

Universidade Federal de Santa Catarina  
Programa de Pós-Graduação em  
Engenharia de Produção

Douglas Hörner

**RESOLUÇÃO DO PROBLEMA DAS P-MEDIANAS  
NÃO CAPACITADO  
COMPARAÇÃO DE ALGUMAS TÉCNICAS HEURÍSTICAS**

Florianópolis

2009

Universidade Federal de Santa Catarina

Programa de Pós-Graduação em

Engenharia de Produção

**RESOLUÇÃO DO PROBLEMA DAS P-MEDIANAS  
NÃO CAPACITADO**  
COMPARAÇÃO DE ALGUMAS TÉCNICAS HEURÍSTICAS

Douglas Hörner

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina como requisito parcial para obtenção do título de Mestre em Engenharia de Produção.

Orientador: Antônio Sérgio Coelho, Dr.

Florianópolis

2009

**RESOLUÇÃO DO PROBLEMA DAS P-MEDIANAS  
NÃO CAPACITADO**  
COMPARAÇÃO DE ALGUMAS TÉCNICAS HEURÍSTICAS

**Douglas Hörner**

Esta dissertação foi julgada e aprovada para a obtenção do Título de **Mestre em Engenharia de Produção** no **Programa de Pós-Graduação em Engenharia de Produção** da Universidade Federal de Santa Catarina.

Florianópolis, 26 de fevereiro de 2009.

---

Prof. Antônio Sérgio Coelho, Dr.  
Coordenador

---

Prof. Antônio Sérgio Coelho, Dr.  
(Orientador)

**BANCA EXAMINADORA**

---

Prof. Sérgio Fernando Mayerle, Dr.  
(Presidente)

---

Profa. Mirian Buss Gonçalves, Dra.  
(Membro)

---

Profa. Daniela Brandão Nascimento, Dra.  
(Membro)

## **AGRADECIMENTOS**

Ao professor Dr. Antônio Sérgio Coelho, pela paciência, confiança e sabedoria transmitida.

Ao professor Dr. Sérgio Fernando Mayerle, pelo auxílio indispensável.

Aos professores pela participação na banca.

A todos os professores e funcionários do PPGE/UFSC.

Aos amigos Neimar Follmann, Leonor Farias Abreu, Raimundo Ronilson Leal do Rosário e Robson Raulino Rautemberg.

Aos meus pais Esmeralda Fischer Hörner e Anselmo Hörner pelo valor dado à educação.

A Louise Dorow pelo apoio constante.

“Vivemos num mundo que é cheio de miséria e ignorância, e o dever simples de cada um e de todos é de tentar e fazer com que o pequeno recanto que possa influenciar seja um pouco menos miserável e um pouco menos ignorante do que era antes”.

Thomas Henry Huxley (1825 – 1895)

## RESUMO

HÖRNER, Douglas. **Resolução do problema das p-medianas não capacitado – Uma comparação de técnicas heurísticas**. 2009. 104 f. Dissertação (Mestrado em Engenharia de Produção), Departamento de Engenharia de Produção e Sistemas, Universidade Federal de Santa Catarina, Florianópolis, Brasil.

Neste trabalho são implementadas e analisadas as heurísticas de Teitz e Bart, Busca Tabu e Algoritmo Genético aplicadas ao problema das p-medianas. O objetivo foi avaliar o desempenho dos algoritmos em redes pequenas, médias e grandes levando-se em consideração o valor da solução e o tempo de resolução. Para isso foram realizados testes em 115 problemas gerados aleatoriamente. Os resultados demonstraram bom desempenho dos algoritmos de Busca Tabu e Algoritmo Genético em redes pequenas e médias. Já em problemas de grande porte, o algoritmo de Teitz e Bart modificado forneceu melhores respostas em menor tempo.

**Palavras – Chave:** teoria da localização, p-medianas, algoritmos heurísticos.

## ABSTRACT

HÖRNER, Douglas. **Resolution of uncapacitated p-median problem – Comparison of heuristics**. 2009. 104 p. Dissertation (Master in Industrial Engineering), Industrial Engineering and Systems Department, Federal University of Santa Catarina, Florianópolis, Brazil.

In this study are developed and analyzed the heuristic of Teitz & Bart, Tabu Search and Genetic Algorithm applied to the p-median problem. The objective was to evaluate the performance of algorithms in small, medium and large networks taking into account the value of the solution and the time spent. The tests were realized in 115 randomly generated problems. The results showed good performance of algorithms, Tabu Search and Genetic Algorithm in small and medium sized networks. The algorithm of Teitz & Bart modified provided better responses in larger networks.

**Keywords:** location theory, p-medians problem, heuristics.

## Lista de Figuras

Figura 3.1 – Interface do programa desenvolvido.....	57
Figura 3.2 – Perfil de desempenho dos algoritmos em relação à qualidade as solução em redes pequenas.....	59
Figura 3.3 – Perfil de desempenho dos algoritmos em relação à qualidade as solução em redes médias.....	60
Figura 3.4 – Perfil de desempenho dos algoritmos em relação à qualidade as solução em redes grandes.....	61
Figura 3.5 – Perfil de desempenho dos algoritmos em relação ao tempo em redes pequenas.....	62
Figura 3.6 – Perfil de desempenho dos algoritmos em relação ao tempo em redes médias.....	63
Figura 3.7 – Perfil de desempenho dos algoritmos em relação ao tempo em redes grandes.....	64
Figura 3.8 – Perfil de desempenho dos algoritmo em relação à qualidade em todos os 115 problemas.....	65
Figura 3.9 – Perfil de desempenho dos algoritmos em relação ao tempo em todos os 115 problemas.....	66
Figura A.1. – Grafo do exemplo.....	79
Figura A.2. – Solução do problema.....	85
Figura A.1. – Grafo orientado.....	96
Figura A.2. – Grafo não orientado.....	96



Figura A.3. – Grafo Misto.....	96
Figura A.4. – Laço.....	97
Figura A.5. – Grafo não orientado conexo.....	99
Figura A.6. – Grafo orientado fortemente conexo.....	99
Figura A.7. – Grafo orientado unilateralmente conexo.....	100
Figura A.8. – Grafo proposto.....	101
Figura A.9. - Grafo parcial.....	101
Figura A.10. – Sub-grafo.....	102
Figura A.11. – Sub-grafo parcial.....	103

## Lista de Tabelas

Tabela 2.1 – Tabela de Complexidade.....	33
Tabela A.1. – Tabela representando a matriz de distâncias mínimas.....	80
Tabela A.2. – Tabela representando a matriz Peso – Distância.....	81
Tabela B.1 – Resultados obtidos e respectivos tempos de execução em redes pequenas.....	87
Tabela C.1 – Resultados obtidos e respectivos tempos de execução em redes médias.....	89
Tabela C.2 – Resultados obtidos e respectivos tempos de execução em redes médias.....	90
Tabela D.1 – Resultados obtidos e respectivos tempos de execução em redes grandes.....	92
Tabela D.2 – Resultados obtidos e respectivos tempos de execução em redes pequenas.....	93

# SUMÁRIO

<b>CAPÍTULO I.....</b>	<b>12</b>
1 INTRODUÇÃO .....	12
1.1 Considerações Iniciais .....	12
1.2 Objetivos.....	14
1.3 Importância .....	15
1.4 Limitações .....	16
1.5 Estrutura do Trabalho.....	16
<b>CAPÍTULO II.....</b>	<b>17</b>
2 REVISÃO DA LITERATURA .....	17
2.1 Localização de facilidades.....	17
2.2 Definição do Problema das p-medianas.....	21
2.3 Formulação Matemática do Problema das p-medianas.....	24
2.4 Trabalhos relevantes que utilizam o modelo das p-medianas .....	25
2.5 Resultados obtidos por Hakimi .....	29
2.6 Complexidade do Problema das p-Mediana.....	33
2.7 Algoritmos Exatos .....	35
2.8 Algoritmos Heurísticos .....	36
2.9 Algoritmos Metaheurísticos .....	38
<b>CAPÍTULO III.....</b>	<b>43</b>
3 IMPLEMENTAÇÃO E RESULTADOS COMPUTACIONAIS .....	43
3.1 Métrica Euclidiana.....	43
3.2 Algoritmo de Teitz e Bart.....	45
3.3 Algoritmo Genético aplicado ao problema das p-medianas .....	47
3.3.1 Representação da Estrutura Cromossômica.....	49
3.3.2 População Inicial e Número de Indivíduos.....	50
3.3.3 Avaliação de Cada Indivíduo ou Fitness.....	50
3.3.4 Seleção dos Cromossomos .....	51
3.3.5 Reprodução.....	51
3.3.6 Cruzamento.....	51
3.3.7 Mutação .....	52
3.3.8 Estrutura do Algoritmo Genético .....	52
3.4 Algoritmo de Busca Tabu para o problema das p-medianas .....	52
3.4.1 Solução Inicial.....	54
3.4.2 Movimentos .....	54
3.4.3 Restrições Tabu.....	54
3.4.4 Critério de Aspiração.....	55
3.4.5 Estratégia de Oscilação.....	55

3.4.6 Estrutura do algoritmo de Busca Tabu.....	55
3.5 Comparação e análise dos resultados.....	56
<b>CAPÍTULO IV.....</b>	<b>67</b>
4 CONCLUSÕES E RECOMENDAÇÕES.....	67
4.1 Conclusões.....	67
4.2 Recomendações.....	69
BIBLIOGRAFIA.....	70
<b>APÊNDICE A.....</b>	<b>78</b>
A. RESOLUÇÃO DE UM PROBLEMA DIDÁTICO PELO MÉTODO DE TEITZ E BART.....	79
A.1. O Exemplo.....	79
A.2. Objetivo.....	80
A.3. Resolução.....	80
<b>APÊNDICE B.....</b>	<b>86</b>
B. RESULTADOS OBTIDOS E RESPECTIVOS TEMPOS DE EXECUÇÃO EM REDES PEQUENAS.....	87
<b>APÊNDICE C.....</b>	<b>88</b>
C. RESULTADOS OBTIDOS E RESPECTIVOS TEMPOS DE EXECUÇÃO EM REDES MÉDIAS.....	89
<b>APÊNDICE D.....</b>	<b>91</b>
D. RESULTADOS OBTIDOS E RESPECTIVOS TEMPOS DE EXECUÇÃO EM REDES MÉDIAS.....	92
<b>ANEXO A.....</b>	<b>94</b>
A. CONCEITOS BÁSICOS DA TEORIA DOS GRAFOS.....	95
A.1. Definições Gerais sobre Teoria de Grafos.....	95
A.1.1. Grafo.....	95
A.1.2. Arcos e Arestas.....	95
A.1.3. Grafos Orientados, Não Orientados e Mistos.....	95
A.1.4. Laço.....	97
A.1.5. Rede.....	97
A.1.6. Vértices Adjacentes.....	97
A.1.7. Grau de um Vértice.....	98
A.1.8. Grau de Entrada e de Saída de um Vértice.....	98
A.1.9. Caminhos e Cadeias.....	98
A.1.10. Circuitos e Ciclos.....	98
A.1.11. Conexidade.....	98
A.1.12. Árvores.....	100
A.1.13. Grafo Ponderado.....	100
A.1.14. Grafo Parcial.....	101
A.1.15. Sub-Grafo.....	102
A.1.16. Sub-Grafo Parcial.....	102

# CAPÍTULO I

## 1 INTRODUÇÃO

### 1.1 Considerações Iniciais

A crescente competitividade em todos os setores da economia faz com que a margem de lucro das instituições seja cada vez mais estreita. Com isso, a busca pela redução dos custos, sem comprometimento do nível de serviço proporcionado aos clientes tem feito, inegavelmente, parte do cotidiano das empresas.

Dentro deste contexto, a logística das empresas tem passado por uma crescente valorização. O que antes da segunda guerra mundial era somente um setor responsável pelo transporte de suprimentos, passou a desempenhar uma função nevrálgica no desempenho das corporações.

A logística, segundo definição do *Concil of Supply Chain Management Professionals* (CSCMP, 2005), “é a parte da cadeia de suprimentos que planeja, implementa e controla a eficiência dos fluxos, direto e reverso, e a armazenagem de produtos, bem como os serviços e informações associadas, cobrindo desde o

ponto de origem até o ponto de consumo, com o objetivo de atender aos requisitos do consumidor”.

Dentre os vários assuntos pertencentes ao escopo da logística cita-se o problema de localização de facilidades, de interesse das mais diferentes áreas de conhecimento tais como a economia, a administração, a pesquisa operacional, a geofísica e a engenharia.

Cada uma das áreas supracitadas apresenta interpretações, problemas e aplicações diferentes. No entanto, o termo facilidade é utilizado na maioria dos textos para os mais diversos fins. Uma facilidade pode ser uma empresa, um depósito, um posto policial, um mercado, uma escola, entre outros.

Saliente-se também que um dos problemas de maior destaque dentro da teoria da localização é o problema das  $p$ -medianas, onde  $p$  é o número de facilidades a serem instaladas. Conhecido também como *minisum*, o problema consiste em minimizar a soma das distâncias de todos os vértices à mediana mais próxima.

Com relação ao problema das  $p$ -medianas, é importante destacar que, à medida que o tamanho do mesmo aumenta, cresce também a complexidade da resolução devido à explosão combinatorial, tornando-se inviável, a partir de um determinado ponto, encontrar a respectiva solução ótima em um tempo considerado aceitável.

No sentido de obter uma solução satisfatória dentro do tempo de que se dispõe, ao longo dos últimos anos têm sido sugeridas várias técnicas heurísticas aplicadas ao problema das  $p$ -medianas. Algoritmos tais como o genético, busca tabu, dentre outros, podem ser adaptados à resolução do referido problema e proporcionar, em diferentes aplicações, uma solução não muito distante da ótima em um tempo de processamento compatível com aquele disponível pelo tomador de decisão.

A literatura acerca do assunto não apresenta, no entanto, qual ou quais as melhores técnicas associadas a diferentes tamanhos de problema. Desta forma, corre-se o risco de escolher, dentre as várias técnicas disponíveis, uma que não seja a mais adequada ao tipo/tamanho do problema de interesse.

Neste sentido, este trabalho consiste na comparação, sob os enfoques *qualidade da solução* e *tempo de processamento*, de diferentes técnicas heurísticas aplicadas ao problema das p-medianas.

## 1.2 Objetivos

O objetivo geral deste trabalho é implementar, avaliar e comparar uma heurística baseada em Algoritmos Genéticos, uma heurística baseada em Busca Tabu e o algoritmo da Substituição de Vértices de Teitz e Bart aplicados ao problema das p-medianas. As principais características a serem avaliadas dos algoritmos são o tempo e a qualidade da solução.

Os objetivos específicos são:

- a) Apresentar o problema das p-medianas.
- b) Elaborar uma revisão bibliográfica do problema das p-medianas mostrando a evolução dos trabalhos mais relevantes.
- c) Implementar os algoritmos de Teitz e Bart, Busca Tabu e Algoritmo Genético.
- d) Analisar os resultados obtidos com o desempenho dos algoritmos baseados em testes em problemas aleatórios de pequeno, médio e grande porte.

### 1.3 Importância

As empresas conseguiram reduzir ao máximo seus custos de produção devido aos avanços na área de produção (Ballou, 2001). Isso faz com que a redução dos custos logísticos, através das ferramentas da otimização, tenha papel essencial nas instituições para aumentar a margem de lucro.

Neste sentido o tempo também é um fator determinante e não pode ser desperdiçado. Sabe-se que problemas classificados como NP – Completo, como é o caso do problema das p-medianas (Garey e Johnson, 1979), podem levar horas, dias, e até anos para serem resolvidos por um computador. Ressalta-se que este custo de tempo computacional não é devido à máquina, mas sim ao problema da explosão combinatorial.

Um problema de médio porte com cem pontos de demanda, ou seja, cem candidatos, buscando-se instalar vinte medianas faz com que as possíveis soluções sejam mais de 500 quintilhões ( $5^{20}$ ). Pensando em um computador que analisa uma solução a cada milionésimo de segundo, isto levaria quase 17 milhões de anos se todas as possibilidades fossem observadas.

Assim, em determinadas circunstâncias pode ser necessária a utilização de algoritmos heurísticos que não pesquisem todas as possibilidades, mas que convirjam para um ponto que não necessariamente é o ponto ótimo do problema, mas sim uma aproximação deste, em tempo viável.

A relevância deste trabalho está em apresentar uma comparação e avaliação dos algoritmos propostos observando o desempenho de cada um em problemas de diferentes tamanhos, fazendo com que, quem venha a utilizar os algoritmos não perca nem em tempo nem em qualidade da solução nas mais diversas aplicações do problema das p-medianas.



## 1.4 Limitações

Este trabalho está limitado ao problema das  $p$ -medianas não capacitado, onde cada facilidade pode atender todos os pontos de demanda próximos a ela como se tivesse capacidade infinita.

Quanto ao tamanho dos problemas gerados, os algoritmos são testados em redes que vão de 50 nós até um máximo de 5000 nós e a distribuição destes pontos é aleatória e não clusterizada.

Além disso, as distâncias entre os pontos são calculadas como se fossem linhas retas (métrica euclidiana) para os quatro algoritmos testados. Este fato se distancia da realidade, mas não é relevante para a análise em questão.

## 1.5 Estrutura do Trabalho

O trabalho está dividido em quatro capítulos incluindo este, onde é apresentada a introdução, os objetivos, a importância e as limitações.

No segundo capítulo é apresentada uma revisão bibliográfica na qual o problema da localização de  $p$ -medianas é definido juntamente com o levantamento dos mais importantes trabalhos nesta área. Também são apresentados os vários métodos de resolução do problema e os resultados apresentados pelos autores.

No terceiro capítulo são apresentadas as implementações dos algoritmos bem como a análise comparativa e avaliação do desempenho.

Finalmente, no último capítulo são apresentadas as conclusões e recomendações para trabalhos futuros.

## CAPÍTULO II

### 2 REVISÃO DA LITERATURA

A seguir será apresentado e definido o problema de localização das  $p$ -medianas e também uma revisão de literatura com alguns dos trabalhos mais importantes neste campo.

Também é apresentado neste capítulo o teorema de Hakimi, a complexidade do problema das  $p$ -medianas e uma classificação das técnicas utilizadas para resolver o problema em algoritmos exatos, heurísticos e metaheurísticos.

#### 2.1 Localização de facilidades

Problemas de localização têm sido estudados pela humanidade desde os tempos mais remotos. De fato, quando uma tribo nômade decide acomodar-se em uma determinada região vários são os fatores que contribuem para essa escolha, como, a água, o alimento, o solo produtivo e a facilidade de acesso. Essa decisão afeta todos os integrantes da tribo.

