

**UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO – DCC**

RICARDO CESAR LIVELLI SALAZAR

**ROTEAMENTO DE VEÍCULOS EM PLATAFORMAS SIG COM
COLÔNIAS DE FORMIGAS**

JOINVILLE, SC

2008

UNIVERSIDADE DO ESTADO DE SANTA CATARINA-UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO – DCC

RICARDO CÉSAR LIVELLI SALAZAR

ROTEAMENTO DE VEÍCULOS EM PLATAFORMAS SIG COM
COLÔNIAS DE FORMIGAS

Relatório Final de Trabalho de Conclusão de Curso (TCC) apresentado ao Curso de Graduação em Ciência da Computação, da Universidade do Estado de Santa Catarina (UDESC), como requisito parcial a disciplina de Trabalho de Conclusão de Curso.

Orientadora : Prof^a Mestre Éverlin Marques

JOINVILLE – SC

2008

AGRADECIMENTOS

Aos meus pais, Lucho e Alicia, pelo apoio e incentivo que me deram durante todo o período da graduação, além da minha querida irmã Estela, esta me ajudando muito no decorrer de minha formação.

Também agradeço a minha esposa, Martha, e filhos Enzo, Arianna e Jair, pela paciência e compreensão que me foi dada durante as horas de ausência da nossa vida em comum.

Aos amigos e colegas, em especial ao Alano, que contribuíram, cada um a sua maneira e em momentos diferentes, para que este trabalho tivesse continuidade.

Agradeço a minha orientadora, Éverlin, por ser minha guia nesta jornada, além da luz do seu conhecimento colaborar na construção do meu trabalho. Mas, sobretudo, agradeço-a pela minha iniciação ao mundo da pesquisa e da ciência.

RESUMO

No mundo globalizado de hoje, as empresas se encontram numa luta constante por reduzir custos operacionais nos diferentes processos produtivos de suas atividades, obtendo diferenciais competitivos no mercado, podendo oferecer produtos e serviços com maiores vantagens na demanda assim como um melhoramento na qualidade do serviço prestado.

Estas necessidades por parte das empresas geram demandas de tecnologias computacionais para apoio na toma de decisões nos diferentes níveis da cadeia produtiva, o que origina investimentos em tecnologias computacionais.

Por tais razões é proposto um Modelo para o Problema de Roteamento de Veículos em plataformas SIG. Este modelo está baseado na meta heurística de múltiplas colônias de formigas conhecida como (MACS-VRPTW). O Problema de Roteamento de Veículos visa atender uma rede de clientes, utilizando-se uma frota de veículos a partir da criação de rotas de custo mínimo, respeitando restrições como a capacidade dos veículos, os horários de atendimento dos clientes, a diversidade de frotas, entre outros. Este modelo consta de três partes: O módulo de geração da rede de atendimento, o módulo de geração de solução inicial e o módulo de aplicação do (MACS-VRPTW), o resultado permitira mostrar a rota ótima através de um mapa onde se poderão visualizar as características da rota gerada assim como acompanhar a distribuição geográfica dos clientes a serem atendidos.

PALAVRAS-CHAVE: SIG, roteirização, colônia de formigas, logística.

ABSTRACT

In the globalized world of today, companies are in a constant struggle to reduce operating costs in the various production processes of its activities, getting diferencias competitive in the market and can offer products and services with greater advantages in demand as well as an improvement in the quality of service provided.

These needs for businesses generate demands for computational technologies to support the decision making at different levels of the production chain, which leads investments in computer technology.

For these reasons it is proposed a Model for the Problem of Vehicle Routing platforms in GIS. This model based on this metaheuristics of multiple colonies of ants known as (MACS - VRPTW). The Problem of Vehicle Routing aims to answer a network of customers, using a fleet of vehicles from the creation of routes of least cost, respecting restrictions as the capacity of the vehicles, the hours of care of customers, diversity of fleets, among other. This model consists of 3 parts: The module generation of the network of care, the module for generating a solution and the initial module of the application (MACS - VRPTW), the results therefore show the optimum route through a map where you can see the characteristics of the route generated as well as monitor the geographic distribution of customers to be served.

KEYWORDS: GIS, Vehicle routing, ant colonies, logistics.

SUMÁRIO

LISTA DE FIGURAS	8
LISTA DE ABREVIATURAS.....	9
LISTA DE FÓRMULAS.....	11
LISTA DE TABELAS	12
1. INTRODUÇÃO	13
1.1. O PROBLEMA DE ROTEIRIZAÇÃO NO CONTEXTO LOGÍSTICO	13
1.2. JUSTIFICATIVAS	14
1.3. OBJETIVOS.....	14
1.3.1. OBJETIVO GERAL.....	14
1.3.2. OBJETIVOS ESPECÍFICOS	15
1.4. DELIMITAÇÕES DO TRABALHO	15
1.5. CONCEITO E DESCRIÇÃO DO PROBLEMA DE ROTEAMENTO.....	15
1.6. MÉTODO DE SOLUÇÃO DO PROBLEMA	16
1.7. FÓRMULAÇÃO DO PROBLEMA DO ROTEAMENTO DE VEÍCULOS	16
1.8. ESTRUTURA DO TRABALHO	17
2. GEOPROCESSAMENTO.....	18
2.1. PRINCÍPIOS	18
2.2. HISTÓRICO.....	18
2.3. A TRADUÇÃO DO MUNDO REAL PARA AMBIENTE COMPUTACIONAL.....	21
2.4. TIPOS DE DADOS EM GEOPROCESSAMENTO	23
2.5. BANCOS DE DADOS GEOGRAFICOS	24
2.6. INDEXAÇÃO DE DADOS ESPACIAIS	31
2.7. SISTEMAS DE INFORMAÇÃO GEOGRÁFICA (SIG).....	32
2.8. O PADRÃO OGIS.....	40
3. TEORIA DOS GRAFOS	42
3.1. DEFINIÇÃO	42
3.2. REPRESENTAÇÃO DE GRAFOS.....	45
3.3. INTRODUÇÃO AOS PROBLEMAS DE ROTEAMENTO	46
3.4. CLASSIFICAÇÃO DOS PROBLEMAS DE ROTEAMENTO.....	47
3.5. PROBLEMA DO CAIXERO VIAJANTE	48
3.6. PROBLEMA DE ROTEAMENTO DE VEÍCULOS (PRV).....	50
3.7. FÓRMULAÇÃO DO PROBLEMA DE ROTEAMENTO DE VEÍCULOS.....	51
3.8. MÉTODOS PARA RESOLUÇÃO DO PRV.....	51
4. OTIMIZAÇÃO POR MULTIPLAS COLONIAS DE FORMIGAS	53
4.1. O ANT SYSTEM (AS).....	54
4.2. O ANT COLONY SYSTEM (ACS)	57
4.3. O ANT COLONY OPTIMIZATION (ACO).....	59
5. MODELO PROPOSTO COM MACS - VRPTW.....	62
5.1. MODELO PARA SOLUÇÃO DO VRPTW.....	62
5.2. ARQUITETURA DO MODELO PARA VRPTW	63

5.3.	MÓDULO DA GERAÇÃO DA REDE	64
5.4.	MÉTODOS HEURÍSTICOS PARA ENCONTRAR A SOLUÇÃO INICIAL	65
5.5.	APLICAÇÃO DO MULTIPLE COLONY SYSTEM (MACS - VRPTW)	68
6.	ARQUITETURA DO SISTEMA	71
6.1.	ESPECIFICAÇÃO DA ARQUITETURA	71
6.2.	CARACTERÍSTICAS DA ARQUITETURA	72
6.3.	DOCUMENTAÇÃO	75
7.	ALGORITMO PARA A ROTEIRIZAÇÃO DE VEÍCULOS.	77
7.1.	ROTINA PRINCIPAL DO MACS-VRPTW	79
7.2.	IMPLEMENTAÇÃO DO ACS-VEI	80
7.3.	IMPLEMENTAÇÃO DO ACS-TIME	82
7.4.	FUNÇÃO SIMULA FORMIGA.....	82
7.5.	FUNÇÃO INSERTION PROCEDURE	83
8.	APLICAÇÕES PRÁTICAS A RESOLVER.....	85
8.1.	CARACTERÍSTICAS DOS PROBLEMAS A RESOLVER	85
8.2.	PLANEJAMENTO DE TESTES	86
8.3.	RESULTADO DE TESTES.....	87
	CONSIDERAÇÕES FINAIS	104

LISTA DE FIGURAS

FIGURA 2.1 DEMONSTRA O PENSAMENTO DA ÉPOCA.....	20
FIGURA 2.2 PARADIGMA DOS QUATRO UNIVERSOS.....	21
FIGURA 2.3. EXEMPLO DE DADO CADASTRAL.....	23
FIGURA 2.4 EXEMPLIFICANDO AS ESTRUTURAS VETORIAL E MATRICIAL (LISBOA, 2001).....	26
FIGURA 2.5 DIFERENTES RESOLUÇÕES PARA UM MAPA MATRICIAL (CÂMARA, 2001).....	27
FIGURA 2.6 ELEMENTOS DA REPRESENTAÇÃO VETORIAL (CÂMARA, 2001).....	28
FIGURA 2.7 ILUSTRAÇÃO DE ASPECTOS TECNOLÓGICOS DE UM SIG.....	33
FIGURA 2.8 ESTRUTURA DE UM SISTEMA DE INFORMAÇÃO GEOGRÁFICA.....	35
FIGURA 2.9 EXEMPLO DE ORGANIZAÇÃO, ARQUITETURA DUAL (SILVA, 2002).....	37
FIGURA 2.10 COMPARATIVA ENTRE DUAS ARQUITETURAS DE SIG (CÂMARA, 2001).....	38
FIGURA 3.1 EXEMPLO DE UM GRAFO.....	42
FIGURA 3.2 EXEMPLO DE UM GRAFO ORIENTADO E DE GRAFO NÃO ORIENTADO.....	43
FIGURA 3.3 EXEMPLO DE UM GRAFO VALORADO (G') E NÃO VALORADO (G'').....	43
FIGURA 3.4 EXEMPLO DE UM GRAFO CONEXO (G') E DESCONEXO (G'').....	44
FIGURA 3.5 GRAFO G COM DOIS SUBGRAFOS (G') E (G'').....	44
FIGURA 3.6 GRAFO E MATRIZ DE ADJACÊNCIA (SKIENA, 1997).....	45
FIGURA 3.7 REPRESENTAÇÃO DO GRAFO E DA LISTA DE ADJACÊNCIA (SKIENA, 1997).....	46
FIGURA 3.8 CLASSIFICAÇÃO DOS PR E ESCALONAMENTO (BODIN, 1983).....	49
FIGURA 3.9 REPRESENTAÇÃO DO PCV SIMÉTRICO (A) E ASSIMÉTRICO (B).....	49
FIGURA 3.10 ILUSTRAÇÃO DE UM ROTEAMENTO DE VEÍCULOS.....	50
FIGURA 4.1 DESCOBERTA DO MENOR CAMINHO.....	60
FIGURA 5.1 ARQUITETURA DO MODELO PROPOSTO.....	64
FIGURA 5.2 INTERVALOS NO TEMPO DO CICLO DE ATENDIMENTO, PARA O VRPTW.....	66
FIGURA 5.3 ALGORITMO MACS-VRPTW PROPOSTO POR (GAMBARDELLA, 1999).....	69
FIGURA 6.1 ARQUITETURA DUAL PARA UM SISTEMA (.....)	71
FIGURA 6.2 ARQUITETURA DA PLATAFORMA SIG.....	74
FIGURA 6.3 INTERFACE ENTRADA DE DADOS PELO USUÁRIO.....	74
FIGURA 6.4 INTERFACE APRESENTANDO O RESULTADO DE UMA ROTEIRIZAÇÃO.....	75
FIGURA 6.5 ILUSTRAÇÃO DA MODELAGEM DE DADOS DO BANCO DE DADOS GEOGRÁFICO.....	76
FIGURA 6.6 MODELO DE CLASSES DO PACOTE PRINCIPAL.....	77
FIGURA 7.1 DIAGRAMA DE BLOCOS DO ALGORITMO NEAREST NEIGHBOUR.....	79
FIGURA 7.2 DIAGRAMA DE BLOCOS DA ROTINA PRINCIPAL DO MACS-VRPTW.....	79
FIGURA 8.1 DISPOSIÇÃO GEOGRÁFICA DOS CLIENTES.....	89
FIGURA 8.2 DISPOSIÇÃO GEOGRÁFICA DAS ROTAS ENCONTRADAS.....	90
FIGURA 8.3 DISPOSIÇÃO GEOGRÁFICA DAS ROTAS PARA O PROBLEMA R201.....	91
FIGURA 8.4 DISTRIBUIÇÃO GEOGRÁFICA DO PROBLEMA C101 COM 25 CLIENTES.....	92
FIGURA 8.5 DISPOSIÇÃO GEOGRÁFICA DAS ROTAS PARA O PROBLEMA C101.....	93
FIGURA 8.6 DISTRIBUIÇÃO GEOGRÁFICA DO PROBLEMA RC101 COM 25 CLIENTES.....	95
FIGURA 8.7 DISPOSIÇÃO GEOGRÁFICA DAS ROTAS PARA O PROBLEMA RC101.....	96
FIGURA 8.8 DISPOSIÇÃO GEOGRÁFICA DAS ROTAS PARA O PROBLEMA R111.....	98
FIGURA 8.9 DADOS DO PROBLEMA R103.....	99
FIGURA 8.10 DISTRIBUIÇÃO GEOGRÁFICA DO PROBLEMA R103 COM 50 CLIENTES.....	100
FIGURA 8.11 DISPOSIÇÃO GEOGRÁFICA DAS ROTAS PARA O PROBLEMA R103.....	101

LISTA DE ABREVIATURAS

AS – Ant System.

ACS - Ant Colony System.

ACO - Ant Colony Optimization.

BDG – Bancos de dados geográficos.

CAD – Computer Aided Design.

DIGEST - Digital Geographic Information Exchange Standard.

ERP – Enterprise resource Planning.

GPS – System position global.

MACS – VRPTW - Multiple Ant Colony System aplicado ao problema de roteamento de veículos com janelas de tempo.

OPENGIS , OGIS – Open Geodata Interoperability Specification.

PIB – Produto Bruto Interno.

PRV – Problema de roteamento de veículos.

VRPTW - Problema de roteamento de veículos com Janelas de tempo.

PCV – Problema do caixeiro viajante.

SIG – Sistemas de informação geográfica.

SBGD – Sistemas Gerenciador de bancos de dados.

SAIF - Spatial Archive and Interchange Format.

SDTS - Spatial Data Transfer Standard.

LISTA DE FÓRMULAS

FÓRMULA 4.1 FÓRMULA DE CALCULO DE ATRATIVIDADE (DORIGO, 1996).	55
FÓRMULA 4.2 FÓRMULA DE PROBABILIDADE.	55
FÓRMULA 4.3 FÓRMULA DE ATUALIZAÇÃO DE FERORMÔNIO (DORIGO, 1996).	56
FÓRMULA 4.4 FÓRMULA DE ATUALIZAÇÃO DE FEROMÔNIO (DORIGO, 1996).	57
FÓRMULA 5.1 ELIMINAÇÃO DE <i>SUBTOUR</i>	63
FÓRMULA 5.2 FÓRMULA PARA ACHAR A DISTÂNCIA ENTRE DOIS PONTOS.	64

LISTA DE TABELAS

TABELA 3.1 CARACTERÍSTICAS DO PROBLEMA DE ROTEAMENTO.....	50
TABELA 4.1 PROBLEMAS DO TSP (DORIGO E GAMBARDILLA, 1997) .	61
TABELA 8.1 RESUMO DOS PROBLEMAS A SEREM ESTUDADOS.....	87
TABELA 8.2 RESUMO DOS RESULTADOS DAS ROTAS.	87
TABELA 8.4 DADOS DO PROBLEMA R201.....	88
TABELA 8.5 ROTAS ENCONTRADAS NO PROBLEMA R101.....	90
TABELA 8.6 ROTAS ENCONTRADAS NO PROBLEMA R201.....	91
TABELA 8.7 ROTAS ENCONTRADAS NO PROBLEMA C101.....	93
TABELA 8.8 DADOS DO PROBLEMA RC101.	94
TABELA 8.9 ROTAS ENCONTRADAS NO PROBLEMA RC101.	95
TABELA 8.10 ILUSTRADO OS DADOS DO PROBLEMA R111.	97
TABELA 8.11 ROTAS ENCONTRADAS NO PROBLEMA R111.....	97
TABELA 8.13 ROTAS ENCONTRADAS NO PROBLEMA R103.....	101

1. INTRODUÇÃO

1.1. O PROBLEMA DE ROTEIRIZAÇÃO NO CONTEXTO LOGÍSTICO

O transporte no Brasil é uma atividade que representa 2,53% do PIB brasileiro (IBGE, 2001), sendo um serviço que permite a viabilidade dos demais setores e influenciando no desenvolvimento econômico do país. O transporte é responsável, pelo abastecimento de matéria-prima às indústrias, pelo envio de produtos aos consumidores, entre outros. O Brasil é um país com uma área geográfica muito grande, com uma infra-estrutura de transporte deficiente, além de uma malha rodoviária como principal meio de transporte de carga. Todos esses elementos demonstram a importância e a necessidade de estudos, estes relacionados sobre como realizar a distribuição de produtos de forma eficiente.

A partir deste contexto, este trabalho foi proposto com o intuito de abordar uma parte do processo de distribuição de produtos, representado pelo Problema de Roteamento de Veículos em plataformas de Sistemas de Informação Geográfica (SIG).

O Problema de Roteamento de Veículos (PRV) pode ser descrito como um problema no qual uma frota de veículos, inicialmente situada em um depósito, deve atender a um conjunto de consumidores num determinado horário, os quais possuem diferentes demandas por produtos a serem distribuídos, utilizando-se a frota (LAPORTE, 1992).

A solução para esse problema pode variar de acordo com o objetivo a ser seguido. Geralmente, a solução por métodos exatos procura encontrar a distância mínima percorrida, capaz de atender a todos os consumidores. Já a solução por métodos heurísticos é feita em duas fases, de tal modo que, inicialmente, é feita a determinação do número mínimo de veículos capaz de atender a todos os consumidores. Em seguida, é realizada a minimização da distância percorrida (GAMBARDELLA, 1999).

O PRV é classificado como um problema NP-Difícil (LENSTRA, 1981), portanto, não existem algoritmos em tempo polinomial para encontrar soluções ótimas. Logo, métodos heurísticos têm sido amplamente utilizados para a solução de problemas-teste de diversas ordens, sofrendo, no entanto, das dificuldades inerentes à indefinição da qualidade da solução determinada.

Este projeto se propõe a resolver um problema de roteamento de veículos- descrito no capítulo 3 - considerando restrições como: homogeneidade e capacidade dos veículos, horário de atendimento, demanda dos clientes que precisam ser satisfeitos para que a distribuição seja realizada. Assim, além de definir o melhor caminho para a distribuição, este projeto indica qual o número de veículos necessários para a entrega, assim como o tempo gasto para percorrer todo o percurso.

1.2. JUSTIFICATIVAS

Os modelos de *roteirização*, de um modo geral, representam uma importante área de pesquisa em logística ou que desafia ao desenvolvimento de novas técnicas computacionais que visem atender estas necessidades, uma vez que os algoritmos de solução exigem um contínuo aperfeiçoamento, agregando elementos que reflitam melhor as operações reais (DASKIN,1985). Com base nesta necessidade, a produção acadêmica sobre os problemas de *roteirização* têm sido bastante numerosa nos últimos 50 anos.

1.3. OBJETIVOS

Os objetivos deste trabalho estão divididos em dois níveis. O primeiro deles compreende o objetivo geral. No segundo nível, aparecem os objetivos específicos.

1.3.1. OBJETIVO GERAL

O presente trabalho tem com objetivo principal propor um modelo de resolução para o Problema de Roteirização de Veículos aplicado em plataformas SIG. Sua aplicação será em redes de transportes, que serão representadas por grafos não orientados considerando os horários de atendimento. Este problema também é conhecido como problema de roteamento de veículos com janelas de

tempo (VRPTW).

1.3.2. OBJETIVOS ESPECÍFICOS

Os objetivos específicos deste trabalho são:

- a) Desenvolver um procedimento para aquisição dos grafos, a partir dos bancos geográficos baseados nos pontos de atendimento, distribuição e na malha viária;
- b) Criação de interfaces de usuário para consultas espaciais no SIG.
- c) Definir os algoritmos de roteirização que devem ser agregados ao modelo;
- d) Desenvolver os procedimentos necessários para execução dos algoritmos de roteamento sobre os grafos gerados;
- e) Aplicar os resultados de roteamento nas interfaces de usuário do SIG.

1.4. DELIMITAÇÕES DO TRABALHO

O trabalho desenvolvido apresenta as seguintes delimitações:

- a) O modelo proposto é aplicável somente no caso de redes de transporte representadas por grafos não orientados;
- b) As demandas por serviço existentes são realizadas nos vértices e são determinísticas;
- c) A restrição considerada no modelo contempla a capacidade dos veículos, assim como a característica heterogênea deles;
- d) É considerado o horário de atendimento dos clientes.

1.5. CONCEITO E DESCRIÇÃO DO PROBLEMA DE ROTEAMENTO

É o processo de determinar uma ou mais rotas ou seqüências de parada a serem cumpridas por uma determinada frota de veículos, tendo como objetivo a visita de pontos geográficos pré-determinados que necessitem de atendimento.

Segundo Laporte (2000), o problema de roteirização de veículos consiste em definir rotas para uma frota de veículos, dando atendimento a determinados

pontos geográficos, além de minimizar os custos envolvidos neste processo e maximizar a capacidade de entrega.

Desde o ponto de vista de otimização, os problemas de roteirização de veículos pertencem à categoria de problemas, conhecida como NP-difícil - o que significa que possuem ordem de complexidade exponencial. Isto implica num esforço computacional que cresce exponencialmente, conforme o tamanho do problema mensurado pelo número de pontos a serem atendidos (CUNHA, 1997).

O problema de roteamento de veículos (PRV) é um problema clássico da literatura que pode ser definido como segue: dado um conjunto de cidades (ou consumidores), cada qual com uma demanda, por um produto e um depósito com veículos de capacidade Q , encontrar as rotas para os veículos, minimizando os custos de transporte.

1.6. MÉTODO DE SOLUÇÃO DO PROBLEMA

A solução proposta neste trabalho está baseada na meta-heurística de Sistema Múltiplas Colônias de Formigas, para o problema de roteamento de veículos com janelas de tempo (MACS-VRPTW). Criado por Gambardella (1999), o MACS-VRPTW é uma meta-heurística que permite solucionar problemas de instâncias maiores e montar soluções robustas. No capítulo 5, será aprofundada a descrição do método citado, assim como a solução proposta.

1.7. FÓRMULAÇÃO DO PROBLEMA DO ROTEAMENTO DE VEÍCULOS

Seja $G = (V, E)$ um grafo não direcionado, onde $V = \{v_0, v_1, \dots, v_n\}$ é o conjunto dos vértices e $E = \{(v_i, v_j) : v_i, v_j \in V, i < j\}$ é o conjunto de arestas. O vértice v_0 representa o depósito, sendo este a base de uma frota de veículos idênticos de capacidade Q , enquanto os vértices remanescentes correspondem às cidades ou aos consumidores. Cada consumidor V_i tem uma demanda não negativa Q_i e $Q_0 = 0$ (LAPORTE, 1992). Neste trabalho, supõe-se que existe um número ilimitado de veículos no depósito.

A cada aresta (V_i, V_j) está associada uma distância não negativa C_{ij} , representando a distância entre os consumidores.

O Problema de Roteamento de Veículos com janelas de tempo consiste em determinar o conjunto de rotas que deverão ser feitas pelos veículos, minimizando os custos de transporte, baseados na distância e respeitando as seguintes restrições:

- a. Cada rota começa e termina no depósito;
- b. Toda cidade (ou consumidor) de V , com exceção do depósito, é visitada somente uma vez por somente um veículo;
- c. A demanda total de qualquer rota não deve superar a capacidade Q de um veículo;
- d. Os clientes (ou consumidores) têm um horário de atendimento também chamado de janela de tempo (LAPORTE, 1992).

A Figura 3.10 ilustra o roteamento de um veículo.

1.8. ESTRUTURA DO TRABALHO

Este trabalho está organizado como segue. No capítulo 1, dá-se uma introdução e visão geral do problema, apresentando uma abordagem da solução proposta, além de identificar objetivos gerais e específicos. No capítulo 2, apresentam-se os conceitos necessários sobre geoprocessamento. Os conceitos de bancos de dados geográficos apresentam-se os conceitos dos sistemas SIG. No capítulo 3, demonstra-se o problema de PRV, sua classificação. A otimização por colônias de formigas assim como a revisão de literatura do método são apresentados no capítulo 4. O modelo proposto, a fórmulação do modelo e a arquitetura são apresentados no capítulo 5. No capítulo 6, são apresentadas as conclusões do trabalho.

2. GEOPROCESSAMENTO

Este capítulo tem por finalidade delinear o entendimento sobre a área de geoprocessamento, assim como apresentar uma descrição da ferramenta computacional utilizada para se fazer geoprocessamento: o Sistema de Informação Geográfica - SIG. Serão apresentados os componentes internos do SIG, as diferentes arquiteturas para armazenamento de dados geográficos e uma topologia de SIG (THOMÉ,1998).

2.1. PRINCÍPIOS

Ao longo da história, o homem vem evoluindo no desenvolvimento de suas idéias, gerando conhecimento em diversas áreas do saber humano. Muitas destas áreas envolvem diretamente ou indiretamente um fator muito importante para todos nós: o meio ambiente. O desenvolvimento de estas áreas traz consigo pontos positivos, pois são mostrados novos horizontes e possibilidades de descoberta. Paralelamente, surgem situações negativas plausíveis de acontecer como desequilíbrios econômicos, ecológicos e sociais. Desta forma, o ser humano, fazendo uso de seu principal atributo - a inteligência - vem criando mecanismos para supervisionar, ratificar e acautelar estes possíveis desequilíbrios. Neste contexto, surge uma área de estudos e aplicações denominada “Geoprocessamento” (THOMÉ,1998).

Geoprocessamento, pela etimológica da palavra, encontra-se definida por (Geo = Terra, Processar = Executar, desempenhar, transformar), ou seja, ciência que se encarrega de processar informações sobre superfícies terrestres, podendo ser assistidas por ferramentas computacionais. Estes procedimentos auxiliam ao homem no planejamento, administração e supervisão do espaço geográfico em que vive. Os sistemas que surgem com esta finalidade são chamados de Sistemas de Informação Geográfica – SIG (THOMÉ,1998).

2.2. HISTÓRICO

O senso de localização do ser humano é uma necessidade constante que provém de muitos anos atrás, pois conhecer e se locomover sobre seu espaço é

uma condição necessária para sua evolução e desenvolvimento. Neste contexto, é que surge seu primeiro instrumento de auxílio: o mapa.

Segundo a definição do Aurélio (1977), o mapa propõe representar uma determinada área geográfica numa superfície plana e em escala menor. Para Robinson (1953), os primeiros mapas surgiram há cerca de 5.000 anos. Sem projeção e sem escala, foram utilizados como esboços por milênios. Porém, na Grécia Antiga (160 – 120 a.C.), é que foram lançados os primeiros fundamentos de ciência cartográfica (BAKKER, 1965). Mais tarde, o comércio e as navegações impulsionaram seu desenvolvimento como produto da necessidade de se guiar nos oceanos.

No Século XVII, a Academia de Ciências de Paris influencia no desenvolvimento da cartografia francesa, focando particularmente nas ciências de matemática, geodésica e astronomia, as quais proporcionam uma maior solidez científica ao desenvolvimento da cartografia. Com ajuda da matemática, os mapas se tornaram mais precisos e representativos, assim como as projeções usadas nos mapas puderam alcançar maiores níveis de exatidão. Desta forma, a cartografia passou a colaborar no estudo da terra como ciência, focando na definição da melhor representação geométrica da terra, sendo esta o “geóide”. Posteriormente, surgem as projeções do planeta que são suas representações em diversas superfícies geométricas. Isto permite a criação de mapas conforme a necessidade de utilização, mantendo características como escala, preservação de direção, área, distância entre outras (THOMÉ, 1998).

