cada nó não designado até cada mediana que ainda possuem espaço.

Vamos considerar as medianas da solução do Algoritmo de Teitz e Bart.

Medianas:  $\{v_3, v_8\}$

A matriz a seguir mostra as distâncias das medianas a cada um dos pontos

nós/ medianas	1	2	3	4	5	6	7	8	9	10
3	1984.5	1810.84	0	1989.83	204.18	215.42	508.78	0	1052.89	1353.99
8	729.91	378.97	0	662.27	1525.51	1647.28	1327.34	0	383.02	378.64

**Passo 2:** Para cada nó  $i$  do passo anterior, obter  $t_i^1$  como sendo a mediana mais próxima de  $i$  e  $t_i^2$  como a segunda mediana mais próxima de  $i$ , com distâncias iguais a  $d_i^1$  e  $d_i^2$  respectivamente.

nós $i$	Distância 1 ( $d_i^1$ )	Mediana 1 ( $t_i^1$ )	Distância 2 ( $d_i^2$ )	Mediana 2 ( $t_i^2$ )
1	729.91	8	1987.5	3
2	378.97	8	1810.54	3
4	662,27	8	1989.83	3
5	204.18	3	1525.51	8
6	215.42	3	1647.28	8
7	508.78	3	1327.34	8
9	383.02	8	1052.89	3
10	378.64	8	1356.99	3

**Passo 3:** Calcular a razão  $d_i = d_i^2 - d_i^1$ , ordenar os nós  $i$  com a razão ( $d_i$ ) decrescente.

$$d_1 = 1984.5 - 729.91 = 1254.59$$

$$d_2 = 1810.84 - 378.97 = 1431.87$$

$$d_4 = 1989.83 - 662.27 = 1327.56$$

$$d_5 = 1525.51 - 204.18 = 1321.33$$

$$d_6 = 1647.28 - 215.42 = 1431.86$$

$$d_7 = 1327.34 - 508.78 = 818.56$$

$$d_9 = 1052.89 - 383.02 = 669.87$$

$$d_{10} = 1353.99 - 378.64 = 975.35$$

Ordem decrescente:  $d_2, d_6, d_4, d_5, d_1, d_{10}, d_7, d_9$ .

$$d_2 = 1431.87$$

$$d_6 = 1431.86$$

$$d_4 = 1327.56$$

$$d_5 = 1321.33$$

$$d_1 = 1254.59$$

$$d_{10} = 975.35$$

$$d_7 = 818.56$$

$$d_9 = 669.87$$

**Passo 4:** A designação é feita priorizando as maiores razões;

Medianas:  $\{v_3, v_8\}$

$v_3 \rightarrow v_6, v_5, v_7$

$v_8 \rightarrow v_2, v_4, v_1, v_{10}, v_9$

### O EXEMPLO 3: ALGORITMO ACO

A matriz de distâncias Euclidianas ( $D_{ij}$ ) é dada em metros.

$$\eta_{ij} = \frac{1}{D_{ij}}$$

$$\eta_{ij} = \begin{bmatrix} \text{inf} & 0.0018 & 0.0005 & 0.0070 & 0.00047 & 0.00045 & 0.0006 & 0.0014 & 0.0010 & 0.0016 \\ 0.0018 & \text{inf} & 0.00055 & 0.0022 & 0.00053 & 0.0005 & 0.0006 & 0.0026 & 0.0013 & 0.0016 \\ 0.005 & 0.00055 & \text{inf} & 0.0005 & 0.0049 & 0.0046 & 0.002 & 0.0007 & 0.001 & 0.0007 \\ 0.007 & 0.0022 & 0.0005 & \text{inf} & 0.00047 & 0.00045 & 0.00056 & 0.0015 & 0.001 & 0.0015 \\ 0.00047 & 0.00052 & 0.0048 & 0.00047 & \text{inf} & 0.0044 & 0.0014 & 0.0006 & 0.0008 & 0.0006 \\ 0.00045 & 0.0005 & 0.0046 & 0.00045 & 0.0043 & \text{inf} & 0.0016 & 0.0006 & 0.00078 & 0.00063 \\ 0.00058 & 0.0006 & 0.0019 & 0.00056 & 0.0014 & 0.0016 & \text{inf} & 0.0007 & 0.104 & 0.0009 \\ 0.0013 & 0.0026 & 0.0007 & 0.0015 & 0.0006 & 0.0006 & 0.00075 & \text{inf} & 0.0026 & 0.0026 \\ 0.001 & 0.0013 & 0.0009 & 0.001 & 0.0008 & 0.0007 & 0.001 & 0.0026 & \text{inf} & 0.0024 \\ 0.0015 & 0.0016 & 0.0007 & 0.0015 & 0.0006 & 0.00063 & 0.0009 & 0.0026 & 0.0025 & \text{inf} \end{bmatrix}$$