Quanto às aplicações dos problemas de localização, Christofides (1975) e Larson e Odoni (1981), elencam alguns exemplos:

- Depósitos;
- Indústrias;
- Escolas;
- Bibliotecas;
- Creches;
- Hospitais;
- Postos de saúde;
- Bombeiros;
- Postos policiais;
- Aeroportos;
- Portos;
- Equipes de resgate;
- Área de carga e descarga em depósitos;
- Usinas elétricas;
- Bancos;
- Caixas eletrônicos;
- Antenas de telefonia celular;
- Pontos de taxi e ônibus.

Formalmente, um dos primeiros trabalhos que se tem registro, trata da melhor localização de atividades agrícolas de uma cidade, objetivando a minimização dos custos com transporte. Esta teoria, de 1826, de autoria do economista alemão Johann Heinrich Von Thünen utiliza-se de circunferências ao redor das regiões produtoras que depois ficaram conhecidas como “anéis de Thünen” (Azzoni, 1982).

O problema matemático clássico de localização foi formulado por Fermat que no século XVII, depois Steiner em 1837, analisou a melhor posição de um ponto central, sendo dados três pontos quaisquer, com o objetivo de a soma destas três distâncias ser a mínima possível (Novaes e Rosseto, 1993).

Segundo Azzoni (1982), o modelo seminal que deu origem a teoria da localização deve-se a Alfred Weber, que trata da localização de indústrias. Este trabalho, de 1909, é o primeiro a apresentar uma teoria geral para a localização de uma firma individualmente, considerando os fornecedores de matéria prima e o mercado consumidor. Para Weber, a melhor localização da indústria, é aquela que proporciona o menor custo de produção possível, sendo o fator transporte primordial desse custo. A extensão deste problema deu origem à metodologia das p-medianas.

Estes primeiros trabalhos têm uma abordagem contínua. No entanto, sabe-se que na maioria das vezes as regiões urbanas não são espaços contínuos. Com o desenvolvimento da teoria dos grafos, introduziram-se então os estudos de localização em redes a partir da década de 1960.

A divisão entre problemas contínuos no plano e problemas em redes é bem caracterizada por Galvão (1981). O autor define a localização de facilidades no plano como um problema em um espaço de solução infinito, ou seja, a facilidade pode ser instalada em qualquer ponto do plano, e com as distâncias calculadas segundo um sistema métrico específico. Já na localização em redes, o espaço de soluções consiste apenas em pontos da rede, e as distâncias são medidas ao longo dos arcos.

As redes, segundo Moreira (1990), apresentam características que facilitam a modelagem dos problemas de localização tornando possível o desenvolvimento de procedimentos eficientes para resolvê-los. Entre estas características pode-se citar a natureza discreta das variáveis, o número finito de vértices e arcos, a conectividade entre estes vértices e arcos além de outras propriedades matemáticas relevantes.

De acordo com Larson e Odoni (1981) os problemas de localização em redes se dividem em três categorias: problemas das p-medianas, dos p-centros e os problemas de cobertura de conjunto.

O problema da localização de  $p$ -medianas tem o objetivo de localizar  $p$  facilidades de tal forma que a distância entre estas  $p$  instalações e os  $n$  locais de demanda seja minimizada. Observa-se que não se trata de um caso emergencial, mas sim de minimização das somas das distâncias percorridas (minisum) (Larson e Odoni, 1981).

A localização de  $p$ -centros se encaixa no caso emergencial, conhecido na literatura como minimax. O objetivo é localizar os  $p$  centros de forma a minimizar a máxima distância a qualquer um dos  $n$  pontos de demanda. Esta abordagem se adéqua a problemas de localização de postos policiais, bombeiros, ambulâncias e outros problemas onde o atendimento deve ser o mais rápido possível (Larson e Odoni, 1981).

O objetivo dos problemas de cobertura de conjuntos é minimizar o número de instalações de forma a atender (cobrir) todos os pontos de demanda e ainda determinar a localização ideal das facilidades (Larson e Odoni, 1981).

Atualmente, como já foi citado acima, os problemas de localização têm diversas aplicações, que vão da simples localização de um depósito, fábrica, escola ou posto policial, passando pela localização de antenas, centrais telefônicas, estações de tratamento de água, até a localização de roteadores em redes de computadores.

Muitas contribuições no campo teórico foram desenvolvidas, os principais estão elencados em Brandeau e Chiu (1989) que apresentam 54 problemas com aplicações diferentes da localização de facilidades.

A seguir será apresentado formalmente o problema das  $p$ -medianas, objeto de estudo desta dissertação.

## 2.2 Definição do Problema das p-medianas

De acordo com Pereira (2005), o problema das p-medianas é um problema de localização-alocação, ou seja, visa determinar a configuração de custo mínimo de instalação de facilidades e de atendimento da demanda de cada cliente em uma rede conectada por um número finito de caminhos.

Ainda segundo Pereira (2005), os dados relevantes do problema das p-medianas são: o número finito de pontos de demanda, o número finito de candidatos a instalação de facilidades, a distância entre os pontos de demanda e o número p de facilidades a serem instaladas.

Antes de apresentar formalmente o modelo, faz-se necessário a apresentação de alguns conceitos primordiais. Estas definições, bem como o modelo matemático do problema, foram formuladas por Christofides (1975).

Inicialmente considera-se que cada vértice  $x_j$  esteja associado a um peso  $v_j$  para indicar o tamanho ou importância desse vértice. Desta forma o problema torna-se geral tendo uma maior abrangência.

Um dado grafo  $G$  é uma estrutura matemática denotada por  $G(X, A)$ , onde  $X$  é o conjunto de  $n$  pontos ou vértices  $x_1, x_2, \dots, x_n$  e  $A$  é o conjunto de  $m$  arcos ou ligações  $a_1, a_2, \dots, a_m$ . Os grafos podem ser orientados, não orientados ou mistos. Esses conceitos são fáceis de visualizar na realidade. Por exemplo, se uma cidade é composta somente de ruas com sentido duplo, o grafo resultante será não orientado. Da mesma forma, se uma cidade ou distrito possui ruas com único sentido e com sentido duplo, o grafo resultante será misto.

Feita a representação do problema em um grafo e a ponderação dos pesos de cada um dos vértices, faz-se necessária a construção da matriz de distâncias mínimas. Esta matriz é a representação matricial dos menores caminhos que ligam um vértice a todos os demais vértices da rede e, obviamente, é uma matriz quadrada de tamanho  $n \times n$ . A matriz peso-distância é a matriz de caminhos

mínimos onde cada vértice é multiplicado pelo seu peso. Como exemplo, em um problema de localização de um depósito, o peso pode ser a quantidade de viagens que saem do depósito até o nó de demanda  $n$  em um determinado período.

Para cada vértice  $x_i \in X$ , sendo distância do vértice  $x_i$  ao vértice  $x_j$  representada por  $d(x_i, x_j)$ , define-se como número de transmissão de saída (out-transmission) e número de transmissão de entrada (in-transmission) respectivamente:

$$\sigma_o(x_i) = \sum_{x_j \in X} v_j \cdot d(x_i, x_j) \quad (2.1)$$

$$\sigma_t(x_i) = \sum_{x_j \in X} v_j \cdot d(x_j, x_i) \quad (2.2)$$

O número  $\sigma_o(x_i)$ , denominado número de transmissão de saída, é a soma dos elementos da linha  $x_i$  da matriz peso-distância e o número de transmissão de entrada  $\sigma_t(x_i)$  é a soma dos elementos da coluna  $x_i$ .

Define-se como mediana de saída (out-median) de um grafo o vértice  $\bar{x}_o$  para o qual:

$$\sigma_o(\bar{x}_i) = \min_{x_i \in X} [\sigma_o(x_i)] \quad (2.3)$$

E de modo análogo, a mediana de entrada (in-median) é:

$$\sigma_t(\bar{x}_i) = \min_{x_i \in X} [\sigma_t(x_i)] \quad (2.4)$$

Generalizando o conceito de uma mediana para o caso de  $p$  medianas, considerando para isso um subconjunto  $X_p \subseteq X$  com  $p$  vértices, tem-se a distância deste subconjunto até os demais vértices  $x_j$  como sendo:

$$d(X_p, x_j) = \min_{x_i \in X_p} [d(x_i, x_j)] \quad (2.5)$$

$$d(x_j, X_p) = \min_{x_i \in X_p} [d(x_j, x_i)] \quad (2.6)$$

Os respectivos números de transmissão de saída e de entrada deste subconjunto  $X_p$  de vértices são:

$$\sigma_o(X_p) = \sum_{x_j \in X} v_j \cdot d(X_p, x_j) \quad (2.7)$$

$$\sigma_t(X_p) = \sum_{x_j \in X} v_j \cdot d(x_j, X_p) \quad (2.8)$$

Assim, os números de saída  $\sigma_o(\bar{X}_p)$  e de entrada  $\sigma_t(\bar{X}_p)$  serão:

$$\sigma_o(\bar{X}_{po}) = \min_{X_p \subseteq X} [\sigma_o(X_p)] \quad (2.9)$$

$$\sigma_t(\bar{X}_{pt}) = \min_{X_p \subseteq X} [\sigma_t(X_p)] \quad (2.10)$$

Desta forma, considerando que cada vértice é uma mediana em potencial, define-se formalmente o problema da localização de p-medianas da seguinte forma:

Seja  $G(X, A)$  um grafo não orientado onde  $X$  é o conjunto de vértices e  $A$  é o conjunto de arestas. Deve se encontrar um conjunto de vértices  $X_p \subset X$  (conjunto de medianas) com cardinalidade  $p$ , tal que, a soma das distâncias de cada vértice restante em  $\{X - X_p\}$  (conjunto de demandas) até seu vértice mais próximo em  $X_p$  seja a mínima possível.



## 2.3 Formulação Matemática do Problema das p-medianas

Para o caso de localização de p-medianas com a utilização de programação matemática inteira, segundo Christofides (1975), tem-se:

$$\text{Minimizar } Z = \sum_{i=1}^n \sum_{j=1}^n d_{ij} \cdot \xi_{ij} \quad (2.11)$$

Sujeito a:

$$\sum_{i=1}^n \xi_{ij} = 1 \quad \forall j = 1, \dots, n \quad (2.12)$$

$$\sum_{i=1}^n \xi_{ii} = p \quad (2.13)$$

$$\xi_{ij} \leq \xi_{ii} \quad \forall i, j = 1, \dots, n \quad (2.14)$$

$$\xi_{ij} \in \{0, 1\} \quad (2.15)$$

Onde:

$n$  é o número de nós do grafo do problema,

$p$  é o número de medianas a ser instalado,

$[d_{ij}]$  a matriz de distâncias ponderadas, onde  $d_{ij}$  é o produto da distância entre os vértices  $x_i$  e  $x_j$  pelo peso  $w_j$ , sendo  $w_j$  a demanda de cada vértice  $x_j$ .

$[\xi_{ij}]$  é a matriz de alocação, que será 1 se o nó  $x_j$  é alocado ao nó  $x_i$  e 0 em caso contrário.

A restrição (2.12) garante que um dado nó  $x_j$  é alocado a somente um nó  $x_i$ , que é mediana. A restrição (2.13) garante o número de medianas desejado. A restrição (2.14) garante que o ponto  $x_j$  só esteja alocado à mediana  $x_i$  e a restrição (2.15) corresponde às condições de integralidade.

Ainda quanto à definição e formulação do problema das p-medianas, Minieka (1977), classifica o problema em medianas e medianas absolutas. O problema das medianas absolutas, objetiva encontrar a melhor localização nos vértices e nos arcos, enquanto que o problema das p-medianas procura a melhor solução somente nos nós do grafo.

Christofides (1975), ainda divide o problema em outros dois casos: p-medianas puro e p-medianas generalizado. O problema puro consiste em não considerar o custo de localização das facilidades. Já o problema das p-medianas generalizado introduz os custos de localização das instalações na função objetivo.

Neste trabalho será utilizado somente o caso das p-medianas puro, já que o objetivo aqui é avaliar os algoritmos.

## **2.4 Trabalhos relevantes que utilizam o modelo das p-medianas**

Neste setor serão apresentados alguns trabalhos relevantes da teoria da localização que fazem uso do problema das p-medianas.

As primeiras generalizações do problema de Weber, ou seja, que tratam da localização de depósitos para indústrias privadas são: Cooper (1963), Kuehn e Hamburger (1963), Eilon et al.(1971), Wesolowsky e Love (1971) e Corneujols et al. (1977).

Talvez o mais importante trabalho da década de 1960 se deva a Hakimi (1964 e 1965). O teorema de Hakimi prova que ao menos um conjunto de pontos ótimos

do problema será constituído de nós do grafo. A demonstração desse teorema, que será apresentada no próximo item, faz com que as heurísticas de busca não percam tempo procurando a solução sobre os arcos do grafo.

O método mais citado na literatura e considerado como um dos melhores em desempenho se deve a Teitz e Bart (1968). Este método é chamado de Algoritmos de Substituição de Vértices (node substitution). Como o nome menciona, o algoritmo consiste em partir de uma solução inicial viável e substituir os vértices do grafo um a um até que não haja mais melhoria. Este método é considerado na literatura como um dos melhores na solução do problema das p-medianas. No entanto, o algoritmo pode convergir para um mínimo local e encerrar a busca.

Tondo (1992), abordou a modelagem da localização de um conjunto de contêineres no estado de São Paulo resolvendo o problema com a utilização da metodologia das p-medianas.

Ao apresentar uma metodologia aplicada ao extrativismo do babaçu, Bezerra (1995), busca racionalizar o trabalho de transporte entre o local de coleta e o de processamento. Para a otimização do problema foram utilizados três algoritmos: o de Floyd para determinar as distâncias mínimas entre cada nó que compõem a rede viária; o de p-medianas de Teitz e Bart para a localização dos postos de coleta, e um Algoritmo Genético para determinar o roteiro para o problema do caixeiro viajante.

Souza (1996), utiliza a abordagem do problema das p-medianas adaptado para o caso emergencial com a justificativa de que em países com menos recursos financeiros como o Brasil, não se tem normalmente um número adequado de equipamentos para atender todos os usuários.

Lima (1996), utilizou o modelo das p-medianas para localizar centros de serviços de saúde, onde se buscou minimizar a distância a ser percorrida pelos usuários. Sua aplicação apresentou uma proposta para os serviços de cardiologia em

Santa Catarina de forma hierárquica, com hospitais locais, centros intermediários e um centro de referência no estado.

O estudo da melhor localização de centros de assistência materna e perinatal no município de Rio de Janeiro é o tema do estudo de Vasconcelos (1997). Neste trabalho o autor utiliza o modelo das p-medianas interligado a sistemas de informações geográficas (SIG). O problema é resolvido com o algoritmo de Teitz e Bart e também por um algoritmo exato proposto por Galvão e Raggi, as duas soluções são comparadas e é apresentado a distribuição espacial dos serviços de saúde.

Lobo (2003), aplicou o problema das p-medianas juntamente com diagramas de Voronoi para determinar um dimensionamento adequado para as unidades de educação infantil. Já Nunes (1998) utilizou o Algoritmo Genético para resolver o problema das p-medianas aplicado a um problema de roteirização calculando os pontos de parada das rotas.

No estudo de Sampaio (1999), utilizou-se o problema de p-medianas capacitado para verificar a necessidade de ampliação ou redução da capacidade de escolas estaduais no município de Curitiba. Enquanto que Corrêa (2000) apresentou um estudo sobre o problema de localização de facilidades, objetivando otimizar a designação de candidatos ao vestibular para os locais de provas mais próximos de suas residências. Trata-se também de problema de p-medianas capacitado, para o qual foram propostas duas heurísticas, uma baseada em um Algoritmo Genético simples que utiliza os operadores genéticos usuais e um operador heurístico chamado hiper-mutação direcionada e a outra heurística é baseada em Busca Tabu.

Silva (2006) faz um levantamento sobre a Logística contemporânea e a conseqüente necessidade de uma rede de distribuição integrada especificamente em uma indústria siderúrgica. Nesse sentido se faz necessário encontrar a melhor disposição dos estoques da empresa, e para isso, o autor

propõe um modelo de localização de instalações levando em conta questões táticas como a distribuição dos estoques na rede logística.

No estudo de Bandeira (2006), é apresentada uma metodologia que identifica e caracteriza áreas candidatas à localização e o número de depósitos a serem instalados. A autora faz uma aplicação em uma empresa de suprimentos industriais.

Pires (2002) faz uso do Algoritmo Genético para encontrar a melhor localização de escolas no município de Coronel Vivida no Paraná, já que o município contava com escolas superlotadas e outras subutilizadas. O modelo das p-medianas foi resolvido para equilibrar a distribuição dos alunos e minimizar a distância a ser percorrida pelos usuários. Lorena et al. (2004), avaliaram em um artigo a melhor localização de escolas de Ensino Fundamental no cidade de Vitória – ES utilizando para isso o modelo de p-medianas capacitado e não-capacitado.

Pizzolato (1994), propôs um método para o problema das p-medianas destinado para redes grandes. A técnica se inicia com a construção de p árvores cujas formas vão progressivamente sendo modificadas de acordo com os sucessivos testes. Neste mesmo trabalho é apresentada uma aplicação da técnica para localização de 85 escolas públicas na cidade do Rio de Janeiro.

A metodologia das p-medianas é utilizada por Rosa (2006) para maximizar o valor presente líquido do projeto de desenvolvimento de campos de petróleo offshore. A representação da superfície do fundo do mar dá-se neste trabalho através de um grafo e é feita a comparação entre a maximização do valor presente líquido e a minimização de custos de investimentos.

Uma comparação entre um algoritmo que usa Busca Tabu e um algoritmo de Relaxação Lagrangeana aplicado ao problema das p-medianas capacitado é feita no trabalho de Ducati (2003). A partir de testes computacionais a autora conclui que, contrariando a maioria dos trabalhos, o algoritmo de Busca Tabu

apresenta um desenvolvimento melhor tanto em tempo de processamento quanto em qualidade de solução.

## 2.5 Resultados obtidos por Hakimi

O teorema de Hakimi, como ficou conhecido, prova a existência de um conjunto formado apenas por vértices que gera um resultado tão bom quanto um conjunto localizado em qualquer parte do grafo. Isso facilita a busca pela melhor solução fazendo com que o algoritmo não pesquise pontos que estão sobre os arcos.

Considerando uma rede não orientada pode-se ignorar os índices  $t$  e  $o$  de entrada e saída respectivamente. E buscando encontrar uma mediana (1-mediana), a pergunta que se faz, é se existe um ponto  $y$  em algum arco, não necessariamente num vértice do grafo de forma que o número de transmissão:

$$\sigma(y) = \sum_{x_j \in X} v_j \cdot d(y, x_j)$$

seja menor que a mediana do grafo. Se tal ponto existir, pode-se chamar este ponto  $\bar{p}$  de “Mediana Absoluta” do grafo  $G$ . Hakimi provou que este ponto  $\bar{y}$  não existe (Larson e Odoni, 1981).