Com o aparecimento da tecnologia digital, os mapas, antes analógicos, passaram a ser representados em formato digital. Isto foi o trabalho inicial dos sistemas computacionais chamados de CAD (*Computer Aided Design*). Estes sistemas reproduzem a forma original em formato digital, tendo características limitadas, onde não era possível realizar mudanças de projeções, associações com banco de dados, entre outras limitações (BAKKER, 1965).

No intuito de demonstrar a importância deste tipo de tecnologia, será apresentado um caso na área de análise espacial, cujo propósito é comprovar claramente a utilidade do geoprocessamento na solução de problemas graves que afetam o homem (THOMÉ, 1998).

Em 1854, na cidade de Londres, aconteceu uma grande epidemia de cólera, onde famílias inteiras foram mortas, assim como muitas casas foram lacradas por não se conhecer as fontes de disseminação da enfermidade.

No entanto, o médico John Snow realizou o seguinte processo: sobre o mapa de ruas da cidade, marcou com “x” todas as fontes de água e com “ponto” as casas onde houve mortes, estas como consequência da epidemia. Com essas duas classes de informações, o médico realizou o que hoje se conhece como análise espacial, onde comprovou que havia muitos “pontos” ao redor de um poço de água da “Broat Street”, representado com um ponto, sendo o dito poço indicado para ser fechado. Como resultado, constatou-se uma diminuição dos casos de cólera e comprovou-se uma associação da água à doença, conforme Figura 2.1 demonstra.

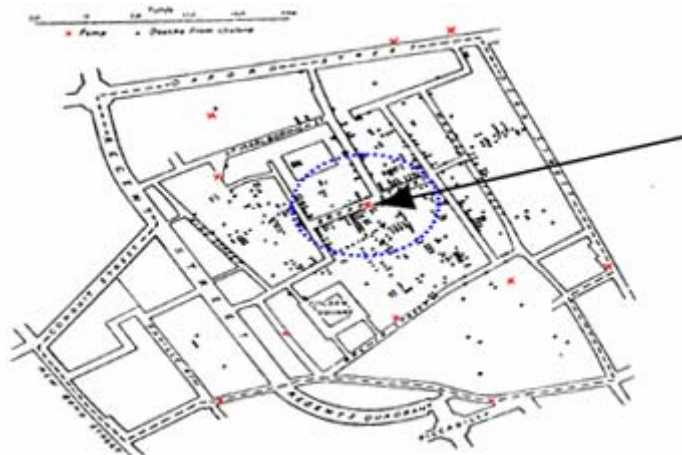


Figura 2.1 Demonstra o pensamento da época.

Neste exemplo, demonstra-se com clareza os benefícios trazidos com o geoprocessamento. De outro ponto de vista, se os dados utilizados neste processo fossem em formato de listagem, o grau de dificuldade para uma tomada de decisão seria maior.

O Geoprocessamento é identificado como uma “tecnologia interdisciplinar”, possibilitando a convergência de várias disciplinas científicas e permitindo o estudo de diferentes fenômenos urbanos e ambientais.

2.3. A TRADUÇÃO DO MUNDO REAL PARA AMBIENTE COMPUTACIONAL

Na abordagem do problema fundamental da ciência da Geoinformação, o entendimento das representações computacionais do espaço, será utilizado o “paradigma dos quatro universos” (GOMES e VELHO, 1995), representado na Figura 2.2, que distingue:

- O universo real, incluindo as entidades da realidade a serem modeladas no sistema;
- O universo matemático (conceitual), incluindo uma definição matemática (formal) das entidades a serem representadas;
- O universo de representação, onde as diversas entidades formais são mapeadas para representações alfanuméricas no computador;
- O universo de implementação, onde as estruturas e algoritmos são escolhidos, baseados em considerações como desempenho, capacidade do equipamento e tamanho da massa de dados. É neste nível que acontece a codificação.

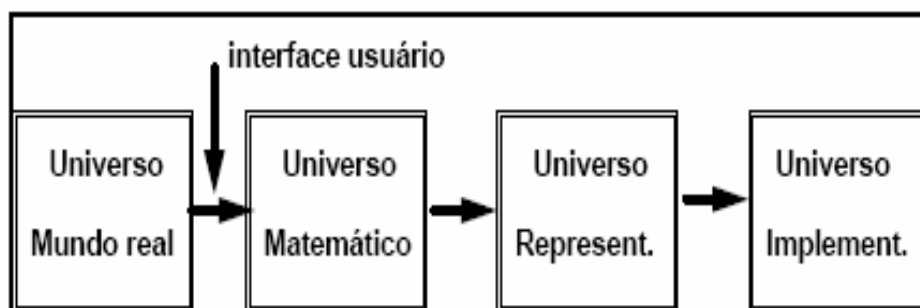


Figura 2.2 Paradigma dos quatro universos.

Para Câmara (2001), a visão apresentada não se limita apenas aos problemas do geoprocessamento, mas unifica a representação de problemas da Computação Gráfica e Processamento de Imagens. Com respeito ao Geoprocessamento, o paradigma representa apropriadamente os problemas da área, como pode se comprovar:

- No universo do mundo real, encontram-se os fenômenos a serem representados (solo, cadastros urbanos, dados topográficos, entre outros);

- No universo conceitual matemático, pode-se distinguir entre as grandes classes formais de dados geográficos e especializar estas classes nos tipos de dados geográficos (modelos numéricos de terreno, entre outros);
- No universo de representação, as entidades formais definidas no universo conceitual são associadas as diferentes representações geométricas. É aqui onde se diferenciam a representação matricial e vetorial.
- No universo de implementação, é onde ocorre a realização do modelo de dados, através de linguagens de programação, onde se escolhe as estruturas que irão implementar as geometrias do universo da representação.

Nesta análise, pode-se apreciar que a interface do usuário de um SIG deve refletir o universo conceitual, escondendo detalhes dos universos de representação e implementação. Desta forma, fica transparente para o usuário, pois ele lida com conceitos mais próximos de sua realidade, minimizando a complexidade envolvida nos diferentes tipos de representação geométrica.

Em Geoprocessamento, o espaço geográfico é modelado segundo duas visões complementares: os modelos de campos e objetos (WORBOYS, 1995).

O modelo de campos enxerga o espaço geográfico como uma superfície contínua, sobre a qual variam os fenômenos a serem observados segundo diferentes distribuições. Por exemplo, um mapa de vegetação descreve uma distribuição que associa, a cada ponto do mapa, um tipo específico de cobertura vegetal, enquanto um mapa geoquímico associa o teor de um mineral a cada ponto (WORBOYS, 1995).

O modelo de objetos representa o espaço geográfico como uma coleção de entidades distintas e identificáveis. Por exemplo, um cadastro espacial dos lotes de um município identifica cada lote como um dado individual, com atributos que o distinguem dos demais. Igualmente, poder-se-ia pensar, como geo-objetos, os rios de uma bacia hidrográfica ou os aeroportos de um estado.

2.4. TIPOS DE DADOS EM GEOPROCESSAMENTO

Câmara (2001), destacou os seguintes tipos de dados utilizados em aplicativos, envolvendo Geoprocessamento: Dados Temáticos, Dados Cadastrais, Redes, Modelos Numéricos de Terreno e Imagens:

- **Dados Temáticos** - Dados temáticos descrevem a distribuição espacial de uma grandeza geográfica, expressa de forma qualitativa. Pode-se citar, como exemplo, o mapa de vegetação e de declividade.
- **Dados Cadastrais** - Um dado cadastral diferencia-se de um temático, pois cada um de seus elementos é um objeto geográfico que possui atributos, podendo ser associado a várias representações gráficas. Por exemplo, os terrenos das casas de uma cidade que possuem atributos como localização, valor, IPTU, podem ter representações diferentes em mapas com diferentes escalas de projeções. Estes atributos são armazenados em sistemas gerenciadores de bancos de dados. Na Figura 2.3 é mostrado um exemplo de dado cadastral, onde se representam as informações de PIB e População, estas relacionadas aos países do mapa. Pode-se destacar que as informações de PIB e Pop podem ser extraídas de um SGBD, relacionando-se aos objetos geográficos países.

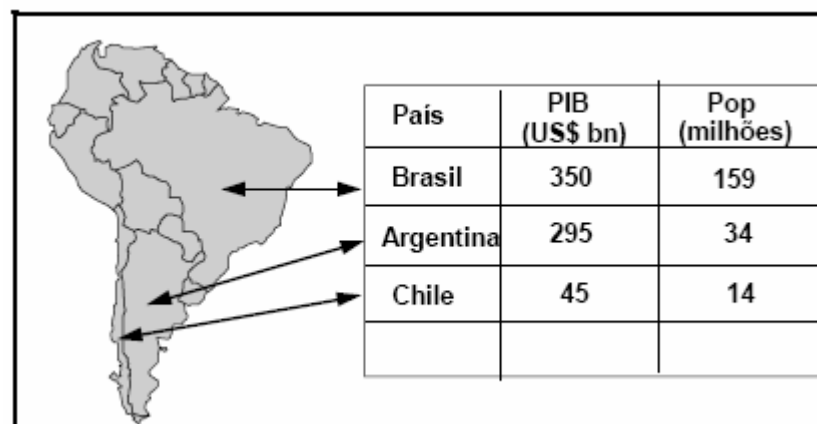


Figura 2.3. Exemplo de dado cadastral.

- **Redes** - Em Geoprocessamento, estes conceitos representam as informações associadas a rodovias; serviços de utilidade pública como água, luz; redes de esgoto, entre outros.

No caso deste tipo de dados, cada objeto geográfico (rua, estrada) possui uma localização geográfica exata, a qual se encontra vinculada a atributos descritos em um banco de dados.

As informações geográficas de redes são armazenadas em coordenadas vetoriais com topologia de arco-nó. As topologias dessas redes formam um grafo.

- **Modelos numéricos de terreno** - O termo modelo numérico de terreno (MNT) é utilizado para denotar representação quantitativa de uma grandeza que sofre variações constantes no espaço;
- **Imagens** - Obtidas a partir de fotografias aéreas, satélites, os “scanners”, entre outras. Estas imagens representam formas de captura indiretas. Cada elemento da imagem, denominado “*pixel*”, tem um valor proporcional à energia eletromagnética refletida ou emitida pela área de superfície terrestre correspondente.

2.5. BANCOS DE DADOS GEOGRAFICOS

Os Bancos de Dados Geográficos (BDG) representam o conjunto de dados usados pelos SIG. Cada SIG tem características próprias de arquitetura, que podem ser diferenciados pela forma como representam os dados geográficos: isoladamente ou por meio de estruturas de relacionamento entre eles (RUSCHEL, 2003). Os sistemas de informações geográficas armazenam grandes volumes de dados, precisando disponibilizar essas informações para consulta e análise, onde se pode apreciar a necessidade de Sistemas Gerenciadores de Banco de Dados (SBGD).

Para Zeiler (1999), um BDG pode representar dados geográficos de quatro formas:

- Objetos discretos através de estrutura vetorial;
- Fenômenos contínuos através de forma matricial;
- Superfícies como redes triangulares;

- Referências para lugares como localizadores e endereços.

Um BDG pode modelar uma série de características integradas como transportes ou redes de utilidades:

- Em um BDG, podem ser definidos relacionamentos gerais ou arbitrários entre as características de seus objetos;
- Um BDG pode vincular o comportamento natural de uma representação geográfica com as tabelas que a armazenam;
- Um BDG pode apresentar versões múltiplas, de modo que pode ser acessada por diferentes usuários ao mesmo tempo.

2.5.1. MODELO DE DADOS GEOGRÁFICOS

Para se construir um Sistema de Informação Geográfica torna-se necessário criar um modelo que represente os fenômenos geográficos. Segundo Câmara (1996) tal modelo representa estes fenômenos como dois grandes tipos de dados: objetos e campos.

Segundo Câmara (1996), o modelo de dados deve fornecer instrumentos para delinear a organização lógica de banco de dados, assim como determinar operações de manipulação de dados permitidas. Este processo de modelagem, quando bem conduzido, produz uma visão abstrata da realidade.

2.5.2. ESTRUTURAS DE DADOS EM BANCOS GEOGRÁFICOS

Segundo Lisboa (2001), dados espaciais podem ser estruturados de diversas maneiras, porém duas abordagens são as mais usadas: a estrutura matricial (ou *raster*) e a estrutura vetorial, conforme Figura 2.4.

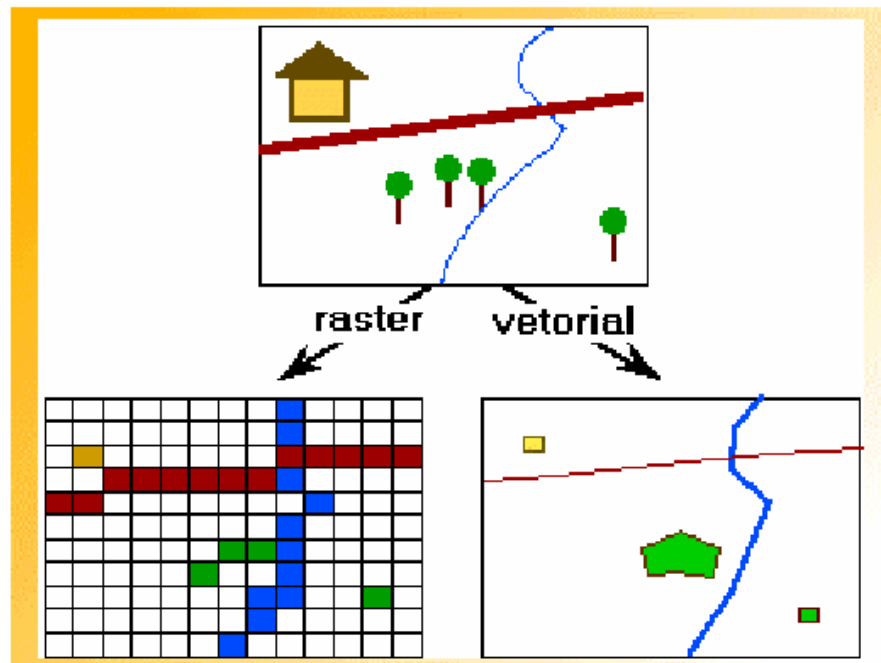


Figura 2.4 Exemplificando as estruturas vetorial e matricial (LISBOA, 2001).

A Figura 2.4 ilustra a diferença entre as representações vetorial e matricial, partindo de uma imagem supostamente real. Na representação matricial, pode-se ver que ocorre uma perda na qualidade da imagem, quando esta é analisada mais detalhadamente.

O modelo de uma base de dados geográfica é o nexa entre a percepção das pessoas, a respeito dos objetos ao seu redor no mundo, e como esses objetos são armazenados em bases de dados relacionais (ZEILER, 1999).

2.5.2.1. ESTRUTURA DE REPRESENTAÇÃO MATRICIAL

Nesta estrutura matricial, os atributos dos fenômenos geográficos estão associados a grupos de células do mesmo valor. O valor armazenado em uma célula representa a característica mais marcante do atributo, sendo esta em toda a área relativa à célula (LISBOA, 2001).

Nesta representação, o espaço é representado como uma matriz $P(m, n)$, sendo composta de m colunas e n linhas. Cada célula possui um número de linha, um número de coluna e um valor correspondente ao atributo estudado,

além de ser acessada individualmente pelas suas coordenadas (CÂMARA, 2001). A resolução de uma imagem matricial corresponde à dimensão linear mínima da célula sendo considerada. Quanto menor a dimensão das células, maior a resolução da imagem matricial e, conseqüentemente, maior a necessidade de memória para ser armazenada. (Figura 2.5).

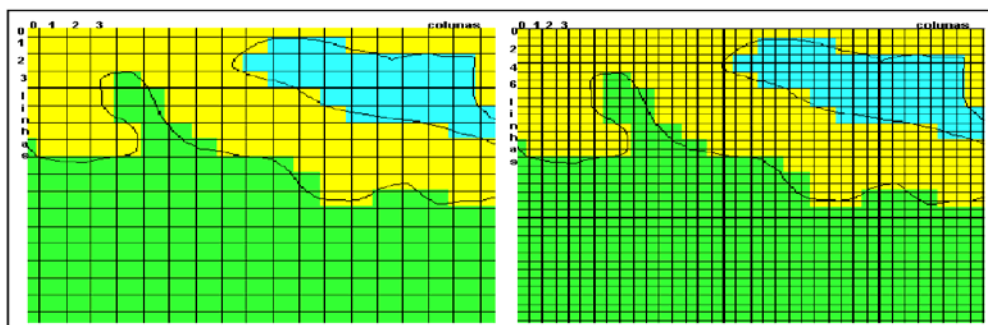


Figura 2.5 Diferentes resoluções para um mapa matricial (CÂMARA, 2001).

Uma área geográfica pode ser representada através de diversas camadas, onde as células de cada camada armazenam os valores associados a uma única variável. Segundo Zeiler (1999), uma célula pode representar uma grande variedade de dados, como a cor de uma fotografia (*pixel*), o reflexo da luz em uma parte do espectro, um valor temático como tipo de vegetação, característica de superfície ou elevação.

Lisboa (2001), em seus estudos, fez um levantamento a respeito de possíveis técnicas de compactação de dados que podem ser empregadas em estruturas matriciais. Algumas delas são:

- Códigos de cadeia (*chain codes*) – nesta técnica, os limites de cada região são armazenados através de uma estrutura, esta contendo uma célula de origem e uma seqüência de vetores unitários. Esses vetores são aplicados nas direções cardinais de cada região (leste, oeste, norte e sul), sendo percorridos no sentido horário;
- Códigos em seqüência (*Run-length codes*) – neste método, para cada linha, o número de ocorrências de células de mesmo valor são armazenados, além do valor correspondente;

- Códigos de bloco (*Block codes*) - são armazenadas, para cada quadrado máximo, que pode ser formado por um conjunto de células de mesmo valor, as coordenadas da célula inferior esquerda dele, a quantidade de células (tamanho) do seu lado, além do valor do atributo;
- Árvores quaternárias (*Quadtree*) – utiliza uma estrutura hierárquica espacial, possuindo uma grande variação de tipos de estruturas. Segundo Lisboa (2001), *quadtree* descreve uma família de estruturas de dados hierárquicas, que se baseiam no princípio da decomposição recursiva do espaço.

2.5.2.2. ESTRUTURA DE REPRESENTAÇÃO VETORIAL

Esta representação utiliza pontos, linhas e polígonos para representar a geometria de entidades geográficas. Pontos são representados por um par de coordenadas, linhas por uma seqüência de pontos e polígonos por uma seqüência de linhas, resultando em polígonos fechados, onde o ponto inicial coincide com o final.

O polígono é uma região do plano limitada por uma ou mais linha poligonais, sendo estas conectadas de modo que o último ponto de uma linha seja idêntico ao primeiro da próxima (CÂMARA, 2001). As linhas poligonais, arcos ou elementos lineares são conjuntos de pontos conectados. Além das coordenadas dos pontos que compõem a linha, deve-se armazenar a informação que indique o tipo que se está tratando, ou seja, a que atributo ela está associada.

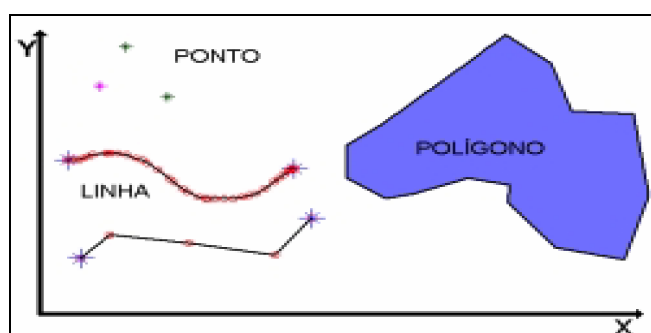


Figura 2.6 Elementos da representação Vetorial (Câmara, 2001).

Entidades lineares geográficas, como ruas e redes de tráfego, são representadas em formato vetorial. As redes são casos especiais de dados vetoriais, onde são utilizados arcos e nós, sendo estes conectados na representação do fluxo e da direção da rede. As operações topológicas e métricas são comuns em representações vetoriais (BORGES, 2002).

Zeiler (1999) define que dados Vetoriais representam a forma das feições precisamente e compactamente, estando em uma lista ordenada de coordenadas com seus atributos associados. Esta representação suporta operações geométricas tal qual para cálculo de uma distância e área, identificação de sobreposição e intersecções, além de encontrar outras feições que são adjacentes ou próximas. Zeiler classifica os dados vetoriais por sua dimensão:

- **Pontos** – são estruturas de dimensão zero que representam feições geográficas muito pequenas para serem descritas com linhas ou áreas. Pontos são armazenados como simples coordenadas x,y com seus atributos;
- **Linhas** – são estruturas unidimensionais que representam feições muito estreitas para serem representadas através de áreas. Linhas são armazenadas como uma série de coordenadas x,y ordenadas e seus respectivos atributos. Os segmentos de uma linha podem ser retos, circulares, elípticos ou conectados;
- **Polígonos** – são estruturas bidimensionais que representam uma parte larga de uma feição geográfica, sendo esta armazenada como uma série de segmentos que encerram uma área, formando um grupo de regiões fechadas;
- **Anotação** – Estes são rótulos descritivos associados às feições, exibindo nomes e atributos.

2.5.2.3. COMPARATIVO ENTRE MODELO MATRICIAL E VETORIAL

Os autores Câmara (2001), Lisboa (2001) e Zeiler (1999) consideram relevante a discussão entre qual tipo de dados deve ser utilizado. Segundo Lisboa (1995), os modelos de representação de dados *raster* e vetorial são

empregados de acordo com a aplicação para a qual são destinados. O primeiro é mais adequado para aplicações focadas às áreas ambientais, enquanto o último, o vetorial, é mais adequado para aplicações relacionadas às invenções humanas, como cadastro de propriedades urbanas e rurais, redes de infra-estruturas entre outras.

Lisboa (1995) e Câmara (2001) pontuam que, para a produção de cartas e em operações onde se requer maior precisão, a representação vetorial é mais adequada. As operações de álgebra de mapas são mais facilmente realizadas no formato matricial. No entanto, para um mesmo grau de precisão, o espaço de armazenamento requerido por uma representação matricial é substancialmente maior.

Câmara (2001) montou uma tabela, ilustrando as principais vantagens e desvantagens do uso das duas formas de representação. O comparativo em questão está reproduzido na Tabela 2.1. Nela, o modelo mais vantajoso para cada caso está apresentado em destaque.

Aspecto	Representação Vetorial	Representação Matricial
Relações espaciais entre objetos	Relacionamentos topológicos entre objetos disponíveis	Relacionamentos espaciais devem ser inferidos
Ligação com banco de dados	Facilita associar atributos a elementos gráficos	Associa atributos apenas a classes do mapa
Análise, Simulação e Modelagem	Representação indireta de fenômenos contínuos. Álgebra de mapas é limitada.	Representa melhor fenômenos com variação contínua do espaço. Simulação e abordagem mais fáceis
Escalas de trabalho	Adequado tanto a grandes quanto a pequenas escalas	Mais adequado para pequenas escalas (1:25.000 e menores)
Algoritmos	Problemas com erros geométricos	Processamento mais rápido e eficiente.
Armazenamento	Por coordenadas (mais eficiente)	Por matrizes.

Tabela 2.1: Comparação entre representações para mapas temáticos (Câmara, 2001).

Não se procura, na tabela 2.1, determinar as condições de implementação (como, por exemplo, formato de armazenamento matricial ou vetorial). As visões conceituais de campo e de objeto, muitas vezes, induzem o usuário de SIG a pensar em uma implementação matricial (*raster*) para campos e vetorial para objetos. Em grande parte dos casos, essa relação acaba ocorrendo de fato; porém, não é necessária. A implementação vetorial, por ser mais completa, tem recursos suficientes para representar fenômenos tanto da visão de campo, quanto de objeto. No entanto, quando um fenômeno geográfico recai na representação grade de células, como é o caso das imagens capturadas por satélite, a implementação matricial é a mais adequada, pois o esforço computacional de gerenciamento de um grande número de polígonos na vetorial é muito maior (RUSHEL, 2003).

Para efeitos do desenvolvimento do trabalho a estrutura a ser utilizada será a Vetorial que conforme foi exposto anteriormente é a que se acondiciona melhor para representar os fenômenos de campos e objetos de uma forma simples e com pouca necessidade de alocação de espaço em banco de dados.

2.6. INDEXAÇÃO DE DADOS ESPACIAIS

Índices são estruturas de dados criadas para otimizar o acesso aos bancos de dados que, normalmente, estão armazenados em memória secundária (usualmente discos). O acesso à memória secundária é bem mais lento, quando comparado à memória principal. No caso de bancos de dados convencionais, os índices aceleram o processamento de consultas, através da utilização de estruturas de dados dinâmicas, tais como Tabelas *Hash* e *B-trees* (Árvores balanceadas), tendo como chave um ou vários campos. Os índices espaciais levam em consideração a localização dos objetos no espaço. Os índices espaciais processam consultas espaciais eficientemente, através da redução do conjunto de dados a ser verificado no processamento (SILVA, 2002).

Normalmente, as consultas espaciais envolvem um teste espacial, no qual verifica-se um predicado espacial, do tipo “dois objetos se interceptam?” ou “dois objetos têm algum tipo de relacionamento espacial específico?”. No caso de

aplicações SIG, processar uma consulta como essa lida com operações geométricas, é computacionalmente custoso e complexo, além do volume dos dados armazenados no disco geralmente ser grande (SILVA, 2002).

2.7. SISTEMAS DE INFORMAÇÃO GEOGRÁFICA (SIG)

O termo Sistemas de Informação Geográfica (SIG) caracteriza os sistemas de informação, cuja principal característica é possibilitar a realização de análises espaciais, envolvendo dados georreferenciados. Estes sistemas são capazes de capturar, armazenar, consultar, manipular, analisar, exibir e imprimir dados, estes referenciados espacialmente sobre a superfície da Terra (LISBOA, 1995).

Os SIG têm sido usados por uma gama de instituições públicas (municipais, estaduais e federais) e privadas, em diversas áreas de aplicação, tais como: ocupação humana - cadastro territorial urbano, roteamento de veículos; uso da terra – gerenciamento de bacias hidrográficas, levantamento topográfico; uso de recursos naturais – gerenciamento costeiro e marítimo; meio ambiente - controle de emissão de efluentes, controle de queimadas; atividades econômicas - distribuição de produtos e serviços (GARAFFA, 1998).

Entre os principais componentes de um SIG, encontra-se o componente de armazenamento, denominado banco de dados geográfico. Devido à complexidade das aplicações que são desenvolvidas baseadas em SIG, um dos problemas mais críticos no desenvolvimento destes sistemas é projetar o banco de dados geográfico, pois ele deve ser projetado com ajuda do modelo de alto nível (modelo conceitual) (JUGURTA, 2000).

De forma geral, o SIG está composto de quatro grandes componentes: captura de dados, armazenamento, análise e apresentação de dados.

2.7.1. CONCEITOS BÁSICOS

Conforme descrito no capítulo 2, Sistemas de Informação Geográfica (SIG) são sistemas computacionais que permitem a captura, armazenamento,

manipulação, recuperação, análise e apresentação de dados referenciados geograficamente (WORBOYS, 1995) - o que se encontra ilustrado na Figura 2.7.

Segundo Câmara (1995), as duas principais características de um SIG são: integrar, em uma mesma base de dados, informações espaciais provenientes de diversas fontes, oferecendo mecanismos para combinar várias informações, através de algoritmos de manipulação e análise. Além disso, podem-se fazer conexões entre diferentes fenômenos com base em relacionamentos espaciais.

Quatro aspectos caracterizam um dado georreferenciado:

- A descrição do fenômeno geográfico;
- Sua posição e localização geográfica;
- Relacionamento espacial com outros fenômenos;
- Instante de tempo em que o fenômeno existe ou é válido.

Estes aspectos são classificados em duas categorias de dados: convencionais e espaciais.

- Dados Convencionais: atributos alfanuméricos usados para armazenar os dados descritivos e temporais;
- Dados Espaciais: atributos que descrevem a geometria, a localização geográfica e os relacionamentos espaciais. Além disso, um SIG pode conter dados pictóricos, que armazenam imagens sobre regiões geográficas (exemplos: fotografia de uma cidade, imagem de satélite).