#### Passo 1:

$\alpha$  e  $\beta = 1$  (parâmetros)

$\rho = 0.8$  (taxa de evaporação)

$t = 1$  (número de iterações)

$m=3$  (número de formigas)

$n=10$  (número de nós)

$NC = 0$  (número de ciclos)

$NCmax = 1$  (número máximo de ciclos)

Para cada aresta (i, j) Atribua um valor inicial  $\tau_{ij}(t)$  para a intensidade da trilha

$\tau_{ij}(t)$  = quantidade de feromônio na aresta (i, j) na iteração t

$\Delta(\tau_{ij}(t)) = 0$  (variação da quantidade de feromônio na aresta (i, j) na iteração t)

Temos  $\tau_{ij}(t)$  como a matriz dos feromônios na iteração inicial;

$$\tau_{ij}(0) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Vamos considerar três formigas, dispostas inicialmente nos nós um, três e cinco respectivamente.

$m = 3$ ;

#### Passo 2:

$s=1$ ; **temos:** (Lista tabu)

Para  $k=1$  até  $m \rightarrow k=1:3 \rightarrow tabu_k(s)$

Coloca-se a cidade inicial de cada formiga na lista tabu;

$tabu_1(1)=1$

$tabu_2(1)=3$

$tabu_3(1)=5$

### **Passo 3:**

Repetir o passo 3 até que a lista tabu esteja completa;

A probabilidade  $p_{ij}^k(t)$  com a qual uma formiga  $k$  escolhe ir da cidade  $i$  para a cidade  $j$  na  $t$ -ésima

iteração do algoritmo é:  $p_{ij}^k(t) = \frac{a_{ij}(t)}{\sum_{l \in N_i^k} a_{il}(t)}$ , onde  $N_i^k$  é o conjunto de nós vizinhos a  $i$  que ainda não

foram visitados pela formiga  $k$ .

Os valores de  $a_{ij}(t)$ , por sua vez, são os formadores da tabela de decisão  $A_i = [a_{ij}(t)]_{|N_i|}$  do nó  $i$  e são obtidos pela composição dos valores locais de feromônio com os valores heurísticos locais:

$a_{ij}(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta}$ , para todo  $j \in N_i$ , sendo:

$N_i$  = conjunto de vizinhos ao nó  $i$ .

$$a_{ij}(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta}$$

Cálculo de  $a_{ij}$

**soma dos elementos da matriz  $\eta_{ij}$  da linha 1 = 0.0149; então:**

para (1, 2):  $0.0018/0.0131=0.137$

para (1, 3) :  $0.0005/0.0144=0.034$

para (1,4):  $0.0070/0.0079=0.886$

para (1,5):  $0.0005/0.0144=0.034$

para (1,6):  $0.0005/0.0144=0.034$

para (1,7):  $0.0006/0.0143=0.041$

para (1,8):  $0.0014/0.0135=0.103$

para (1,9):  $0.0010/0.0139=0.071$

para (1,10):  $0.0016/0.0133=0.120$

**soma dos elementos da matriz  $\eta_{ij}$  da linha 2 = 0.0118; então:**

para (2, 1):  $0.0018/0.01=0.18$

para (2, 3) :  $0.0006/0.011=0.05$

para (2,4):  $0.0023/0.009=0.242$

para (2,5):  $0.0005/0.011=0.044$

para (2,6):  $0.0005/0.011=0.044$

para (2,7):  $0.0006/0.011=0.053$

para (2,8):  $0.0026/0.009=0.286$

para (2,9):  $0.0013/0.010=0.123$

para (2,10):  $0.0016/0.010=0.156$

**soma dos elementos da matriz  $\eta_{ij}$  da linha 3 = 0.0154; então:**

para (3, 1):  $0.0005/0.0149=0.0335$   
 para (3, 2):  $0.0006/0.0148=0.0405$   
 para (3,4):  $0.0005/0.0149=0.0335$   
 para (3,5):  $0.0049/0.0105=0.4666$   
 para (3,6):  $0.0046/0.0108=0.4259$

para (3,7):  $0.002/0.0134=0.1495$   
 para (3,8):  $0.0007/0.0147=0.0476$   
 para (3,9):  $0.0009/0.0145=0.0620$   
 para (3,10):  $0.0070.0147=0.0476$