Teorema 1: Existe pelo menos um vértice  $x$  do grafo  $G$  para o qual  $\sigma(x) \leq \sigma(y)$  para qualquer ponto arbitrário  $y$  em  $G$ .

Uma forma de demonstrar o Teorema de Hakimi, baseada em Larson e Odoni (1981), é:

Hipótese: supõe-se que a localização ótima de uma facilidade é um ponto  $x$  entre os nós  $p$  e  $q$ , ou seja, pertencente ao arco  $(p, q)$ , tal que  $J(x) < J(p)$ , para todo  $x$ .

Seja  $G(X, A)$  um grafo não orientado. Seja  $k$  positivo e inteiro. O conjunto de  $k$  pontos distintos, do grafo  $G$ , é dado por  $X_k = \{x_1, x_2, \dots, x_k\}$ . E a distância mínima,  $d(X_k, j)$ , entre os pontos  $x_i$  de  $X_k$  e o vértice  $j$  de  $G$ , dado por:

$$d(X_k, j) = \min_{x_i \in X_k} d(x_i, j)$$

A definição do conjunto de  $k$ -medianas do grafo é dado por:  $k$  pontos de  $X_k^*$  em  $G$ , se para todo  $X_k$  de  $G$  temos:

$$J(X_k^*) \leq J(X_k)$$

Onde:

$$J(X_k) = \sum_{j=1}^n h_j \cdot d(X_k, j)$$

Segue abaixo a demonstração do teorema de Hakimi para  $k = 1$ :

A distância  $d(x, j)$  entre  $x$  e um vértice  $j$  de  $X$  é:

$$d(x, j) = \text{Min}\{d(x, p) + d(p, j), d(x, q) + d(q, j)\}$$

$P$  é o conjunto de nós que são mais eficientemente acessado por  $p$  e  $Q$  é o conjunto de pontos que são melhor acessados por  $q$ .

$$X = P \cup Q$$

Assume-se agora que mais usuários são acessados através de  $p$ :

$$\sum_{j \in P} h_j \geq \sum_{j \in Q} h_j$$

Utilizando os conceitos apresentados, pode-se escrever:

$$J(x) = \sum_{j \in N} h_j \cdot d(x, j)$$

$$J(x) = \sum_{j \in P} h_j [d(x, p) + d(p, j)] + \sum_{j \in Q} h_j [d(x, q) + d(q, j)]$$

$$J(x) = \sum_{j \in P} h_j [d(x, p) + d(p, j)] + \sum_{j \in Q} h_j [d(p, q) - d(x, p) + d(q, j)]$$

$$J(x) = \sum_{j \in P} h_j d(x, p) + \sum_{j \in P} h_j d(p, j) + \sum_{j \in Q} h_j d(p, q) - \sum_{j \in Q} h_j d(x, p) + \sum_{j \in Q} h_j d(q, j)$$

$$J(x) = d(x, p) \left( \sum_{j \in P} h_j - \sum_{j \in Q} h_j \right) + \sum_{j \in P} h_j d(p, j) + \sum_{j \in Q} h_j [d(p, q) + d(q, j)]$$

Da definição de distância  $d(p, j)$ , tem-se:

$$d(p, j) \leq d(p, q) + d(q, j)$$

Então:

$$J(x) \geq d(x, p) \left( \sum_{j \in P} h_j - \sum_{j \in Q} h_j \right) + \sum_{j \in P} h_j d(p, j) + \sum_{j \in Q} h_j d(p, j)$$

$$J(x) = d(x, p) \left( \sum_{j \in P} h_j - \sum_{j \in Q} h_j \right) + \sum_{j \in N} h_j d(p, j)$$

$$J(x) = d(x, p) \left( \sum_{j \in P} h_j - \sum_{j \in Q} h_j \right) + J(p)$$

Assumimos inicialmente que:



$$\sum_{j \in P} h_j - \sum_{j \in Q} h_j \geq 0$$

Isto implica que o produto:

$$d(x, p) \left( \sum_{j \in P} h_j - \sum_{j \in Q} h_j \right)$$

é maior ou igual a zero e isso nos leva a concluir que:

$$J(x) \geq J(p)$$

o que contradiz nossa hipótese.

Assim prova-se que ao menos uma boa solução movendo a facilidade de  $x$  para  $p$ , o que completa a demonstração.

O segundo teorema é uma generalização deste que acabou de ser demonstrado:

Teorema 2: existe pelo menos um subconjunto  $X_p \subseteq X$  contendo  $p$  vértices, tais que  $\sigma(X_p) \leq \sigma(Y_p)$  para qualquer conjunto arbitrário de  $Y_p$  de  $p$  vértices localizados nos arcos ou nos vértices do grafo  $G(X, A)$ .

O desenvolvimento destes dois teoremas têm sido fundamental para o desenvolvimento de todos os algoritmos elaborados para o problema das  $p$ -medianas, já que eles fazem com que possamos restringir a busca somente aos vértices de  $G$ .

## 2.6 Complexidade do Problema das p-Mediana

Os problemas denominados combinatoriais apresentam como característica principal a dificuldade de se encontrar uma solução por meio de algoritmos exaustivos, devido à explosão combinatorial, o que implicaria em um tempo computacional enorme. Segundo Mercado (2001), com o objetivo de distinguir os problemas de fácil e de difícil solução classificam-se os problemas de otimização combinatorial em Polinomiais (P) e Não-Polinomiais (NP).

Os problemas polinomiais possuem algoritmos que, na pior das hipóteses, precisam de um número de operações necessárias para encontrar a solução ótima limitado por uma função de tempo polinomial.

Um problema classificado como Não-Polinomial (non deterministic polynomial time algorithm) é um problema computável, cujas soluções até então conhecidas, são de ordem exponencial ou fatorial e não se sabe se existe uma solução melhor de complexidade polinomial.

A importância do estudo da complexidade pode ser facilmente observada na tabela abaixo. Como geralmente existem vários algoritmos para solução de um determinado problema, o exemplo mostra cinco algoritmos (A, B, C, D e E), de complexidades diferentes para resolver um mesmo problema. Vamos supor que uma operação leve um milissegundo (1ms). A tabela abaixo mostra o tempo de execução de cada algoritmo.

Tabela 2.1: Tabela de complexidade.

$n$	$T_A(n) = n$	$T_B(n) = n \log n$	$T_C(n) = n^2$	$T_D(n) = n^3$	$T_E(n) = 2^n$
2	0.002s	0.001s	0.004s	0.008s	0.004s
4	0.004s	0.002s	0.016s	0.064s	0.016s
8	0.008s	0.007s	0.064s	0.512s	0.256s
16	0.016s	0.019s	0.256s	4.096s	1.055 min.
32	0.032s	0.048s	1.024s	32.768s	1.65 meses
64	0.064s	0.116s	4.096s	4.369 min.	$5,9 \cdot 10^6$ séc.
128	0.128s	0.270s	16.384s	34.952 min.	$10^{26}$ séc.

Na tabela,  $n$  é o tamanho da entrada e  $T_K(n)$  é a complexidade de cada um dos  $K$  algoritmos supostos.

Apesar do desenvolvimento de computadores mais rápidos, torna-se cada vez mais importante desenvolver algoritmos eficientes, devido ao constante aumento no tamanho dos problemas a serem resolvidos.

A eficiência de um algoritmo é medida com base em duas variáveis, tempo e espaço. O espaço representa a quantidade de memória a ser utilizada pelo algoritmo para a solução do problema. O tempo é quantidade de tempo requerida para solucionar o problema, mas não o tempo real do processo, pois isso seria diferente de computador para computador, mas o tempo medido em quantidade de operações dominantes.

Uma das notações mais utilizadas para análise de algoritmos é o estudo do limite superior (upper bound) para o comportamento assintótico de uma função, representado por  $O(n)$ . Esta classificação consiste basicamente em determinar o pior dos casos possíveis para cada algoritmo. Por exemplo, se no lançamento de um dado comum deseja-se obter face três, a complexidade seria constante, ou seja,  $O(6)$ .

Define-se que a função custo  $C(n)$  é  $O(F(n))$  se existe constantes positivas  $c$  e  $n_0$  tais que:

$$C(n) \leq c \cdot F(n) \text{ quando } n \geq n_0.$$

A notação  $O(F(n))$  diz que existe um ponto  $n_0$  tal que para todos os tamanhos de entrada  $n$  superiores a  $n_0$ , o custo computacional é inferior a algum múltiplo de  $F(n)$ , ou seja,  $F(n)$  cresce mais rapidamente que  $C(n)$ .

Assim, dizer que a complexidade de um algoritmo é  $O(n^2)$  significa dizer que a quantidade de tempo utilizada pelo algoritmo para solução do problema é no máximo uma função quadrática que depende do volume  $n$  de entradas.

Hakimi (1978) demonstrou que para um grafo de  $n$  vértices a complexidade do problema das  $p$ -medianas é da ordem de  $O\left(\frac{n^p}{p!}\right)$ .

## 2.7 Algoritmos Exatos

Os algoritmos exatos buscam encontrar o ótimo do problema, fato que em problemas de médio e grande porte faz o algoritmo se tornar inviável devido ao tempo de processamento da solução.

Os métodos exatos, além da enumeração exaustiva, utilizam-se de técnicas de busca em árvore (Branch-and-Bound ou Tree Search), relaxação da programação linear, métodos duais e métodos baseados em relaxação lagrangeana (Galvão, 1981).

A técnica da enumeração exaustiva, ou simplesmente busca exaustiva, consiste em avaliar todas as soluções viáveis até que seja encontrada a solução ótima. Este método requer que seja calculado o valor da função objetivo para cada uma das possíveis configurações, e isto torna o algoritmo inutilizável em problemas de médio e grande porte.

Um dos trabalhos de maior relevância nesta categoria dos algoritmos exatos se deve a Järvinen et al. (1972). Neste algoritmo de busca em árvore proposto pelos autores, todos os nós do grafo são inicialmente rotulados como “abertos” e um limite inferior definido é utilizado para “fechar” sucessivamente as facilidades, até que exatamente  $p$  vértices estejam abertos. O processo termina quando todas as possíveis soluções são avaliadas.

El-Shaieb (1973), Khumawala (1972) e Neebe (1978) também propõem algoritmos branch-and-bound para o problema das  $p$ -medianas, no entanto, estes trabalhos se limitam a poucas dezenas de nós.

Os primeiros trabalhos que fazem uso da relaxação da programação linear foram desenvolvidos por ReVelle e Swain (1970), Swain (1974), Garfinkel et al. (1974), e Galvão e Raggi (1989). Nestes algoritmos as restrições de integralidade (2.15) são trocadas por  $\xi_{ij} \geq 0$  para todos os valores de  $i$  e  $j$  até  $n$ . Embora a grande maioria dos casos resulte em 0 ou 1, existe a possibilidade de  $\xi_{ij}$  assumir valores fracionários. Outro problema deste tipo de formulação é o excessivo tamanho do problema de programação linear correspondente.

Algoritmos que utilizam o dual do problema foram desenvolvidos por Bilde e Krarup (1977), Erlenkotter (1978), e Galvão (1980).

Marsten (1972) foi quem primeiro publicou um trabalho utilizando relaxação lagrangeana para resolver o problema das  $p$ -medianas. O trabalho mostra que a solução ótima do problema é um ponto extremo de um poliedro  $H$ . O algoritmo faz uso dos multiplicadores de Lagrange para construir um caminho através dos pontos extremos do poliedro  $H$ . Este caminho gera sucessivamente soluções para valores decrescentes de  $p$ . Apesar de o método ser computacionalmente atraente, pode-se encontrar valores fracionários de  $p$  e se pode não encontrar solução para determinados valores de  $p$ .

Narula et al. (1977) e Christofides e Beasley (1982) também apresentaram algoritmos para o problema das  $p$ -medianas baseado em relaxação lagrangeana.

## 2.8 Algoritmos Heurísticos

Devido à necessidade de solução de problemas cada vez maiores, os algoritmos exatos se tornaram ineficientes devido ao problema da explosão combinatorial. Surgiram então os algoritmos heurísticos. As heurísticas são métodos ou algoritmos exploratórios para a solução de problemas de otimização combinatorial. Algoritmos heurísticos não garantem que o ótimo seja encontrado, mas resolvem problemas de grande porte em tempo viável e encontrando uma solução satisfatória.

O primeiro algoritmo heurístico para o problema das  $p$ -medianas foi proposto por Maranzana (1964). A técnica ficou conhecida como Método da Partição de Vértices por buscar sucessivos vértices únicos de  $m$  subconjuntos destinos, cada um associado com uma origem. Estes conjuntos vão sendo modificados e o processo é repetido até que se tenha uma solução satisfatória.

A heurística gulosa (greedy heuristic), também conhecida como método da adição de vértices, foi proposta inicialmente por Kuehn e Hamburger (1963) e consiste em adicionar recursivamente medianas a solução. Iniciando com uma mediana e adicionando outras seqüencialmente até que  $p$  medianas tenham sido incluídas. A escolha do vértice a ser incorporado na solução é feita com base no nó que produz o máximo decréscimo na função objetivo. Este procedimento é normalmente utilizado como passo inicial dos algoritmos que precisam de uma solução inicial viável.

Cornuejols et al. (1977) deram importante contribuição no campo teórico. O estudo apresenta métodos heurísticos e relaxamentos para os problemas de localização não-capacitados.

O algoritmo mais citado na literatura para solução do problema das  $p$ -medianas é o concebido por Teitz e Bart (1968). Conhecido como Método da Substituição de Vértices. A busca se inicia com uma solução qualquer viável e vários vértices são testados, o vértice candidato que propiciar a maior redução no custo da função objetivo entra na solução. O procedimento é repetido até que não se obtenha mais melhorias. Esta técnica será estudada com mais detalhes no próximo capítulo.

Eilon e Galvão (1978) apresentaram um algoritmo que é uma melhoria do método da substituição de vértices, utilizando para isso o conceito de  $\lambda$ -optimalidade.

Muitas outras técnicas heurísticas são derivadas do algoritmo proposto por Teitz e Bart. Podendo-se citar as de Whitaker (1983), Densham e Rushton (1992) e Resende e Werneck (2002).

O método de Pizzolato (1994) consiste em construir  $p$  árvores disjuntas. As árvores vão modificando suas formas, conforme testes efetuados sobre suas raízes e vértices podendo ocorrer à eliminação de uma árvore ou a partição em duas, bem como modificações nas raízes e transferência de vértices de uma para outra árvore.

Hribar e Daskin (1997) elaboraram uma heurística híbrida que utiliza o algoritmo guloso juntamente com programação dinâmica. O algoritmo encontra várias soluções boas, que provavelmente fazem parte da solução ótima, ao longo da busca e contabiliza a frequência que eles fizeram parte da solução. A heurística também identifica os locais que não devem fazer parte da solução, tratando de excluí-los. O método foi testado em problemas onde o mínimo é conhecido chegando ao ótimo em muitos deles.

## **2.9 Algoritmos Metaheurísticos**

As metaheurísticas são métodos de busca inteligentes flexíveis, pois possuem uma estrutura com componentes genéricos que são adaptados aos diversos tipos de problemas de otimização. Esses métodos possuem certa facilidade em incorporar novas situações e exploram o espaço de solução permitindo a escolha de soluções piores que as já encontradas, na tentativa de superar a otimalidade local. Assim, uma metaheurística pode ser implementada para guiar determinada técnica a comandar o processo de busca da solução.

As metaheurísticas se dividem em duas categorias: técnicas de busca local e de busca populacional.

Os métodos de busca local se baseiam no conceito de vizinhança, partindo de uma solução inicial e explorando soluções que podem ser alcançadas em um movimento. As técnicas de Busca Tabu e Simulated Annealing são exemplos de busca local.

As metaheurísticas de busca populacional partem de um conjunto de soluções denominado população inicial e sobre essa população são aplicados os diversos operadores com o objetivo de melhorar os indivíduos e gerar novos integrantes. Como exemplo dessa categoria, pode-se citar os Algoritmos Genéticos e a técnica Scatter Search.

Nas últimas duas décadas, a maioria dos trabalhos apresentados aplicados ao problema das  $p$ -medianas, fazem uso de metaheurísticas. Entre as principais utilizadas estão os Algoritmos Genéticos, Busca Tabu, Simulated Annealing, Ant Colony, GRASP e Redes Neurais.

A Busca Tabu é uma metaheurística que guia os procedimentos de busca local para explorar soluções em todo o espaço possível, isso permite ao algoritmo ir além do ótimo local. O procedimento de Busca Tabu foi inicialmente introduzido por Glover (1986) para resolução de problemas de otimização combinatorial.

O primeiro trabalho a aplicar o conceito de Busca Tabu ao problema de  $p$ -medianas se deve a Schilling et al. (1996). A busca proíbe alguns movimentos de acordo com um determinado critério, e estes movimentos são chamados de tabu. Esta restrição faz com que pontos visitados freqüentemente não sejam novamente utilizados.

Sun (2006) elabora e testa um algoritmo de Busca Tabu em problemas com até 2000 vértices. E em todos os problemas, o algoritmo de Busca Tabu obteve melhores resultados em menos tempo computacional.

Os Algoritmos Genéticos (AG) formam uma classe de algoritmos baseada na teoria da evolução das espécies de Charles Darwin. Estas heurísticas também



são conhecidas como algoritmos evolucionários. Os primeiros trabalhos neste campo se devem a John Holland na década de 70.

Como se sabe, a teoria da evolução se baseia na seleção natural. De acordo com a teoria darwiniana, uma população que se encontra sujeita a um ambiente qualquer, sofrerá sua influência de maneira que os mais adaptados ao meio terão maiores chances de sobrevivência.

Hosage e Goodchild (1986) foram os primeiros pesquisadores a aplicar o Algoritmo Genético ao problema das p-medianas com 20 vértices e 3 medianas, eles geraram 100 problemas aleatórios e encontraram a solução ótima entre 70% e 90% dos casos. No entanto o algoritmo trabalha com a representação da solução binária, gerando um espaço com  $2^{20}$  possibilidades, fazendo com que se desperdice memória e requerendo operações desnecessárias.

Dibble e Densham (1993) elaboraram uma representação mais adequada de um cromossomo para o problema das p-medianas, onde cada cromossomo tem exatamente p genes e cada gene traz o número de uma mediana. Os resultados obtidos por este algoritmo genético foram praticamente os mesmos dos obtidos pelo método de Teitz e Bart, porém o algoritmo genético requereu um tempo de processamento maior.

Nunes (1998) e Sampaio (1999) fizeram uso do Algoritmo Genético proposto por Mayerle (1996). Este algoritmo difere dos anteriores por utilizar o operador genético de seleção dado por:

$$Seleção(i) = \left\{ i = \left\lfloor \frac{-1 + \sqrt{1 + 8r}}{2} \right\rfloor \right\}$$

Onde cada indivíduo  $i$  é associado a um número randômico, com  $rnd() \in [0, 1)$ ,  $r$  é um número pertencente ao intervalo  $[0; \frac{m^2+m}{2})$ ,  $m$  é o tamanho da população, fazendo com que  $r = rnd() \cdot (\frac{m^2+m}{2})$ .