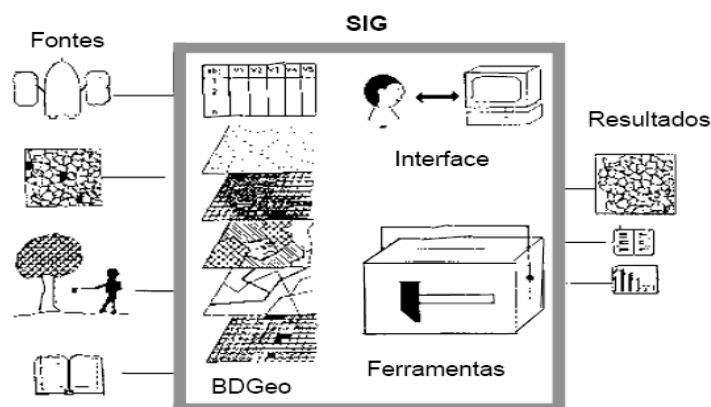


Figura 2.7 Ilustração de aspectos tecnológicos de um SIG.

2.7.2. ÁREAS DE APLICAÇÃO

O número de problemas, onde os SIG vêm sendo usados, aumentam com o tempo. Em Ramirez (1994) é apresentada uma relação de diversas áreas de aplicação de SIG organizadas em cinco grupos:

- Ocupação Humana: Redes de infra-estrutura; planejamento e supervisão de limpeza urbana, cadastramento territorial urbano, mapeamento eleitoral, rede hospitalar e, como tema do trabalho, roteamento de veículos entre outros;
- Uso da Terra: Planejamento agropecuário, estocagem e escoamento da produção agrícola, classificação de solos, gerenciamento de bacias hidrográficas, planejamento de barragens;
- Uso de Recursos Naturais: Controle de extrativismo vegetal e mineral, classificação de poços petrolíferos, planejamento de gasodutos, gerenciamento costeiros e marítimos;
- Meio Ambiente: Controle de queimadas, estudos de modificações climáticas, acompanhamento de emissão e ação de poluentes, gerenciamento florestal de desmatamento;
- Atividades Econômicas: Planejamento de marketing, pesquisas socioeconômicas, distribuição de produtos e serviços, transporte de matéria prima.

2.7.3. CARACTERÍSTICAS DE UM SIG

Uma das principais características de um SIG é sua capacidade de manipular dados gráficos (cartográficos) e não-gráficos (descritivos) de forma integrada, provendo uma forma consistente para análise e consulta. É possível, desta forma, ter acesso às informações descritivas de um fenômeno geográfico, a partir de sua localização e vice-versa. Além disso, podem-se fazer conexões entre diferentes fenômenos com base em relacionamentos espaciais (LISBOA, 2001).

Segundo Davis & Câmara (2004), pode-se indicar que um SIG tem os seguintes componentes: Interface com usuário; entrada e integração de dados; funções de processamento gráfico e de imagens; visualização e plotagem; armazenamento e recuperação de dados (organizado sob a forma de um banco de dados geográfico). A Figura 2.8 ilustra a disposição desses componentes na hierarquia de um SIG.

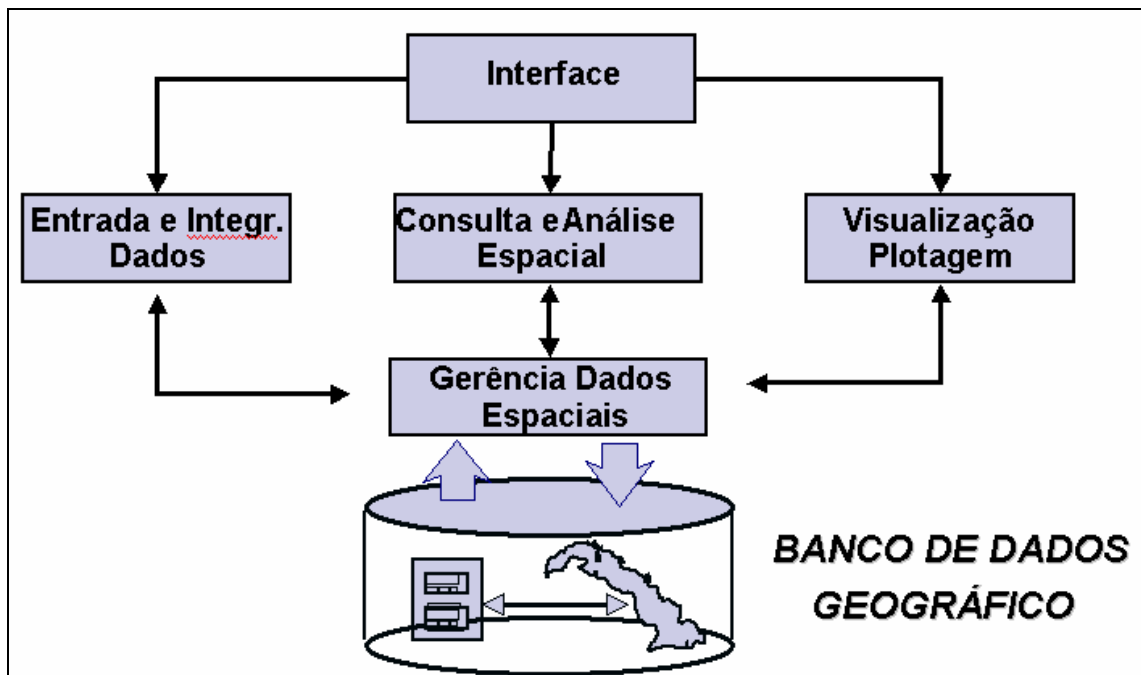


Figura 2.8 Estrutura de um Sistema de Informação Geográfica

Existem três tipos de interfaces que são utilizadas nos SIG: linguagem de comandos, menus hierárquicos e sistemas de janelas. Estes tipos de interfaces acompanham a evolução das técnicas empregadas em sistemas de informações em geral (LISBOA, 2001).

Os componentes de um SIG interagem entre si, como demonstra a Figura 2.8. Câmara (1995) propõe que, no nível mais próximo ao usuário, a interface com o sistema é operado e controlado. No nível intermediário, um SIG deve ter mecanismos de processamento de dados espaciais (entrada, edição, análise, visualização e saída). Na camada mais baixa, um sistema de gerenciamento de banco de dados geográficos fornece o armazenamento e disponibiliza a recuperação dos dados espaciais e de seus atributos. A maioria dos dados

gerados por uma aplicação SIG é apresentada na forma de mapas (LISBOA, 2000).

O componente responsável pelo armazenamento e recuperação de dados geográficos provê as estruturas de dados, possibilitando: compactação de imagens, armazenamento de relacionamentos espaciais e acesso aos dados através de índices espaciais entre outras (LISBOA, 2000).

2.7.4. ARQUITETURAS DE UM SIG

A principal diferença entre SIGs é a forma em que os dados geográficos são gerenciados. Segundo Câmara (2001), existem basicamente três diferentes arquiteturas de SIGs que utilizam os recursos de um SGBD: dual, integrada baseada em SGBDs relacionais e integrada baseada em extensões espaciais sobre SGBDs objeto-relacionais.

Serão analisadas principais características e diferenças existentes nestas formas de implementação.

2.7.4.1. ARQUITETURA DUAL

Atualmente é a arquitetura mais empregada na construção de SIG e será usada no desenvolvimento deste trabalho já que diferencia claramente a parte espacial e não espacial mantendo sua integridade. Em linhas gerais, um SIG implementado, seguindo essa estratégia, armazena, através de tabelas de um SGBD, atributos convencionais de todos os objetos, além de arquivos normais para a componente espacial deles (CÂMARA, 1996). Assim, explica-se o nome “Arquitetura dual”.

A entrada dos atributos não-espaciais é feita por meio de um SGBD relacional. Para cada entidade gráfica inserida no sistema, é imposto um identificador único ou rótulo, através do qual se relaciona com seus respectivos atributos não-espaciais, que se encontram armazenados em tabelas de dados no SGBD (Câmara, 2001). A Figura 2.9 ilustra esta forma de ligação.

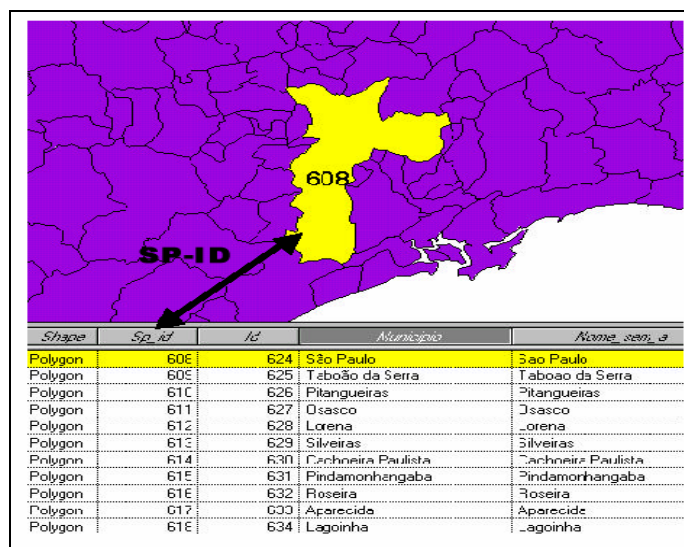


Figura 2.9 Exemplo de organização, Arquitetura Dual (SILVA, 2002).

Segundo Silva (2002), nesta abordagem a integridade do sistema pode ser comprometida, caso ocorra algum problema com a conexão entre a parte espacial e não espacial. Esta situação pode ser evitada quando o SIG implementa recursos de segurança contra falhas.

Câmara (2001) aponta, como uma das principais vantagens deste tipo de abordagem, a possibilidade de utilizar uma diversidade de SGBDs disponíveis no mercado. Já como principais desvantagens:

- Dificuldade no controle e manipulação dos dados espaciais;
- Dificuldade em manter integridade entre a componente espacial e a componente alfa-numérica;
- Consultas mais lentas, pois são processadas separadamente;
- Falta de interoperabilidade entre os dados.

2.7.4.2. ARQUITETURA INTEGRADA

Em Câmara (2001), é concluído que a arquitetura integrada consiste em armazenar todo o dado espacial em um SGBD, tanto sua componente espacial como a parte alfanumérica. A principal vantagem deste tipo de abordagem é a utilização dos recursos de um SGBD para controle e manipulação de dados espaciais, como gerência de transações, controle de integridade e concorrência.

Sendo assim, a manutenção de integridade entre a componente espacial e alfanumérica é feita pelo SGBD.

Silva (2002) analisou que a utilização de uma abordagem integrada assegura uma forte ligação entre os dados espaciais e não espaciais, reduzindo as dificuldades em manter a integridade. Além disto, os dados espaciais podem se beneficiar das estruturas de segurança que SGBDs relacionais oferecem, como por exemplo *backup* e recuperação de falhas.

A Figura 2.10 mostra a diferença estrutural da implementação de um SIG e uma Arquitetura Dual e em Arquitetura Integrada.

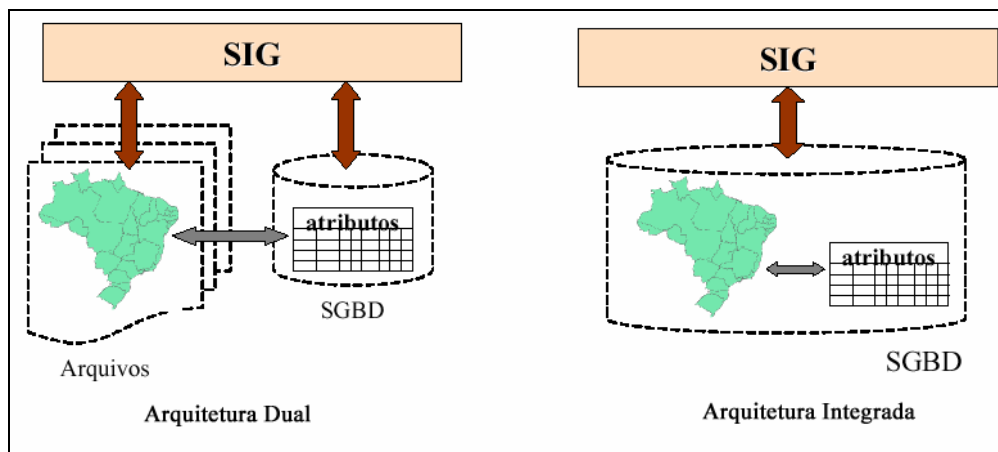


Figura 2.10 Comparativa entre duas arquiteturas de SIG (CÂMARA, 2001).

Existem duas alternativas para implementação de arquiteturas integradas: baseadas em SGBDs relacionais e baseadas em extensões espaciais sobre SGBDs objeto-relacionais.

A arquitetura integrada baseada em um SGBD relacional utiliza campos longos, chamados de BLOBs (*Binary Large Objects*), para armazenar a componente espacial do dado. Suas principais desvantagens são:

- Não captura dos dados espaciais: os dados são guardados e tratados como uma cadeia binária, portanto, não é possível conhecer a semântica de seu conteúdo;
- Os métodos de acesso espacial e otimizador de consultas precisam ser implementados pelo SIG;
- Limitação na utilização da linguagem SQL para a manipulação de dados espaciais.

A outra forma de arquitetura é a integrada baseada em extensões espaciais sobre SGBDs objeto-relacionais (SGBDOR). Estas extensões possuem funcionalidades e procedimentos que permitem armazenar, acessar e analisar os dados espaciais. Câmara (2001) aponta como desvantagens, dessa arquitetura, a falta de mecanismos de controle de integridade, estes atuantes sobre os dados espaciais, e a falta de padronização entre as extensões da linguagem SQL utilizada por elas.

Ainda segundo Câmara (2001), um SGBDOR que possui uma extensão para tratar dados espaciais, deve ter as seguintes características:

- Fornecer tipos de dados espaciais como ponto, linha e região, em seu modelo de dados e manipulá-los assim como os tipos alfanuméricos básicos;
- Estender a linguagem de consulta SQL para suportar operações e consultas espaciais sobre os tipos de dados espaciais;
- Adaptar uma série de funções em níveis mais internos a fim de manipular estes tipos mais eficientemente, tais como métodos de armazenamento e acesso (indexação espacial) e métodos de otimização de consultas (junção espacial).

Portanto, além dos Tipos de Dados Espaciais, as extensões espaciais fornecem operadores e funções que são utilizados, juntamente com a linguagem de consulta do SGBD. Além disso, proporcionam métodos de acesso eficiente aos dados espaciais através de estruturas e técnicas de indexação como *R-tree* e *QuadTree*.

Todas estas extensões baseiam-se nas especificações do OpenGIS que será tratado na seguinte seção, porém possuem variações relevantes entre os modelos de dados, semântica dos operadores espaciais e mecanismos de indexação. O OpenGIS é uma associação formada por organizações públicas e privadas envolvidas com SIGs, dedicada à criação e gerenciamento de uma arquitetura padrão para geoprocessamento.

2.8. O PADRÃO OGIS

O OpenGIS está composto por empresários, comunidade acadêmica, governo e organizações privadas da comunidade SIG e de informática, promovendo o desenvolvimento do padrão OGIS (*Open Geodata Interoperability Specification*).

Em geral, padrões facilitam o compartilhamento, a integração e a transferência de dados. Padrões para SIG usualmente incluem padrões para linguagens de especificação, transferência de dados, geocodificação e documentação de meta dados e formatos. Além desses, estão surgindo também padrões para qualidade de dados e bibliotecas de objetos espaciais. Esta seção apresenta um breve panorama do padrão OGIS (CÂMARA, 1996).

O objetivo do OGIS é permitir que o usuário tenha acesso a dados geográficos, local ou remotamente, sem estar limitado por modelos de dados proprietários, formatos de arquivos ou SIG que gerenciam os dados, em uma dada localidade (CARGILL, 1994).

O projeto do OGIS utiliza três conceitos básicos:

- Primeiro, todos os dados, processos e servidores são objetos, baseados em uma tecnologia de objetos distribuídos que tratam da interoperabilidade, comunicação e gerência de objetos e serviços distribuídos;
- Segundo, um modelo de objetos geográficos para informações espaciais é proposto. Este modelo é baseado no SAIF e é compatível com o SDTS e o DIGEST, padrões para transferência de dados espaciais entre sistemas de computadores possivelmente diferentes;
- Terceiro, o OGIS permitirá serviços a serem implementados, fornecendo ligações entre o modelo de objetos e formatos de dados privados, definindo uma interface de procedimentos padrão para cada elemento do modelo e provendo base ampla e consistente de funções de Geoprocessamento;

Aplicações em conformidade com OGIS poderão usar dados acessíveis em diversos sistemas.

A continuação no seguinte capítulo, será apresentado conceitos sobre as teoria de Grafos e sua representação assim como será abordado o Problema do Caixeiro Viajante e será revisado o problema clássico de Roteamento de Veículos. Cabe mencionar a necessidade de entendimento do capítulo para um melhor entendimento do trabalho.

3. TEORIA DOS GRAFOS

O objetivo deste item é apresentar uma definição sobre grafos, alguns conceitos básicos necessários na solução do Problema de Roteamento de Veículos. Baseados nesses princípios, também são apresentados alguns problemas clássicos que utilizam o conceito de grafos, demonstrando a aplicação da teoria em soluções de problemas reais.

3.1. DEFINIÇÃO

Um grafo $G=(V,E)$ é composto por um conjunto finito não-vazio V , correspondendo aos vértices, e um conjunto E de pares não-ordenados de elementos distintos de V , estes correspondentes às arestas (SZWARCFITER, 1984). Cada aresta $e \in E$ será denotada pelo par de vértices (v,w) que a compõe. A Figura 3.1. Representa um grafo $G=(V,E)$, onde V é o conjunto composto pelos vértices $\{1,2,3,4\}$ e E composto pelo conjunto de arestas $\{(1,2),(2,3),(3,2),(3,4)\}$.

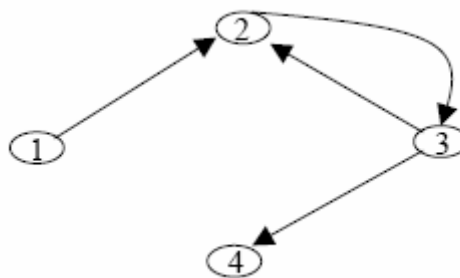


Figura 3.1 Exemplo de um grafo.

Quanto à definição que se refere ao fluxo representado em um grafo, ele é dito orientado ou direcionado quando suas arestas são representadas por setas. Neste caso, o sentido do fluxo permitido entre os vértices é o indicado pela seta. Um grafo é dito não orientado ou não direcionado quando suas arestas são

representadas por linhas sem setas. Logo, são possíveis ambos sentidos entre os vértices (VIANA, 1998).

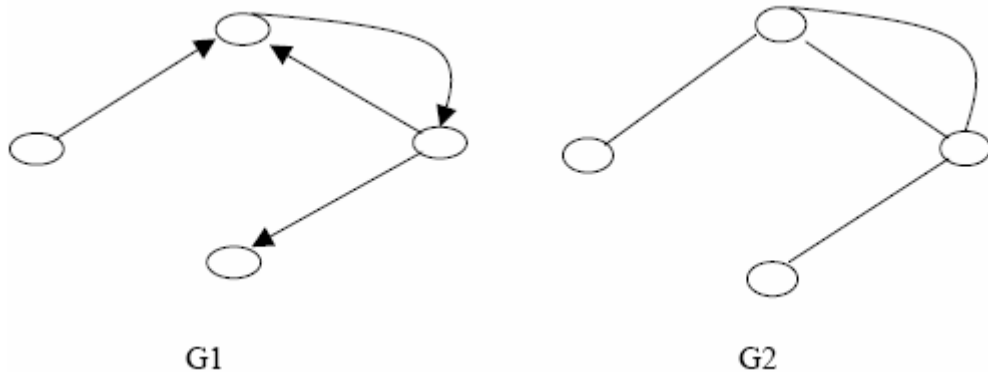


Figura 3.2 Exemplo de um grafo orientado e de grafo não orientado.

Um grafo é dito completo se ele contém todas as arestas possíveis. Caso contrário, ele é incompleto (SCHAFFER, 1997).

Dado um grafo $G = (V, E)$, se existe uma aresta unindo dois vértices V_i e V_j , sendo $i \neq j$, então estes dois vértices são ditos adjacentes. O grau de um vértice v é igual ao número de vértices adjacentes a v . O grau máximo entre todos os vértices de V é o grau do grafo (VIANA, 1998).

Um grafo é dito valorado se tem valores associados aos seus vértices ou arestas. Estes valores correspondem a características ou propriedades dos problemas como, por exemplo, distâncias, capacidades ou custos (VIANA, 1998).

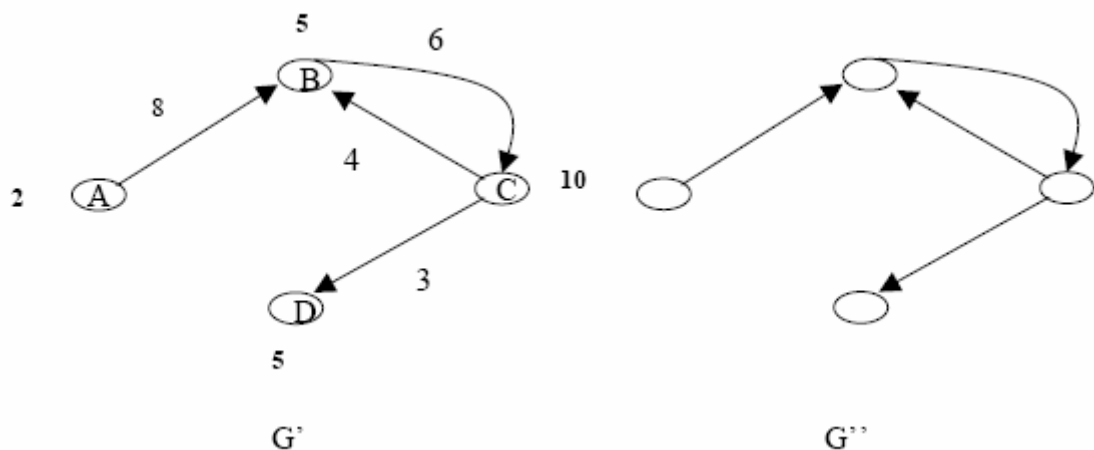


Figura 3.3 Exemplo de um grafo valorado (G') e não valorado (G'').

Um grafo é dito conexo se, para todo par de vértices, há pelo menos um caminho entre eles. Caso contrário é dito não-conexo (SZWARCFITER, 1984).

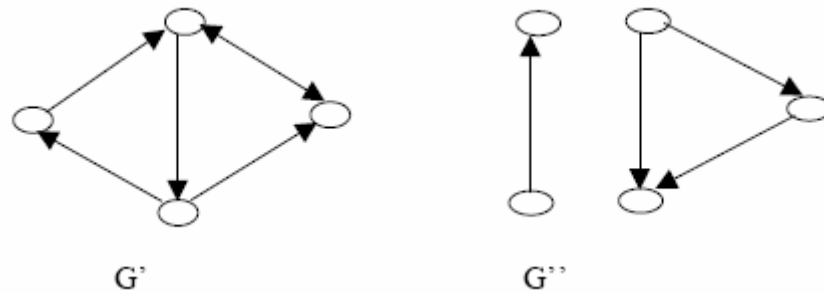


Figura 3.4 Exemplo de um grafo conexo (G') e desconexo (G'').

Caminho é uma seqüência de vértices v_1, \dots, v_k tal que $(v_j, v_{j+1}) \in E, 1 \leq j < k$. Um caminho de k vértices é formado por $k-1$ arestas. Se todos os vértices do caminho forem distintos, ele é chamado caminho simples ou elementar (SZWARCFITER, 1984). Em um grafo não-orientado, o conceito de caminho pode ser substituído por cadeia (VIANA, 1998). Um ciclo é um caminho v_1, \dots, v_k, v_{k+1} , sendo $v_1 = v_{k+1}$ e $k \geq 3$. Uma árvore é o nome dado a um grafo conexo e sem ciclos.

Um subgrafo é obtido através da seleção de um conjunto de vértices e/ou arestas de um grafo. Um grafo $G' = (V', E')$ é um subgrafo de $G = (V, E)$, se $V' \subset V$ e $E' \subset E$ (VIANA, 1998).

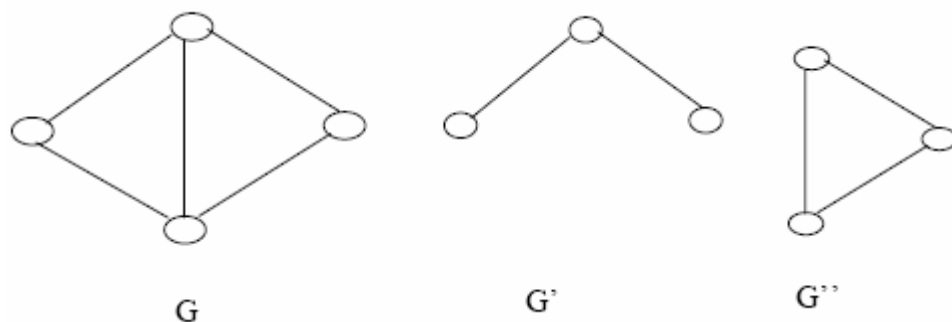


Figura 3.5 Grafo G com dois subgrafos (G') e (G'').

3.2. REPRESENTAÇÃO DE GRAFOS

Há duas maneiras que são geralmente utilizadas para se representar um grafo: Matriz de Adjacência e Lista de Adjacência.

Uma Matriz de Adjacência (Figura 3.6) é uma matriz M , de ordem $n \times n$, onde n é a quantidade de vértices, em que $M[i, j] = 1$. Isto ocorre se existir a aresta ligando estes dois vértices. Caso não exista, $M[i, j] = 0$.

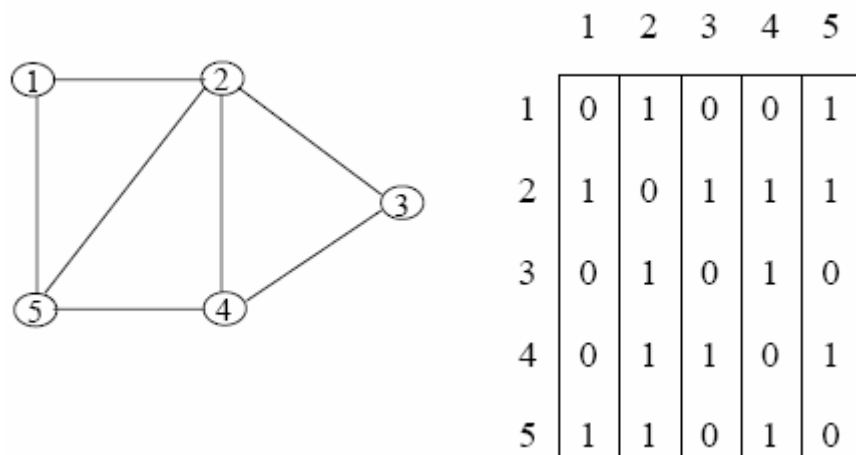


Figura 3.6 Grafo e matriz de adjacência (SKIENA, 1997).

Como principal vantagem deste tipo de estrutura, alguns autores citam a facilidade de se verificar se existe uma relação entre dois vértices. Entretanto, matrizes são estruturas esparsas, pois representam mesmo arestas que não existem (BRASSARD e BRATLEY, 1996), (VIANA, 1998).

A Lista de Adjacência (Figura 3.7) consiste numa seqüência de n elementos, onde n representa a quantidade de nodos, em que cada elemento aponta para os seus vértices adjacentes através de uma lista encadeada. Cada elemento desta lista tem um identificador e um ponteiro para o próximo elemento.

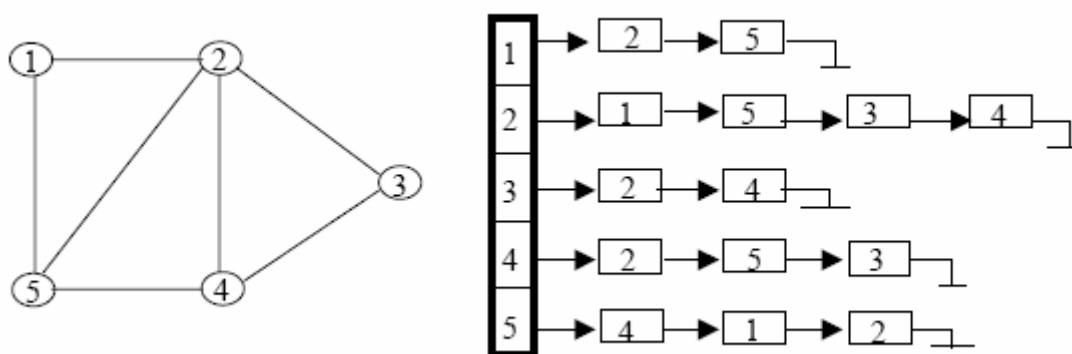


Figura 3.7 Representação do grafo e da lista de adjacência (SKIENA, 1997).