E assim por diante, até completar a matriz  $a_{ij}$

$$a_{ij} = \begin{bmatrix} 0 & 0.14 & 0.04 & 0.9 & 0.03 & 0.03 & 0.04 & 0.10 & 0.07 & 0.12 \\ 0.18 & 0 & 0.05 & 0.24 & 0.05 & 0.04 & 0.05 & 0.29 & 0.13 & 0.16 \\ 0.03 & 0.04 & 0 & 0.03 & 0.46 & 0.43 & 0.15 & 0.05 & 0.07 & 0.05 \\ 0.83 & 0.17 & 0.03 & 0 & 0.03 & 0.03 & 0.04 & 0.11 & 0.07 & 0.11 \\ 0.03 & 0.04 & 0.52 & 0.03 & 0 & 0.44 & 0.11 & 0.05 & 0.07 & 0.05 \\ 0.03 & 0.04 & 0.49 & 0.03 & 0.45 & 0 & 0.13 & 0.05 & 0.06 & 0.05 \\ 0.07 & 0.07 & 0.26 & 0.06 & 0.18 & 0.21 & 0 & 0.09 & 0.12 & 0.10 \\ 0.11 & 0.24 & 0.05 & 0.13 & 0.05 & 0.05 & 0.06 & 0 & 0.24 & 0.24 \\ 0.09 & 0.12 & 0.09 & 0.09 & 0.08 & 0.07 & 0.09 & 0.27 & 0 & 0.26 \\ 0.14 & 0.14 & 0.06 & 0.14 & 0.06 & 0.05 & 0.07 & 0.26 & 0.24 & 0 \end{bmatrix}$$

**Cálculo de  $p_{ij}^k$**

$$p_{ij}^k(t) = \frac{a_{ij}(t)}{\sum_{l \in N_i^k} a_{il}(t)}$$

**soma dos elementos da matriz  $a_{ij}$  da linha 1: 1.47**

para (1, 2):  $0.14/1.47=0.0952$   
 para (1, 3):  $0.04/1.47=0.0272$   
 para (1,4):  $0.9/1.47=0.6122$   
 para (1,5):  $0.03/1.47=0.0204$   
 para (1,6):  $0.03/1.47=0.0204$

para (1,7):  $0.04/1.47=0.0272$   
 para (1,8):  $0.1/1.47=0.0680$   
 para (1,9):  $0.07/1.47=0.0476$   
 para (1,10):  $0.12/1.47=0.0816$

**soma dos elementos da matriz  $a_{ij}$  da linha 2: 1,19**

para (2, 1):  $0.18/1,19=0.1512$   
 para (2, 3):  $0.05/1,19=0.0420$   
 para (2,4):  $0.24/1,19=0.2016$   
 para (2,5):  $0.05/1,19=0.0420$   
 para (2,6):  $0.04/1,19=0.0336$

para (2,7):  $0.05/1,19=0.0420$   
 para (2,8):  $0.29/1,19=0.2436$   
 para (2,9):  $0.13/1,19=0.1092$   
 para (2,10):  $0.16/1,19=0.1344$

**soma dos elementos da matriz  $a_{ij}$  da linha 3: 1,31**

para (3, 1):  $0.03/1,31=0.0229$   
 para (3, 2):  $0.04/1,31=0.0305$   
 para (3,4):  $0.03/1,31=0.0229$   
 para (3,5):  $0.46/1,31=0.3511$   
 para (3,6):  $0.43/1,31=0.3282$

para (3,7):  $0.15/1,31=0.1145$   
 para (3,8):  $0.05/1,31=0.0381$   
 para (3,9):  $0.07/1,31=0.0534$   
 para (3,10):  $0.05/1,31=0.0381$

E assim por diante, até completar a matriz  $p_{ij}^k$ .