Mercado (2001) elabora um Algoritmo Genético aplicado ao problema das  $p$ -medianas e compara o algoritmo ao método da substituição de vértices. A autora obteve resultados melhores em redes de tamanho pequeno com ganhos de 1%. Já em redes médias e grandes, obtiveram-se resultados equivalentes com tempo computacional maior.

Erkut et al. (2003) apresentaram importante trabalho aplicando o Algoritmo Genético ao problema das  $p$ -medianas. Neste simples e rápido algoritmo, a função de fitness é utilizada como função objetivo para determinar ou não a permanência de um vértice na solução. Outra característica interessante deste algoritmo é que ele permite passagens por soluções inviáveis, este fato permite que a busca ocorra de modo mais rápido.

Correa et al. (2004) apresentam um algoritmo baseado na teoria darwiniana da evolução que introduz um novo operador genético denominado hipermutação. Este operador faz com que uma porcentagem dos cromossomos da geração futura tenha uma melhoria na sua aptidão.

Além da Busca Tabu e do Algoritmo Genético, outras metaheurísticas tem sido aplicadas ao problema.

O estudo de Galvão e Chiyoshi (2000) apresenta uma análise estatística do Simulated Annealing aplicado ao problema de  $p$ -medianas, no qual o algoritmo usado combina elementos do método de substituição de vértices de Teitz e Bart com a metodologia geral Simulated Annealing.

O trabalho de Merino et al. (2003) utiliza o conceito de redes neurais para elaborar três algoritmos aplicados ao problema das  $p$ -medianas. Segundo os autores a eficiência dos algoritmos depende do número de pontos de demanda e do número de facilidades a serem instalados.

Resende e Werneck (2004) apresentam um algoritmo baseado na metaheurística GRASP (Greedy Randomized Adaptive Search Procedure). Cada

iteração deste algoritmo aplica um algoritmo guloso randômico seguido por um processo de busca local que memoriza as melhores soluções. Ao fim do algoritmo, as melhores soluções são combinadas entre si.

A produção de trabalhos que utilizam as metaheurísticas para solução de problemas de otimização combinatorial tem crescido consideravelmente na literatura e duas delas, Busca Tabu e Algoritmos Genéticos serão apresentados no próximo capítulo.

## CAPÍTULO III

### 3 IMPLEMENTAÇÃO E RESULTADOS COMPUTACIONAIS

Nesta seção é apresentada a métrica euclidiana, são descritos os algoritmos utilizados para a solução do problema das p-mediana não capacitado e é feita a comparação entre eles.

#### 3.1 Métrica Euclidiana

A análise dos algoritmos será feita considerando as distâncias entre os pontos como linhas retas, para isso é importante definir métrica e métrica euclidiana.

Matematicamente, métrica em um conjunto  $M \neq \emptyset$  é uma função  $d: M \times M \rightarrow R$ , que associa a cada par  $(x, y)$  uma distância  $d$ , tal que:

$$i) d(x, y) \geq 0, \text{ com } d(x, y) = 0 \Leftrightarrow x = y$$

$$ii) d(x, y) = d(y, x)$$

$$iii) d(x, z) \leq d(x, y) + d(y, z)$$

As métricas mais utilizadas e conhecidas são a métrica euclidiana e a métrica retangular.

A métrica euclidiana utiliza-se do princípio de que o caminho mais curto entre dois pontos é uma reta e consiste na mensuração desta reta. Já a métrica retangular se encaixa nos caminhos perpendiculares das grandes metrópoles planejadas.

A distância euclidiana (DE) entre dois pontos A e B quaisquer é dada por:

$$DE_{AB} = [(x_B - x_A)^2 + (y_B - y_A)^2]^{1/2}$$

A distância retangular (DR) é dada por:

$$DR_{AB} = [|x_B - x_A| + |y_B - y_A|]$$

Quando utilizada uma das métricas, em situações reais, raramente o trajeto entre dois pontos é exatamente uma reta ou um retângulo, para isso faz-se uso de coeficientes de correção. Esse fator de correção, segundo Novaes (1989), pode ser obtido por regressão linear simples, tendo assim:

$$D_{AB} = a + bDE_{AB}$$

Ou

$$D_{AB} = a + bDR_{AB}$$

Nos estudos de Love et al (1988) e Novaes (1989) recomenda-se o uso do fator de correção de 30% sobre a métrica euclidiana em malhas urbanas para se obter uma aproximação da distância efetiva, ou seja,  $a = 0$  e  $b = 1,3$ .

Love et al (1988), comentam que é conveniente ressaltar que as métricas acima são dois casos especiais para valores  $\rho = 1$  e  $\rho = 2$  das distâncias de Minkowski:

$$D_{\rho(AB)} = [|x_B - x_A|^\rho + |y_B - y_A|^\rho]^{1/\rho}, \quad 1 \leq \rho \leq \infty$$

Para o objetivo proposto nesse trabalho, onde se considera o problema das  $p$ -medianas sem restrições de percurso, utilizar-se-á a métrica euclidiana.

### 3.2 Algoritmo de Teitz e Bart

O método de Teitz e Bart, ou método da Substituição de Vértices (Node Substitution), é um algoritmo que foi apresentado por Michael B. Teitz e Polly Bart no artigo intitulado Heuristic methods for estimating the generalized vertex median of a weighted graph, em 1968. O método é citado na literatura como um dos melhores algoritmos para o problema das  $p$ -medianas.

Informalmente, o algoritmo consiste em partir de uma solução aleatória inicial e a partir dela comparar um a um os nós que estão fora da solução atual, substituindo os nós mais bem colocados na função objetivo.

O procedimento pode ser descrito da seguinte forma:

**Passo 1)** Selecionar um conjunto  $S$  de  $p$  vértices, formando uma aproximação inicial do conjunto ótimo  $\bar{X}_p$ . Chamar todos os vértices  $x_j \notin S$  de vértices “não testados” e os  $x_i \in S$  de “testados”.

**Passo 2)** Selecionar um vértice não testado  $x_j \notin S$ , compute a “redução”  $\Delta_{ij}$  no número de transmissão, se  $x_i$  é substituído por  $x_j$ , isto é, compute:

$$\Delta_{ij} = \sigma(S) - \sigma(S \cup \{x_j\} - \{x_i\})$$

**Passo 3)** Encontre  $\Delta_{i_0j} = \max_{x_i \in S} [\Delta_{ij}]$

- i) Se  $\Delta_{i_0j} \leq 0$  rotule o vértice  $x_j$  como testado e volte ao passo 2.
- ii) Se  $\Delta_{i_0j} > 0$  efetuar  $S \leftarrow S \cup \{x_j\} - \{x_{i_0}\}$ , rotular  $x_j$  como testado e volte ao passo 2.

**Passo 4)** Repetir 2 e 3 até que todos os vértices de  $X$  estejam rotulados como “testados”. Este procedimento é referido como ciclo.

Se, durante o último ciclo nenhuma substituição foi feita no passo 3(ii), vá ao passo 5. Caso contrário, se foi feita alguma substituição, rotule todos os vértices como “não testados” e retorne ao passo 2.

**Passo 5)** Pare. O conjunto  $S$  atual é o conjunto de  $p$ -medianas  $\bar{X}_p$ .

Observa-se que o passo 1 do algoritmo consiste em selecionar um número  $p$  de medianas para fazerem parte da solução inicial do problema denominado conjunto  $S$ , considerando uma solução como uma aproximação do conjunto ótimo. Ainda neste passo os nós que não fazem parte da solução inicial,  $x_j \notin S$ , são rotulados como “não testados”.

No passo 2 é feita a seleção de um vértice  $x_j \notin S$  qualquer para ser comparado aos vértices  $x_i$  que estão na solução. É feita a comparação com o vértice  $x_j$  entre todos os vértices  $x_i$  e são computadas as reduções, ou acréscimos, no número de transmissão ( $\sigma(S)$ ), que é representado por  $\Delta_{ij}$ .

O passo 3 é responsável pela escolha da redução máxima ( $\Delta_{i_0j}$ ). Se a redução máxima é positiva, o nó  $x_j$  é substituído e passa a fazer parte da nova solução, caso contrário, o nó  $x_j$  é rotulado como testado e volta-se ao passo 2.

O ciclo é caracterizado no passo 4. É nesta etapa que os passos 2 e 3 são repetidos até que não haja mais alterações durante o último ciclo completo. Caso isso ocorra, o passo 5 é responsável pelo fim do algoritmo e o conjunto solução  $\bar{X}_p$  é considerado a melhor aproximação do ótimo.

O algoritmo de Teitz e Bart se baseia no princípio de otimização local, caracterizando a idéia de  $\lambda$ -*otimalidade* utilizada em vários problemas de

otimização combinatorial que foi generalizado posteriormente por Lin e Kernighan (1973) para o problema do caixeiro viajante.

No algoritmo de Teitz e Bart modificado fez-se uso de um raio de busca que limitava a substituição de nós. Este raio impede que pontos distantes da mediana atual sejam testados, o que reduz significativamente o tempo de processamento da busca sem que haja perdas significativas na qualidade da solução. Este método é destinado principalmente para problemas de grande porte e foi baseado em Mayerle (2005).

### **3.3 Algoritmo Genético aplicado ao problema das p-medianas**

O primeiro pesquisador a utilizar a teoria Darwiniana (1859) para resolver problemas de otimização combinatorial foi John Holland, da Universidade de Michigan, ainda na década de 1970. Esta técnica de busca baseia-se no mecanismo da seleção natural, onde os indivíduos mais aptos de uma população são os que sobrevivem. Em cada nova geração de soluções, os indivíduos são recombinados e alterados por operadores genéticos, produzindo um método de busca rápido e eficiente.

Para Goldberg (1986), Algoritmos Genéticos são algoritmos computacionais de busca baseados no mecanismo de evolução natural e na genética. Em algoritmos Genéticos, uma população de possíveis soluções para o problema em questão evolui de acordo com operadores probabilísticos concebidos a partir de metáforas biológicas, de modo que há uma tendência de que, na média, os indivíduos representem soluções cada vez melhores à medida que o processo evolutivo continua.

Segundo Koza (1992), um Algoritmo Genético é um algoritmo matemático altamente paralelo que transforma um conjunto de objetos matemáticos individuais com respeito ao tempo, usando operações modeladas de acordo com o princípio Darwiniano de reprodução e sobrevivência do mais apto.



De modo mais específico, o Algoritmo Genético é um processo iterativo que mantém uma população de estruturas (chamadas de indivíduos, cromossomos ou strings), que representam possíveis soluções de um determinado problema. Durante o processo evolutivo, em cada geração, os indivíduos da população atual são avaliados de acordo com o valor de sua aptidão para a solução do problema. Para cada indivíduo é dada uma nota, ou índice, refletindo sua habilidade de adaptação (fitness) a determinado ambiente. Uma porcentagem dos mais adaptados é mantida, enquanto que outros indivíduos são descartados. Os indivíduos mantidos pela seleção podem sofrer modificações em suas características fundamentais através de mutações e cruzamento (crossover) ou recombinação genética, gerando descendentes para a próxima geração, caracterizando a reprodução (Mercado, 2001).

Os elementos básicos dos Algoritmos Genéticos são: população, indivíduos, fitness, seleção, crossover, mutação e reprodução.

A população de um problema é o conjunto de indivíduos que fazem parte da iniciação do algoritmo. Esses elementos são usados para gerar as próximas gerações de indivíduos. A população inicial normalmente é gerada por um processo aleatório.

Um indivíduo (ou cromossomo ou string) é a representação de uma solução viável. Essa representação pode ser elaborada de diversas formas. As primeiras aplicações utilizavam-se da representação binária, onde cada vértice faz (1), ou não (0), parte da solução. No caso do problema das p-mediana, em um problema com 30 vértices, um indivíduo seria uma cadeia de 30 genes, onde estas 30 posições (genes) são os vértices e o número contido nesta posição traz a informação de qual mediana atende o vértice. Dibble e Densham (1993) propõem ainda uma representação onde cada gene tem exatamente p espaços, e estes espaços correspondem a uma instalação utilizada como mediana, ou seja, nesta representação não aparecem os vértices que não são medianas.

O fitness é a função que avalia cada um dos indivíduos da população. Esta avaliação é essencial para decidir se o indivíduo permanece na próxima geração ou se ele é descartado, ou seja, quanto melhor o valor da função objetivo, maiores são as chances de o indivíduo sobreviver no ambiente, reproduzir-se e gerar descendentes.

Os operadores genéticos mais comuns são: a seleção, a reprodução, o cruzamento e a mutação. A reprodução consiste basicamente em copiar indivíduos da população atual para a futura geração.

A seleção é o processo pelo qual se escolhem os indivíduos da próxima geração. Existem várias técnicas para efetuar a seleção, a mais difundida é a regra da roleta, onde cada indivíduo recebe uma porcentagem de acordo com o seu fitness e os próximos indivíduos são escolhidos por sorteio.

Considerado o operador genético mais importante, o crossover, consiste em recombinar as características dos pais durante a reprodução, permitindo que as próximas gerações herdem essas características.

Usada em uma taxa sempre menor que o crossover, a mutação consiste na alteração em uma única posição, ou seja, apenas um gene será alterado aleatoriamente na próxima geração.

Nos próximos parágrafos o Algoritmo Genético e seus operadores utilizados no algoritmo proposto serão detalhados.

### **3.3.1 Representação da Estrutura Cromossômica**

Neste trabalho será utilizada a representação da solução através de vetores, onde cada vetor é um cromossomo com vários números inteiros, os números nos “alelos” são as medianas que servem o vértice, se o número no alelo e o número do vértice coincidirem, significa que o vértice é uma mediana.

Para exemplificar, em um problema com sete vértices onde se busca localizar três medianas, um possível solução é o cromossomo (2, 2, 5, 5, 5, 7, 7).

Observe que a posição (alelo) dois é ocupada pelo número dois, isso quer dizer que o vértice dois é mediana do problema, já a posição 6 é ocupada pelo número 7, isso quer dizer que o vértice 6 é servido pelo vértice 7.

### **3.3.2 População Inicial e Número de Indivíduos**

O algoritmo genético trabalha com o tamanho da população constante e se a população inicial for um número exagerado, o algoritmo se tornará lento, no entanto com um número pequeno de cromossomos o algoritmo terá pouca diversificação fazendo com que soluções possíveis não sejam exploradas.

Quanto à formação da população inicial, adotou-se neste trabalho a geração aleatória de cromossomos, assim os vértices são escolhidos para serem medianas aleatoriamente.

Em relação ao tamanho da população, optou-se por fixar a população em 2000 indivíduos. Segundo Mercado (2001), este número cria uma grande quantidade de opções para a busca mantendo a rapidez do algoritmo.

### **3.3.3 Avaliação de Cada Indivíduo ou Fitness**

A avaliação de cada integrante da população consiste em determinar a aptidão de sobrevivência do cromossomo, no caso dos algoritmos genéticos aplicados para resolução de problemas de otimização, isso é determinado pelo valor que a solução/cromossomo tem na função objetivo. No caso específico do problema das p-medianas onde se deseja obter o menor número de transmissão possível, os cromossomos mais aptos serão aqueles que terão os menores valores na função objetivo, tendo maiores chances de sobreviver.

No algoritmo, logo após a criação da população inicial, é avaliado o fitness de cada indivíduo e eles são colocados em uma lista ordenados por este valor.

### **3.3.4 Seleção dos Cromossomos**

Após serem classificados pelo fitness, são selecionados os cromossomos que farão parte da reprodução. Os escolhidos passarão pelos operadores genéticos para formar a próxima geração.

Para realizar a seleção na técnica testada foram efetuados dois sorteios de três cromossomos cada, onde os dois melhores classificados são cruzados formando assim o próximo sucessor. Este processo é feito até se formar a nova população.

### **3.3.5 Reprodução**

A reprodução é o processo pelo qual são gerados novos descendentes através dos operadores de cruzamento e de mutação.

### **3.3.6 Cruzamento**

O operador de cruzamento ou crossover consiste em recombinar as características dos pais durante a reprodução, gerando assim novos indivíduos para as futuras gerações. Este operador é considerado o operador genético predominante e é usado com uma taxa sempre maior que o operador de mutação.

O Algoritmo Genético elaborado neste trabalho mantém as características que coincidem nos dois pais, ou seja, os genes que se repetem tanto no primeiro cromossomo quanto no segundo são mantidos no cromossomo filho que fará parte da futura população. Os genes que são diferentes nos dois cromossomos são preenchidos com o valor zero e os genes dos pais, que são medianas em somente um cromossomo, são armazenados em uma lista para serem utilizados para completar as medianas faltantes.

### 3.3.7 Mutação

O operador genético de mutação modifica, em uma taxa muito baixa, aleatoriamente um único gene. Isso permite que novos indivíduos sejam introduzidos na população.

### 3.3.8 Estrutura do Algoritmo Genético

O algoritmo pode ser representado da seguinte forma:

#### **Início** Algoritmo

Representar uma solução na estrutura de um cromossomo;

Construir uma população inicial;

Avaliar o Fitness dos indivíduos;

#### **Repetir**

Selecionar ancestrais da população;

Efetuar o crossover entre os selecionados;

Efetuar a mutação;

Avaliar o Fitness dos descendentes gerados;

**Até** que uma solução satisfatória seja encontrada;

#### **Fim** Algoritmo.

A estrutura acima foi adaptada de Mercado (2001).

## 3.4 Algoritmo de Busca Tabu para o problema das p-medianas

Fred Glover, durante a década de 1980, desenvolveu um método para solução de problemas de otimização combinatorial denominado Busca Tabu. Este método é conhecido por, na maior parte das vezes, superar o problema da convergência local.

Como se sabe, a palavra “tabu” significa algo proibido, ou que deve ser inibido. Em se tratando de problemas de otimização combinatorial, os “tabus”, são movimentos proibidos que não serão realizados a menos que um critério de aspiração permita o movimento. Assim, a Busca Tabu, direciona a procura em espaços que ainda não foram pesquisados, tendendo a evitar um mínimo local. As restrições tabu são controladas por uma lista que memoriza os últimos movimentos executados e o tempo que o movimento permanece proibido é contado em iterações.

Segue abaixo a apresentação dos principais elementos da Busca Tabu.

Qualquer solução, ótima ou não, é denominada configuração e o movimento é o processo pelo qual se gera uma nova solução, onde um novo vértice ocupa o lugar de um nó que é retirado da solução.

O conjunto de movimentos candidatos é o conjunto de todas as possibilidades de movimentos excluindo-se a solução atual.

Os movimentos proibidos fazem parte de um conjunto denominado restrições tabu. É este conjunto que difere a Busca Tabu de outras técnicas. Estes movimentos não permitidos, no caso do problema das p-mediana, podem ser um conjunto de vértices que foram recentemente visitados ou movimentos que não satisfizeram certos critérios.