A grande vantagem deste tipo de representação é que não há necessidade de se representar arestas que não existam (VIANA, 1998).

3.3. INTRODUÇÃO AOS PROBLEMAS DE ROTEAMENTO

Como foi visto, um Problema de Roteamento pode ser conceituado como um agrupamento de clientes que devem ser atendidos por uma frota de veículos, considerando-se algumas restrições (LAPORTE, 1992). Os Problemas de Roteamento são freqüentemente abordados no contexto da área de logística, sendo definidos como a provisão de bens e serviços de pontos de suprimento aos de demanda (BODIN, 1983).

Um sistema de logística abarca todas as atividades necessárias para administrar toda a cadeia de fluxo de bens e serviços, desde o ponto onde são produzidos até onde são consumidos. Têm-se, como atividade primordial, o transporte, o controle dos estoques e o processamento de pedidos (BALLOU, 1993).

Os custos de transporte representam entre um e dois terços do total dos custos logísticos, o que faz com que muitas empresas considerem uma importância maior a esta atividade. Sendo assim, a necessidade de otimizar os gastos gerados nesta atividade é de muita importância para as empresas. A busca pela minimização de custos nesta atividade, com ajuda de algum modelo de roteamento, é um problema constante de tomada de decisão no setor logístico, pois a otimização deve levar em consideração as seguintes variáveis: redução de tempo nos percursos com trajetos mais econômicos, minimização

das distâncias, custos envolvidos e quantidade de veículos. A importância dos problemas de roteamento é evidente, devido à magnitude dos custos associados. Nesse contexto, um pequeno percentual, que consiga se reduzir do custo com transporte e distribuição, representa uma significativa economia (BODIN, 1983).

3.4. CLASSIFICAÇÃO DOS PROBLEMAS DE ROTEAMENTO

Os problemas de roteamento consideram duas variáveis principais: a definição da rota e o escalonamento dos veículos (BODIN,1983). O primeiro mostra a seqüência dos lugares que tem que ser visitados. Já o segundo, representa o tempo que tem que ficar em cada visita.

Segundo Bodin (1983), ele classifica os problemas de roteamento e escalonamento em três grupos: problemas de roteamento, de escalonamento e de roteamento e escalonamento. Os autores Laporte (1992), Bodin (1983) e Ballou (1993), definem os problemas de roteamento como aqueles problemas onde não existem restrições, quanto aos tempos de execução da atividade, além das visitas poderem ser feitas dentro de um período curto de tempo. Neste caso, as restrições temporais podem ser ignoradas para se obter um problema “puro” de roteamento de veículo. Esse tipo de problema se encontra em atividades simples que não requerem muito tempo. Os modelos que implementam esse tipo de problema estão concentrados apenas na otimização da rota.

A continuação será apresentada as características do problema de roteamento de veículos que será abordado neste trabalho o qual encontra como base a classificação realizada por Bodin (1983), sendo demonstrada na Tabela 3.1.

Características	Opção
1. Tamanho da Frota	Múltiplos veículos.
2. Tipo da frota disponível.	Homogênea.
3. Localização dos veículos.	Um depósito.
4. Natureza das demandas	Determinística.
5. Localização das demandas.	Nos nodos.
6. Tipo de Rede.	Não direcionada.
7. Restrições da capacidade do veículo.	Mesma capacidade.

8. Tempo máximo das rotas.	Não imposto.
9. Operação	Coletas.
10. Custo	Custo variável.
11. Objetivos	Minimizar o custo total das rotas. Minimizar a soma dos custos fixos e variáveis. Minimizar o número de veículos requeridos. Maximizar a utilização baseada nas prioridades de atendimento.

Tabela 3.1. Características do Problema de roteamento de veículos abordado

3.5. PROBLEMA DO CAIXERO VIAJANTE

A introdução ao estudo do problema de roteamento de veículos, neste trabalho, será feita através do estudo de um outro problema, este também bastante conhecido na área da otimização combinatória (BODIN, GOLDEN e ASSAD, 1983): o Problema do Caixeiro Viajante. A relação entre o Problema do Caixeiro Viajante e o Problema de Roteamento de Veículos é apontada em diversos trabalhos como Golden, Magnanti e Nguyen (1977) e Christofides (1976) entre outros. Segundo Golden, Magnanti e Nguyen (1977), o Problema do Caixeiro Viajante pode ser interpretado como um problema de roteamento de veículos com apenas um depósito e um veículo, cuja capacidade excede a demanda total. Este modelo pode ser estendido para considerar mais veículos, mais depósitos, diferentes capacidades de veículos, entre outras variáveis. Segundo Nelson (1985), apenas a presença de veículos com restrições (tempo e/ou capacidade) separa o Problema de Roteamento de Veículos do PCV. A maioria dos modelos de Problema de Roteamento de Veículos são variantes e extensões do PCV (GOLDEN, MAGNANTI e NGUYEN, 1977). Segundo Thangiah e Petrovic (1997), o Problema de Roteamento de Veículos é mais complexo do que o PCV, pois cada rota no Problema de Roteamento de Veículos é um PCV, ou seja, se existir k veículos, então, para se obter o melhor conjunto de rotas para os k veículos, k problemas do caixeiro viajante deveriam ser resolvidos.

O Problema do Caixeiro Viajante envolve um veículo que deve, partindo do depósito, visitar um conjunto de cidades, passando uma única vez em cada ponto e retornar ao depósito, sendo que percorreu o menor trajeto possível. A Figura 3.8. demonstra uma ilustração espacial do PCV:

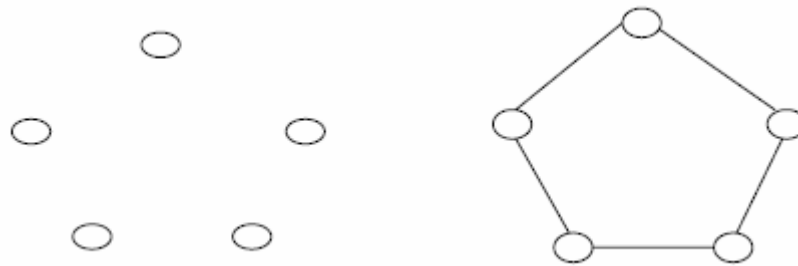


Figura 3.8. Classificação dos PR e escalonamento (BODIN,1983).

O PCV pode ser ilustrado por um grafo $G = (V, E)$, onde V representa o conjunto de cidades e E o conjunto de arestas ligando as cidades. Com cada aresta $uv \in E$ está associado um custo (COOK, 1998) e o objetivo é determinar o menor ciclo num dado grafo (BRASSARD e BRATLEY, 1996). Se os custos entre dois vértices não dependem da direção, o problema é dito simétrico, caso contrário é dito assimétrico. A seguir, a Figura 3.9 representando o PCV no caso simétrico e no caso assimétrico.

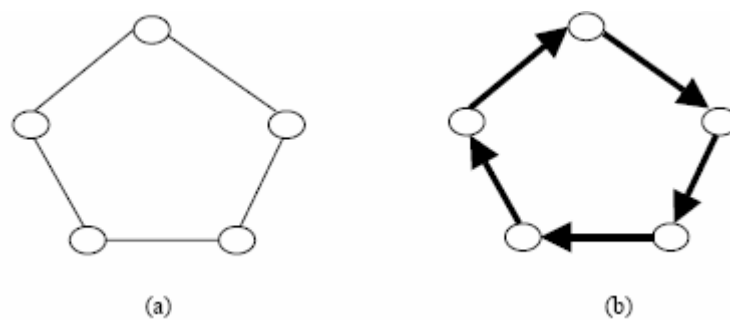


Figura 3.9. Representação do PCV simétrico (a) e assimétrico (b).

O PCV é classificado como NP-Completo (BODIN, GOLDEN e ASSAID,1983), (GAREY e JOHNSON, 1999). Ou seja, um problema que não possui solução em um tempo polinomial. Isso faz com que a dificuldade na resolução aumente de forma alarmante conforme se cresce o número de nós no grafo em questão. Para se ilustrar esta dificuldade, um problema contendo apenas 46 nós, sendo

resolvido por um computador capaz de realizar uma comparação a cada nanosegundo (um bilionésimo de segundo), levaria cerca de 356 séculos para encontrar todas as soluções possíveis.

3.6. PROBLEMA DE ROTEAMENTO DE VEÍCULOS (PRV)

O Problema de Roteamento de Veículos (PRV) pertence à classe *Np-Hard* (BREEDAM, 2001), sendo um dos problemas de Otimização Combinatória mais estudados, devido a sua relevância prática e considerável dificuldade. Num problema de roteamento de veículos se deseja minimizar a distância total viajada ou o número de veículos usados, para assim então minimizar a distância destes veículos. O PRV é o nome genérico dado a uma classe de problemas, envolvendo a visita de clientes por veículos.

Na literatura é possível encontrar uma grande variedade de problemas de roteamento que apresentam diferentes características (BODIN, GOLDEN e ASSAD, 1981), como, por exemplo, é possível citar: o Problema do Caixeiro Viajante (BODIN, 1981), o Problema de Roteamento com um Único Depósito e Múltiplos Veículos, o Problema de Roteamento com Múltiplos Depósito e Múltiplos Veículos, o Problema de Roteamento com janelas de tempo, além de outros (BODIN e LEVY, 1991), sendo este último abordado neste trabalho. A seguir, a Figura 3.10 representando o roteamento de um veículo.



Figura 3.10. Ilustração de um roteamento de veículos.

3.7. FÓRMULAÇÃO DO PROBLEMA DE ROTEAMENTO DE VEÍCULOS

Um típico PRV pode ser definido como o problema de designar rotas de custo mínimo de um depósito até um conjunto de clientes, estes geograficamente separados, respeitando algumas restrições como, por exemplo: a capacidade do veículo, tempo de duração das rotas, relações de precedência entre as cidades, horário de atendimento dos clientes a visitar, (LAPORTE,1992) entre outras.

A capacidade dos veículos pode ser homogênea ou heterogênea. Uma matriz $C = (C_{ij})$, não negativa representa distância, custo ou tempo. Geralmente os custos da viagem são os mesmos em ambas as direções. Sendo assim, uma vez que $C_{ij} = C_{ji}$ para todos V_i, V_j , o problema é dito simétrico (TOTH e VIGO,2002). Associado a cada vértice, há um peso D_i , não negativo, representando a demanda do cliente V_i e o peso total, este designado para uma rota e não podendo exceder a capacidade Q_v do veículo. O problema pode permitir que um cliente seja atendido por mais de um veículo; pode ter como restrição que um cliente deva ser atendido por um único veículo. Todas as rotas devem iniciar e terminar no depósito. Em alguns problemas, existe também uma limitação quanto ao tempo de duração da rota. Nestes casos, é definido um valor T_{ij} para representar cada par de vértices (V_i, V_j) , além de um valor T definido como sendo o tempo limite de duração da rota.

Neste trabalho, é utilizada a versão do problema com janelas de tempo, também conhecida como VRPTW (LAPORTE,1992), em que se considera, como restrição principal, o horário de atendimento dos clientes a visitar.

3.8. MÉTODOS PARA RESOLUÇÃO DO PRV

Embora existam métodos exatos (LAPORTE,1992), para a resolução do PRV, a complexidade computacional dos problemas permite que esses métodos sejam executados apenas para instâncias pequenas do problema. Uma vez que o PRV é *Np-Hard* e seu tempo de execução cresce exponencialmente, em relação ao tamanho de sua entrada (NELSON,1985), para se obter “boas soluções” num tempo aceitável, heurísticas são utilizadas. Este é o motivo pelo qual a maioria dos pesquisadores vem concentrando seus esforços no desenvolvimento das

heurísticas (NELSON, 1985). Uma vez que a maioria dos modelos de roteamento de veículo são variantes e extensões do PCV (GOLDEN, MAGNANTI e NGUYEN, 1977), as heurísticas para o PRV são freqüentemente derivadas dos métodos para resolução do PCV (LAPORTE, 1992).

Diversas famílias de heurísticas têm sido propostas para o PRV. Laporte (2000) classifica em heurísticas clássicas, desenvolvidas entre 1960 e 1990, e meta-heurísticas.

O grupo das heurísticas clássicas, geralmente é dividido em métodos construtores de rotas, métodos de duas fases e métodos de melhoramento/troca de rotas (GENDREAU, HERTZ e LAPORTE, 1994). Entre as heurísticas clássicas, encontram-se os trabalhos de Clarke e Wright (1964), Gillet e Miller (1974), que segundo Laporte (2000), estes métodos executam uma exploração limitada do espaço de procura, assim mesmo se concentram na construção de rotas baseadas no critério das economias (XU e KELLY, 1996).

No grupo das meta-heurísticas, a ênfase é na execução de uma exploração profunda das regiões mais promissoras do espaço de busca. Meta-heurísticas são também conhecidas como heurísticas de otimização global, uma vez que aceitam deixar um ótimo local, admitindo uma deterioração do valor da função objetivo (BREEDAM, 2001). Estes métodos, geralmente, combinam sofisticadas regras de geração de vizinhança, estrutura de memória e recombinação de soluções (LAPORTE, 2000). Segundo Prins (2004), meta-heurísticas oferecem soluções muito melhores, especialmente para instâncias de grande escala deste problema. Na literatura é possível encontrar trabalhos que utilizam *Simulated Annealing* (BREEDAM, 2001), Algoritmos Genéticos (BAKER e AYECHWE, 2003), entre outras.

Neste trabalho, será estudado o método múltiplas colônias de formigas para problemas de roteamento de veículos com janelas de tempo, também conhecido como (MACS-VRPTW) proposto por Gambardella (1999). No próximo capítulo, será realizada uma revisão da literatura correspondente aos métodos existentes baseados em colônias de formigas os que fundamentam o método proposto para a solução do VRPTW.

4. OTIMIZAÇÃO POR MÚLTIPLAS COLÔNIAS DE FORMIGAS

As formigas são seres vivos relativamente simples. Não entanto cabe mencionar que suas colônias apresentam um nível de sofisticação estrutural e funcional muito organizado. Isso é possível graças às formigas serem indivíduos sociais, ou seja, que interagem entre si e que unem esforços para atender suas necessidades (Gambardella, 1997). Um exemplo disso é a busca por alimento. Uma formiga poderia vagar pela superfície à procura de alimento, eventualmente teria sucesso e retornaria à colônia com o produto encontrado. Uma nova formiga, porém, precisaria passar pelo mesmo processo aleatório para encontrar comida, tendo as mesmas chances de sucesso. O que se observa, entretanto, é uma colaboração mútua das formigas para otimizar esta procura por alimento.

As formigas para se comunicar na colônia, utilizam uma técnica de comunicação indireta chamada de estigmergia que consiste em uma troca de informações através de substâncias químicas, no caso das formigas o feromônio.

Cada formiga, ao se movimentar, deposita certa quantidade de feromônios em seu caminho. Os feromônios são substâncias que uma grande parte dos seres vivos é capaz de produzir, e que são detectáveis por outros indivíduos da mesma espécie. Uma formiga, que sai em busca por alimento, tende a seguir caminhos, onde essa concentração de feromônios é maior, ou seja, por onde um maior número de formigas tenha passado. Isso pode ser caracterizado como um conhecimento coletivo acumulado, pois conforme o tempo passa, as contribuições individuais de cada formiga, que isoladamente não são muito significativas, acumulam-se e passam a indicar a melhor solução global para a situação.

Com base neste comportamento das formigas, desenvolveu-se uma área de estudos de algoritmos, que utilizam este princípio, na resolução de problemas.

O assunto foi primeiramente abordado em 1991, por Dorigo, Maniezzo & Coloni, que propuseram um modelo denominado Ant System (AS) para a resolução do problema do caixeiro viajante. Em 1996, Dorigo e Gambardella criaram o Ant Colony System (ACS) a partir do AS, o que representou um novo avanço no desempenho dos algoritmos, aplicando este modelo também ao TSP e atingindo resultados motivadores. Em 1997, Bullnheimer, Hartl e Strauss

aplicaram pela primeira vez a otimização baseada em colônias de formigas ao problema de roteirização de veículos (VRP), através de um modelo baseado no AS, que ficou denominado por AS-VRP. Muitos outros modelos surgiram ao longo da década de 90, em aplicações tais como o problema de seqüenciamento de tarefas, de coloração de grafos, e mesmo no TSP e no VRP.

4.1. O ANT SYSTEM (AS)

O AS foi o primeiro algoritmo desenvolvido baseado na conduta de formigas no ano de 1991 por Dorigo, Maniezzo & Colorni, que serviu de base para o desenvolvimento de muitos outros modelos, como o ACS e o AS-VRP. Sua aplicação original foi no problema do caixeiro viajante (TSP). Este algoritmo simula ciclos de vida de cada formiga, onde cada uma delas é criada em um nó inicial, faz suas escolhas de qual caminho percorrer, de acordo com uma regra probabilística e, quando tiver finalizado o percurso de todos os nós do problema, ela morre, sendo excluída do problema. O registro do caminho por onde a formiga passou consta em sua memória, e é utilizado para a avaliação da qualidade das soluções, ou seja, para a determinação do comprimento total percorrido. Além disso, esta memória também indica a localização dos arcos onde os feromônios devem ser depositados, o que só é feito depois da formiga ter terminado seu ciclo de vida. A regra probabilística, de decisão para os algoritmos do tipo AS, leva em consideração tanto a quantidade de feromônios acumulada em cada arco, quanto o comprimento destes arcos. Estas quantidades de feromônios representam o conhecimento acumulado pela colônia, sendo resultado de decisões tomadas por todas as formigas precedentes. Os comprimentos dos arcos, por sua vez, representam as informações físicas do problema, que são estáticas e independem das formigas precedentes. Pode-se dar um peso diferenciado, de acordo a contribuição da densidade de feromônios e do comprimento do arco, através dos parâmetros alfa e beta, estes ilustrados na Fórmula 4.1.

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

Fórmula 4.1 Fórmula de calculo de atratividade (Dorigo, 1996).

Este cálculo da atratividade é para uma formiga que se encontra em um nó i qualquer, sendo para todo j pertencente a N_i , onde N_i é o conjunto dos nós adjacentes ao nó i . Onde cada arco ij incidente ao nó i , tem-se o valor de a_{ij} calculado, e, como o denominador da Fórmula 4.1, tem-se o somatório de todos os valores de a_{ij} dos arcos incidentes ao nó i . Este valor de a_{ij} , em particular, representa uma fração deste somatório de valores de a . Na equação da Fórmula 4.1, a variável τ_{ij} representa a quantidade de feromônio acumulada no arco ij , e a variável η_{ij} é igual ao inverso do comprimento do arco ij , ou seja, quanto maior a concentração de feromônios em um arco e, quanto menor seu comprimento, maior o valor de a do arco. Assim mesmo, o parâmetro alfa define a importância que é dada à quantidade de feromônios no arco. Já o parâmetro beta define a importância do comprimento do arco. Uma vez calculados os valores de a de todos os arcos incidentes a i , pode-se definir a probabilidade de uma formiga k de escolher caminhar o arco ij como conforme Fórmula 4.2:

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

Fórmula 4.2 Fórmula de probabilidade de formiga escolher um caminho (Dorigo, 1996).

Na Fórmula 4.2 temos N_i^k como sendo o subconjunto de N_i que contém somente os nós que ainda não foram visitados pela formiga k , ou seja, os nós que não constam na memória da formiga. Esta probabilidade é demonstrada como uma razão entre o valor de α de um arco e a soma dos valores de α dos arcos que são factíveis, ou seja, que levam a nós que ainda não foram visitados.

Uma vez que todas as k formigas já tenham percorrido seus caminhos em sua totalidade, atualiza-se a quantidade de feromônios nos arcos, onde se comprovar a ocorrência de dois fenômenos: a evaporação e a deposição dos feromônios. Ambos os eventos podem ser modelados por uma mesma equação, conforme foi proposto por (Dorigo, 1996) ilustrado na Fórmula 4.3:

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

Fórmula 4.3 Fórmula de atualização de ferormônio (Dorigo, 1996).

O parâmetro ρ define a velocidade de evaporação dos feromônios. Já o termo $\Delta\tau_{ij}$ é o somatório de todos os $\Delta\tau_{ij}^k$ de cada arco ij , ou seja, é a soma de todas as contribuições individuais de cada formiga k que tenha passado pelo arco ij na iteração em questão; $\Delta\tau_{ij}^k$ é a quantidade de feromônios que a formiga k deposita no arco ij , e é definida como o inverso do comprimento total do percurso que a formiga k percorreu, ou seja, quanto menor comprimento L_k do caminho encontrado, maior será a contribuição em feromônios da formiga k aos arcos que ela percorreu. Um arco que é percorrido por um número grande de formigas, recebe uma quantidade maior de feromônios proporcionais à qualidade das soluções encontradas. A Fórmula 4.4 explana toda a fórmula passo-a-passo.

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{1}{L_k(t)}, & \text{Se } (i, j) \in \text{percurso feito pela formiga } k \text{ na} \\ & \text{iteração } t. \\ 0, & \text{Caso contrário.} \end{cases}$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t), \quad \text{Onde } m \text{ é o número de formigas.}$$

Fórmula 4.4 Fórmula de atualização de feromônio (Dorigo, 1996).

Esta é a idéia fundamental do AS, aplicado ao problema do caixeiro viajante. A continuação verá outra meta-heurística que foi desenvolvida tendo como base o AS conhecida como *Ant Colony System*.

4.2. O ANT COLONY SYSTEM (ACS)

O *Ant Colony System* é uma meta-heurística que aperfeiçoa o *Ant System*, possuindo um desempenho muito superior e possibilitando ser aplicado a problemas de instâncias maiores. Este algoritmo data de 1996, sendo inicialmente desenvolvido por Dorigo e Gambardella (1997).

Segundo Rafael (2006), a principal diferença entre o ACS e o AS refere-se à forma de como é feita a atualização dos feromônios. No caso do AS, ela é realizada de forma global e unificada, após a finalização de cada iteração, realizando simultaneamente a evaporação de feromônios em todos os arcos, além da deposição de novos feromônios por todos os caminhos, proporcionalmente à qualidade dos caminhos encontrados. Já no ACS, a atualização dos feromônios se dá de forma local, após a ação de cada formiga, representando a evaporação dos feromônios pelo seu caminho, quanto de forma global, esta após o fim de cada iteração, premiando apenas o melhor caminho encontrado pelas formigas. Neste caso, existe uma certa quantidade de feromônios proporcional à qualidade da solução encontrada. Outra diferença entre os algoritmos, entretanto, é a regra de decisão das formigas. No modelo AS, esta regra é considerada meramente probabilística. No ACS, a regra é um

pouco mais complexa, sendo dita pseudoaleatória, pois envolve uma parte determinística, ou exploratória, e uma parte probabilística, ou desbravadora, parte esta que é definida nos mesmos moldes do AS. Assim, mesmo no ACS, temos um parâmetro extra que se chama q_0 , variando de 0 a 1, definindo quanto poder de exploração de novos caminhos quer se dar às formigas, sendo ilustrado nas equações das fórmulas dos tópicos 4.2, que são as mesmas do modelo AS. A diferença está porque que o expoente α da variável τ_{ij} , que representa a importância dada à densidade de feromônios em cada arco, não é utilizado, ou seja, é igual a 1.

As equações dos tópicos 4.2 fazem com que a formiga tenha a possibilidade de conhecer novos caminhos, representando o poder desbravador das formigas, viabilizando a descoberta de possíveis rotas de comprimentos menores e mais interessantes. Quanto menor for o valor do parâmetro q_0 , maior é esse poder desbravador. Por outro lado, se o número aleatório sorteado q for menor que q_0 , a formiga tomará sua decisão baseada no conhecimento disponível sob a forma de depósitos de feromônios e distâncias. Em outras palavras, a regra de decisão da formiga será, para todo j pertencente a N_i^k , ou seja, a formiga escolherá o nó j que maximize o valor de a_{ij} .

Outra grande diferença para o modelo AS, é na atualização dos feromônios. O modelo ACS faz uma atualização local em cada arco, ou seja, sempre que uma formiga percorre um arco. Por outro lado, no modelo AS, a atualização de feromônios inclui também na mesma expressão uma atualização global de feromônios, garantindo que fossem feitos diferentes depósitos de feromônios nos arcos, conforme a qualidade das soluções encontradas por cada formiga. No caso do ACS, esta regra é dada pelo reforço de feromônios. Neste caso, não é feito por cada formiga, mas sim de forma unificada e centralizada. Ao final de cada iteração, o melhor caminho encontrado é o único que tem sua densidade de feromônios incrementada, representando uma diferença importante para o modelo AS.

O ACS foi a base para o desenvolvimento do MACS-VRPTW, algoritmo que será visto na seção 5.5, sendo aplicado ao problema de roteirização de veículos com janelas de tempo, este proposto por Gambardella (1999).

4.3. O ANT COLONY OPTIMIZATION (ACO)

O algoritmo da colônia de formigas foi proposto por Dorigo (1996), propondo um método computacional para solução de problemas combinatórios baseado na observação do comportamento das formigas se movendo entre a colônia e o ninho. Como será mostrado a continuação, a inteligência da colônia de formigas que resolve o problema não emerge de um indivíduo, mas sim da soma de todas as experiências de todos os indivíduos da colônia. Fox (2004) demonstra esta característica ao dizer “se você deseja entender a inteligência de enxames não olhe para uma abelha ou formiga específica. Você observa padrões de interação que emergem em um nível macro, inclusive aqueles gerados no nível micro aonde tudo parece caótico”.

Dorigo (1996) e Lerman (2001) enumeram algumas características adicionais do algoritmo de colônia de formigas:

- escalável, pois a arquitetura de uma unidade é a mesma para um milhar de unidades (Lerman , 2001);
- versátil, pois pode ser usado por várias versões do mesmo problema; por exemplo, uma variação do problema do caixeiro viajante conhecida como sendo o problema assimétrico do caixeiro viajante (Dorigo, 1996);
- robusto, podendo ser aplicado com mudanças mínimas à outros problemas de otimização combinatória (Dorigo , 1996);
- dependente de população, permitindo a exploração da realimentação positiva como um mecanismo de busca (Dorigo, 1996).

Tal algoritmo tem sido usado em pesquisas visando a otimização na gerência de redes de comunicação, conforme trabalho de Bieszczad (1998). Parpinelli (2002) apresentam uma aplicação deste algoritmo no campo de mineração de dados (*data mining*).

Talbi (2001) demonstra a simplicidade do agente ao resumir como um conjunto de formigas consegue determinar o menor caminho entre cada ponto: “Enquanto percorrem seu caminho, as formigas deixam uma trilha de feromônio no solo. O papel desta trilha é guiar as outras formigas até o alvo. Para uma formiga, o caminho é escolhido de acordo com a quantidade de feromônio. Posteriormente, a quantidade desta substância química vai diminuindo com a

ação do tempo e a quantidade de feromônio restante depende da quantidade de alimento encontrada e do número de formigas que se utilizaram desta trilha.”

Esta forma de comunicação indireta (através do meio), é uma das características mais sobre salientes deste algoritmo. A dinâmica deste algoritmo é ilustrada na figura 4.1. Com o propósito de ir do ponto A ao ponto B, inicialmente as formigas vão para o caminho da esquerda como para o caminho da direita (figura 4.1(a)). Posteriormente, com a evaporação do feromônio, a trilha da esquerda se torna mais rica em feromônio, pois maior numero de formigas passa por ele em menos tempo (figura 4.1(b)). Após algum tempo, o caminho da esquerda conterá uma maior proporção de feromônios que o caminho da direita, o que levará as formigas a escolhê-lo com maior freqüência.