$$p_{ij}^k = \begin{bmatrix} 0 & 0.09 & 0.02 & 0.61 & 0.02 & 0.02 & 0.03 & 0.07 & 0.05 & 0.08 \\ 0.15 & 0 & 0.04 & 0.20 & 0.04 & 0.04 & 0.04 & 0.24 & 0.11 & 0.13 \\ 0.03 & 0.03 & 0 & 0.03 & 0.36 & 0.33 & 0.11 & 0.04 & 0.05 & 0.04 \\ 0.58 & 0.12 & 0.02 & 0 & 0.02 & 0.02 & 0.03 & 0.08 & 0.05 & 0.08 \\ 0.03 & 0.03 & 0.39 & 0.03 & 0 & 0.33 & 0.08 & 0.04 & 0.05 & 0.04 \\ 0.03 & 0.03 & 0.37 & 0.03 & 0.34 & 0 & 0.10 & 0.03 & 0.04 & 0.04 \\ 0.06 & 0.06 & 0.23 & 0.06 & 0.15 & 0.18 & 0 & 0.08 & 0.11 & 0.09 \\ 0.10 & 0.21 & 0.05 & 0.11 & 0.04 & 0.04 & 0.05 & 0 & 0.20 & 0.21 \\ 0.08 & 0.10 & 0.07 & 0.08 & 0.07 & 0.06 & 0.08 & 0.24 & 0 & 0.22 \\ 0.12 & 0.12 & 0.05 & 0.12 & 0.05 & 0.04 & 0.06 & 0.22 & 0.21 & 0 \end{bmatrix}$$

**Para s=2;**

*Para k=1*, (formiga 1 que está inicialmente no nó 1), ir para o nó com maior probabilidade. Se a formiga já visitou a cidade, ir para a próxima com maior probabilidade; neste caso a maior probabilidade é (1,4):  $0.9/1.47=0.6122$ , portanto, a formiga 1 vai para o nó 4.

$$\text{tabu}_1(2)=4$$

*Para k=2*, (formiga 2 que está inicialmente no nó 3), ir para o nó com maior probabilidade. Se a formiga já visitou a cidade, ir para a próxima com maior probabilidade; neste caso a maior probabilidade é (3,5):  $0.46/1.31=0.3511$ , portanto, a formiga 2 vai para o nó 5.

$$\text{tabu}_2(2)=5$$

*Para k=3*, (formiga 3 que está inicialmente no nó 5), ir para o nó com maior probabilidade. Se a formiga já visitou a cidade pegar a próxima com maior probabilidade; neste caso a maior probabilidade é (5,3):  $0.52/1.33=0.3909$ , portanto, a formiga 3 vai para o nó 3.

$$\text{tabu}_3(2)=3$$

**s=3;**

*Para k=1*, (formiga 1 que está inicialmente no nó 1 e passou para o nó 4), a partir do nó 4 ir para o nó com maior probabilidade excluindo o nó já visitado, ou seja, se a formiga já visitou o nó pegar o próximo com maior probabilidade, neste caso a maior probabilidade é 0.12 portanto, a formiga 1 vai para o nó 2.

$$\text{tabu}_1(3)=2$$

*Para k=2*, (formiga 2 que está inicialmente no nó 3 e passou para o nó 5), a partir do nó 5 ir para o nó com maior probabilidade excluindo o nó já visitado, ou seja, se a formiga já visitou o nó pegar o próximo com maior probabilidade, neste caso a maior probabilidade é 0.36, portanto, a formiga 2 vai para o nó 6.

$$\text{tabu}_2(3)=6$$

Para  $k=3$ , (formiga 3 que está inicialmente no nó 5 e passou para o nó 3), a partir do nó 3 ir para o nó com maior probabilidade excluindo o nó já visitado, ou seja, se a formiga já visitou o nó pegar o próximo com maior probabilidade, neste caso a maior probabilidade é 0.33, portanto, a formiga 3 vai para o nó 6.

$$\text{tabu}_3(3)=6$$

**s=4 temos:**

Para formiga 1, temos que ela se moverá para a cidade 8 com probabilidade 0.24, portanto temos para a lista tabu:  $\text{tabu}_1(4) = 8$ ;

Para formiga 2 temos que ela se moverá para a cidade 7 com probabilidade 0.10, portanto temos para a lista tabu:  $\text{tabu}_2(4) = 7$ ;

Para formiga 3 temos que ela se moverá para a cidade 7 com probabilidade 0.10, portanto temos para a lista tabu:  $\text{tabu}_3(4) = 7$ ;

**s=5 temos:**

Para formiga 1, temos que ela se moverá para a cidade 10 com probabilidade 0.21, portanto temos para a lista tabu:  $\text{tabu}_1(5) = 10$ ;