Para permitir exceções, o critério de aspiração permite que um movimento que esteja na lista de restrições tabu seja utilizado. Um critério de aspiração bastante comum é o de ignorar a restrição sempre que o valor da configuração (solução) resultante do movimento for melhor que a melhor solução encontrada até o momento.

O processo de Busca Tabu aplicado ao problema das p-mediana pode, de forma bastante simples, ser descrito como:

A partir de uma configuração inicial, avalia-se a função objetivo para esta configuração. Gera-se um subconjunto de movimentos candidatos da solução atual. Se o melhor destes movimentos não é um movimento pertencente à lista tabu, ou se o movimento satisfaz o critério de aspiração, este vértice responsável por este movimento passa a fazer parte da nova configuração corrente.

O algoritmo aqui testado é uma adaptação do algoritmo apresentado por Schilling et al.(1996). Nos próximos parágrafos o Algoritmo de Busca Tabu utilizado para resolver o problema das  $p$ -medianas será especificado.

### **3.4.1 Solução Inicial**

Tal como o Algoritmo Genético apresentado anteriormente, o Algoritmo de Busca Tabu também se inicia com uma solução aleatória inicial contendo exatamente  $p$  vértices.

### **3.4.2 Movimentos**

Como estratégia para fugir de um ponto de ótimo local, a Busca Tabu permite durante a busca visitar soluções ineficazes. No caso do problema das  $p$ -medianas o algoritmo pesquisa em soluções contendo  $p - 1$  ou  $p + 1$  medianas.

Para isso a Busca Tabu conta com dois movimentos básicos, ADD e DROP. O movimento construtivo (ADD) consiste em adicionar uma mediana na solução vigente, já o movimento destrutivo (DROP) elimina uma mediana da solução. Outro movimento frequentemente utilizado chama-se SWAP; nele são feitas as trocas sem diminuir ou aumentar o número de medianas.

### **3.4.3 Restrições Tabu**

As restrições tabu são aplicadas para prevenir que movimentos voltem para soluções recentemente visitadas. Assim, quando um determinado nó passa a

fazer parte da solução, ele entra também na lista tabu permanecendo por  $p$  iterações, onde o nó fica proibido de entrar na solução.

#### **3.4.4 Critério de Aspiração**

O critério de aspiração existe para burlar a lista tabu, desta forma, elaborou-se uma memória onde é gravada a melhor solução até o momento. Se um nó que pertence à lista tabu é testado e produz o resultado melhor que o contido na memória, este movimento é permitido e o nó entra na solução.

#### **3.4.5 Estratégia de Oscilação**

O algoritmo permite aumentar ou diminuir o número de medianas, isso resulta em soluções não factíveis, mas permite que a busca “escape” de mínimos locais. Frequentemente, aumentando uma mediana ao problema, o valor da função objetivo diminui para além do mínimo encontrado até o momento, fazendo com que novas configurações sejam testadas.

#### **3.4.6 Estrutura do algoritmo de Busca Tabu**

O algoritmo conta com uma inicialização onde os parâmetros são definidos. Nesta inicialização, o número máximo de iterações é fixado como o dobro do número de nós do problema. Outro parâmetro importante é o número de estabilidade, aqui fixado em 20%, ou seja, se em 20% do número máximo de iterações não houver melhora, o algoritmo pára.

O algoritmo de Busca Tabu possui quatro procedimentos: o procedimento ESCOLHA\_MOVIMENTO, o procedimento ADD, o procedimento DROP e o procedimento SWAP. Os procedimentos ADD e DROP são responsáveis, respectivamente, pelos movimentos de acrescentar e retirar um nó da solução. O procedimento SWAP realiza a substituição de um nó, de modo análogo ao proposto por Teitz & Bart. No procedimento ESCOLHA\_MOVIMENTO é definido o movimento que será executado.



A estrutura do algoritmo pode ser descrito como:

**Início** Algoritmo

Gerar uma solução inicial factível;  
Calcular o valor da função objetivo da solução;  
Gerar a lista tabu;

**Repetir**

Executar o procedimento ESCOLHA\_MOVIMENTO;  
Realizar o movimento escolhido;  
Atualizar a FO e a lista tabu;  
Realizar o movimento SWAP;  
Atualizar a FO e a lista tabu;

**Até** que o n<sup>o</sup> máximo de iterações seja alcançado.

**Fim** Algoritmo

### 3.5 Comparação e análise dos resultados

As quatro heurísticas foram desenvolvidas em Delphi 7.0 e todos os problemas foram realizados em um computador Intel Celeron de 2.66GHz, com 2 GB de memória em um sistema operacional Windows XP. A interface do programa desenvolvido é apresentada na figura 3.1.

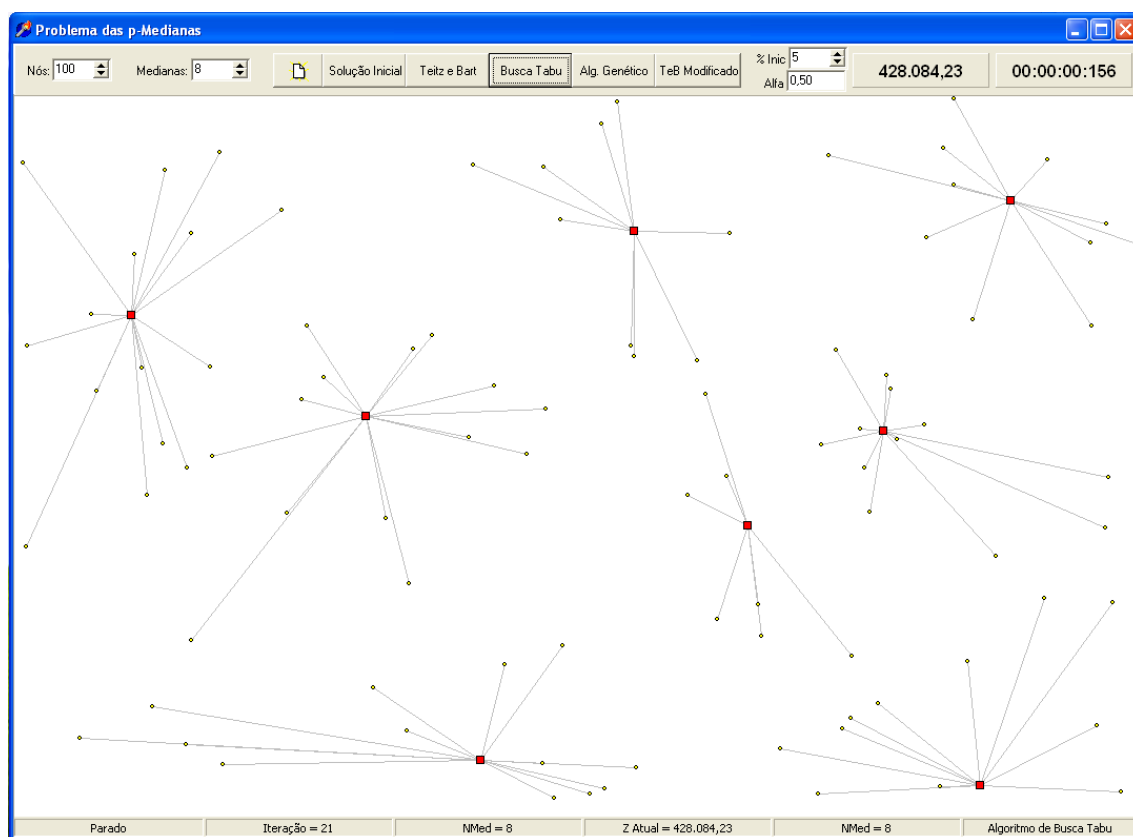


Figura 3.1: Interface do programa desenvolvido.

O sistema é composto por duas entradas principais, o número de pontos (nós) e o número de medianas. No painel superior encontram-se seis botões, o primeiro gera o problema com as entradas desejadas, o segundo cria uma solução aleatória para o problema e os quatro seguintes disparam as heurísticas testadas, na seguinte ordem: Teitz e Bart (T&B), Busca Tabu (BT), Algoritmo Genético (AG) e Teitz e Bart Modificado (T&B-M). No canto superior direito e em negrito pode-se ver o valor da função objetivo e o tempo decorrido.

Para avaliar a eficiência e eficácia das técnicas propostas foram gerados 115 problemas aleatórios. Estes problemas foram resolvidos pelos algoritmos com pesos aleatórios entre um e cem, ou seja, cada ponto pode ser visitado no mínimo uma vez e no máximo 100 vezes.

Neste trabalho, convencionou-se o termo redes pequenas para problemas com até 100 nós. Já problemas de 200 a 700 nós são considerados de médio porte e problemas com mais de 800 pontos são considerados grandes.

Quando um problema é gerado com 200 nós e 5 medianas também é definido aleatoriamente o peso de cada nó, isso na prática pode ser exemplificado como uma empresa que deseja instalar 5 depósitos (medianas) nas suas 200 cidades de atendimento (nós ou pontos). Nesse sentido, o peso seria a quantidade de visitas em cada cidade, sempre buscando a menor soma das distâncias totais percorridas.

Esta pesquisa teve como objetivo classificar o comportamento dos algoritmos aplicados ao problema das  $p$ -medianas quanto a solução e quanto ao tempo necessário, e para isso foi utilizada a metodologia de *performance profile*, proposta por Dolan e Moré (2002).

Basicamente, nesta metodologia, compara-se o desempenho de  $n_s$  métodos em um conjunto de  $n_p$  problemas utilizando alguma medida de desempenho  $m_{s,p}$ . Esta medida de desempenho ( $m_{s,p}$ ) pode ser o número de iterações, ou número de cálculo de funções, ou tempo de processamento em que o método  $s$  usou para resolver o problema  $p$  (Casali, 2007). No trabalho proposto, utilizou-se como medida de desempenho a qualidade da solução obtida pelos algoritmos e o tempo de processamento.

Ainda segundo Casali (2007), definiu-se inicialmente para cada problema o valor  $\overline{m}_p = \min_s \{m_{s,p}\}$ , igual à medida de desempenho alcançada pelo método que se revelou mais eficiente para o problema  $p$ . Para cada método  $s$  e para cada problema  $p$ , definiu-se a taxa de desempenho do método  $s$  no problema  $p$ : se o método  $s$  não resolve o problema  $p$ ,  $r_{s,p} = \infty$ , caso contrário:

$$r_{s,p} = \frac{m_{s,p}}{\overline{m}_p}$$

Assim, para avaliar o desempenho do método  $s$  no conjunto de  $n_p$  problemas constrói-se uma curva de distribuição  $\rho_s: [1, +\infty) \rightarrow [0,1]$ , com:

$$t \in [1, +\infty) \mapsto \rho_s(t) = \frac{1}{n_p} \text{card}\{p | r_{s,p} < t\}$$

De acordo com a metodologia *performance profile*, um algoritmo  $s_1$  é melhor que  $s_2$ , quando a curva  $s_1$  estiver sempre acima da curva do algoritmo  $s_2$ . O início da curva mostra a porcentagem de problemas em que o método obteve o melhor resultado encontrado pelos quatro algoritmos e a parte final revela a confiabilidade dos métodos. Ou seja,  $\rho_s(1)$  é a porcentagem dos problemas em que o método  $s$  foi o campeão e  $\rho_s(t)$  é a porcentagem dos problemas em que  $s$  fez menos de  $t$  vezes o número de iterações do campeão. Por exemplo,  $\rho_s(10) = 78\%$  significa que em 78% dos problemas o número de iterações de  $s$  foi inferior a 10 vezes o número do método campeão.

O gráfico abaixo mostra o desempenho, em relação à qualidade da solução, dos quatro algoritmos em redes pequenas.

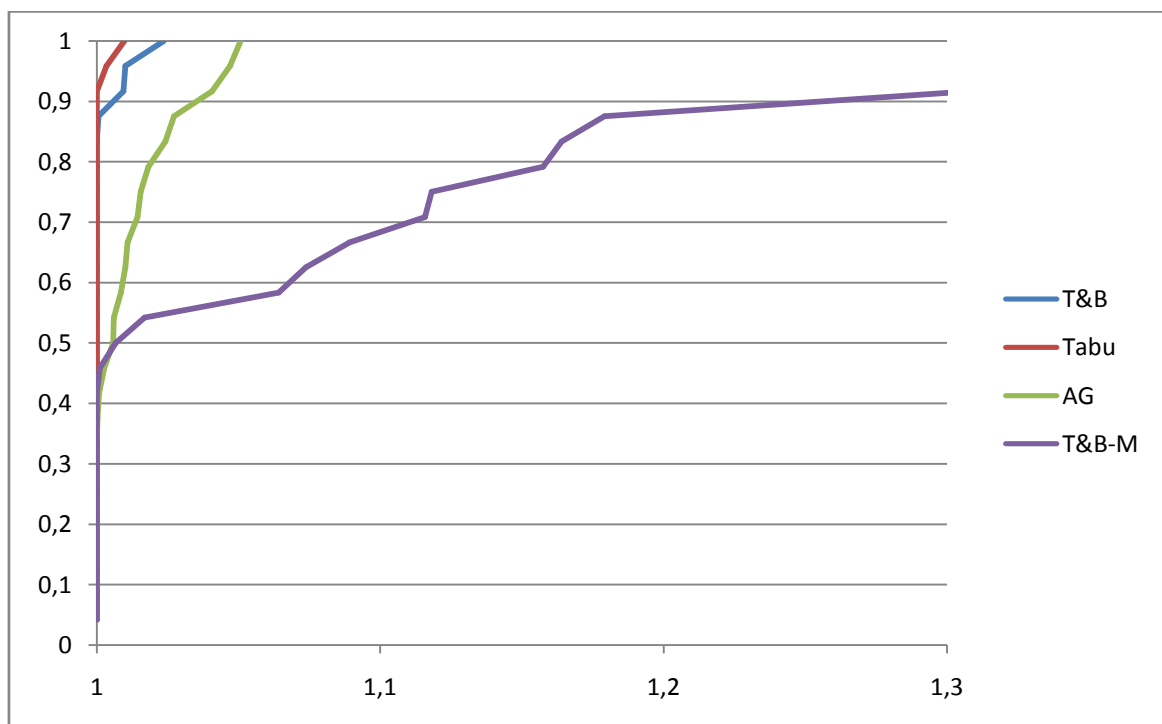


Figura 3.2 – Perfil de desempenho dos algoritmos em relação à qualidade as solução em redes pequenas.

Na figura 3.2, percebe-se que, nos 24 problemas gerados, o algoritmo de BT obteve o melhor resultado em 91,66% dos problemas contra 83,33% do algoritmo de T&B, 41,67% de T&B-M, e 33,33% do AG. O algoritmo BT, quando não encontrou o melhor resultado, obteve um erro máximo de 0,96% em relação ao melhor encontrado, seguido por T&B com 2,35%, AG com 5,06% e T&B-M com 53,70%.

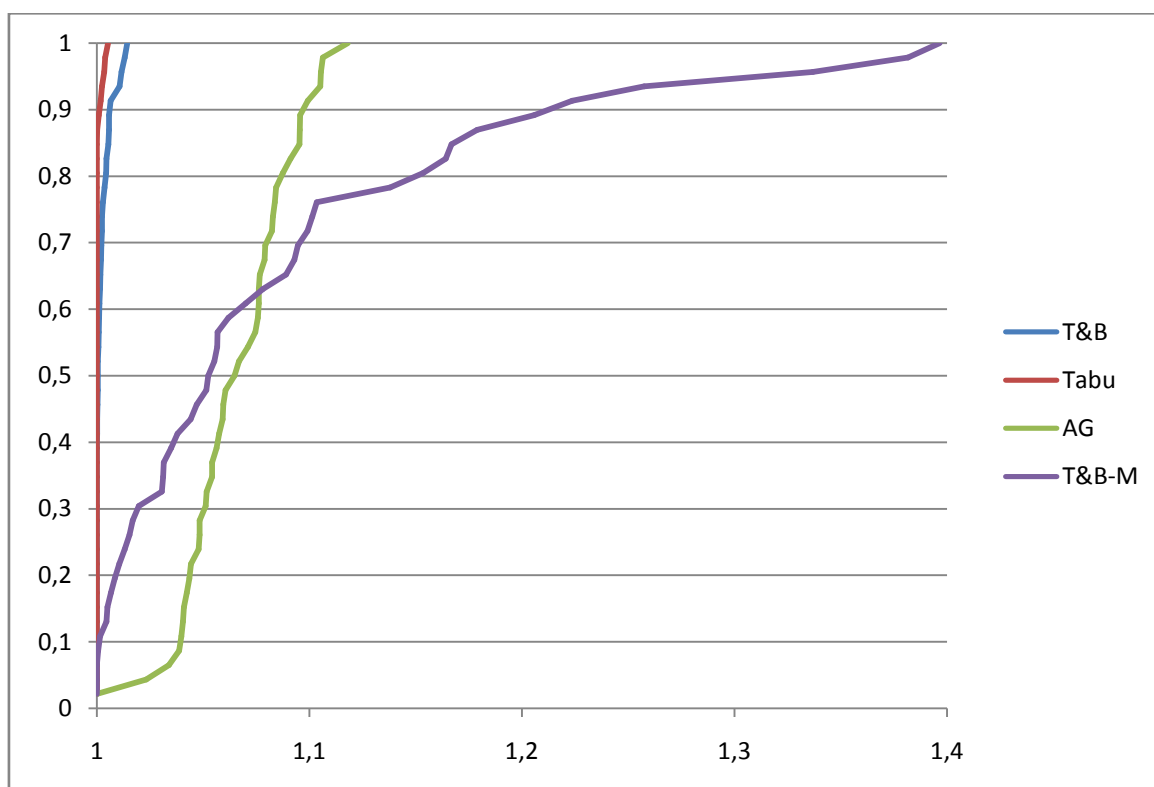


Figura 3.3 – Perfil de desempenho dos algoritmos em relação à qualidade da solução em redes médias.

A figura 3.3 mostra o desempenho dos algoritmos, quanto à qualidade da solução, em 46 problemas de porte médio. Percebe-se que o algoritmo de BT prevalece sobre os demais, sendo o campeão em 84,78% dos problemas com erros de no máximo 0,52%. O algoritmo de T&B obteve o melhor resultado em 39,13% com erros menores que 1,42%.