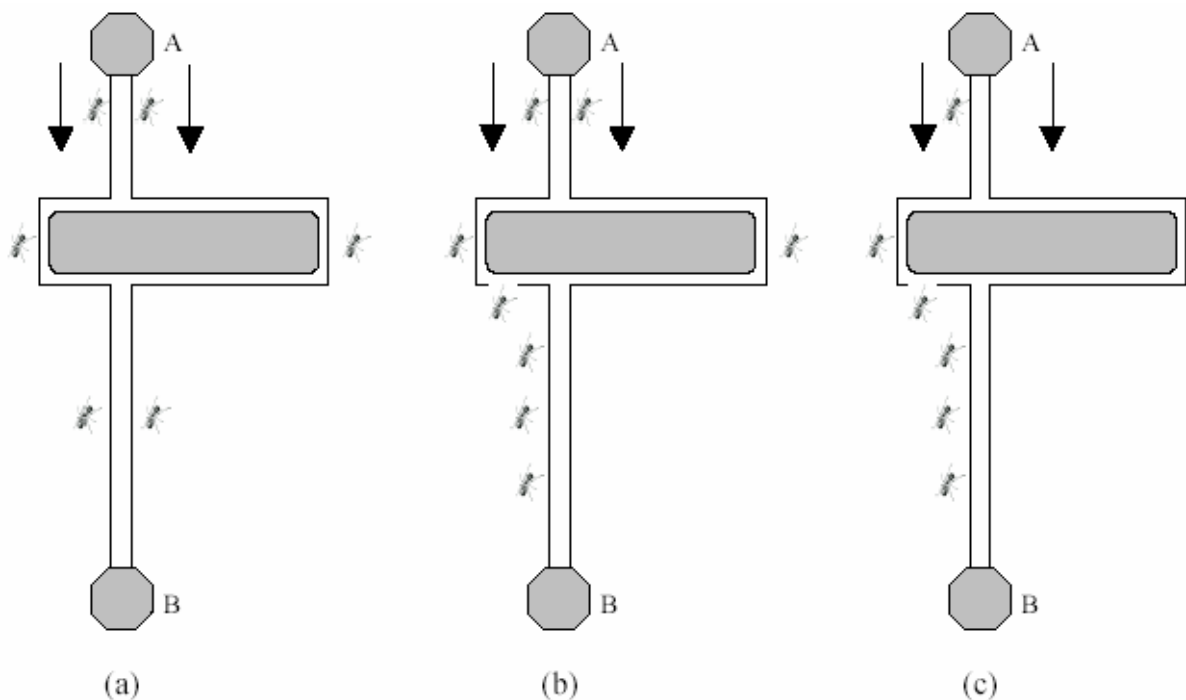


Figura 4.1 Descoberta do menor caminho através de informações de feromônio (Dorigo,1996)

Dorigo e Gambardella (1997) aplicaram o algoritmo da otimização através da colônia de formigas no problema do caixeiro viajante, e compararam os resultados obtidos com outras técnicas como algoritmos genéticos, programação evolucionária, *simulated annealing* e *annealing genetic algorithms*. Os resultados obtidos por Dorigo e Gambardella (1997) são reproduzidos na Tabela 4.1. Nesta

tabela, foram aplicados os algoritmos mencionados em quatro problemas do caixeiro viajante, sendo registrados o tamanho do melhor percurso encontrado.

Os resultados apresentados por Dorigo mostram que o algoritmo da colônia de formigas como sendo recomendável para a resolução do TSP de grande porte. Onde podemos destacar ACS como o sistema de colônia de formigas.

Nome do Problema	ACS	AG	EP	SA	ANG	Melhor resultado
Oliver30 (30 cidades)	420	421	420	424	420	420
Eil50 (50 cidades)	425	428	426	443	436	425
Eil75 (75 cidades)	535	545	542	580	561	535
KroA100 (100 cidades)	21 282	21 761	N/A	N/A	N/A	21 282

Tabela 4.1 Resultado da aplicação de vários algoritmos distintos para problemas do TSP (Dorigo e Gambardella,1997) .

No capítulo seguinte, será apresentado o modelo proposto para a solução do problema baseado no (MACS-VRPTW). Assim mesmo cabe destacar que o modelo proposto esta baseado na necessidade do setor logístico de empresas que precisam de soluções para a roteirização de seus veículos. Assim mesmo cabe destacar que o algoritmo (MACS-VRPTW), tem como característica principal, controlar a restrição de horários de atendimento de clientes também conhecido como janelas de tempo, restrição muito usada no setor logístico.

5. MODELO PROPOSTO COM MACS - VRPTW

O modelo utilizado neste trabalho foi proposto por (GAMBARDELLA, 1999) e é definido num grafo não direcionado e completo, sendo $G = (V, A)$, onde $V = (V_0, V_1 \dots V_n)$ é um conjunto de vértices e $A = \{(V_i, V_j) : V_i, V_j \in V, i \neq j\}$ é um conjunto de arcos. O vértice V_0 representa o depósito, onde uma frota de N_v veículos de capacidade homogênea é alocada e todos os outros vértices representam clientes. Uma matriz não-negativa $C = (C_{ij})$ é definida em A , representando a distância entre os vértices. O problema é simétrico, pois se considera que os custos são iguais em ambas as direções. Associado a cada vértice, há um peso D_i não negativo, representando a demanda do cliente V_i . As rotas devem ser construídas de tal maneira que cada cliente seja atendido por um único veículo. Além disto, todas as rotas devem iniciar e terminar no depósito, além do atendimento do cliente que deve ser realizado dentro do horário estabelecido para atendimento a esse público.

5.1. MODELO PARA SOLUÇÃO DO VRPTW

A função objetivo do modelo proposto está baseada na formulação típica de Bargarosoglu e Ozgur (1999), onde deve-se considerar a restrição de horário de atendimento e consiste na seguinte formulação:

$$\text{Minimizar: } \sum_i \sum_j \sum_v C(i, j) X_{ij}^v \quad (1).$$

Sujeito a:

$$\sum_i \sum_v X_{ij}^v = 1 \forall j \quad (2).$$

$$\sum_j \sum_v X_{ij}^v = 1 \forall i \quad (3).$$

$$\sum_i X_{ip}^v - \sum_j X_{pj}^v = 0 \forall p, v \quad (4).$$

$$\sum_i d_i (\sum_i X_{ij}^v) \leq Q_v \forall v \quad (5).$$

$$\sum_{j=1}^n X_{0j}^v \leq 1 \forall v \quad (6).$$

$$\sum_{i=1}^n X_{i0}^v \leq 1 \forall v \quad (7).$$

$$X_{ij}^v \in Z \leq 1 \forall i, j, v \quad (8).$$

Onde X_{ij}^v são variáveis binárias que indicam se o (v_i, v_j) é utilizado pelo veículo v . A função objetivo de minimização do distância/custo/tempo aparece na equação (1). As restrições representadas nas equações (2) e (3), juntas, garantem que cada vértice é atendido por um único veículo. A equação (4) garante que um veículo deixa o vértice, onde foi atender a demanda, tão logo alcance este vértice. A capacidade do veículo é expressa na equação (5). As restrições (6) e (7) expressam que a disponibilidade do veículo não pode ser excedida. A equação (8) representa a eliminação de *subtour*, onde Z pode ser definido por:

$$Z = \left\{ (X_{ij}^v) : \sum_{i \in B} \sum_{j \in B} X_{ij}^v \leq |B| - 1 \forall B \subset V, |B| \geq 2 \right\}$$

Fórmula 5.1 Eliminação de *subtour* Bargarosoglu e Ozgur (1999).

5.2. ARQUITETURA DO MODELO PARA VRPTW

A arquitetura proposta (Figura 5.1) para a implementação do modelo foi baseada, primeiramente, na geração da rede, onde se faz a aquisição da informação de entrada, a partir da seleção de pontos geográficos no mapa. Posteriormente, gera-se uma solução inicial através da heurística Nearest Neighbour conforme proposto por (GAMBARDELLA, 1999), sequentemente se procura uma solução mais ótima através da técnica de pesquisa a ser estudada: colônia de formigas, assim executando-a e gerando os resultados desejados.

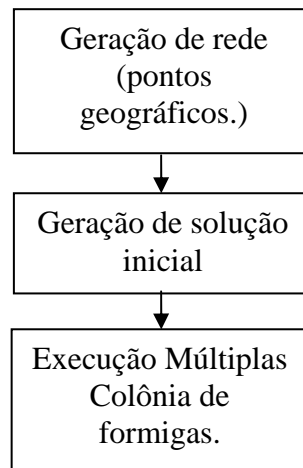


Figura 5.1. Arquitetura do Modelo Proposto.

Serão descritos os módulos que integram a arquitetura proposta com maiores detalhes no tópico seguinte.

5.3. MÓDULO DA GERAÇÃO DA REDE

Este módulo é o responsável pela geração das redes que serão utilizadas no módulo que gera a solução inicial. Nesta aplicação, será realizada a aquisição de dados provenientes da seleção de pontos geográficos diretamente do mapa, os quais representaram um grupo de clientes com um depósito, onde será iniciada a saída dos veículos. Estas informações compreenderão as referências de latitude e longitude dos clientes e do depósito, assim como as distâncias entre estes.

Para cada par de vértices com coordenadas $(lat_1 - ln g_1)$, $(lat_2 - ln g_2)$ é calculada a distância entre pontos. O resultado deste módulo é uma matriz com as distâncias entre quaisquer dois vértices. A continuação a fórmula 5.2 que mostra a distância entre dois pontos.

$$d(P, Q) = \sqrt{(lat_1 - lat_2)^2 + (ln g_1 - ln g_2)^2}$$

Fórmula 5.2 Fórmula para achar a distância entre dois pontos.

5.4. MÉTODOS HEURÍSTICOS PARA ENCONTRAR A SOLUÇÃO INICIAL

Este módulo utiliza a rede obtida na etapa anterior para gerar uma Solução Inicial para o problema.

A heurística, que será usada para achar a solução inicial, é o *Nearest Neighbor* aplicado ao TSP. Para o caso do VRP, a idéia principal é a mesma, no entanto deve-se ajustar conforme as restrições do problema. No *Nearest Neighbor* também conhecido como o método do “Vizinho mais Próximo” (Tyagi,1968), (Cook,1998), onde são ordenados são os nós adjacentes ao nó atual, em ordem decrescente de custos para serem alcançados. Sendo assim, para cada nó visitado, será criada uma nova lista dos nós mais próximos, ou seja, os de menor custo para serem alcançados. Será utilizado o termo “distância” entre dois nós para indicar o custo da aresta que os conecta, apesar de sabermos que este custo está relacionado ao tempo necessário para se percorrer tal arco. O algoritmo pode ser descrito por:

1. Construa uma lista com os nós adjacentes ao nó atuais ainda não visitados, em ordem decrescente de distância.
2. Tome desta lista o nó *i* com o menor custo
 - a. Se o nó *i* puder ser incluído na rota atual sem que a solução se torne infactível
 - i. Inclua o nó *i* na rota
 - ii. Defina o nó *i* como o nó atual e reinicie o algoritmo
 - b. Se o nó *i* não puder ser incluído na rota atual.
 - c. Elimine o nó *i* da lista de nós adjacentes ao nó atual e retorne ao passo 2.
3. Se a lista estiver vazia, tome o depósito (inclua o nó inicial na rota), e reinicie as restrições de capacidade, tempo de ciclo, etc., com o depósito como nó atual.
4. Se não houver mais nós a serem visitados, o algoritmo está terminado.

Esta é a forma mais simples e intuitiva do *Nearest Neighbor*. Entretanto, para abordar o problema de forma mais eficiente, é preciso fazer algumas alterações neste algoritmo. Para considerar as restrições de janelas de tempo, é necessário alguns ajustes para um bom funcionamento do algoritmo. Estes ajustes estão relacionados ao custo da distância, cujo valor deve ser ponderado para sua

posterior seleção. Esta escolha leva em conta, não só o custo de se percorrer o arco, mas também o tempo que pode levar, no caso, se chegue a este dado nó antes do início de sua janela de tempo. Além disto, considera-se também a urgência de se atender a este nó, devido à proximidade do fim de sua janela de tempo. Conforme sugerido por Pellegrini (2005), deve-se dar pesos para cada um destes aspectos. Conforme Figura 5.2, onde se representa duas situações distintas: na primeira, há espera e na segunda não.

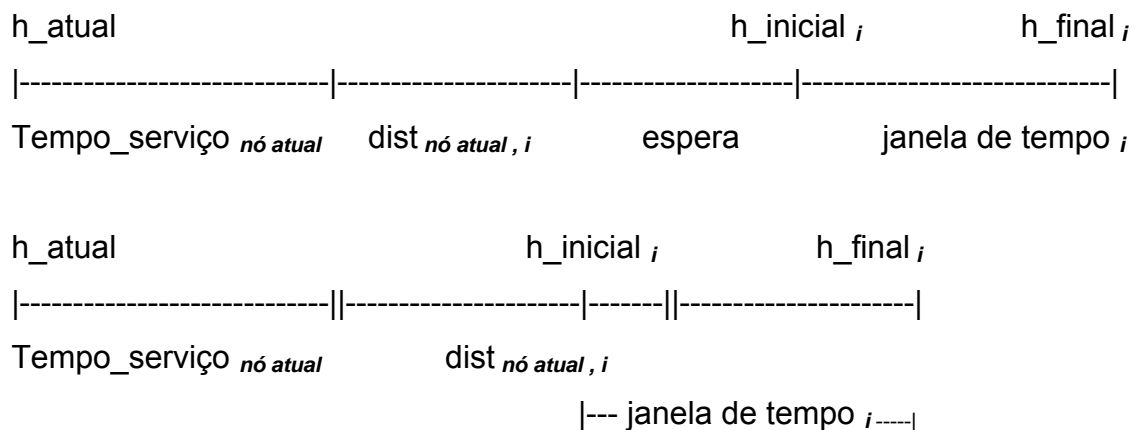


Figura 5.2. Intervalos no tempo do ciclo de atendimento, para o VRPTW.

A espera ligada ao nó i pode ser caracterizada por: $(h_inicial_i - (h_atual + tempo_serviço_{nó_atual} + dist_no_atual, i))$, onde

- $h_inicial_i$ é o horário de início da janela de tempo do nó i ,
- h_atual é a hora atual (que corresponde ao início do serviço do nó atual),
- $tempo_serviço_{nó_atual}$ é o tempo que o nó atual leva para realizar seu serviço, e
- $dist_{nó_atual, i}$ é a distância do nó atual para o nó i . É importante lembrar que, para o VRP, esta distância i deve ser entendida como o tempo necessário para se percorrer o arco $nó_atual - i$. Se esta equação der um resultado de um número negativo, tem-se uma situação em que não há espera, ou seja, define-se $espera = 0$. Formalmente, tem-se então:

$$espera = \text{Max}((h_inicial_i - (h_atual + tempo_serviço_no_atual + dist_no_atual, i)), 0)$$

Dessa forma, quando

$(h_atual + tempo_serviço_{no_atual} + dist_{no_atual, i}) > h_inicial_i$, tem-se que $espera = 0$, situação esta que corresponde à segunda situação na Figura 5.2.

Define-se urgência como:

$(h_final_i - (h_atual + tempo_serviço_{no_atual} + dist_{no_atual, i}))$, sendo ela positiva se for possível alcançar até o nó i antes que sua janela de tempo se feche. Na figura 5.2, ambas as situações mostram uma urgência positiva, pois o final da janela de tempo i é posterior a $(h_atual + tempo_serviço_{no_atual} + dist_{no_atual, i})$, ou seja, é possível atender o cliente i dentro da janela de tempo. Dessa forma, para o algoritmo de vizinho mais próximo (*Nearest Neighbor*), deve-se procurar pelo nó que minimize um certo valor $x = a * distância + b * espera + c * urgência$, onde a , b e c são parâmetros, e ainda que apresente $urgência > 0$, que ainda não tenha sido visitado e que não viole quaisquer outras restrições do problema, como capacidade, tempo de ciclo, entre outros.

Para execução do algoritmo, é necessário definir os dados de entrada que representem as características físicas do problema, bem como os dados das janelas de tempo e capacidade de carga dos veículos. Sendo assim, usaremos como argumentos: uma matriz $N \times N$, contendo todos os custos C_{ij} entre todos os pares de nós do problema (matriz dist); dois vetores $1 \times N$, contendo os valores iniciais b_i e finais e_i das janelas de tempo de cada um dos N nós (vetores $h_inicial$ e h_final); um vetor $1 \times N$ com as demandas d_j de cada nó, lembrando que a demanda do nó 1, que convencionamos ser o depósito, deve ser igual a zero (vetor demanda); uma variável com o valor da capacidade máxima de carga dos veículos, considerados todos iguais (cap); finalmente, um vetor $1 \times N$ contendo os tempos de serviço s_i de cada nó, incluindo do depósito (vetor tempo_serv).

A saída do algoritmo será composta de três partes: o vetor caminho, contendo a seqüência de nós do melhor caminho encontrado; a variável *comp*, com o custo total de tal caminho (comprimento total do caminho, que conforme mencionado acima, é equivalente ao tempo total que um veículo precisará rodar para percorrer o caminho encontrado) e a variável *carros*, com o número de veículos utilizados na solução encontrada. Cabe mencionar que a variável *comp* representa o tempo total em trânsito, que é proporcional à distância total percorrida pelos veículos. Se dobrarmos o valor de *comp*, estaremos aumentando em dobro o tempo de trânsito, além de dobrar a distância total percorrida. O valor de *comp* poderia indicar o custo com combustível, a quilometragem total dos carros, o número de horas trabalhadas, entre outros. Dessa forma, a saída do algoritmo permite compreender o problema de diversas formas. Entretanto, o que se deseja minimizar juntamente com o número de veículos utilizados é o comprimento total das rotas, que corresponde ao somatório dos custos C_{ij} de cada aresta percorrida, desconsiderando os tempos de serviço de cada nó. A isto corresponde a variável *comp*.

5.5. APLICAÇÃO DO MULTIPLE COLONY SYSTEM (MACS - VRPTW)

O MACS-VRPTW é uma extensão do ACS para o problema de roteirização de veículos com janelas de tempo (RAFAEL, 2006). O objetivo do problema consiste em minimizar, tanto a quantidade de veículos utilizada, quanto a distância total percorrida por eles. Estes objetivos são de caráter hierárquico, sendo o primeiro mais importante. Esta hierarquia é comumente empregada na literatura, sendo aqui também utilizada. A idéia fundamental do modelo reside na geração de duas colônias de formigas, cada uma visando a otimização de um dos objetivos do problema. As melhores soluções são compartilhadas entre as duas colônias, mas as informações de feromônios são independentes. Dessa forma, ambas as colônias trabalham de forma distinta, porém simultânea, na procura por soluções mais interessantes, tendo, os resultados de uma colônia, grande influência nos caminhos tomados pela outra. Esta é a essência do MACS-VRPTW (RAFAEL, 2006). Esta é a idéia proposta por Gambardella

(1999) a qual se encontra ilustrada pela Figura 5.3 onde se destaca as rotinas principais do algoritmo: ACS_VEI e ACS_TIME.

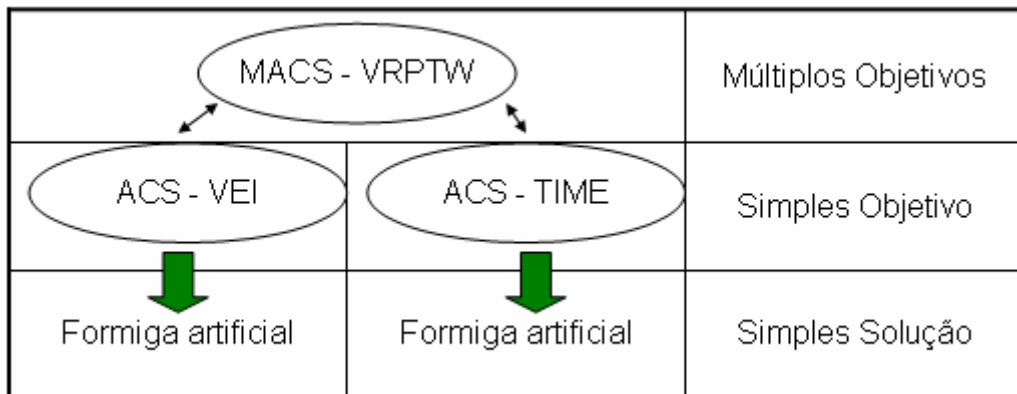


Figura 5.3. Algoritmo MACS-VRPTW proposto por (GAMBARDELLA ,1999).

Para simplificar o desenvolvimento, pode-se desenvolver um algoritmo que trabalhe por ciclos, nos quais as colônias serão ativas alternadamente. Assim, a solução encontrada por uma é usada pela outra, e vice-versa. Basicamente, cada colônia será simulada por uma função, que deverá construir rotas e analisar a qualidade destas.

A diferença nos objetivos das duas colônias se focará, principalmente, na forma com que esta análise dos caminhos será feita. Entretanto, a construção dos caminhos em ambas as colônias será realizado de forma parecida, já que a regra de decisão de cada formiga é sempre a mesma, independentemente de qual colônia ela seja. Sendo assim, ambas as colônias usarão uma mesma função auxiliar para a construção das rotas.

Este processamento das rotas, percorrido por cada formiga por esta função auxiliar, será realizado igual ao funcionamento do ACS, segundo o qual uma matriz de distâncias/custos e uma de feromônios é analisada na procura pelo arco mais atraente. Entretanto, como o VRP envolve rotas múltiplas, e não mais um único caminho passando por todos os nós, é necessário que o processamento das matrizes seja adaptado. Isto é feito através da inserção de depósitos virtuais nestas matrizes. Estes depósitos são cópias idênticas do depósito real, nas mesmas coordenadas geográficas, e as matrizes deverão possuir tantos depósitos quantos veículos forem utilizados. Dessa forma, o funcionamento do ACS no VRP é equivalente àquele no TSP, resultando em um caminho que incluirá, uma seqüência proposta de nós.

Quando um veículo regressa ao depósito, a capacidade e o tempo decorrido são zerados, possibilitando o início de uma nova rota. O modelo precisa ser inicializado com alguma solução inicial. Isto normalmente é feito com um método simples, como o *Nearest Neighbour* visto anteriormente.

A solução inicial passa pelo chamado ACS_VEI, que é um algoritmo que procura encontrar alguma solução factível, utilizando um veículo a menos que na solução inicial. Este algoritmo usa uma primeira colônia de formigas. Posteriormente, a solução encontrada passa para a segunda colônia de formigas, conhecido como o algoritmo ACS_TIME, que tem como objetivo minimizar o tempo total gasto pelo dado número de veículos encontrado pela primeira colônia. Cabe recordar aqui que, minimizar o tempo total é o mesmo que minimizar a distância total percorrida. Terminada a execução deste segundo algoritmo, o ciclo é reiniciado. Esse ciclo continua até que algum critério de parada seja atingido (RAFAEL, 2006).

A rota ótima obtida pelo processamento do algoritmo mencionado será visualizada numa interface do sistema que permitira acompanhar a rota em formato georeferenciado.

6. ARQUITETURA DO SISTEMA

Neste capítulo são expostas as especificações do ambiente baseado na arquitetura dual conforme foi citado no capítulo 2. Esta arquitetura está focada no desenvolvimento de um sistema para plataformas SIG para apresentar as informações de roteamento baseado nas informações geográficas em formato de mapa.

6.1. ESPECIFICAÇÃO DA ARQUITETURA

Um SIG com arquitetura Dual utiliza um SGBD relacional para armazenar os atributos convencionais dos objetos geográficos na forma de tabelas. No modelo relacional, os registros são organizados em forma de tabela: onde as linhas correspondem aos dados e as colunas correspondem aos atributos.

A entrada dos atributos não-espaciais é feita por meio de um SGBD relacional e para cada entidade gráfica inserida no sistema é imposto um identificador único ou rótulo, através do qual é feita uma ligação lógica com seus respectivos atributos não-espaciais armazenados em tabelas de dados no SGBD, como ilustrado na Figura 6.1.

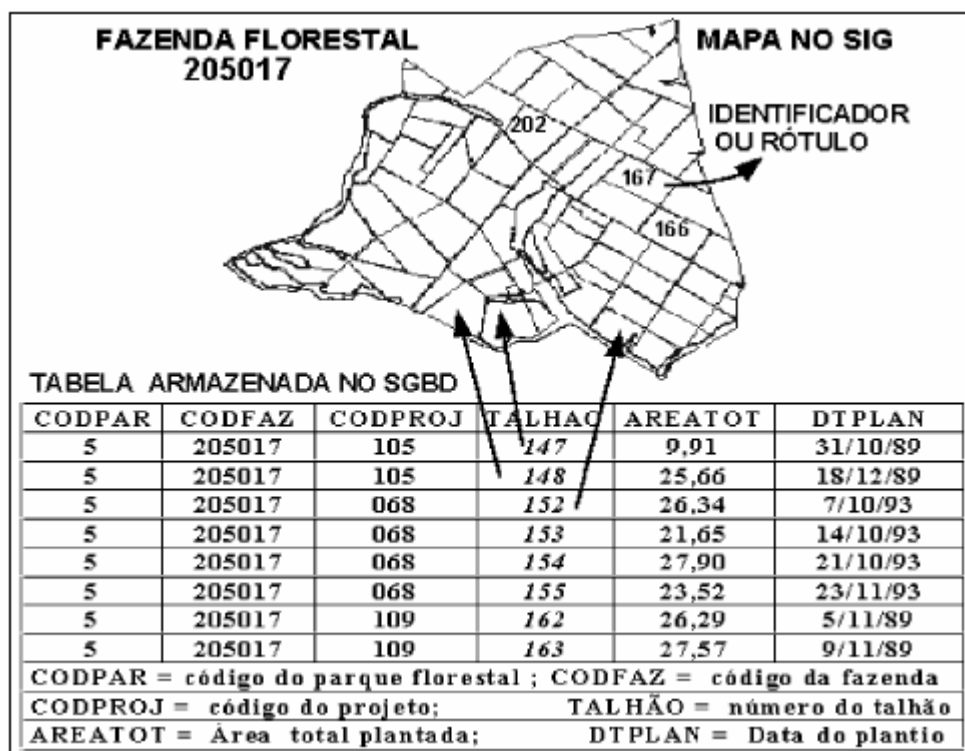


Figura 6.1. Arquitetura Dual para um sistema (LISBOA, 2001).

A Figura 6.1 ilustra as ligações lógicas criadas entre os rótulos dos talhões de um mapa florestal e seus atributos correspondentes (registros no “campo” ou coluna TALHÃO) numa tabela de banco de dados. O mesmo tipo de relacionamento lógico pode ser feito em outros casos, como por exemplo: moradores em um lote, lotes em uma quadra, quadras em bairro, bairros em uma cidade; hidrantes de segurança ou telefones públicos ao longo de uma avenida; postos de serviço e restaurantes ao longo de uma rodovia.

A principal vantagem desta estratégia é poder utilizar os SGBDs relacionais de sistemas transacionais. Exemplos de sistemas comerciais baseados em estratégia dual são o ARC/VIEW, MGE e o SPRING (CÂMARA, 2001).

Nesta arquitetura os dados serão acessados num formato vetorial o que permite que os objetos do banco de dados possam ser acessados a partir de consultas SQL, e assim trabalhar com dados complexos.

As características deste tipo de arquitetura são:

- Fornecer tipos de dados espaciais, como ponto, linha e região;
- Estender a linguagem de consulta SQL para suportar operações e consultas espaciais;
- Adaptar outras funções de níveis mais internos para manipular eficientemente, tais como métodos de armazenamento e acesso (indexação espacial) e métodos de otimização de consultas (junção espacial).

Existem basicamente três extensões comerciais disponíveis no mercado: Oracle Spatial (Ravada e Sharma, 1999), IBM DB2 Spatial Extender (IBM, 2001), Informix Spatial Datablade (IBM, 2002) e o projeto de extensão espacial chamada PostGIS (Ramsey, 2002) para o SGBD PostgreSQL, que é objeto-relacional, gratuito e de código fonte aberto o qual será usado neste trabalho.

6.2. CARACTERÍSTICAS DA ARQUITETURA

A arquitetura de desenvolvimento está dividida em duas grandes áreas:

6.2.1. PLATAFORMAS DO SISTEMA SIG

A plataforma SIG é composta por diferentes tecnologias em sua maioria de código aberto, estas tecnologias permitem poder realizar a leitura e a apresentação em camadas de informações geográficas a partir de informações vetoriais armazenadas num banco de dados relacional. Para esta plataforma do sistema SIG será usado as seguintes tecnologias:

- Arquivos em formato Shapes da cidade de Joinville. Estes arquivos serão transformados para PostgreSQL. Estes arquivos tem as informações geográficas organizadas em formato de registros onde os objetos geográficos se encontram nas filas e a descrição de estes objetos se encontram nas colunas.
- Software Opensource QuantumGIS 0.9.2 também de código de fonte aberto, que auxiliará na conversão dos arquivos *shapes*.
- Banco de dados relacional PostgreSQL 8.0 com suporte para sistemas SIG de código de fonte aberto Postgis 3.1.: nele serão armazenadas as informações geográficas provenientes dos arquivos *shapes*.
- Servidor de mapas Opensource MapServer 4.8.3 + PHP 5.0 para disponibilizar as informações.
- Servidor WEB : Apache version 2.0.58
- Servidor IIS para disponibilizar as interfaces do sistema de roteamento.
- Software Opensource GeoServer 1.6.3 para sobreposição de camadas de informações geográficas.
- API Google Maps 2.0 para apresentação dos dados e dos mapas.