Para formiga 2, temos que ela se moverá para a cidade 9 com probabilidade 0.11, portanto temos para a lista tabu:  $\text{tabu}_2(5) = 9$ ;

Para formiga 3, temos que ela se moverá para a cidade 9 com probabilidade 0.11, portanto temos para a lista tabu:  $\text{tabu}_3(5) = 9$ ;

**E assim por diante até completar a lista tabu.**

**LISTA TABU:**

**Para a formiga 1**

$\text{Tabu}_1(1) \rightarrow \text{Tabu}_1(2) \rightarrow \text{Tabu}_1(3) \rightarrow \text{Tabu}_1(4) \rightarrow \text{Tabu}_1(5) \rightarrow \text{Tabu}_1(6) \rightarrow \text{Tabu}_1(7) \rightarrow \text{Tabu}_1(8) \rightarrow \text{Tabu}_1(9) \rightarrow \text{Tabu}_1(10)$

**Para a formiga 2**

$\text{Tabu}_2(1) \rightarrow \text{Tabu}_2(2) \rightarrow \text{Tabu}_2(3) \rightarrow \text{Tabu}_2(4) \rightarrow \text{Tabu}_2(5) \rightarrow \text{Tabu}_2(6) \rightarrow \text{Tabu}_2(7) \rightarrow \text{Tabu}_2(8) \rightarrow \text{Tabu}_2(9) \rightarrow \text{Tabu}_2(10)$

**Para a formiga 3**

$\text{Tabu}_3(1) \rightarrow \text{Tabu}_3(2) \rightarrow \text{Tabu}_3(3) \rightarrow \text{Tabu}_3(4) \rightarrow \text{Tabu}_3(5) \rightarrow \text{Tabu}_3(6) \rightarrow \text{Tabu}_3(7) \rightarrow \text{Tabu}_3(8) \rightarrow \text{Tabu}_3(9) \rightarrow \text{Tabu}_3(10)$

Cada formiga tem um *tour*  $T^k$  descrito por  $\text{tabu}_k(s)$ ;

**Para a formiga 1**

$T^1 = 1 \rightarrow 4 \rightarrow 2 \rightarrow 8 \rightarrow 10 \rightarrow 9 \rightarrow 7 \rightarrow 3 \rightarrow 5 \rightarrow 6$  (os números representam os nós)

**Para a formiga 2**

$T^2 = 3 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 9 \rightarrow 8 \rightarrow 10 \rightarrow 2 \rightarrow 4 \rightarrow 1$

**Para a formiga 3**

$T^3 = 5 \rightarrow 3 \rightarrow 6 \rightarrow 7 \rightarrow 9 \rightarrow 8 \rightarrow 10 \rightarrow 2 \rightarrow 4 \rightarrow 1$



**Para k=2**

$$\Delta \tau_{ij}^2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/3978.88 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/3978.88 & 0 & 0 & 0 & 0 & 0 \\ 1/3978.88 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/3978.88 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/3978.88 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3978.88 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3978.88 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3978.88 & 0 & 0 \\ 0 & 1/3978.88 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

**Para k=3**

$$\Delta \tau_{ij}^3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/3966.52 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/3966.52 & 0 & 0 & 0 & 0 \\ 1/3966.52 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3966.52 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/3966.52 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3966.52 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3966.52 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3966.52 & 0 & 0 \\ 0 & 1/3966.52 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Temos que:

$$\Delta \tau_{ij}(t) = \sum_{k=1}^m \Delta \tau_{ij}^k(t)$$

**Logo:**

$$\Delta \tau_{ij}(1) = \begin{bmatrix} 0 & 0 & 0 & 0.0003 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0005 & 0 & 0 & 0 & 0.0003 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.00052 & 0.00025 & 0 & 0 & 0 & 0 \\ 0.0005 & 0.0003 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.00025 & 0 & 0 & 0.00052 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0005 & 0 & 0 & 0 \\ 0 & 0 & 0.0003 & 0 & 0 & 0 & 0 & 0 & 0.0005 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0008 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0003 & 0.0005 & 0 & 0 \\ 0 & 0.0005 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0003 & 0 \end{bmatrix}$$

**Passo 5:**

Calcular  $\tau_{ij}(t) = (1 - \rho)\tau_{ij}(t) + \Delta \tau_{ij}(t)$ , com  $\rho = 0,8$  (taxa de evaporação).