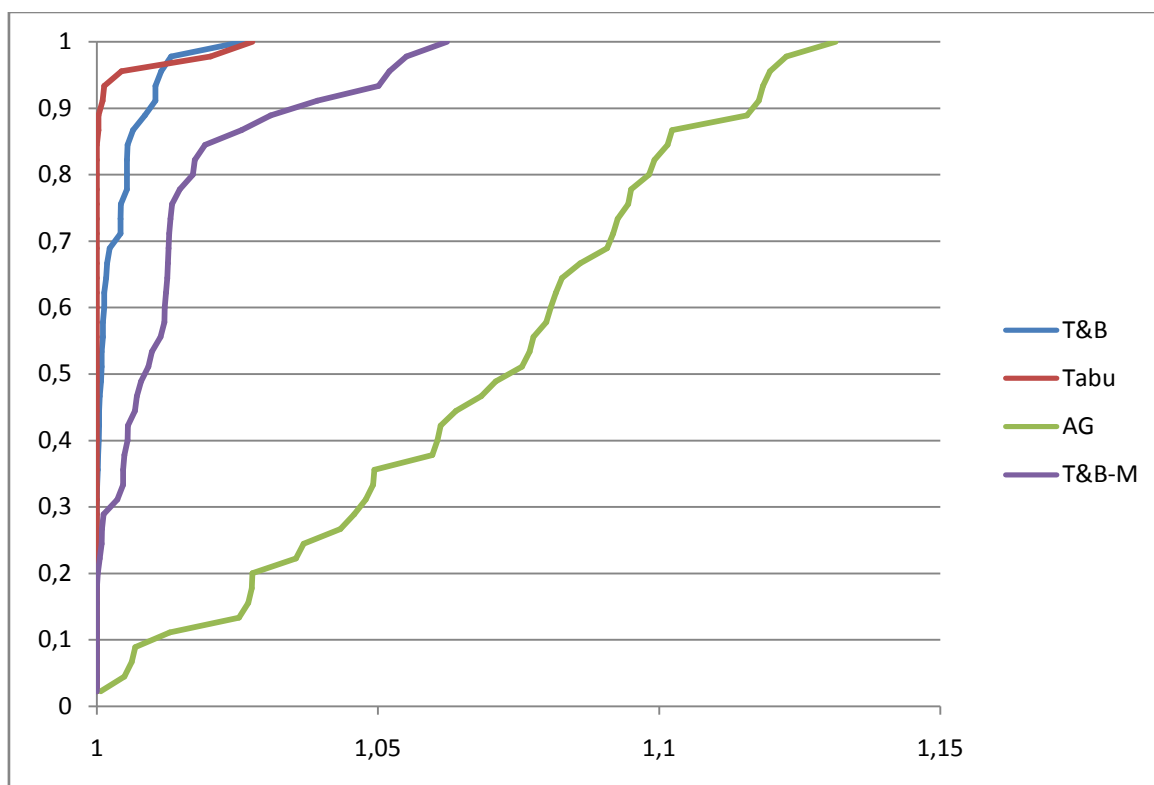


Figura 3.4 – Perfil de desempenho dos algoritmos em relação à qualidade da solução em redes grandes.

O perfil de desempenho apresentado na figura 3.4 demonstra que o algoritmo de BT continua encontrando o melhor resultado na maioria dos problemas, 84,44%. O algoritmo de T&B obteve o melhor resultado em 31,11%. No que diz respeito aos erros máximos, o algoritmo de T&B e BT se mostraram os mais confiáveis, 2,59% e 2,76% respectivamente.

Vale ressaltar, o bom desempenho do algoritmo T&B-M em redes grandes. Apesar de o erro máximo estar na casa dos 6%, o algoritmo apresenta uma boa solução em tempo computacional menor que os concorrentes. O algoritmo obteve  $\rho_s(1,0170) = 80\%$ , ou seja, resolveu 80% dos problemas com um erro inferior a 1,70%.

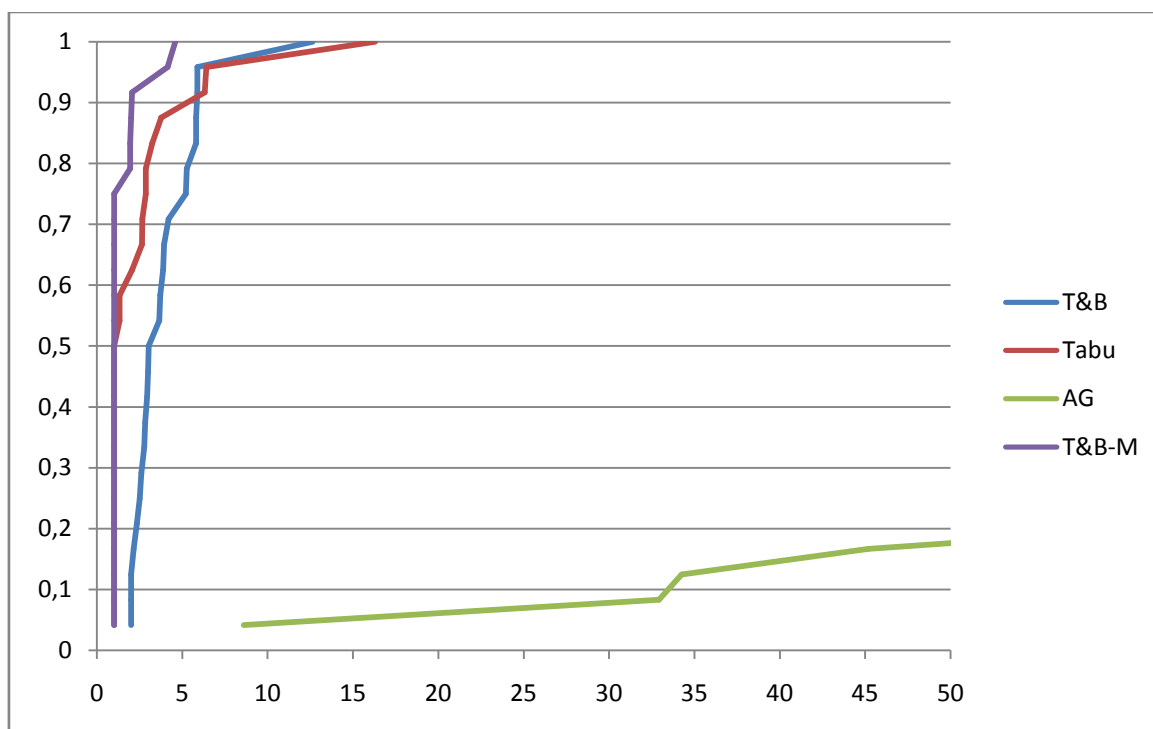


Figura 3.5 – Perfil de desempenho dos algoritmos em relação ao tempo em redes pequenas.

A figura 3.5 demonstra o desempenho dos quatro algoritmos testados em relação ao tempo em redes pequenas. O algoritmo T&B-M obteve o menor tempo em 75% dos problemas, seguido por 50% do algoritmo BT. O algoritmo BT obteve  $\rho_s(6,41) = 95,83\%$ , ou seja, em 95,83% dos problemas precisou-se de no máximo 6,41 vezes o tempo do algoritmo campeão. Já o algoritmo T&B foi ainda melhor, com  $\rho_s(5,87) = 95,83\%$ . O algoritmo de T&B-M obteve no pior caso:  $\rho_s(4,58) = 100\%$ .

O algoritmo genético apresentou os piores resultados quanto ao tempo de processamento nesse conjunto de problemas, principalmente, porque a versão aqui apresentada utilizou um número fixo de cromossomos (2000 cromossomos). Esse número pode ser alterado de acordo com o tamanho do problema a ser resolvido, no entanto, como os outros três algoritmos não foram adaptados de acordo com o tamanho da rede, optou-se por usar o número de cromossomos do AG fixo.

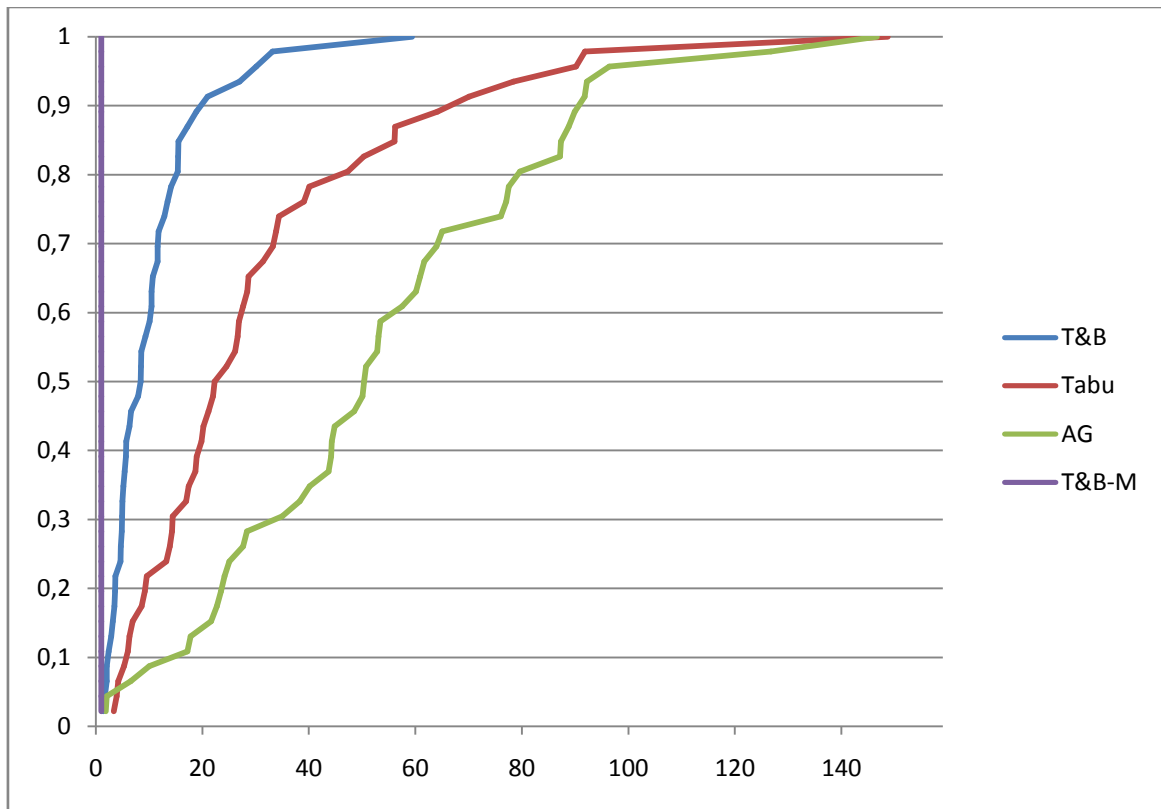


Figura 3.6 – Perfil de desempenho dos algoritmos em relação ao tempo em redes médias.

Em redes médias, avaliando-se o tempo (figura 3.6), o algoritmo de T&B-M foi o melhor em todos os 46 problemas testados, seguido pelo algoritmo T&B, por BT e pelo AG. Nos piores casos, os algoritmos AG e BT chegam a precisar de mais de 100 vezes o tempo do algoritmo campeão.



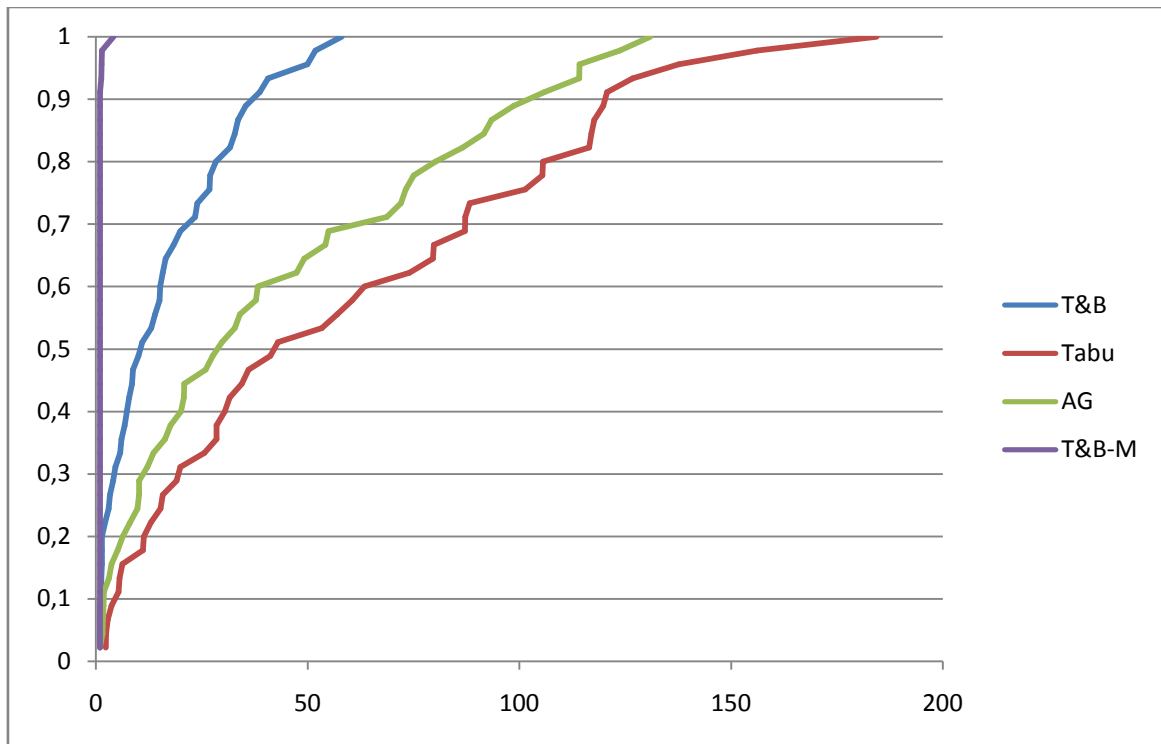


Figura 3.7 – Perfil de desempenho dos algoritmos em relação ao tempo em redes grandes.

Na figura 3.7, permanece sendo retratada a superioridade do método de T&B-M em relação ao tempo. As técnicas AG e BT permanecem sendo a pior alternativa no quesito tempo de processamento.

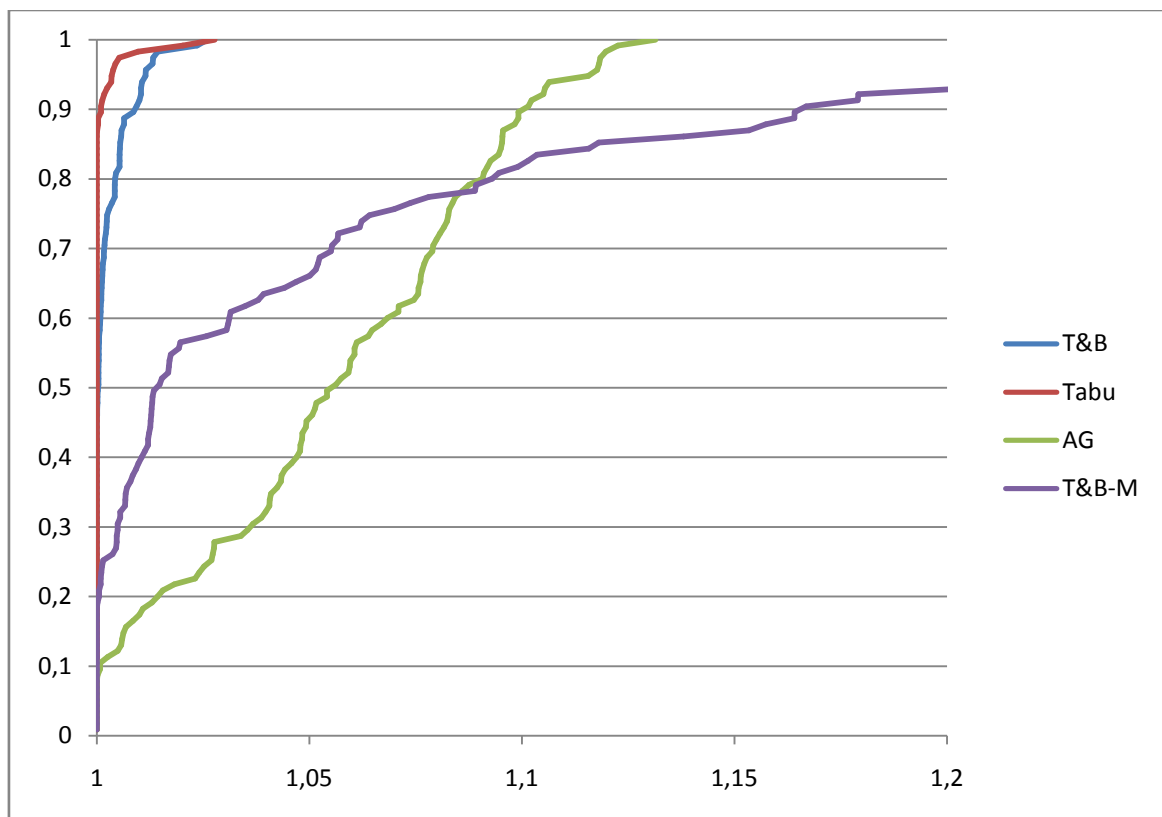


Figura 3.8 – Perfil de desempenho dos algoritmos em relação à qualidade da solução em todos os 115 problemas.

A figura 3.8 apresenta o desempenho dos algoritmos, em relação à qualidade da solução, em todos os 115 problemas testes. A técnica de BT foi campeã em 86,08% das vezes apresentando um erro máximo de 2,76% em relação ao melhor encontrado. A heurística de T&B obteve o melhor resultado em 45,21% dos problemas, com erros nunca maiores que 2,59%. A heurística T&B-M encontrou o melhor resultado em 17,39% dos testes, com erro máximo de 53,70%. Vale ressaltar que este erro se deve ao desempenho de T&B-M em redes pequenas, para as quais o algoritmo não foi projetado. O AG encontrou o melhor resultado em apenas 6,95% dos problemas, com erro máximo de 13,14%.