Figura 6.2 ilustra a plataforma do sistema SIG descrito neste trabalho.

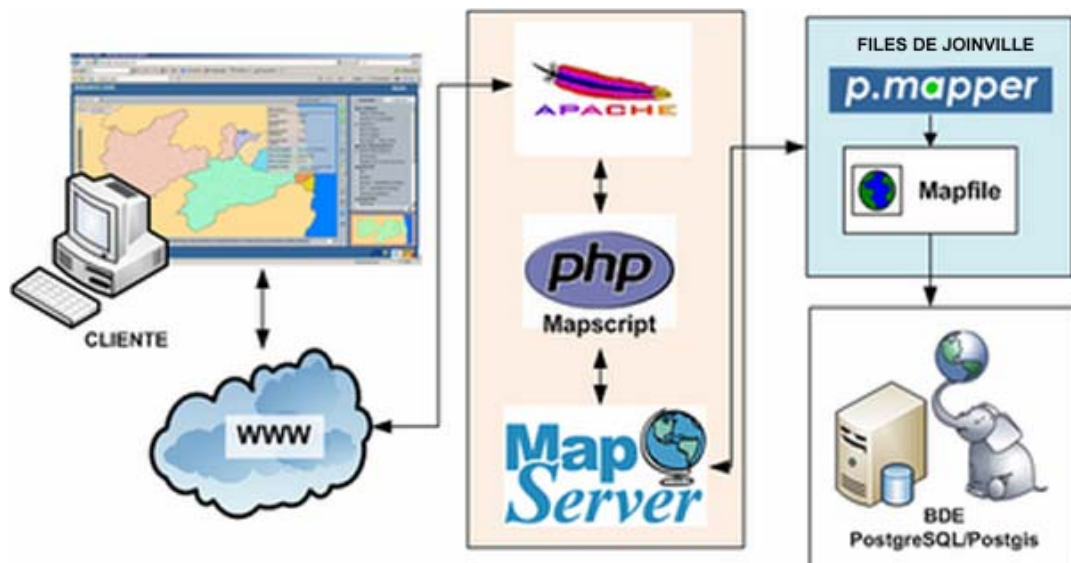


Figura 6.2. Arquitetura da plataforma SIG.

6.2.2. ALGORITMOS DE ROTEIRIZAÇÃO

Os algoritmos de roteirização estão projetados para trabalhar a partir das informações geográficas (latitude e longitude) provenientes dos mapas, onde os algoritmos realizaram seu processo e devolveram o resultado para ser apresentado pelo sistema em formato de camada vetorial.

O desenvolvimento dos algoritmos de roteirização será dividido nos seguintes módulos:

- Módulo de aquisição de dados por leitura de um arquivo físico ou por informação de entrada por usuário. Conforme se ilustra na Figura 6.3



Figura 6.3. Interface entrada de dados pelo usuário.

Os balões (amarelos e azuis) na figura 6.3 são os locais de entrega definidos pelo usuário no mapa – cada balão gera coordenadas geográficas no quadro ao lado. O conjunto dessas coordenadas constitui a matriz sobre a qual se aplicará o algoritmo Nearest Neighbor para obtenção da rota. A rota obtida é o tracejado em negrito na figura 6.4 – o traço destacado foi sobreposto ao mapa existente.

- Módulo de processamento que será encarregado de analisar as informações conforme os algoritmos apresentados no próximo capítulo.
- Módulo de entrega de resultados: permite o salvamento dos resultados num arquivo texto (para posterior leitura do sistema SIG).

A Figura 6.4 ilustra a apresentação do resultado do algoritmo de roteirização aplicado ao mapa.



Figura 6.4. Interface apresentando o resultado de uma roteirização.

6.3. DOCUMENTAÇÃO

As informações contidas em banco de dados se encontram organizadas conforme modelo da Figura 6.5, onde se ilustra o relacionamento das informações entre as tabelas das informações geográficas. Adicionalmente às informações relacionadas, há o campo *the_geom* que armazena as informações de coordenadas geográficas.

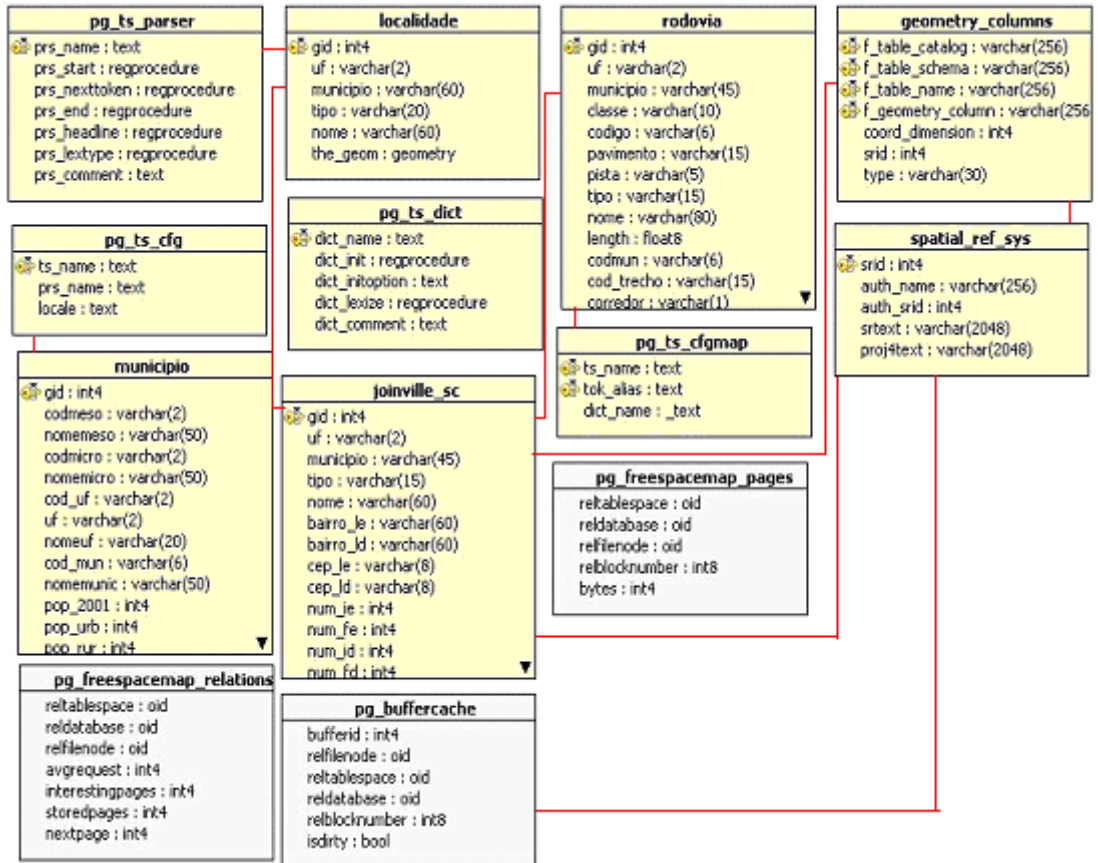


Figura 6.5. Ilustração da modelagem de dados do banco de dados geográfico.

A organização de classes do pacote principal se encontra ilustrada na Figura 6.6 onde se destaca o relacionamento de classes entre a classe PontosRouting e a classe Rotas. Tais classes se encarregam de representar os objetos de roteirização propriamente ditos. As outras classes são auxiliares ao processamento de roteirização.

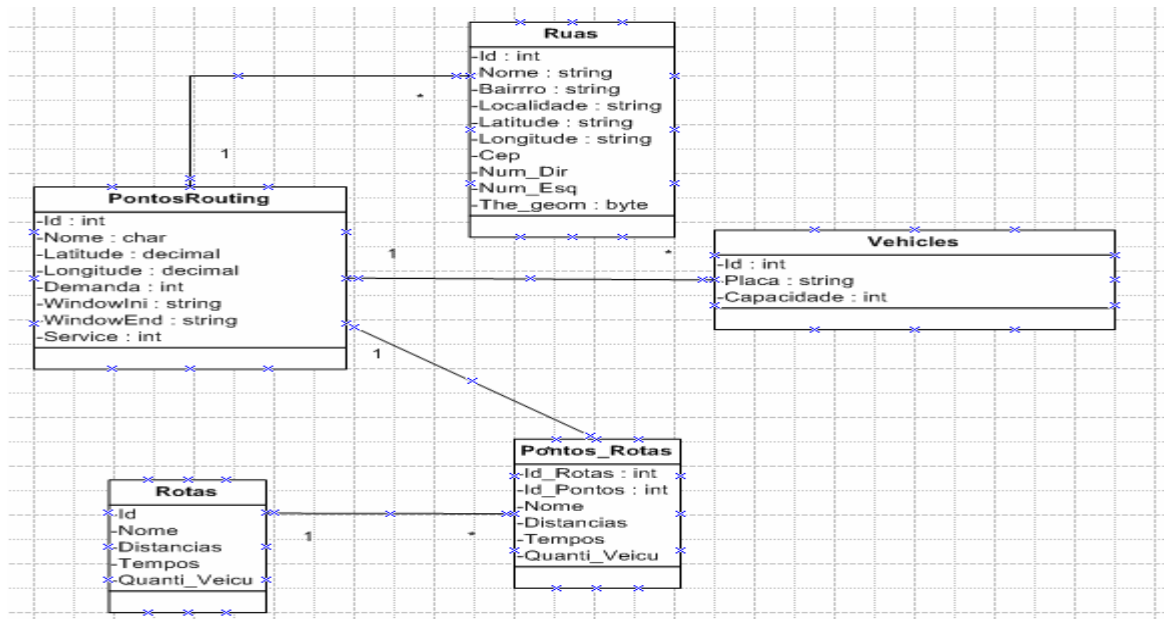


Figura 6.6. Modelo de Classes do pacote principal.

7. ALGORITMO PARA A ROTEIRIZAÇÃO DE VEÍCULOS.

Esse capítulo descreve funcionamento do algoritmo proposto para o problema de roteirização de veículos com janelas de tempo. A formalização do problema foi feita no capítulo 5.1.

- Primeiramente, é preciso definir de que forma serão os dados de entrada do algoritmo MACS-VRPTW, e qual será a forma de sua saída. Os dados de entrada serão obtidos a partir de uma interface no mapa para a seleção de coordenadas geográficas assim como os dados das janelas de tempo e capacidade de carga dos veículos. Sendo assim, estes dados serão:
 - Uma matriz $N \times N$ com todos os custos C_{ij} entre todos os pares de nós do problema;
 - Dois vetores $1 \times N$ contendo os valores iniciais e finais das janelas de tempo de cada um dos N nós;
 - Um vetor $1 \times N$ com as demandas de cada nó, lembrando que a demanda do nó 1, que for convencionado como depósito, deve ser igual a zero;
 - Uma variável com o valor da capacidade máxima de carga dos veículos, considerados todos com mesma capacidade;

- Finalmente, um vetor $1 \times N$ contendo os tempos de serviço si de cada nó, incluindo o depósito. Estes dados são suficientes para identificar completamente o problema.
- O algoritmo, por sua vez, é composto de três saídas:
 - O vetor caminho, contendo a seqüência de nós do melhor caminho encontrado;
 - A variável comp, com o custo total de tal caminho (comprimento total do caminho, é equivalente ao tempo total que um veículo precisará rodar para percorrer o caminho encontrado)
 - E a variável carros, com o número de veículos utilizados na solução encontrada.

Cabe ressaltar que a variável comp representa o tempo total em trânsito, que é diretamente proporcional à distância total percorrida pelos veículos; se dobrar o valor de comp, ou seja, se dobrar o tempo de trânsito, também seria dobrada a distância total percorrida, dado que nossos veículos são considerados como tendo uma velocidade média constante.

Dessa forma o valor da variável comp nós poderá auxiliar a encontrar outros tipos de informações como, por exemplo, o custo com combustível ou a quilometragem total dos carros.

Entretanto, o que se deseja minimizar juntamente com o número de veículos utilizados é o comprimento total das rotas, que corresponde ao somatório dos custos C_{ij} de cada aresta percorrida, desconsiderando os tempos de serviço de cada nó. A isso corresponde a variável comp.

Inicialmente, com base nas seis estruturas de dados do problema, é chamada a função auxiliar `nearest_neighbour`, conforme foi explicado na seção 5.4 do trabalho o qual determina um caminho factível aplicada ao VRP. Sendo assim, as variáveis de saída caminho, comp e carros são inicializadas com os valores referentes ao caminho encontrado pelo Nearest Neighbour. A continuação se apresenta o diagrama de blocos do algoritmo na Figura 7.1

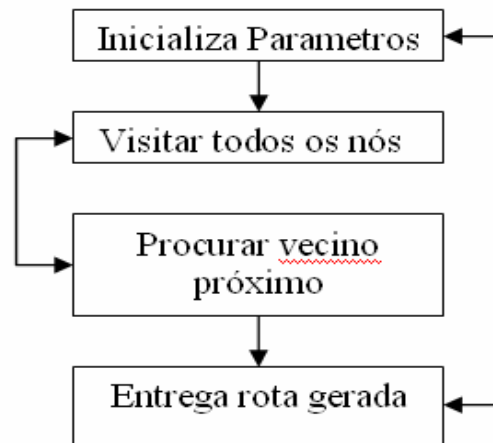


Figura 7.1. Diagrama de Blocos do algoritmo Nearest Neighbour.

7.1. ROTINA PRINCIPAL DO MACS-VRPTW

Primeiramente, com base nos seis dados do problema é executado o algoritmo Nearest Neighbour, que determina um caminho factível. Sendo assim as variáveis caminho, carros e comp são inicializadas com os valores referentes ao caminho encontrado pelo algoritmo Nearest Neighbour. Este algoritmo é o encarregado de chamar ciclicamente aos algoritmos de acs-vei e acs-time conforme se demonstra na Figura 7.2.

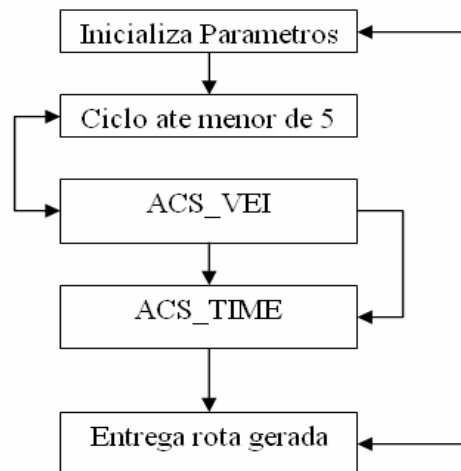


Figura 7.2. Diagrama de blocos da Rotina Principal do MACS-VRPTW.

Conforme o algoritmo apresentado a variável `flag` fornece um critério de término para a rotina. Onde o ciclo do `acs_vei` e do `acs_time` seja realizado por um dado número de vezes sem que seja encontrada uma solução factível com um número de veículos inferior ao utilizado atualmente, a rotina é terminada, e o melhor caminho até o momento é fornecido como resposta, bem como seu comprimento e o número de carros utilizados.

7.2. IMPLEMENTAÇÃO DO ACS-VEI

Abaixo segue o algoritmo criado para a função `acs-vei` a qual recebe os mesmos dados de entrada da função `nearest neighbour` (vetor caminho), seu comprimento total (variável `comp`), o número de veículos que este caminho utiliza (variável `carros`), o flag de finalização de execução, e o comprimento do caminho encontrado pelo `Nearest Neighbour` (variável `comp_nn`, que será usada nas atualizações de ferormônios). Na primeira vez que o ciclo principal do `MACSVRPTW` é executado, estes dados referentes ao melhor caminho encontrado até o momento dizem respeito ao caminho encontrado pelo `Nearest Neighbour`, fazendo com que o valor da variável `comp_nn` seja igual ao valor da variável `comp`. Conforme o ciclo é executado, entretanto, estas informações vão sendo atualizadas com as soluções encontradas.

Na inicialização da função, deseja-se definir um caminho, mesmo que infactível, composto por $(\text{carros} - 1)$ veículos, para posteriormente avaliar-se se é possível chegar a uma solução factível a partir deste caminho, com este número reduzido de veículos. Para esse fim, o primeiro `while` é introduzido. Se existem M veículos na solução inicial caminho, significa dizer que o nó correspondente ao depósito (o nó 0 por convenção) aparece $M + 1$ vezes na seqüência de nós correspondente ao vetor caminho, pois é preciso que todas as rotas se iniciem e terminem no depósito. Sendo assim, este primeiro `while` descobre qual a posição j do vetor caminho que abriga o M -ésimo nó 0, descartando os nós seguintes, que correspondem à última rota e ao último retorno ao depósito.

Por exemplo, se possuímos uma solução `caminho = [0 3 4 0 2 6 7 0 5 8 0]`, temos que `carros = 3` (pois existem 3 rotas, sendo elas 0-3-4-0, 0-2-6-7-0 e 0-5-8-

0); pode-se ver que o depósito aparece 4 vezes no caminho, pois existem 3 veículos em uso.

Na inicialização, é definido o parâmetro ρ , que representa a importância do caminho mais recente frente ao conhecimento acumulado referente aos caminhos passados. Esta variável é o ρ que será usado na atualização dos feromônios.

Podemos destacar que para as instâncias do problema o valor escolhido foi de 0.1, ou seja, a cada evaporação de feromônios, cai em um 10%; no caso da deposição de feromônios, dá-se um peso de 90% ao valor de feromônios atual e 10% ao valor relacionado ao melhor caminho encontrado na iteração em questão. Podemos destaca que quando maior o valor de ρ , mais rápida acontece a evaporação de feromônios, e menos peso se dá para as informações passadas.

No ciclo principal da função, define-se o número máximo de iterações que serão executadas. Escolheram-se o valor $cont = 20$ para tal. Cada uma destas iterações será composta da simulação de $k = 10$ formigas. Basicamente, para cada formiga simulada, uma função auxiliar `simula_formiga` é chamada. Esta função recebe como argumentos as variáveis ρ , $comp_nn$, o vetor IN , as matrizes $dist_acs$ e feromônios, os vetores $h_inicial$, h_final , $demanda$ e $tempo_serv$, e a variável cap . Ou seja, a função recebe as informações mais recentes sobre as concentrações de feromônios e os dados físicos do problema. Esta função calcula então, através das regras de decisão das formigas, o caminho que ela percorreria pelo problema, e fornece como resposta um caminho chamado de candidato, bem como seu comprimento. Esta função também será utilizada pela outra colônia de formigas, pois a regra de decisão das formigas, e conseqüentemente a forma com que os caminhos são construídos, não muda de uma colônia para outra. A diferença entre as colônias está na forma com que estas soluções fornecidas pelo `simula_formiga` são analisadas e avaliadas, como foi explicado anteriormente.

7.3. IMPLEMENTAÇÃO DO ACS-TIME

O Algoritmo criado para a função *acs-time* tem muita semelhança com o algoritmo *acs-vei*, onde seu objetivo é de procurar uma rota factível em cima da rota com um veículo a menos encontrado pela função *acs-vei*. A função *acs-time* tem os mesmos argumentos que a função *acs-vei* assim como as saídas dos vetores caminhos e as variáveis *carros* e *comp*. Na inicialização do algoritmo o *flag* é incrementado em uma unidade, desta forma se finaliza a lógica do funcionamento do algoritmo. Se a função *acs_vei* obtiver melhorias, *flag* é subtraído de uma unidade; em seguida, a função *acs_time* incrementa-o de uma unidade. Enquanto houver melhorias no *acs-vei*, este ciclo continua. No momento que o *acs_vei* não for mais capaz de obter melhorias, a variável *flag* será incrementada seguidamente pelo *acs_time*, até que alcance o valor máximo definido na rotina principal, o que fará com que o MACS-VRPTW seja terminado.

7.4. FUNÇÃO SIMULA FORMIGA

Vamos a descrever a função *simula formiga* que será chamada desde as duas funções principais o *acs-vei* e o *acs-time*. A função *simula_formiga* tem por objetivo definir o caminho que uma formiga percorrerá, com base nas informações atuais de feromônios que são parte dos argumentos da função e nas regras de decisões aleatórias apresentadas. Esta função recebe como argumentos as matrizes de distâncias (*dist_acs*) e de feromônios (*feromônios*) com as dimensões expandidas de acordo com o número de depósitos virtuais que serão usados. Dessa forma, a função *simula_formiga* deve simplesmente percorrer os nós, incluindo os depósitos virtuais, segundo a regra de decisão das formigas, de forma a compor um caminho. O caminho encontrado pode ser ou não factível, ou seja, pode não passar por todos os nós. Entretanto, as consequências da não-factibilidade do caminho encontrado são analisadas pelas funções que chamam o *simula_formiga*, especificamente pela *acs_vei* e pela *acs_time*. No caso da *acs_vei*, por exemplo, a *simula_formiga* é chamada para tentar encontrar um caminho com um veículo a menos que o atual. Caso a

resposta da *simula_formiga* seja um caminho não factível, mas visitando mais clientes que o caminho atual, a solução é mantida pelo *acs_vei*. No caso do *acs_time*, entretanto, só interessam as soluções factíveis. Dessa forma, se a resposta da *simula_formiga* for um caminho não-factível, esta será descartada. Cabe destacar o uso dos parâmetros *q0*, *beta* e *const*.

- O primeiro é o parâmetro que define a probabilidade de cada formiga ser exploradora, bem como a probabilidade da formiga se utilizar do conhecimento passado para tomar suas decisões. Isto está de acordo com o descrito na seção 5, quando o ACS foi detalhado. Quanto maior for o valor de *q0*, menos desbravadora será a formiga.
- O parâmetro *beta*, por sua vez, define a importância relativa do comprimento ponderado dos arcos frente às concentrações de feromônios.
- A variável *const* é somente um indicativo de que já se chegou ao fim da construção do caminho, que será fundamental para a terminação da execução da função.

Assim mesmo a definição de atratividade desde o ponto de vista de uma formiga é uma espécie de comprimento ponderado dos arcos. Ao invés de olharmos simplesmente para os comprimentos, deseja-se levar em consideração também o possível tempo de espera que possa incorrer devido ao início das janelas de tempo, bem como a urgência devido à proximidade do fim das janelas de tempo dos clientes.

7.5. FUNÇÃO INSERTION PROCEDURE

Será abordado o funcionamento da função *insertion_procedure* que foi chamada pelo *simula_formiga* na tentativa de chegar a uma condição de factibilidade de uma solução. Inicialmente, cria-se o vetor *qtd* de demandas, que será igual ao vetor *demandas*, com a diferença de que as demandas que já foram atendidas pelo *caminho_atual* estão zeradas. Dessa forma, o vetor indica quais nós ainda precisam ser visitados. A factibilidade desta configuração é testada, com o auxílio dos contadores *h* e *c*, respectivamente de tempo decorrido e de capacidade utilizada de carga. Se a solução for infactível, um flag é acionado com o valor um, para indicar tal fato. Se for factível, avalia-se o novo

comprimento do caminho. Se ele for menor que o melhor caminho encontrado até o momento, este novo caminho é definido como o *novo_candidato_temp*. Reinicia-se então o ciclo de inserção do nó, tentando-se inseri-lo em outra posição do vetor caminho. Ao final deste ciclo, o *novo_candidato_temp* deverá conter o caminho original acrescido deste nó, na configuração que apresentou o menor comprimento. Faz-se então *novo_candidato = novo_candidato_temp*, e a demanda do nó em questão é zerada no vetor *qtd*, para indicar que ele já foi visitado.

Estas são todas as funções necessárias à implementação do MACSVRPTW. Na seção seguinte, apresentaremos alguns resultados obtidos na simulação de problemas com o auxílio destas rotinas desenvolvidas como forma de atestar sua eficiência.

8. APLICAÇÕES PRÁTICAS A RESOLVER

Nesta capítulo, testam-se cenários de problemas conhecidos para avaliar o desempenho dos algoritmos desenvolvidos neste trabalho. As características dos dados de um problema podem influenciar em muito o funcionamento de um algoritmo. Dessa forma, é de interesse aplicar os algoritmos em problemas de características diversas, com a finalidade de demonstrar a robustez do modelo e o desempenho para diferentes instâncias de dados.

8.1. CARACTERÍSTICAS DOS PROBLEMAS A RESOLVER

Os principais fatores que influenciam o funcionamento dos algoritmos nos problemas de roteirização de veículos com janelas de tempo são: a disposição geográfica dos nós, o horizonte de tempo do problema, o percentual de clientes que possuem janelas de tempo e as características das janelas de tempo propriamente ditas.

- Quanto ao primeiro fator, podem-se ter problemas com nós dispostos aleatoriamente em uma região, problemas com nós agrupados em certas partes desta região (clusters), ou ainda uma combinação de nós agrupados e distribuídos aleatoriamente.
- O segundo fator, por sua vez, diz respeito ao máximo de tempo que define o horizonte da simulação, Por exemplo, apenas algumas horas na manhã, ou um dia inteiro; quanto maior este horizonte, mais clientes tendem a serem atendidos em uma rota, e a restrição de capacidade de carga ganha importância, se tornando mais atuante.
- O terceiro fator obviamente restringe as possibilidades de construção de rotas,
- E o quarto fator, finalmente, define que para um conjunto de nós com janelas de tempo mais estreitas, as rotas tendem a servir um número menor de clientes, enquanto que para um conjunto de nós com janelas de tempo mais largas, mais clientes podem ser servidos por uma mesma rota, reduzindo o número de veículos necessários para cobrir a região.

Problemas documentados com estas diversas características podem ser encontrados na Internet em Benchmarks, (2008). Os problemas são classificados segundo cada um dos fatores descritos acima. A nomenclatura de cada problema permite identificar a que categoria pertence.

A classificação inicial é feita dividindo-se os problemas nos grupos R, C e RC, correspondendo às características de distribuição geográfica aleatória (random), agrupada (clustered) e à combinação das duas.

Além disso, encontra-se ainda a classificação em R1 e R2, C1 e C2, RC1 e RC2, objetivo deste item é apresentar uma definição sobre grafos, alguns conceitos básicos necessários na solução do Problema de Roteamento de Veículos. Baseados nesses princípios, também são apresentados alguns problemas clássicos que utilizam o conceito de grafos, demonstrando a aplicação da teoria em soluções de problemas reais sendo o valor 1 contem a característica de um horizonte de tempo estreito, e 2 de um horizonte mais amplo. Dessa forma, existem seis classificações primárias dos problemas. Para concluir a nomenclatura dos problemas, soma-se a estes algarismos um par de números referentes a diferentes arranjos de janelas de tempo (mais estreitas ou mais largas). Dessa forma, o problema R101, por exemplo, será um problema com distribuição aleatória de pontos, com horizonte de tempo estreito, e com as janelas de tempo segundo a configuração 01.

8.2. PLANEJAMENTO DE TESTES

Cada uma das seis classificações primárias de problemas possui cerca de 10 arranjos diferentes de janelas de tempo propostas no website citado. Neste capítulo, será feita a análise de alguns problemas de configurações diversas. Primeiramente, será feita a simulação dos problemas com disposição de pontos geográficos randômicos com 25 nós cada, com o objetivo de analisar a influência do horizonte de tempo nas soluções encontradas.

Em seguida, serão simulados os problemas cluster, com 25 nós cada, para analisar as influências da distribuição física dos pontos. Posteriormente, será feita uma confrontação dos resultados obtidos nos problemas destacando a diferenciação das janelas de tempos entre os problemas com igual distribuição

de pontos geográficos. Finalmente, ocorrerá a análise de um problema maior, de 50 pontos, para avaliação de desempenho do algoritmo.

A tabela abaixo resume a forma como a análise das soluções a ser realizada:

Fator a ser analisado	Problemas a serem analisados
Horizonte de tempo	R101 e R201
Disposição geográfica dos nós	R101, C101 e RC101
Arranjo das janelas de tempo	R101 e R111
Influência do tamanho do problema	R103

Tabela 8.1 Resumo dos problemas a serem estudados.