$$\tau_{ij}(1) = (1 - 0.8)\tau_{ij}(1) + \Delta\tau_{ij}(1)$$

$$\tau_{ij}(1) = 0,2 \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0.0003 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0005 & 0 & 0 & 0 & 0.0003 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.00052 & 0.00025 & 0 & 0 & 0 & 0 \\ 0.0005 & 0.0003 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.00025 & 0 & 0 & 0.00052 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0005 & 0 & 0 & 0 \\ 0 & 0 & 0.0003 & 0 & 0 & 0 & 0 & 0 & 0.0005 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0008 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0003 & 0.0005 & 0 & 0 \\ 0 & 0.0005 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0003 & 0 \end{bmatrix} = \begin{bmatrix} 0.2 & 0.2 & 0.2 & 0.2003 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2005 & 0.2 & 0.2 & 0.2 & 0.2003 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2005 & 0.20025 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2005 & 0.2003 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2005 & 0.2 & 0.2 & 0.2005 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2005 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2003 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2005 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2008 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2003 & 0.2005 & 0.2 & 0.2 \\ 0.2 & 0.2005 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2003 & 0.2 \end{bmatrix}$$

### **Passo 6**

Se  $NC < NC_{max}$

Esvazie a lista tabu e vá para o passo 2

Caso contrário

Imprima o menor *tour* e pare

Neste caso o menor *tour* é:  $T^1 = 1 \rightarrow 4 \rightarrow 2 \rightarrow 8 \rightarrow 10 \rightarrow 9 \rightarrow 7 \rightarrow 3 \rightarrow 5 \rightarrow 6$ , que é a ordem em que os pontos devem ser visitados, com comprimento  $L^1 = 3641.42$  metros;

## APÊNDICE II

### COORDENADAS GEOGRÁFICAS DOS 281 PONTOS DE PARADA

Na relação a seguir são fornecidos os números dos vértices suas coordenadas com seus respectivos pesos

Tabela 1: Coordenadas geográficas para a Primeira Região Vila C

<i>Nº (nó)</i>	<i>x(longitude)</i>	<i>y(latitude)</i>	<i>Demanda</i>
1	743470,14	7182859,31	2
2	744066,94	7182523,03	4
3	743273,11	7182768,99	2
4	743785,39	7182533,7	1
5	744856,29	7182690,27	4
6	745318,82	7182457,37	1
7	745102,42	7182515,51	3
8	745414,82	7182539,97	2
9	745165,86	7182514,97	3
10	745350,99	7182504,19	1
11	743463,8	7182602,66	2
12	744541	7182476,51	2
13	744979,8	7182719,11	1
14	743945,87	7182566,8	2
15	745197,81	7183374,6	1
16	743931,01	7183408,49	2
17	745021,69	7182640,45	1
18	743746,09	7183366,21	1
19	743436,86	7182929,18	2
20	743487,87	7181805,13	1
21	743231,15	7181768,54	1
22	743698,52	7181738,5	1
23	743757,46	7181920,92	1

Tabela 2: Coordenadas geográficas para a Segunda Região Vila B

<i>Nº (nó)</i>	<i>x(longitude)</i>	<i>y(latitude)</i>	<i>Demanda</i>
1	742006,34	7177890,05	4
2	742664,74	7178747,77	5
3	742730,17	7178621,76	4
4	742649,58	7178430,08	5
5	742046,64	7178339,39	3
6	742419	7178406,83	1
7	742665,78	7178255,8	4
8	742543,23	7177830,7	1
9	742454,95	7177877,86	4
10	742421,27	7178008,4	5
11	742546,12	7177943,63	4
12	742533,27	7178005,75	3
13	742208,11	7178164,66	4
14	742146,45	7177953,65	5
15	742179,9	7178057,1	3
16	742321,08	7178415,68	3
17	742222,99	7178384,21	4
18	742110,57	7178363,77	3
19	742588,94	7180901,22	1