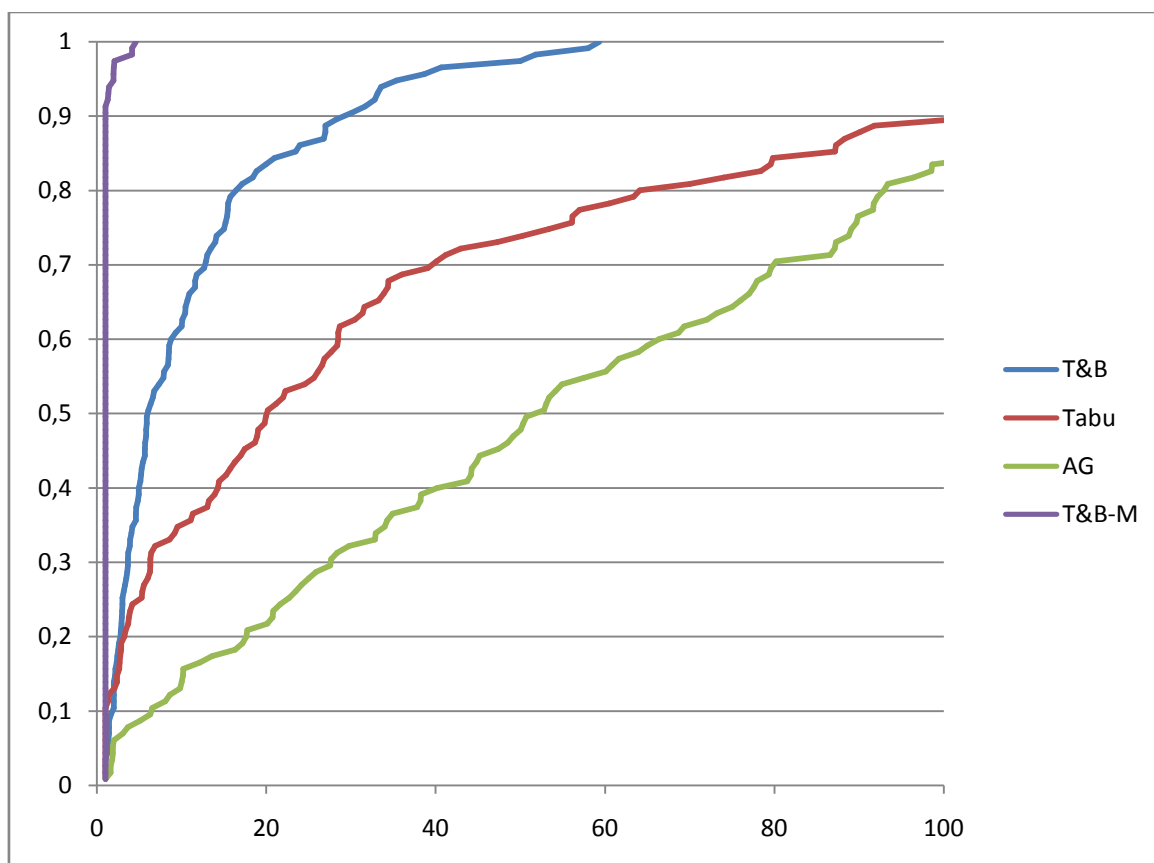


Figura 3.9 – Perfil de desempenho dos algoritmos em relação ao tempo em todos os 115 problemas.

A figura 3.9 apresenta o desempenho dos quatro algoritmos em todos os 115 problemas em relação ao tempo de processamento. Percebe-se a superioridade do algoritmo T&B-M, que resolveu 91,30% dos problemas com o melhor tempo e obteve um pior caso de 4,58. O algoritmo T&B obteve desempenho inferior, seguido pelos algoritmos de BT e AG.

## **CAPÍTULO IV**

### **4 CONCLUSÕES E RECOMENDAÇÕES**

Neste capítulo são apresentadas as conclusões finais do trabalho e as recomendações para trabalhos futuros.

#### **4.1 Conclusões**

Neste trabalho foi apresentado o problema de localização de p-medianas não capacitado. O estudo foi motivado pela crescente busca por métodos mais rápidos e mais precisos na solução e pela falta de um trabalho na literatura que indique o método mais adequado para o tamanho específico do problema. Assim, foram implementados e testados os algoritmos de Teitz e Bart, Busca Tabu, Algoritmo Genético e Algoritmo de Teitz e Bart Modificado em 115 problemas, pequenos, médios e grandes, gerados aleatoriamente.

Em problemas pequenos, de até 100 nós, as quatro técnicas tiveram um bom desempenho. Entretanto, os algoritmos BT e AG apresentaram soluções melhores que os demais, chegando a obter um ganho de 2,3%. O Algoritmo Genético proposto neste trabalho, teve seu melhor desempenho em redes pequenas, em problemas médios e grandes, a metaheurística não obteve bons resultados. Quanto ao tempo, todos os algoritmos resolveram os problemas em

menos de um segundo, com exceção do Algoritmo Genético, que em seu pior caso, gastou 14 segundos para resolver um dos problemas testados.

Para redes de médio porte, o algoritmo de Busca Tabu apresentou os melhores resultados, chegando a um ganho de até 1,4% sobre as demais heurísticas. No entanto, o tempo necessário para a resolução do problema é maior. Como as metaheurísticas (AG e BT) fazem uso de estratégias para fugir de mínimos locais, isso acaba requerendo um tempo computacional maior, deixando o algoritmo lento. O pior caso, um problema de 700 nós e 70 medianas, o algoritmo de Busca Tabu necessitou de quase 9 minutos para encontrar a solução. É interessante ressaltar que dependendo das especificidades da aplicação é que o usuário deve decidir entre utilizar o algoritmo preciso ou o mais rápido.

O algoritmo de Teitz e Bart Modificado teve seu melhor desempenho em problemas de grande porte. Em relação ao tempo o algoritmo é melhor que os demais, este fato se deve ao raio utilizado pelo algoritmo em cada mediana para evitar que sejam testados pontos distantes da mediana atual. Quanto à qualidade da solução, em redes grandes, o método produz resultados satisfatórios gerando perdas de no máximo 6,22% em relação ao melhor valor encontrado. O algoritmo T&B-M obteve  $\rho_s(1,0170) = 80\%$ , resolvendo 80% dos problemas grandes com erros inferiores a 1,70%.

De forma geral, tendo em vista apenas a qualidade da solução encontrada e num total de 115 problemas teste, o algoritmo de Busca Tabu obteve a melhor solução em 86,08% dos problemas. O método de Teitz e Bart conseguiu o melhor resultado em 45,21%, dos problemas, seguido por 17,39% de Teitz e Bart Modificado e por 6,95% do algoritmo genético.

## 4.2 Recomendações

Em trabalhos futuros, recomenda-se o uso de novas metaheurísticas adaptadas para resolver o problema. Técnicas como Simulated Annealing, Scatter Search, Ant Colony e Redes Neurais precisam ser testadas para talvez encontrar meios mais rápidos e acurados na resolução dos problemas.

Outra alternativa para o problema de localização de p-medianas são os algoritmos híbridos. Técnicas mistas costumam gerar boas soluções iniciais, e isso pode trazer agilidade na busca.

Também seria interessante encontrar meios para tornar o algoritmo de Busca Tabu mais rápido. O algoritmo apresentou a melhor solução em 86,08% dos problemas testados, mas o tempo gasto, geralmente, é maior que os demais.

## BIBLIOGRAFIA

AZZONI, C. R. Teoria da Localização: uma análise crítica. São Paulo, IPE – USP, 1982.

BALLOU, R. H. Gerenciamento da Cadeia de Suprimentos. Porto Alegre: Bookman, 2001.

BANDEIRA, R. A. de M. Proposta de uma Sistemática de Análise para a localização de Depósitos. Dissertação de Mestrado. UFRGS, Porto Alegre, 2006.

BEASLEY, J. E. A note on solving large p-medians problems. European Journal of Operational Research, 21, 1985, p. 270 – 273.

BEASLEY, J. E. Lagrangian heuristics for location problems. European Journal of Operational Research, 65, p. 383 – 399, 1993.

BEZERRA, O. B. Localização de Postos de Coleta para o Apoio ao escoamento de Produtos Extrativistas – Um Estudo de Caso Aplicado ao Babaçu. Dissertação de Mestrado. UFSC, Florianópolis, 1995.

BILDE, O.; KRARUP, J. Sharp Lower Bounds and Efficient algorithms for Simple Plant Location Problem. Annals of Discrete Mathematics, 1, p. 79 – 97, 1977.

BINFARÉ, N. J.. Método para localização de uma sede móvel na realização de inventário florestal. Dissertação de Mestrado. UFSC, 1993.

CASALI, R. M.. Um novo tratamento para restrições de equilíbrio em problemas de programação matemática. Tese de Doutorado. UFSC, Florianópolis, 2007.

CHIYOSHI, F.; GALVÃO, R. D. A statistical analysis of simulated annealing applied to the p-median problem. Annals of Operations Research, 96, p. 61 – 74, 2000.

CHRISTOFIDES, N. Graph Theory: an algorithmic approach. London: Academic Press, 1975.

CHRISTOFIDES, N.; BEASLEY, J. E. A tree search algorithm for the p-median problem. European Journal of Operational Research, 10, p. 196 – 204, 1982.

CORRÊA, E. S. Algoritmos Genéticos e Busca Tabu aplicados ao problema das p-medianas. Dissertação de Mestrado. UFPR, Curitiba, 2000.

COOPER, L. L. Location-Allocation Problems. Operations Research, 11 (1963) p. 331 – 343.

CORNUEJOLS, G.; FISCHER, M. L.; NEMHAUSER, G. L. Location of Bank Accounts to Optimize Float: An analytic Study of Exact and Approximate Algorithms. Management Science, 23, (1977), p. 789 – 810.

DASKIN, M. S. Logistics: An overview of the state of the Art and Perspectives on Future Research. Transp. Research 19, (1985), p. 383 – 398.

DENSHAM, P. J.; RUSHTON, G. A more efficient heuristic for solving large p-medians problems. Paper in Regional Science, 71, p. 307 – 329, 1992.

DIBBLE, C.; DENSHAM, P. J. Generation interesting alternatives in GIS and SDSS using genetic algorithms. GIS/LIS symposium, University of Nebraska, Lincoln, 1993.

DOLAN, E. D.; MORÉ, J. J. Benchmarking optimization software with performance profiles. Mathematical Programming, v. 91, n. 2, p. 201 – 213, jan. 2002.

DUCATI, E. A. Busca Tabu aplicada ao problema de localização de facilidades com restrições de capacidade. Dissertação de Mestrado. Unicamp, Campinas, 2003.

EILON, S.; GALVÃO R. D. Single and Double Vertex Substitution in Heuristic Procedures for the p-median Problem. Management Science, 24, p.1763 – 1766, 1978.



EILON, S.; WATSON-GANDY, C. D. T.; CHRISTOFIDES, N. Distribution Management: Mathematical Modeling and Practical Analyses. London: Griffin Publishing, 1971.

EL-SHAIEB, A. M. A new algorithm for locating sources among destinations. Management Science, 20, p. 221 – 231, 1973.

ERKUT, E.; ALP, O.; DREZNER, Z. An efficient genetic algorithm for the p-median problem. Annals of Operations Research, 122, p. 21 – 42, 2003.

ERLENKOTTER, D. A dual-based procedure for uncapacitated facility location. Operations Research, 26, p. 992 – 1009, 1978.

GALVÃO, R. D. A dual-bounded algorithm for the p-median problem. Operations Research, 28, p. 1112 – 1121, 1980.

GALVÃO, R. D. Modelos e Algoritmos para Problemas de Localização em Redes. Pesquisa Operacional, 1, p.3 – 22, 1981.

GALVAO, R. D.; RAGGI, L. A. A method for solving to optimality uncapacitated location problems. Annals of Operations Research, 18, p. 225 – 244, 1989.

GAREY, M. R.; JOHNSON, D. S. Computers and Intractability: a guide to the theory of NP – Completeness. San Francisco: W. H. Freeman and Co., 1979, p. 340.

GARFINKEL, R. S.; NEEBE, A. W.; RAO, M. R. An algorithm for the m-median plant location problem. Transportation Science, 8, p. 217 – 236, 1974.

GLOVER, F. Future paths for integer programming and linkage to artificial intelligence. Computers & Operations Research 13, 533–549, 1986.

GLOVER, F. Tabu Search, Part I, ORSA Journal on Computing, 1, nº 3, p. 190 – 206, 1989.

GLOVER, F. Tabu Search, Part II, ORSA Journal on Computing, 2, nº 1, p. 4 – 32, 1990.

GLOVER, F.; LAGUNA, M. Tabu Search. Kluwer Academic Publishers, Hingham, 1997.

GOLDBARG, M. C.; LUNA, H. P. L. Otimização Combinatória e Programação Linear – Modelos e Algoritmos. Rio de Janeiro, Editora Campus, 2000.

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

HAKIMI, S. L. Optimal Locations of Switching Centers and the Absolute Centers and Medians of a Graph. Operations Research, 12, (1964), p. 450 – 459.

HAKIMI, S. L. Optimal Distribution of Switching Centers in a Communications Network and Some Related Graph Theoretic Problems. Operations Research 13, (1965), p. 462 – 475.

HAKIMI, S. L. Recent developments in Location Theory. In: International Symposium on Location Decisions, Banff, Alberta, Canada, 1978.

HANSEN, P.; MLADENOVIC, N. Variable Neighborhood search for the p-median. Location Science, 5, p. 207 – 226, 1997.

HANSEN, P.; MLADENOVIC, N.; PEREZ-BRITO, D. Variable neighborhood decomposition search. Journal of Heuristics, 7, p. 335 – 350, 2001.

HOLLAND, J. H. Adaptation in Natural and Artificial Systems. University of Michigan Press, 1975.

HOSAGE, C. M.; GOODCHILD, M. F. Discrete space location-allocation solutions from genetic algorithms. Annals of Operational Research, 6, p. 35 – 46, 1986.

HRIBAR, M.; DASKIN, M. A dynamic programming heuristic for the p-median problem. European Journal of Operational Research, 101, p. 499 – 508, 1997.

JÄRVINEN, P.; RAJALA J.; SINERVO, H. A branch-and-bound algorithm for seeking the p-median. Operations Research, 20, p. 173 – 178, 1972.

KHUMAWALA, M. B. An efficient branch-and-bound algorithm for the warehouse location problem. Management Science, 18, p. 712 – 731, 1972.

KUEHN, A. A.; HAMBURGER, M. A Heuristic Program for Locating Warehouses. Management Science, 9, p. 643 – 666, 1963.

LARSON, R. C.; ODONI, A. R. Urban Operations Research. New Jersey: Prentice-Hall, 1981.

LIMA, F. S. Distribuição Espacial de Serviços especializados de saúde. Dissertação de Mestrado. Florianópolis, UFSC, 1996.

LIN, S.; KERNIGHAN, B. W. An effective heuristic Algorithm for the Traveling Salesman Problem, Operations Research, v. 21, p. 498 – 516, 1973.

LOBO, D. da S. Localização de Unidades de Educação Infantil: Uma aplicação para creches Municipais de Florianópolis. Dissertação de Mestrado. UFSC, Florianópolis, 1998.

LOBO, D. da S. Dimensionamento e otimização locacional de unidades de educação infantil. Tese de Doutorado. UFSC, Florianópolis, 2003.

LOVE, R. F.; MORRIS, J. G. Facilities Location: models and methods. Elsevier Science Publishing Co., New York, 1988.

LORENA, L. A. N.; FURTADO, J. C. Constructive genetic algorithm for clustering problems. Evolutionary Computation, 9, p. 309 – 328, 2001.

MARANZANA, F. E. On the Location of supply points to minimize transport costs. Operations Research Quarterly, 15, p. 261 – 270, 1964.

MARSTEN, R. E. An algorithm for finding almost all of the medians of a Network. Discussion Paper, 23, Northwestern University, 1972.

MAYERLE, S. F. Um algoritmo genético para solução do problema do caixeiro viajante. Artigo de circulação interna do departamento de Engenharia de Produção e Sistemas da UFSC, 1996.

MAYERLE, S. F. Notas de aula, UFSC, Programa de Pós-Graduação em Engenharia de Produção, Florianópolis, SC, 2005.

MERCADO, N. B. G. Técnica de Busca Baseada em Algoritmo Genético para Localização de p-Medianas. Dissertação de Mestrado. UFSC, Florianópolis, 2001.

MERINO, E. D.; PÉREZ, J. M.; ARAGONÉS, J. J. Neural networks algorithms for the p-median problem. In ESSAN 2003: Proceedings of 11<sup>o</sup> European Symposium on Artificial Neural Networks, Belgium, April 2003, p. 385 – 391, 2003.

MINIEKA, E. The Centers and Medians of a graph. *Operations Research*, 25, p. 641 – 650, 1977.

MLADENOVIC, N.; MORENO-PEREZ, J. A.; MORENO-VEGA, J. M. A chain-interchange heuristic method. *Yugoslav Journal of Operations Research*, 6, p. 41 – 54, 1996.

MOREIRA, F. M. R. Estudo de Localização de Facilidades Logísticas. Rio de Janeiro. Dissertação de Mestrado, IME, 1990.

MORGADO, A. V. Contribuição Metodológica ao Estudo de Localização de Terminais Rodoviários Regionais Coletivos de Carga. Tese de Doutorado. UFRJ, Rio de Janeiro, 2005.

NARULA, S. C.; OGBU, U. I.; SAMUELSSON, H. M. An algorithm for the p-median problem. *Operations Research*, 25, p. 709 – 713, 1977.

NEEBE, A. W.. 1978. A branch and Bound Algorithm for the p-median transportations problem. *Journal of the Operations Research Society*, v.29. p.989-995.

NOVAES, A. G.; *Sistemas Logísticos: transporte, armazenagem e distribuição física de produtos*. Ed. Edgard Blücher, SP, 1989.

NOVAES, A. G.; ROSSETO, C. F. Localização de depósitos numa rede logística com o auxílio de GIS. *Anais do VII Congresso de Pesquisa e Ensino em Transportes*, São Paulo, Vol. 1, p. 605 – 617, 1997.

NUNES, L. F. Algoritmos Genéticos aplicados na Abordagem de um Problema Real de Roteirização de Veículos. Dissertação de Mestrado. UFPR, Curitiba, 1998.

PEREIRA, M. A. Um método Branch-and-Price para problemas de localização de p-medianas. Tese de Doutorado. INPE, São José dos Campos, 2005.

PIRES, L. A. Algoritmo Genético Aplicado na Localização de Escolas do Município de Coronel Vivida – PR. Dissertação de Mestrado. UFPR, Curitiba, 2002.

PIZZOLATO, N. D. A Heuristic for Large-Size  $p$ -Median Location Problems with Application to School Location. *Annals of Operations Research*, 50, p. 473 – 485, 1994.

PIZZOLATO, N. D.; BARROS, A. G.; BARCELOS, F. B.; CANEN, A. G. Localização de Escolas Públicas: Síntese de Algumas Linhas de Experiências no Brasil. In: SOBRAPO, 24, Rio de Janeiro, 2004.

RABUSKE, M. A.. Introdução à teoria dos Grafos. Editora da UFSC, Florianópolis, SC, 1992.

RESENDE, M. G. C.; WERNECK, R. F. On the implementation of a swap-based local search procedure for the  $p$ -median problem. In: Ladner, Richard E. (Ed.), *Proceedings of the 5<sup>o</sup> Workshop on Algorithm Engineering and Experiments*, SIAM, Philadelphia, p. 119 – 127, 2003.

RESENDE, M. G. C.; WERNECK, R. F. A hybrid heuristic for the  $p$ -median problem. *Journal of Heuristics*, 10, p. 59 – 88, 2004.

ReVelle, C. S.; SWAIN, R. W. Central Facilities Location. *Geographical Analysis*, 2, p. 30 – 42, 1970.

SCHILLING, D. A.; ROLLAND, E. e CURRENT, J. R. An efficient Tabu Search Procedure for the  $p$ -median problem. *European Journal of Operational Research*, 96, 1996, p. 329 – 342.