A Tabela 8.2 abaixo nós resume os resultados obtidos nos testes realizados realizando um comparativo entre tamanho de rotas e número de veículos utilizados para o atendimento de cada rota.

	Problema	Nearest Neighbort		MACS-VRPTW		Indicador	
		Veículos (X)	Cumprimento (Y)	Veículos (Z)	Cumprimento (W)	(X - Z)	(Y / W)
25 Nós	R101	7	469,00	7	460,90	0	1,01
	R201	2	503,70	2	406,00	0	1,24
	C101	3	167,37	3	157,58	0	1,06
	RC101	3	257,41	3	252,48	0	1,01
	R111	5	432,21	4	410,69	1	1,05
50 Nós	R103	10	827,61	9	697,34	1	1,18

Tabela 8.2. Resumo dos resultados das rotas.

8.3. RESULTADO DE TESTES

Cada problema será simulado até 3 vezes apresentando a melhor solução encontrada. Além disso, será tirada uma média das soluções encontradas - serão apresentados os resultados em formatos de tabelas e gráficos estatísticos de comparação tomando em conta variáveis como horizonte de tempo, disposição geográfica dos nós, arranjo das janelas de tempo e influencia do problema.

Para efeitos de ilustração abaixo são apresentados os dados referentes ao problema R101 e R201. CUST é o número do nó (customer), XCOORD e

YCOORD são as suas coordenadas x e y as quais serão alteradas para coordenadas de latitude e longitude reais tomando como referencia as coordenadas de latitude -26.301571461651235 e longitude -48.848304748535156, DEMAND é a sua demanda, READY TIME e DUE DATE são os extremos da sua janela de tempo, e SERVICE TIME é o seu tempo de serviço. Para este problema, a capacidade de cada veículo é de 200 unidades. Para a composição da matriz de distâncias, é preciso se fazer o simples cálculo das distâncias entre todos os pares de nós. O Tratamento mencionado de Latitude e Longitude será aplicado a todas as instancias utilizadas neste trabalho.

CUST NO.	XCOORD.	YCOORD.	DEMAND	READY TIME	DUE DATE	SERVICE TIME
1	35.00	35.00	0.00	0.00	230.00	0.00
2	41.00	49.00	10.00	161.00	171.00	10.00
3	35.00	17.00	7.00	50.00	60.00	10.00
4	55.00	45.00	13.00	116.00	126.00	10.00
5	55.00	20.00	19.00	149.00	159.00	10.00
6	15.00	30.00	26.00	34.00	44.00	10.00
7	25.00	30.00	3.00	99.00	109.00	10.00
8	20.00	50.00	5.00	81.00	91.00	10.00
9	10.00	43.00	9.00	95.00	105.00	10.00
10	55.00	60.00	16.00	97.00	107.00	10.00
11	30.00	60.00	16.00	124.00	134.00	10.00
12	20.00	65.00	12.00	67.00	77.00	10.00
13	50.00	35.00	19.00	63.00	73.00	10.00
14	30.00	25.00	23.00	159.00	169.00	10.00
15	15.00	10.00	20.00	32.00	42.00	10.00
16	30.00	5.00	8.00	61.00	71.00	10.00
17	10.00	20.00	19.00	75.00	85.00	10.00
18	5.00	30.00	2.00	157.00	167.00	10.00
19	20.00	40.00	12.00	87.00	97.00	10.00
20	15.00	60.00	17.00	76.00	86.00	10.00
21	45.00	65.00	9.00	126.00	136.00	10.00
22	45.00	20.00	11.00	62.00	72.00	10.00
23	45.00	10.00	18.00	97.00	107.00	10.00
24	55.00	5.00	29.00	68.00	78.00	10.00
25	65.00	35.00	3.00	153.00	163.00	10.00

Tabela 8.3. Dados do problema R101.

O problema R201, por sua vez, possui a mesma disposição geográfica dos pontos, e as mesmas demandas e tempos de serviço, porém com um horizonte de simulação mais amplo:

CUSTOMER CUST NO.	XCOORD.	YCOORD.	DEMAND	READY TIME	DUE DATE	SERVICE TIME
0	35	35	0	0	1000	0
1	41	49	10	707	848	10
2	35	17	7	143	282	10
3	55	45	13	527	584	10
4	55	20	19	678	801	10
5	15	30	26	34	209	10
6	25	30	3	415	514	10
7	20	50	5	331	410	10
8	10	43	9	404	481	10
9	55	60	16	400	497	10
10	30	60	16	577	632	10
11	20	65	12	206	325	10
12	50	35	19	228	345	10
13	30	25	23	690	827	10
14	15	10	20	32	243	10
15	30	5	8	175	300	10
16	10	20	19	272	373	10
17	5	30	2	733	870	10
18	20	40	12	377	434	10
19	15	60	17	269	378	10
20	45	65	9	581	666	10
21	45	20	11	214	331	10
22	45	10	18	409	494	10
23	45	5	29	206	325	10
24	65	35	3	704	847	10
25	65	20	6	817	956	10

Tabela 8.4. Dados do problema R201.

A capacidade dos veículos para o problema R201 também será definida em 200 unidades.

A disposição geográfica dos clientes da instância do Problema R101 e R201 com 25 clientes

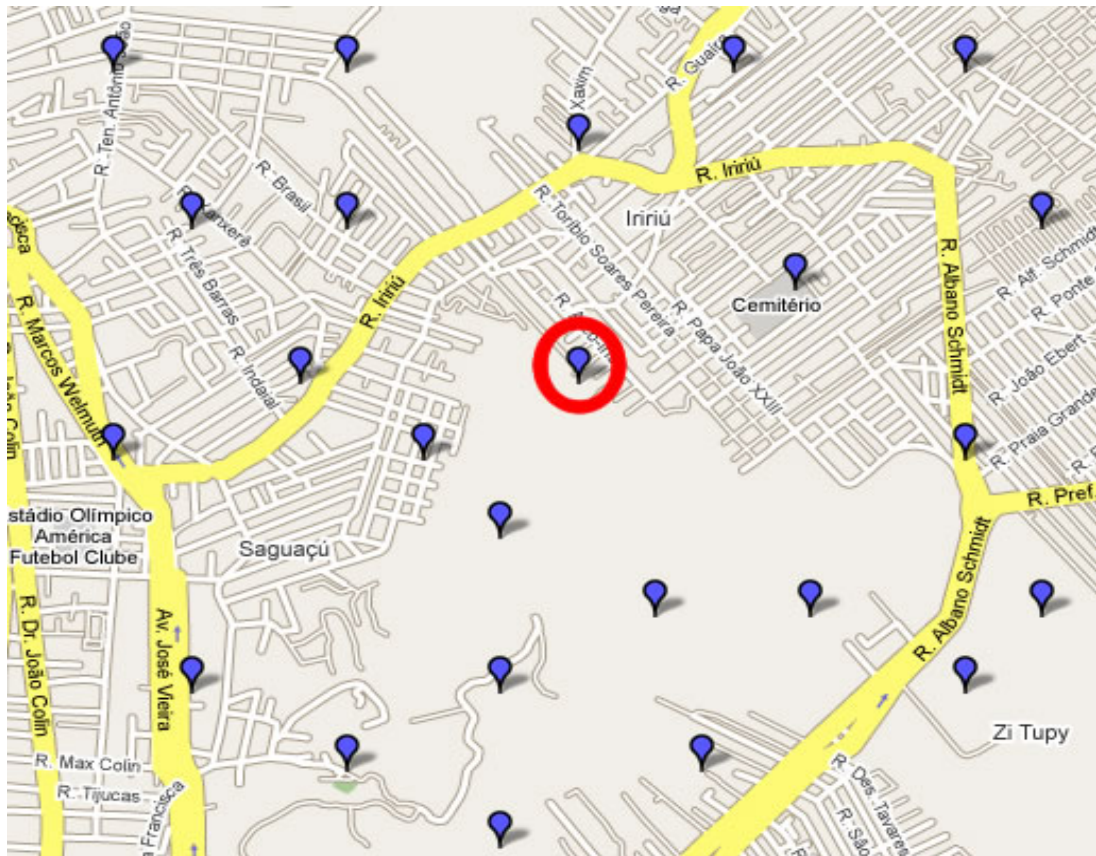


Figura 8.1. Disposição geográfica dos clientes do problema R101 ou R201 com 25 clientes.

O círculo representa o depósito (nó zero). Para aplicarmos os algoritmos do MACS-VRPTW aos problemas acima, bem como as demais instancias utilizada neste trabalho, definiremos os seguintes parâmetros:

- $\rho = 0,1$;
- $\beta = 1$;
- $q_0 = 0,9$.

Os parâmetros do Nearest Neighbour:

- $a = 0,6$ (representando o peso que se da à distância);
- $b = 0,2$ (peso que se da à espera) e
- $c = 0,2$ (peso que se da à urgência).

Além disso, cada simulação das funções `acs_time` e `acs_vei` será feita com até 20 iterações com $k = 10$ formigas cada uma.

8.3.1. RESULTADOS DO PROBLEMA R101

A melhor solução encontrada pelo MACS-VRPTW corresponde ao seguinte conjunto de rotas:

#	Rota (Seqüência de clientes a visitar)	Distância percorrida em kilometros	Carga
1	0 14 2 16 6 13 0	86,40	72
2	0 5 15 18 10 0	86,00	62
3	0 23 25 0	58,00	62
4	0 21 22 4 0	43,90	35
5	0 12 9 3 24 0	65,90	48
6	0 11 19 8 17 0	67,00	51
7	0 7 20 1 0	53,70	25

Tabela 8.5. Rotas encontradas no problema R101.

Onde o nó 0 é o depósito, com um comprimento total de 460,90 kilometros e 7 veículos. A Figura 8.2 ilustra a disposição das rotas encontradas:

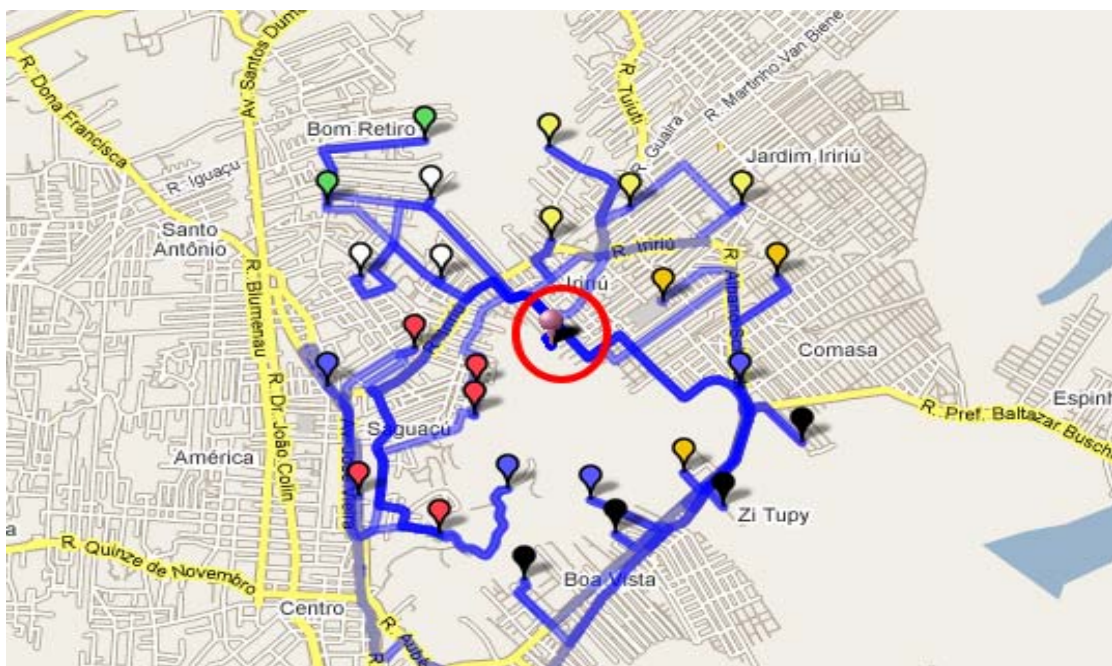


Figura 8.2. Disposição geográfica das rotas encontradas para o problema R101.

O tempo total inclui os tempos gastos nos deslocamentos, bem como os tempos de serviço nos clientes. Para efeito de comparação, podemos aplicar o algoritmo do Nearest Neighbour ao problema. A solução encontrada por este método possui um comprimento de 469,00 quilômetros, também com 7 carros, portanto um valor 1,01% mais elevado que a média do MACS-VRPTW.

O Nearest Neighbour por ser um método determinístico, não há necessidade de aplicá-lo mais de uma vez a cada problema, em vista que sua resposta será sempre a mesma.

8.3.2. RESULTADOS DO PROBLEMA R201

A melhor solução encontrada pelo MACS-VRPTW corresponde ao seguinte conjunto de rotas:

#	Rota (Sequência de clientes a visitar)	Distância percorrida em quilômetros	Carga
1	0 5 14 2 15 21 3 12 11 19 7 16 18 8 6 17 0	200,60	183
2	0 9 22 3 20 10 1 13 4 25 24 0	205,40	126

Tabela 8.6. Rotas encontradas no problema R201.

Onde o nó 0 é o depósito, com um comprimento total de 406,00 quilômetros e 2 veículos. A Figura 8.3 ilustra a disposição das rotas encontradas:

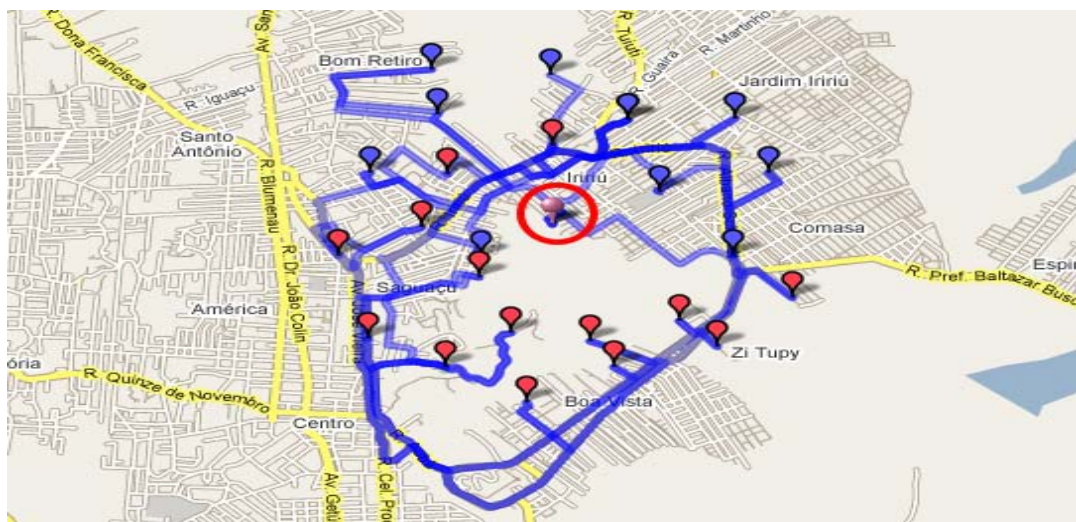


Figura 8.3. Disposição geográfica das rotas encontradas para o problema R201.

Aplicando o algoritmo do Nearest Neighbour ao problema. A solução encontrada por este método possui um comprimento de 503,70 quilômetros, com 2 carros, portanto um valor 19,39% mais elevado que a média do MACS-VRPTW.

8.3.3. RESULTADOS DO PROBLEMA C101

Na Figura 8.4 se ilustra a distribuição geográfica dos nós para o problema C101 com 25 clientes onde se pode apreciar a concentração de 3 grupos de clientes assim mesmo o nó com o círculo vermelho representa o depósito o qual se encontra distante das agrupações de nós.

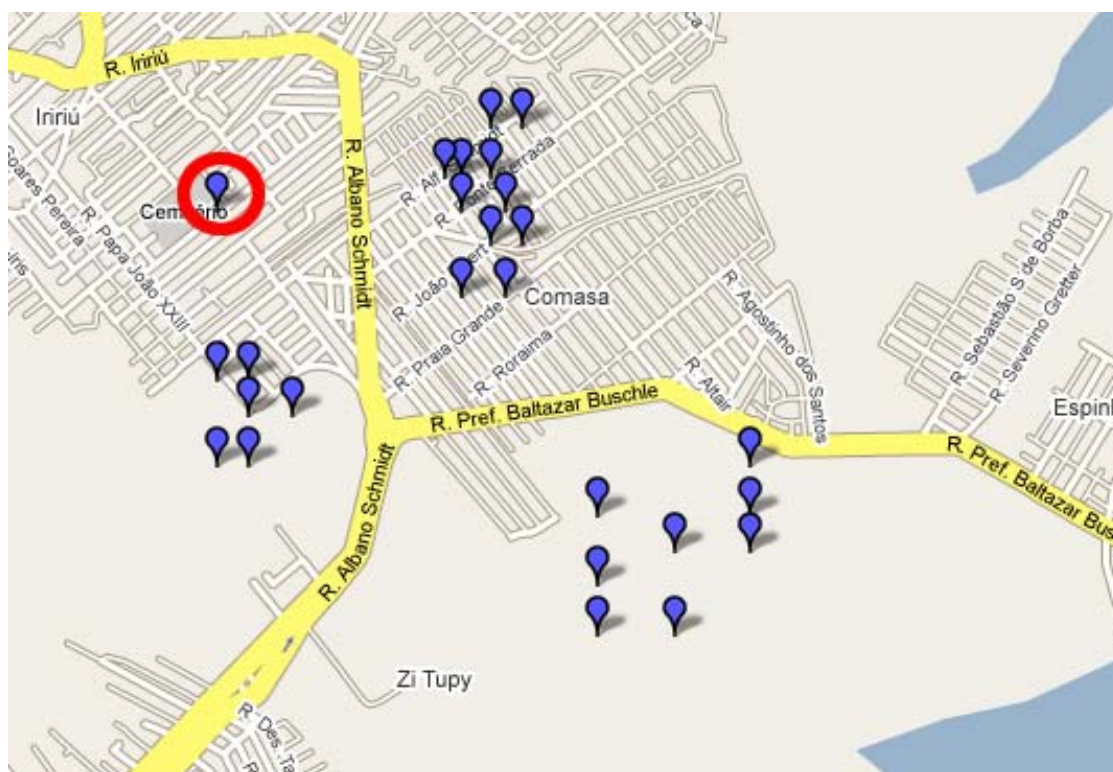


Figura 8.4 Distribuição geográfica do problema C101 com 25 clientes.

Para aplicarmos o algoritmo MACS-VRPTW foram alterados os valores dos parâmetros a, b, c do Nearest Neighbour para os seguintes valores : ($a = 0.5$; $b = 0.25$ e $c = 0.25$), a melhor solução encontrada pelo MACS-VRPTW corresponde ao seguinte conjunto de rotas:

#	Rota (Seqüência de clientes a visitar)	Distância percorrida em kilometros	Carga
1	0 5 3 7 8 10 11 9 6 4 2 1 0	38,50	160
2	0 20 24 25 19 15 16 14 12 22 0	69,52	200
3	0 13 17 18 23 21 0	49,56	100

Tabela 8.7. Rotas encontradas no problema C101.

Onde o nó 0 é o depósito, com um comprimento total de 157,58 kilometros e 3 veículos. A Figura 8.5 ilustra a disposição das rotas encontradas:

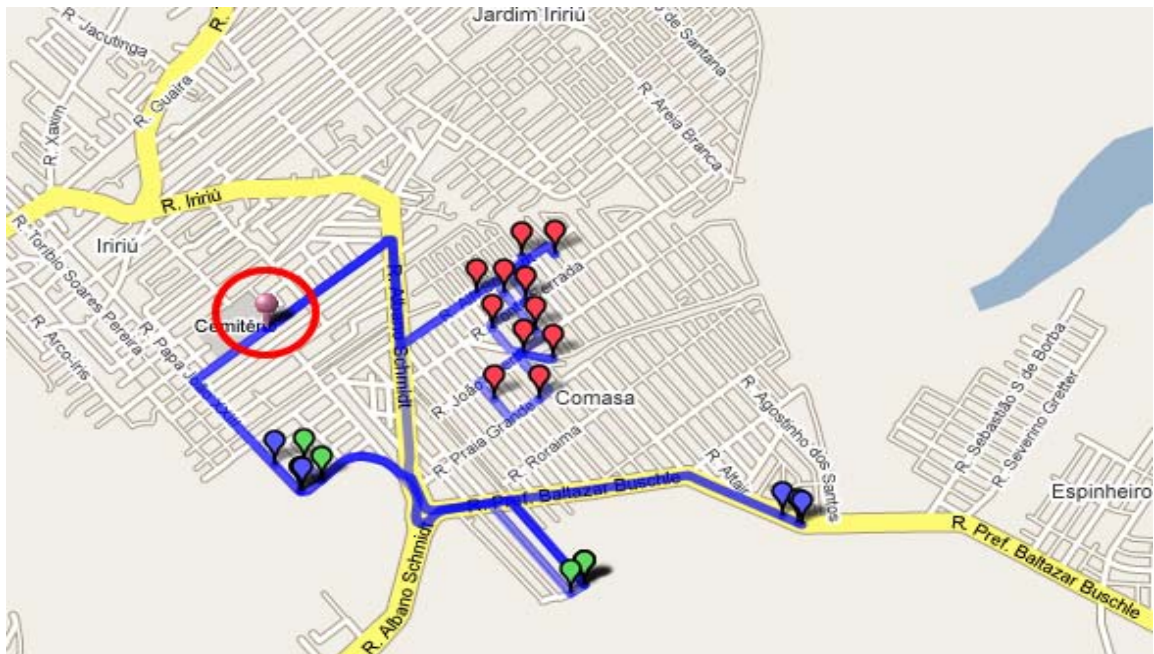


Figura 8.5. Disposição geográfica das rotas encontradas para o problema C101.

Aplicando o algoritmo do Nearest Neighbour ao problema. A solução encontrada por este método possui um comprimento de 167,37 kilometros, com 3 carros, portanto um valor 9,40% mais elevado que a média do MACS-VRPTW. Pode-se apreciar que as rotas encontradas tendem a ser mais heterogêneas cada uma de elas tentando procurar atender a todos os nós de sua região assim mesmo pode-se claramente perceber que quando os clientes se encontram agrupados em áreas distintas o algoritmo cria rotas direcionando para estas

áreas traduzidas em visitar todos os clientes do local e depois retornar ao depósito, isto normalmente ocorre por que existe uma distância considerável entre clientes de áreas diferentes considerando não extrapolar as restrições de capacidade e janelas de atendimento dos clientes.

8.3.4. RESULTADOS DO PROBLEMA RC101

A continuação se apresenta a Tabela 8.8 que ilustra os dados do problema RC101, onde os dados de coordenadas cartesianas serão traduzidos em coordenadas geográficas conforme foi exposto anteriormente.

CUSTOMER CUST NO.	XCOORD.	YCOORD.	DEMAND	READY TIME	DUE DATE	SERVICE TIME
0	40	50	0	0	240	0
1	25	85	20	145	175	10
2	22	75	30	50	80	10
3	22	85	10	109	139	10
4	20	80	40	141	171	10
5	20	85	20	41	71	10
6	18	75	20	95	125	10
7	15	75	20	79	109	10
8	15	80	10	91	121	10
9	10	35	20	91	121	10
10	10	40	30	119	149	10
11	8	40	40	59	89	10
12	8	45	20	64	94	10
13	5	35	10	142	172	10
14	5	45	10	35	65	10
15	2	40	20	58	88	10
16	0	40	20	72	102	10
17	0	45	20	149	179	10
18	44	5	20	87	117	10
19	42	10	40	72	102	10
20	42	15	10	122	152	10
21	40	5	10	67	97	10
22	40	15	40	92	122	10
23	38	5	30	65	95	10
24	38	15	10	148	178	10
25	35	5	20	154	184	10

Tabela 8.8 Dados do problema RC101.

Na Figura 8.6 se ilustra a distribuição geográfica dos nós para o problema RC101 com 25 clientes onde se pode apreciar uma combinação de clientes localizados de forma aleatória e clientes agrupados em regiões a concentração onde o nó com o círculo vermelho representa o depósito que é considerado como o ponto de início de todas as rotas.

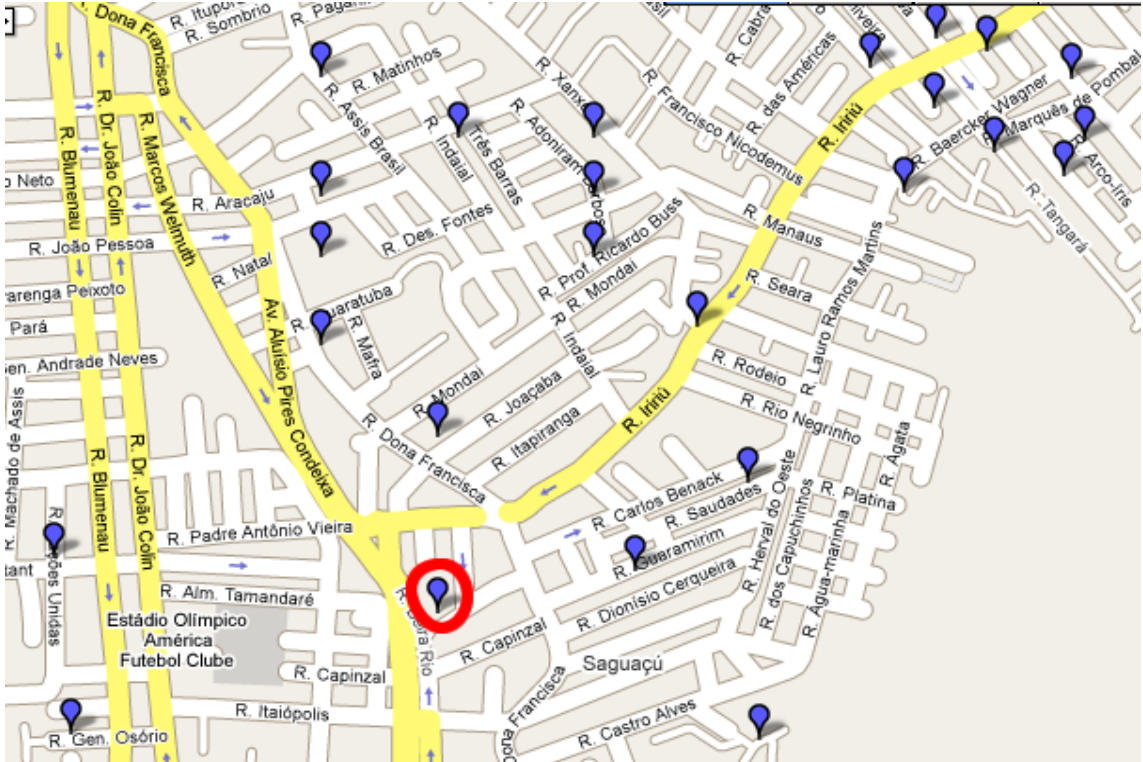


Figura 8.6 Distribuição geográfica do problema RC101 com 25 clientes.

Este problema também utiliza veículos com capacidade de 200 unidades. Aplicando o algoritmo MACS-VRPTW com os mesmos parâmetros do problema C101. Onde a melhor solução encontrada pelo MACS-VRPTW corresponde ao seguinte conjunto de rotas:

#	Rota (Seqüência de clientes a visitar)	Distância percorrida em quilômetros	Carga
1	0 14 15 11 12 16 9 10 13 17 0	92,32	190
2	0 2 5 6 7 8 3 1 4 0	78,18	120
3	0 23 21 19 18 22 20 24 25 0	81,98	180

Tabela 8.9. Rotas encontradas no problema RC101.