Tabela 3: Coordenadas geográficas para a Terceira Região Vila B

<i>Nº (nó)</i>	<i>x(longitude)</i>	<i>y(latitude)</i>	<i>Demanda</i>
1	743200.57	7177516.42	2
2	744065.78	7177662.29	1
3	745432.21	7177636.05	4
4	744775.45	7177888.54	5
5	744896.94	7177885.7	2
6	743247.22	7177853.93	4
7	743470.35	7177677.76	2
8	743186.57	7177084.73	1
9	743232.14	7177669.79	5
10	743460.81	7177016.01	5
11	743499.01	7177993.43	10
12	744362.71	7177198.74	5
13	744505.78	7177337.44	5
14	744508.4	7177434.38	9
15	744707.28	7177217.69	2
16	744839.71	7177293.16	3
17	744842.21	7177368.85	4
18	745179.41	7177350.97	4
19	745176.04	7177273.45	5
20	744846.67	7177520.86	5
21	745181.96	7177505.17	5
22	744848.79	7177590.71	2
23	745184.3	7177571.63	5
24	743203.62	7177300.24	5
25	743160.32	7177023.94	2
26	743196.64	7177238.79	1
27	743204.39	7177373.19	2
28	743525.65	7177017.29	5
29	743232.61	7177772.3	2
30	743249.31	7177922.55	3
31	742901.01	7177329.75	1
32	742288.71	7176955.58	2
33	742922.59	7177595.37	5
34	745319.81	7177418.9	5
35	745367.71	7177121.22	4
36	745380.93	7177415.01	5
37	745583.79	7177187.45	3
38	745578.72	7177321.78	4
39	745563.46	7177708.14	5
40	745559.69	7177776.56	2
41	745558.89	7177855.09	2
42	745555.07	7177920.74	3
43	745390.01	7177635.59	4
44	745188.72	7177721.18	3
45	745190.36	7177795.35	7
46	745192.2	7177865.51	4
47	745194.76	7177944.28	8
48	745054.18	7178064.47	5
49	745049.73	7177653.84	5
50	744987.48	7177657.44	5
51	745003.06	7178065.4	5

52	744857.82	7177961.85	5	105	743822.04	7177075.61	9
53	744854.28	7177814.44	10	106	743996.51	7177434.19	5
54	744851.01	7177742.45	5	107	744337.63	7177447.13	9
55	745004.86	7178132.8	3	108	744675.5	7177450.41	5
56	744959.91	7178200.18	2	109	745015.01	7177436.19	14
57	744648.36	7177677.51	9	110	745176.67	7177292.83	5
58	743961.57	7177433.59	5	111	745180.23	7177426.08	5
59	743943.55	7177093.64	4	112	745186.47	7177644.25	5
60	743821.41	7177132.58	2	113	745194.8	7178205.36	1
61	743821.67	7177208.32	4	114	745186.97	7178509.69	2
62	743823.92	7177285.86	4	115	742949.36	7177899.68	1
63	743826.21	7177349.86	3	116	743225.1	7177605.26	1
64	744165.75	7177215.27	5	117	744163.83	7177140.8	5
65	744168.59	7177294.35	5	118	744169.03	7177441.2	5
66	744171.55	7177425.76	5	119	744178.9	7177659.92	9
67	744308.84	7177446.65	5	120	744178.4	7178642.98	1
68	744305.41	7177182.24	2	121	744178.42	7178781.53	1
69	744315.6	7177663.89	5	122	742966.85	7177032.38	3
70	744324.4	7178038.11	5	123	743105.24	7177022.17	4
71	743831.66	7177511.08	2	124	744562.77	7177229.88	1
72	744173.27	7177520.24	5	125	745013.47	7177184.68	10
73	744185.43	7177956.29	5	126	745584.31	7177094.15	1
74	744182.46	7177809.17	5	127	745573.53	7177404.39	4
75	744179.47	7177737.18	5	128	745567.2	7177622.48	5
76	744520.16	7177894.44	5	129	745548.04	7178085.89	4
77	744659.69	7178052.3	10	130	744501.72	7177222.37	3
78	745156.44	7177393.26	3	131	744518.83	7177821.5	12
79	744060.14	7177108.52	5	132	744522.98	7178048.64	1
80	744031.25	7177435.34	5	133	744302.08	7178499.1	1
81	744516.98	7177750.72	5	134	744176.85	7178710.74	3
82	743481.23	7177492.84	5	135	744181.3	7178571.81	1
83	743485.58	7177578.35	3	136	745650.14	7177924.22	1
84	743335.38	7176817.55	5	137	745825.27	7178102.96	3
85	743148.89	7176887.45	5	138	744423.46	7178198.85	1
86	743467.72	7177088.54	3	139	745280.62	7178207.18	7
87	743464.33	7177224.68	5	140	745277.78	7178311.29	1
88	744607.87	7178213.65	1	141	743720.53	7178480.77	1
89	744597.61	7178691.66	1	142	743474.82	7178322.68	1
90	744524.04	7177969.19	4	143	743219.52	7178174.01	1
91	745579.19	7177256.5	4	144	745634.22	7178244.4	1
92	743476.23	7177356.54	1	145	743735.65	7178130.13	3
93	743038.49	7177991.65	1	146	743716.23	7178705.29	1
94	745681.33	7177994.46	1	147	743942.8	7178334.17	1
95	745916.83	7178006.45	1	148	746156.78	7178032.83	1
96	743224.93	7177565.24	2	149	743600.64	7178249.89	2
97	743122.29	7177990.43	1	150	743252.23	7178236.53	2
98	743431.45	7177997.13	2	151	743726.32	7178400.01	1
99	743308.77	7177995.97	2	152	743940.8	7178408.71	1
100	744163.88	7178031.18	14	153	743204.61	7178568.36	1
101	744589.78	7178049.26	5	154	743711.86	7178848.84	1
102	744861.21	7178055.38	6	155	744837.14	7177213.77	5
103	743356.69	7178096.08	2	156	744842.47	7177443.97	4
104	743822.3	7177426.9	4	157	744850.13	7177663.96	7