Onde o nó 0 é o depósito, com um comprimento total de 252,48 quilômetros e 3 veículos. A Figura 8.7 ilustra a disposição das rotas encontradas:

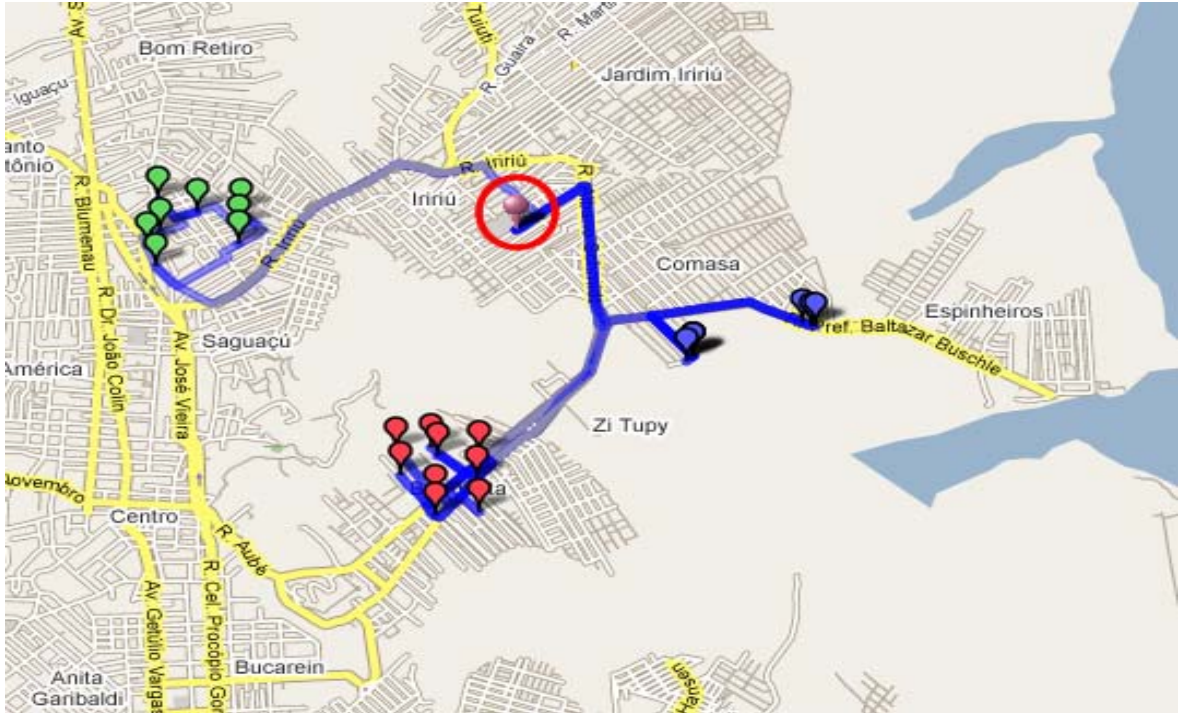


Figura 8.7. Disposição geográfica das rotas encontradas para o problema RC101.

A solução encontrada por este método possui um comprimento de 257,41 quilômetros, com 3 carros, portanto, um valor 1,95% mais elevado que a média do MACS-VRPTW.

Podemos perceber que a melhor rota encontrada se parece muito a forma de como foi resolvido o problema da instancia C101, onde primeiro serão atendidos todos os clientes próximos entre si numa determinada região para depois voltar para o depósito, sempre e quando a capacidade e as janelas o permitam.

8.3.5. RESULTADOS DO PROBLEMA R111

Este problema apresenta a mesma distribuição geográfica de clientes do problema R101 conforme foi ilustrado na Tabela 8.1 bem como as demandas, tempos de serviço e capacidade dos veículos existindo uma única diferença o espaçamento entre as janelas de tempo dos clientes. A continuação os dados do problema são ilustrados na Tabela 8.10.

CUSTOMER CUST NO.	XCOORD.	YCOORD.	DEMAND	READY TIME	DUE DATE	SERVICE TIME
0	35	35	0	0	230	0
1	41	49	10	15	204	10
2	35	17	7	18	202	10
3	55	45	13	54	187	10
4	55	20	19	138	169	10
5	15	30	26	20	199	10
6	25	30	3	76	131	10
7	20	50	5	21	170	10
8	10	43	9	87	112	10
9	55	60	16	88	115	10
10	30	60	16	107	150	10
11	20	65	12	57	86	10
12	50	35	19	15	192	10
13	30	25	23	147	180	10
14	15	10	20	32	187	10
15	30	5	8	50	81	10
16	10	20	19	29	139	10
17	5	30	2	124	189	10
18	20	40	12	47	136	10
19	15	60	17	32	146	10
20	45	65	9	79	182	10
21	45	20	11	18	195	10
22	45	10	18	76	127	10
23	55	5	29	58	87	10
24	65	35	3	58	190	10
25	65	20	6	153	186	10

Tabela 8.10. Ilustrado os dados do problema R111.

Neste problema podemos reparar que existe um espaçamento muito maior entre as janelas de tempo do problema R101 onde suas janelas têm um espaçamento de 10 unidades. O que contribuirá com uma maior flexibilidade na geração das rotas todas. Para este problema os dados dos parâmetros (a,b,c) serão de 0,70, 0,15 e 0,15 respectivamente. Os demais parâmetros de simulação serão mantidos iguais.

Este problema também utiliza veículos com capacidade de 200 unidades. Aplicando o algoritmo MACS-VRPTW, onde a melhor solução encontrada pelo MACS-VRPTW corresponde ao seguinte conjunto de rotas:

#	Rota (Seqüência de clientes a visitar)	Distância percorrida em quilômetros	Carga
1	0 16 24 23 7 8 14 22 0	115,08	97
2	0 11 8 18 16 14 17 5 0	100,71	107
3	0 19 9 10 20 3 12 1 0	118,64	100
4	0 2 24 4 25 0	76,26	35

Tabela 8.11. Rotas encontradas no problema R111.

Onde o nó 0 é o depósito, com um comprimento total de 410,69 quilômetros e 4 veículos. A Figura 8.8 ilustra a disposição das rotas encontradas:

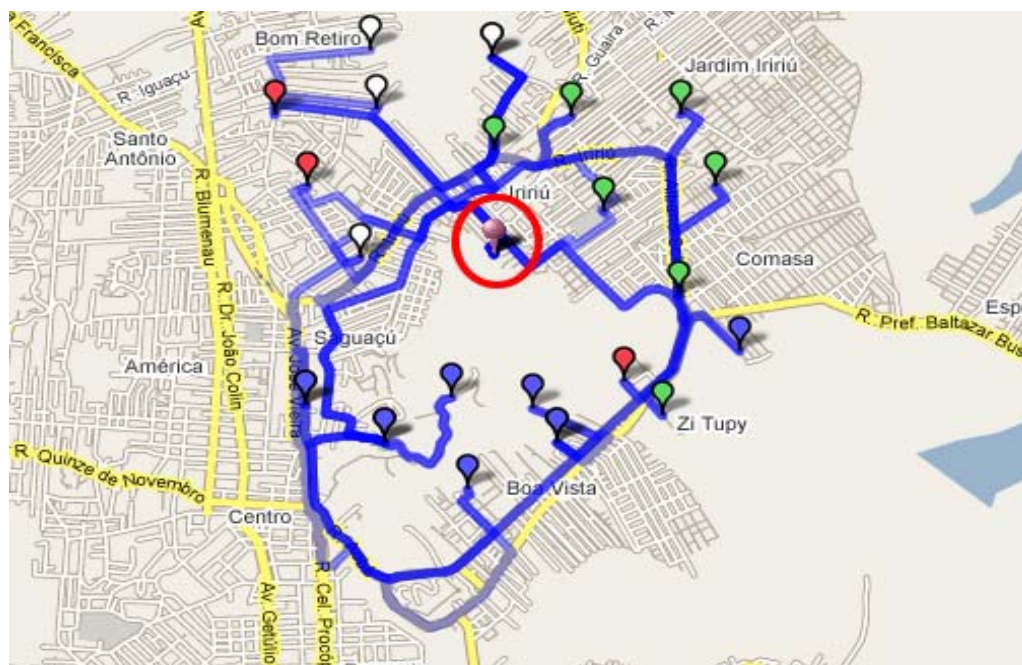


Figura 8.8. Disposição geográfica das rotas encontradas para o problema R111.

Aplicando o algoritmo do Nearest Neighbour ao problema. A solução encontrada por este método possui um comprimento de 432,21 quilômetros, com 5 carros, portanto um valor 4,97% mais elevado que a média do MACS-VRPTW.

Podemos perceber que janelas de tempo mais espaçadas possibilitam o surgimento de novas rotas nas quais as distâncias físicas dos clientes ganham importância sobre as outras restrições gerando rotas mais definidas onde cada rota assume os clientes pertencentes a uma área de região diferente.

Todas as simulações realizadas foram utilizando um computador com as seguintes características:

- Processador Pentium Dual 2 Core duo. 1.73 Ghz.
- Dois GIGAS de RAM.

Agora faremos uma comparação de desempenho com um problema com uma quantidade maior de nós (50 clientes). O Problema escolhido é a instancia R103 que se apresenta a continuação.

8.3.6. RESULTADOS DO PROBLEMA R103

A continuação se apresenta a Tabela 8.12 que ilustra os dados do problema R103, onde os dados de coordenadas cartesianas serão traduzidos em coordenadas geográficas conforme foi exposto anteriormente.

CUSTOMER CUST NO.	XCOORD.	YCOORD.	DEMAND	READY TIME	DUE DATE	SERVICE TIME
0	35	35	0	0	230	0
1	41	49	10	0	204	10
2	35	17	7	0	202	10
3	55	45	13	0	197	10
4	55	20	19	149	159	10
5	15	30	26	0	199	10
6	25	30	3	99	109	10
7	20	50	5	0	198	10
8	10	43	9	95	105	10
9	55	60	16	97	107	10
10	30	60	16	124	134	10
11	20	65	12	67	77	10
12	50	35	19	0	205	10
13	30	25	23	159	169	10
14	15	10	20	0	187	10
15	30	5	8	61	71	10
16	10	20	19	0	190	10
17	5	30	2	157	167	10
18	20	40	12	0	204	10
19	15	60	17	0	187	10
20	45	65	9	0	188	10
21	45	20	11	0	201	10
22	45	10	18	97	107	10
23	55	5	29	68	78	10
24	65	35	3	0	190	10
25	65	20	6	172	182	10
26	45	30	17	0	208	10
27	35	40	16	37	47	10
28	41	37	16	0	213	10
29	64	42	9	0	190	10
30	40	60	21	71	81	10
31	31	52	27	0	202	10
32	35	69	23	0	186	10
33	53	52	11	37	47	10
34	65	55	14	0	183	10
35	63	65	8	143	153	10
36	2	60	5	41	51	10
37	20	20	8	0	198	10
38	5	5	16	83	93	10
39	60	12	31	44	54	10
40	40	25	9	85	95	10
41	42	7	5	97	107	10
42	24	12	5	31	41	10
43	23	3	7	0	185	10
44	11	14	18	69	79	10
45	6	38	16	32	42	10
46	2	48	1	0	184	10
47	8	56	27	0	185	10
48	13	52	36	0	192	10
49	6	68	30	108	118	10
50	47	47	13	0	203	10

Figura 8.12 Dados do problema R103.

#	Rota (Sequência de clientes a visitar)	Distância percorrida em quilômetros	Carga
1	0 27 28 26 40 21 22 2 13 37 16 5 18 0	91,67	178
2	0 6 7 48 47 19 46 17 14 43 12 0	95,93	137
3	0 1 50 33 3 29 24 25 31 0	97,47	92
4	0 30 20 32 10 35 34 0	89,51	82
5	0 42 15 41 4 0	68,84	91
6	0 8 49 0	93,39	39
7	0 45 36 11 9 0	62,56	49
8	0 44 38 0	54,99	34
9	0 39 23 0	42,98	60

Tabela 8.13. Rotas encontradas no problema R103.

Onde o nó 0 é o depósito, com um comprimento total de 697,34 quilômetros e 9 veículos. A Figura 8.10 ilustra a disposição das rotas encontradas:

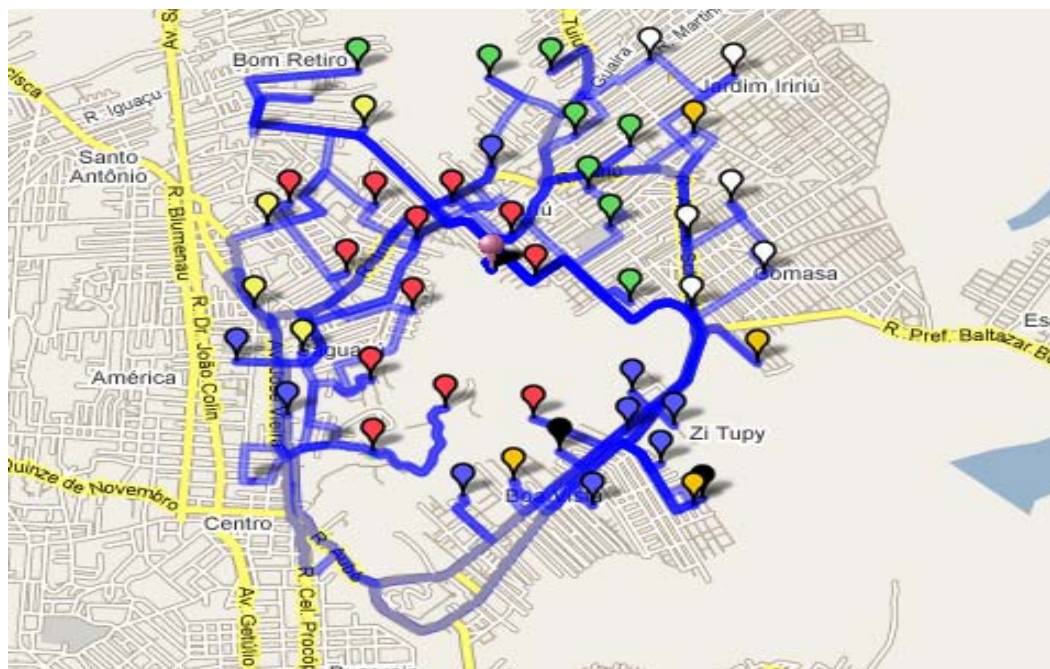


Figura 8.10. Disposição geográfica das rotas encontradas para o problema R103.

Aplicando o algoritmo do Nearest Neighbour ao problema. A solução encontrada por este método possui um comprimento de 827,61 quilômetros, com 10 carros, portanto um valor 18,68% mais elevado que a média do MACS-VRPTW uma diferença expressivamente maior o qual se deve a que é uma instancia com um numero muito maior de clientes possibilitando pequenos ganhos em diferentes trajetos entre as visitas dos clientes que em seu conjunto representam a diferença entre os métodos. O Processamento do algoritmo levou aproximadamente 3 minutos que é um tempo muito maior que o tempo levado para os algoritmos com instancias de 25 clientes.

Os seis problemas apresentados com diversas características envolvem muitos dos diferentes fatores que afetam o desempenho dos algoritmos. A análise das soluções dos problemas nós permite poder entender melhor que as características particulares de cada um dos problemas influem significativamente na geração das rotas assim mesmo nós permite poder criar novos modelos para cada um dos problemas.

A aplicação dos algoritmos em instancias com 100 ou mais nós se mostra ineficiente em vista que temos que esperar mais de 1 hora para a obtenção de uma resposta de qualidade mediana. Isto pode acontecer por que o desenvolvimento de este trabalho visa a roterização de veículos em plataformas de informação geográfica visando o funcionamento correto do algoritmo assim como sua aplicação das rotas nas informações geográficas do mapa. Pode-se obter uma performance maior se usáramos alguma linguagem de mais baixo nível que nós permite obter um desempenho maior na execução dos algoritmos.

Assim mesmo podemos realizar um analises de sensibilidade em cima dos outros parâmetros dos algoritmos tais como q_0 que foi fixado em 0,9. Veremos a continuação as mudanças apresentadas na execução do algoritmo se o valor de q_0 fora mudado para 0,7 e 0,5. Para exemplificar foi tomado o problema de instancia R101, onde o valor de q_0 foi igual a 0,7, nós apresento um valor muito de 462,90 quilômetros, valor muito próximo da primeira experiência com R101 em quanto com o valor alterado para 0,5 apresento um acréscimo de 4 minutos na execução do algoritmo com uma distancia muito próxima de primeira experiência. Com o que podemos perceber que é importante manter o valor de q_0 com um valor maior a 0,5 para obter um maior desempenho. Diversos tipos de análises poderiam ser aplicados a este tipo de problemas, levando em

consideração fatores como próprios da instancias dos problemas ou definição de valores para os parâmetros dos algoritmos visando obter o maior desempenho na execução. Assim mesmo cabe considerar a construção de novos métodos que nós permitam obter novos resultados.

CONSIDERAÇÕES FINAIS

A importância da roteirização de veículos, bem como a diversidade de problemas do mundo real e a sua complexidade computacional, têm provocado os pesquisadores no sentido de se buscarem novos algoritmos que permitam obter melhores resultados, resolvendo problemas de dimensões crescentes e cada vez mais complexos, além de incorporar restrições que permitam tornar mais realistas os modelos que os representam.

Neste trabalho, foi proposto um modelo para o Problema de Roteamento de Veículos em plataformas SIG, que tem por objetivo gerar rotas de custo mínimo que atendam a todos os clientes da rede, respeitando as restrições definidas para o modelo. Neste caso, foram restrições baseadas no modelo de MACS-VRPTW, baseado em colônia de formigas múltiplas proposto por Gambardella, Taillard e Agazzi em 1999.

Para isto, foi apresentado um embasamento teórico das tecnologias a serem usadas no desenvolvimento deste trabalho, tais como geoprocessamento, onde se detalha sobre os princípios e se descreve um histórico sobre o tema, assim como se explica os tipos de dados utilizados desta tecnologia. Também foram apresentados, os conceitos básicos sobre bancos de dados geográficos e suas definições de modelos geográficos, além de tipos de estruturas usadas e um comparativo entre as mesmas. Foram apresentadas as definições básicas, características e arquiteturas de sistemas SIG, mas também foram abordados sucintamente os conceitos envolvidos em teoria dos grafos. Para completar o trabalho, fez-se uma introdução ao problema de roteamento de veículos, assim como a definição do modelo da solução proposta, onde se explica as técnicas a serem usadas, baseada na meta heurística Colônia de Formigas Múltiplas.

Na proposta deste trabalho foram citados objetivos específicos referentes a aquisição de dados através de interfaces de usuário para consultas geográficas assim como objetivos que se referem à geração de algoritmos que permitam realizar a roteirização de veículos, objetivos que foram alcançados através da construção do algoritmo de roteirização que foi apresentado e minuciosamente estudado, através das aplicações dos algoritmos aos problemas propostos e dos

análises dos resultados foi possível perceber que os modelos são capazes de chegar a uma solução próxima da ótima.

Os algoritmos de colônias de formigas sem dúvida representam uma ótima ferramenta para a resolução de problemas de otimização combinatória.

Considera-se ainda a necessidade de trabalhos futuros, estes referentes à integração de sistemas de roteirização a sistemas e bancos de dados das empresas, entre os quais os de pedidos, cadastro de clientes e faturamento e até ferramentas do tipo ERP – “*Enterprise Resource Planning*”. Além disso, espera-se que os sistemas de roteirização venham a se integrar também aos sistemas de rastreamento de veículos via GPS, possibilitando a alteração dinâmica, e em tempo real, de roteiros, de forma a atender novas solicitações, além de proverem uma retro-alimentação dos dados das viagens, estas realizadas de forma a permitir ajustes e aprimoramentos das bases de dados de distâncias e tempos gastos na viagem.

Assim mesmo, considera-se ainda a importância da forma de representação dos dados especiais de clientes e do sistema viário para a qualidade dos roteiros gerados. A forma de representação de dados em sistemas de informação geográfica produzem resultados mais precisos e realistas. Em contrapartida, para problemas no ambiente urbano surgem considerações a serem feitas em cima de velocidades permitidas e tempos de viagem, assim como condições de trânsito as quais podem refletir em variações substanciais.

REFERÊNCIAS

ANDRÉ C., Paula C., Eliane G e Virgilio F. (2007) Sistemas de roteirização de veículos e gestão da cadeia de suprimentos: uma abordagem analítica. Mestrado do Programa de Engenharia de Produção / COPPE / UFRJ. 690p. (Dissertação de Mestrado).

ASSAD, A.A. (1988) Modeling and implementation issues in vehicle routing. In: Vehicle Routing: Methods and Studies, B.L.Golden, A.A.Assad (eds), North Holland, Amsterdam, p. 7-46.

BAKER, Barrie M; Ayechew, M.A. “A genetic algorithm for the vehicle routing problem”. Computer & Operations Research 30, 787-800, 2003.

BAKKER, Mucio P.R. de, Cartografia – Noções Básicas, Marinha do Brasil – Hidrografia e Navegação –1965.

BARBAROSOGLU, Gulay, OZGUR, Demet. “A tabu search algorithm for the vehicle routing problem”, Computers & Operations Research 26, 255-270, 1999.

BENCHMARKINH, <http://www.top.sintef.no/vrp/benchmarks.html> , acesso o 04/03/2008.

BIESZCZAD, A.; PAGUREK, B; WHITE, T. (1998). Mobile Agents for Network Management. IEEE Communications Surveys, vol. 1, no. 1. Disponível em citeseer.ist.psu.edu/article/bieszczad98mobile.html.

BODIN, L.; LEVY, L. The Arc Partitioning Problem. European Journal of Operational Research, v. 53, n. 3, p. 393-401, Aug. 1991.

BODIN,L.D., Golden, B. “Classification in Vehicle Routing and Scheduling”. Networks 11: 97-108, 1981.

BRASSARD, Gilles; Bratley, Paul. Fundamentals of Algorithmics. EUA: Prentice Hall, 1996.

BREEDAM, Alex V. "Comparing descent heuristics and metaheuristics for the vehicle routing problem". Computer & Operations Research 28, 289-215, 2001.

BULLNHEIMER, b., Hartl, r. f., Strauss, c. Applying the Ant System to the Vehicle Routing Problem. In: 2nd International Conference on Metaheuristics MIC-97, Sophia-Antipolis, França, 21- 24 de julho de 1997. Meta-heuristics: Advances and Trends in Local Search Paradigms for Optimization, 1999, pp. 285-296.

C. Cargill. Standard View - ACM Perspectives on Standardization. Communications of the ACM, 2(3), 1994.

CÂMARA, Gilberto; Casanova, Marco A.; Hemerly, Andrea S.; Magalhaes Geovane C.; Medeiros, Cláudia M. B.; Anatomia de Sistemas de Informação Geográfica. Rio de Janeiro, 1996.

CHRISTOFIDES, Nicos; Mingozzi, Aristide; TOTH Paolo. The Vehicle Routing Problem. In: Christofides, Nicos. Combinatorial Optimization UMI, 1979.

CLARKE, G; Wright, J.W. "Scheduling of vehicles from a central depot to a number of delivery points". Operations Research 12: 568-581, 1964.

COOK, William J; Cunningham, William H; Pulleyblank, William r; Schrijver, Alexander. Combinatorial Optimization. Willey, 1998.

CUNHA, C.B. (1991) Algoritmos para roteamento e programação de veículos no contexto da distribuição física. São Paulo: EPUSP, Departamento de Engenharia de Transportes. 178p. (Dissertação de Mestrado).

DASKIN, M. Logistics: Overview of the State of the Art and Perspectives on Future Research. Transportation Research, v. 19, part A, p. 383 – 398, Sep-Nov. 1985.

DORIGO, M.; Gambardella, L. M. Ant Colonies for the Travelling Salesman Problem. Biosystems, Vol. 43, No. 2. pp. 73- 81. Julho 1997.

DORIGO, M.; Maniezzo V.; Colomi, A. Positive feedback as a search strategy. Milão: Dipartimento di Elettronica e Informatica, Politecnico di Milano, Itália, 1991. Relatório Técnico.

FISHER, M.L.,JAIKUMAR, R. “A generalized assignment heuristic for vehicle routing” Networks 11, 109-124 ,1981.

FOX, S. (2004); Swarm Stupidity. IEE Manufacturing Engineer. Dezembro(2003)-Janeiro (2004), pp. 26-29.

GAMBARDELLA, I. m.; Taillard, é.; Agazzi, g. MACS-VRPTW: a multiple Ant Colony System for vehicle routing problems with time windows. In: New Ideas in Optimization. Londres: McGraw-Hill, 1999. pp. 63-76.

GAREY, Michael R., J., David S. “Computers and Intractability – A Guide to the Theory of NP-Completeness”. New York: W.H.Freeman and Company,1999.

GILLET, B.E; Miller, L.R. “A heuristic algorithm for the vehicle dispatch problem” Operations Research 22, 240-349,1974.

GOLDEN,B.L; M., T.L.; NGUYEN, H.Q. “Implementing Vehicle Routing Algorithms” Networks 7:113-148 ,1977.

L. FILHO (2002) Reutilização de projetos de bancos de dados geográficos. Universidade Federal de Viçosa.

LAPORTE, G.; M. Gendreau; J.Y. Potvin e F. Semet (2000) Classical and modern heuristics for the vehicle routing problem, International Transactions in Operational Research, v.7, n4/5, p.285-300.

LENSTRA, J. & **RINNOOY KAN**, A. (1981), Complexity of vehicle routing and scheduling problems, *Networks*, v.11, p. 221-228.

LERMAN, K.; **GALSTYAN**, A.; **MARTINOLI**, A.; **IJSPEERT**, A. (2001). A Macroscopic Analytical Model of Collaboration in Distributed Robotic Systems. *Artificial Life*, no. 7, pp. 375-393.

LISBOA FILHO, Jugurta; *Introdução a SIG - Sistemas de Informações Geográficas*. Porto Alegre, Viçosa, UFRGS, 1995.

MOLE, R.H.; **J**, S.R. "A sequential route-building algorithm employing a generalized savings criterion" *Operations Research Quarterly* 27,503-511,1976.

NELSON, Marvin D; Nygard, Kendall E.; Griffin, John H.; Shreve, Warren E. "Implementing Techniques for the vehicle routing problem" *Computers & Operations Research* 12,273-283, 1985.

PARPINELLI, R.S.; **LOPES**, H.S.; **FREITAS**, A.A.(2002). An Ant Colony Algorithm for Classification Rule Discovery. In: H. Abbass, R. Sarker, C. Newton. (Eds.) *Data Mining: a Heuristic Approach*, pp. 191-208. London: Idea Group Publishing.

PATRICIA, P. Belfiore (2006) XIII SIMPEP - Bauru, SP, Brasil, 6 a 8 de Novembro de 2006. Problema de roteirização de veículos com janelas de tempo: revisão da Literatura.

PELLEGRINI, P. Application of Two Nearest Neighbor Approaches to a Rich Vehicle Routing Problem. Bruxelas: IRIDIA, Université Libre de Bruxelles, 2005. Relatório Técnico.

POSTGRES Global Development Group; PostgreSQL 8.0.3 Documentation. POSTGRESQL, 2005. Disponível em: <<http://www.postgresql.org>>. Acesso em: set.2007.

PRINS, Christian. "A simple and effective evolutionary algorithm for the vehicle routing problem". Computers & Operations Research 31, 1985-2002, 2004.

RAFAEL, Lorenzo Santos (2006) Dissertação de Mestrado em Engenharia Industrial – Pontifícia Universidade Católica do Rio de Janeiro 2006.

RAUL, S. Wazlawick (2004) Análise e Projeto de Sistemas de Informação Orientados a Objetos, p. 23-25.

REFRACTIONS Research; PostGIS Manual. Refractions Research, 2006. Disponível em: <<http://postgis.refractions.net>>. Acesso em: set.2007.

ROGÉRIO Thomé (1998), Interoperabilidade em geoprocessamento: conversão entre modelos conceituais de sistemas de informação geográfica e comparação com o padrão open gis. Dissertação de Mestrado em Computação Aplicada, orientada pelo Dr. Gilberto Câmara, aprovada em setembro de 1998.

SHAFFER, Clifford A. A practical introduction to data structures and algorithm analysis. New Jersey: Prentice Hall, 1997.

SKIENA, Steven S. The Algorithm Design Manual. New York: Springer-Verlag, 1997

SZWARCFITER, Jayme Luiz. Grafos e algoritmos computacionais. Rio de Janeiro: Campus, 1984.

THANGIAH, Sam R; Ptroviw, Pavel. Introduction to Genetic Heuristics and vehicle Routing Problems with Complex Constraints. In: Woodruff, David, L. Advances in Computacional and Stochastic Optimization, Logic programming , and Heuristic search: Interfaces in Computer Science and Operations research. Kluwer Academic Publishers, 1997.

TYAGI, M., "A Pratical Method for the Truck Dispatching Problem". J. of the Operations Research Society of Japan, 10,76-92,1968.

VIANA, Valdisio. Meta-heurísticas e Programação Paralela em Otimização Combinatória. Fortaleza: EUFC, 1998.

XU, Jiefeng; **KELLY**, James P. "A Network Flow-Based Tabu Search Heuristic for the Vehicle Routing Problem" Transportation Science 30, 379-393, 1996.