158	744821.83	7177964.66	1
159	745321.58	7177637.15	3
160	745334.67	7178076.57	3
161	744912.55	7178604.28	2
162	745103.77	7177859.91	4

163	750657.59	7179593.84	1
164	747067.08	7178352.07	1
165	743097.17	7178462.41	1
166	743497.28	7178481.76	1
167	743035.58	7179523.38	1

Tabela 4: Coordenadas geográficas para a Quarta Região Centro

<i>Nº (nó)</i>	<i>x(longitude)</i>	<i>y(latitude)</i>	<i>Demanda</i>
1	743523.9	7172600.81	3
2	743594.98	7173817.17	4
3	744983.74	7175532.83	1
4	742177.01	7172243.25	2
5	743228.63	7172920.54	3
6	742407.03	7172943.8	3
7	745094.36	7174253.4	1
8	741998.44	7173747.38	2
9	742243.97	7172223.87	4
10	742751.87	7173367.33	3
11	742436.24	7173198.81	3
12	742506.23	7173822.22	1
13	742558.61	7174106.98	2
14	742319.07	7172545.16	1
15	742293.95	7172392.3	2
16	742722.98	7171825.7	1
17	742748.24	7172093.4	2
18	742795.46	7172631.33	1
19	742824.38	7172962.08	1
20	742842.41	7173200.66	2
21	743018.42	7171991.19	2
22	743096.06	7172849.68	1
23	743108.44	7173023.1	2
24	743915.51	7173228.51	1
25	742483.1	7173241.37	2
26	743030.34	7173212.94	2
27	743895.95	7173931.44	5
28	744443.23	7174437.75	4
29	742161.34	7173534.77	2
30	742443.73	7171858.18	5
31	742512.27	7172725.15	4
32	742646.81	7168301.27	2
33	743613.44	7172411.06	2
34	744376.65	7169355.85	1
35	742296.12	7173250.00	2
36	743162.11	7175142.16	2
37	742169.22	7173522.31	3
38	742036.19	7173842.14	1
39	746499.16	7173433.81	1
40	745238.42	7174750.75	1
41	745254.78	7174105.13	1
42	745632.82	7174079.71	2
43	746197.79	7174359.64	1
44	745853.3	7174189.88	2
45	745979.98	7173894.13	1
46	745916.71	7174205.03	1
47	746112.58	7172737.74	1
48	744193.5	7173766.52	2
49	744667.01	7173143.32	1
50	744494.05	7173155.73	1
51	747688.59	7174986.02	3
52	743310.75	7169201.41	1
53	745622.28	7173081.75	2
54	745650.84	7174041.82	1
55	748097.89	7175558.79	1
56	747236.19	7173803.81	1
57	743155.25	7168223.96	1
58	746038.86	7174208.94	1
59	743748.83	7172957.54	1
60	747574.72	7174449.64	1
61	744063.73	7171329.87	2
62	744021.49	7173005.51	2
63	741844.48	7174697.49	1
64	744226.55	7171771.78	1
65	743309.43	7175453.82	1
66	743474.23	7168428.72	1
67	743381.79	7175822.27	1
68	744680.37	7173262.84	2
69	743812.43	7174420.96	1
70	744198.45	7171838.1	1
71	742432.75	7174990.71	1
72	747495.4	7173516.36	1
73	743773.94	7175756.00	1