ROSA, V. R. Otimização em Localização de Plataformas de Produção. Dissertação de Mestrado. UFRJ, Rio de Janeiro, 2006.

ROSING, K. E.; REVELLE, C. S. Heuristic Concentration: Two stage solution construction. *European Journal of Operational Research*, 97, p. 75 – 86, 1997.

SAMPAIO, M. E. C. e S. Aplicação de Metaheurísticas ao Problema de Localização de Escolas de Ensino Fundamental. Dissertação de Mestrado. UFPR, Curitiba, 1999.

SILVA, M. V. da. Localização de Estoques na Rede de Distribuição de uma Empresa do Setor Siderúrgico. Dissertação de Mestrado, UFMG, Belo Horizonte, 2006.

SOUZA, J. C. Dimensionamento, localização e escalonamento de serviços de atendimento emergencial. Tese de Doutorado, UFSC, Florianópolis, 1996.

SUN, M. Solving the uncapacitated facility location problem using the tabu search. *Computers & Operations Research*, 33, p. 2563 – 2589, 2006.

SWAIN, R. W. A parametric decomposition approach for the solution of uncapacitated location problems. *Management Science*, 21, p. 189 – 198, 1974.

TONDO, C. M. Um modelo matemático para a localização estratégica de terminais de contêineres no interior: aplicação ao estado de São Paulo. Tese de Doutorado. USP, São Paulo, 1992.

VASCONCELOS, M. M. Modelos de Localização e Sistemas de Informações Geográficas na Assistência Materna e Perinatal: Uma aplicação no município do Rio de Janeiro. Tese de Doutorado, UFRJ, Rio de Janeiro, 1997.

TEITZ, M. B.; BART, P. Heuristic methods for estimating the generalized vertex median of a weighted graph. *Operations Research*, 16, p. 955 – 961, 1968.

WESOLOWSKY, G. O.; LOVE, R. F. Location of Facilities with Rectangular Distances among Point and Area Destinations. *Naval Research Logistic Quarterly*, 18 (1971) p. 83 – 90.

WHITAKER, R. A. A fast algorithm for the greedy interchange of large-scale clustering and median location problems. *INFOR*, 21, p. 95 – 108, 1983.

## APÊNDICE A

## A. RESOLUÇÃO DE UM PROBLEMA DIDÁTICO PELO MÉTODO DE TEITZ E BART

### A.1. O Exemplo

Com o intuito de apresentar o problema das p-medianas de forma clara e simples será proposto um exemplo didático a ser resolvido pelo método mais citado na literatura, a heurística de Teitz e Bart.

Dado o Grafo:

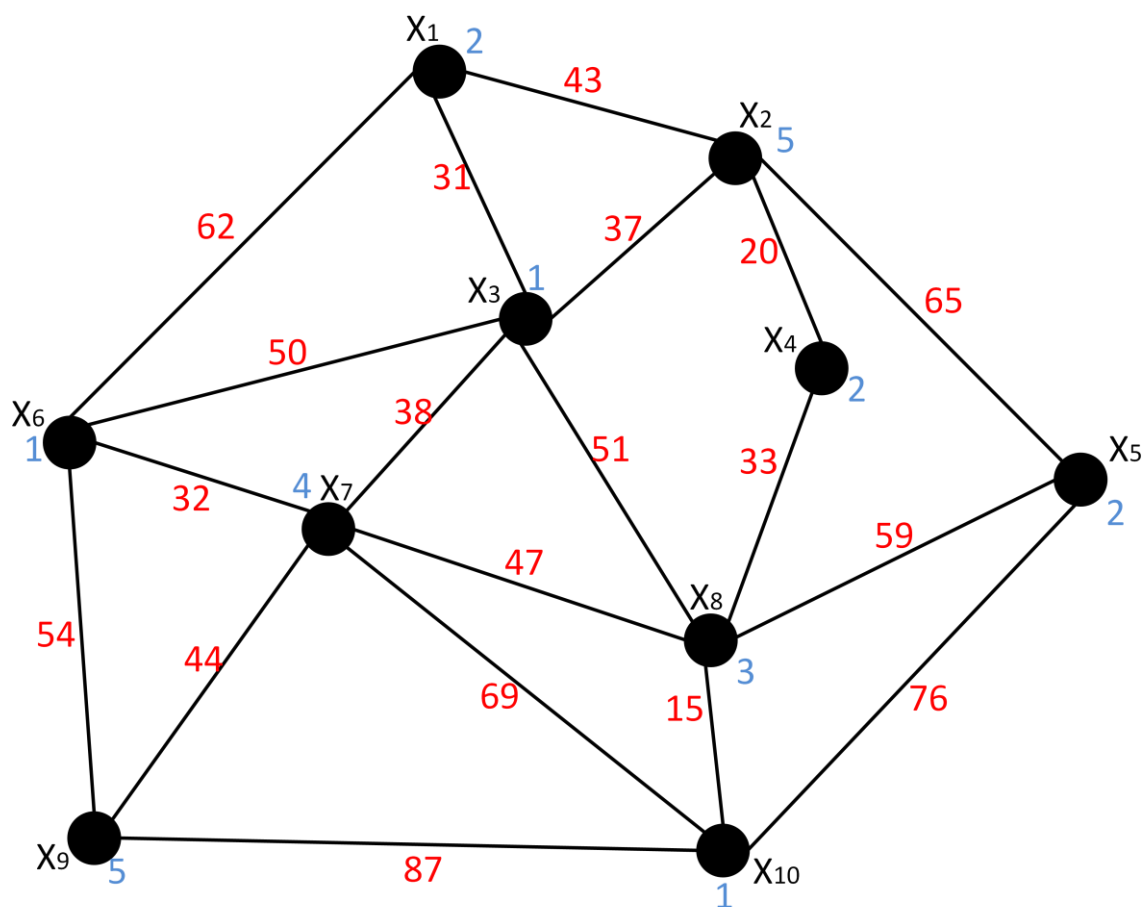


Figura A.1.: Grafo do exemplo.

Os números em vermelho são as distâncias entre os vértices e os números em azul são os pesos de cada vértice. Metaforicamente pode-se imaginar que os



números vermelhos são as distâncias em quilômetros entre as cidades (pontos ou nós) e os números azuis são quantas vezes a cidade deve ser visitada.

## A.2. Objetivo

O objetivo é encontrar a localização de 3 medianas de modo que a soma das distancias percorridas seja a menor possível.

## A.3. Resolução

Neste grafo  $G(X, A)$ , temos  $X = \{x_1, x_2, \dots, x_{10}\}$  e  $A = \{a_1, a_2, \dots, a_{19}\}$  e temos ainda o conjunto de “pesos”  $h_j$ , ou seja, em nosso exemplo temos  $h_1 = 2$ ,  $h_2 = 5$  e assim por diante.

A matriz de distâncias mínimas deste grafo é:

Tabela A.1.: Tabela representando a matriz de distâncias mínimas

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
x1	0	43	31	63	108	62	69	82	113	97
x2	43	0	37	20	65	87	75	53	119	68
x3	31	37	0	57	102	50	38	51	82	66
x4	63	20	57	0	85	107	80	33	124	48
x5	108	65	102	85	0	138	106	59	150	76
x6	62	87	50	107	138	0	32	79	54	94
x7	69	75	38	80	106	32	0	47	44	62
x8	82	53	51	33	59	79	47	0	91	15
x9	113	119	82	124	150	54	44	91	0	87
x10	97	68	66	48	76	94	62	15	87	0

A matriz distância multiplicada pelos respectivos pesos de cada vértice é:

Tabela A.2.: Tabela representando a matriz Peso – Distância

0	215	31	126	216	62	276	246	565	97
86	0	37	40	130	87	300	159	595	68
62	185	0	114	204	50	152	153	410	66
126	100	57	0	170	107	320	99	620	48
216	325	102	170	0	138	424	177	750	76
124	435	50	214	276	0	128	237	270	94
138	375	38	160	212	32	0	141	220	62
164	265	51	66	118	79	188	0	455	15
226	595	82	248	300	54	176	273	0	87
194	340	66	96	152	94	248	45	435	0

Se o objetivo fosse encontrar uma mediana, isso poderia ser feito pelo método exato, bastando somar as linhas da matriz. A linha que obtivesse a menor soma seria a mediana, o que resultaria no nó sete.

Relembrando o problema formulado com programação inteira:

$$\text{Min } Z = \sum_{i=1}^n \sum_{j=1}^m d_{ij} \xi_{ij}$$

sujeito a:

$$\sum_{i=1}^n \xi_{ij} = 1 \quad \forall j = 1, \dots, n$$

$$\sum_{i=1}^n \xi_{ii} = p$$

$$\xi_{ij} \leq \xi_{ii} \quad \forall i, j = 1, \dots, n$$

$$\xi_{ij} \in \{0,1\}$$

E o número de transmissão:

$$\sigma(x_i) = \sum_{x_j \in X} v_j \cdot d(x_i, x_j)$$

$$\sigma(\bar{x}) = \min_{x_i \in X} [\sigma(x_i)]$$

Número de transmissão para subconjuntos:

$$\sigma(X_p) = \sum_{x_j \in X} v_j \cdot d(X_p, x_j)$$

$$\sigma(\bar{X}_p) = \min_{X_p \subset X} [\sigma(X_p)]$$

Como se deseja instalar três facilidades e resolvendo pelo método de Teitz e Bart, temos:

Passo 1) Selecionar arbitrariamente S com p vértices:

$$S = \{x_1, x_2, x_3\}$$

$$X - S = \{x_4, x_5, x_6, x_7, x_8, x_9, x_{10}\}$$

Neste instante, tem-se:

$$\sigma(X_3) = FO = 1001.$$

Isso significa dizer que as medianas são os pontos  $x_1$ ,  $x_2$  e  $x_3$ . E o total de quilômetros percorridos para atender todas as cidades é de 1001 km.

Passo 2) Selecionar algum nó não testado (por exemplo,  $x_4$ ) e calcular:

$$\Delta_{ij} = \sigma(s) - \sigma(S \cup \{x_j\} - \{x_i\})$$

$$\Delta_{ij} = \sigma(s) - \sigma(S \cup \{x_4\} - \{x_j\})$$

$$\Delta_{14} = 1001 - 951 = 50$$

$$\Delta_{24} = 1001 - 1114 = -113$$

$$\Delta_{34} = 1001 - 1211 = -210$$

Passo 3) Encontrar o  $\Delta_{i_0j} = \max_{x_i \in S} [\Delta_{ij}]$

$$\Delta_{i_0j} = \max_{x_i \in S} [\Delta_{14}, \Delta_{24}, \Delta_{34}] = 50$$

Passo 3.ii)  $\Delta_{i_0j} > 0$ , fazer:

$$S \leftarrow S \cup \{x_4\} - \{x_1\}$$

E rotular  $x_4$  como “testado” e volte ao passo 2 até que não se tenham mais nós rotulados como “não testados”.

Atualmente:  $FO = 951$  e  $S = \{x_2, x_3, x_4\}$

Passo 2) Selecionar algum nó “não testado” ( $x_9$ ) e calcular  $\Delta_{ij}$ .

$$\Delta_{29} = 951 - 794 = 157$$

$$\Delta_{39} = 951 - 630 = 321$$

$$\Delta_{49} = 951 - 653 = 298$$

Passo 3)  $\Delta_{i_0j} = \max_{x_i \in S} [\Delta_{29}, \Delta_{39}, \Delta_{49}] = 321$

Passo 3.ii)  $\Delta_{i_0j} > 0$ , fazer:

$$S \leftarrow S \cup \{x_9\} - \{x_3\}$$

Rotular  $x_9$  como “testado” e vá ao P2.

Atualmente:  $FO = 630$  e  $S = \{x_2, x_4, x_9\}$

Passo 2) Selecionar um nó “não testado” ( $x_1$ ) e calcular  $\Delta_{ij}$ .

$$\Delta_{21} = 630 - 678 = -48$$

$$\Delta_{41} = 630 - 658 = -28$$

$$\Delta_{91} = 630 - 1211 = -581$$

Passo 3)  $\Delta_{i_0j} = \max_{x_i \in S} [\Delta_{21}, \Delta_{41}, \Delta_{91}] = -28$

Passo 3.i)  $\Delta_{i_0j} \geq 0$ , faça:

Rotule  $x_1$  como testado e vá ao passo 2.

Atualmente:  $FO = 630$  e  $S = \{x_2, x_4, x_9\}$

Testados:  $x_1, x_3,$

Atualmente:  $FO = 620$  e  $S = \{x_2, x_5, x_9\}$

Após testar os pontos  $x_1, x_3, x_4, x_6$  sem sucesso, houve uma redução no vértice  $x_7$ , assim:

Atualmente:  $FO = 528$  e  $S = \{x_2, x_7, x_9\}$

Os pontos 1, 3, 4, 5, 6, 7, 8 foram testados e no vértice 8, houve outra redução.

Atualmente:  $FO = 526$  e  $S = \{x_2, x_8, x_9\}$

Todos os demais vértices fora do conjunto solução foram testados e não houve melhoria. Então:

Solução: FO = 526

O passo 4 manda repetir os passos 2 e 3. Este procedimento é denominado “ciclo”. Se durante o último ciclo, não houver melhora na função objetivo, vá ao passo 5.

Passo 5) Pare.

A solução para este problema são os pontos  $x_2$ ,  $x_8$  e  $x_9$ , que resulta na seguinte configuração:

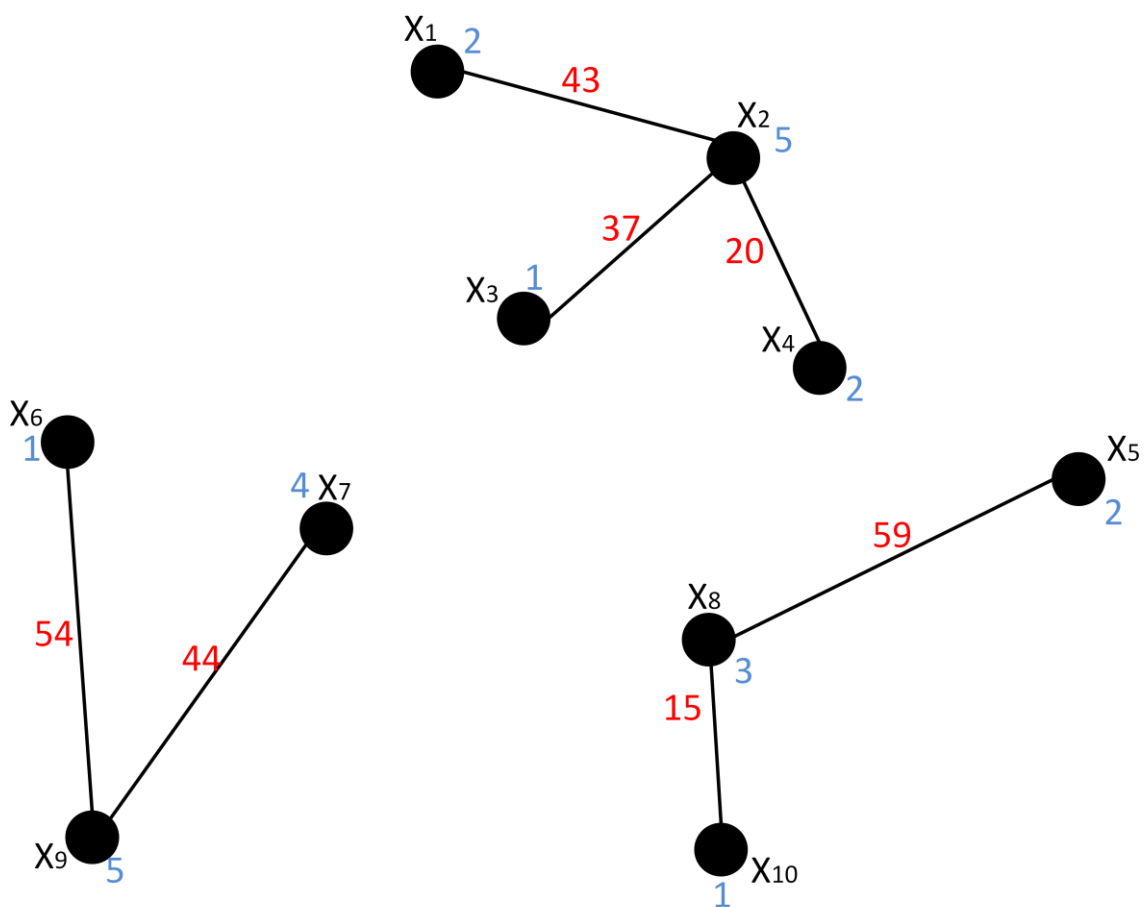


Figura A.2.: Solução do problema.

## **APÊNDICE B**

## B. RESULTADOS OBTIDOS E RESPECTIVOS TEMPOS DE EXECUÇÃO EM REDES PEQUENAS

A tabela abaixo apresenta o comportamento das técnicas propostas em redes pequenas. Um problema 50 x 3, por exemplo, possui 50 pontos candidatos e deseja-se encontrar a melhor localização para 3 medianas. Os resultados em negrito são os melhores obtidos seguidos pelo respectivo tempo de solução.

Tabela B.1: Resultados obtidos e respectivos tempos de execução em redes pequenas.

Problema	Teitz e Bart		Busca Tabu		Alg Genético		T&B Modificado	
50 x 1	<b>716.285,45</b>	00:00:031	<b>716.285,45</b>	00:00:015	<b>716.285,45</b>	00:00:947	<b>716.285,45</b>	00:00:011
50 x 2	<b>402.049,90</b>	00:00:063	<b>402.049,90</b>	00:00:016	<b>402.049,90</b>	00:01:062	<b>402.049,90</b>	00:00:016
50 x 3	<b>355.701,39</b>	00:00:047	<b>355.701,39</b>	00:00:016	359.549,06	00:01:110	<b>355.701,39</b>	00:00:031
50 x 4	<b>262.575,00</b>	00:00:079	<b>262.575,00</b>	00:00:015	264.074,73	00:02:094	262.845,64	00:00:062
50 x 5	<b>229.450,96</b>	00:00:063	<b>229.450,96</b>	00:00:015	<b>229.450,96</b>	00:02:956	244.163,63	00:00:031
50 x 6	<b>183.755,83</b>	00:00:063	183.858,18	00:00:017	185.052,87	00:03:140	205.246,32	00:00:078
50 x 7	142.479,45	00:00:094	143.822,21	00:00:016	143.977,08	00:03:471	158.300,67	00:00:015
50 x 8	<b>176.351,89</b>	00:00:062	<b>176.351,89</b>	00:00:016	176.351,89	00:04:188	189.347,59	00:00:032
50 x 9	150.265,37	00:00:094	<b>146.805,26</b>	00:00:016	<b>146.805,26</b>	00:04:887	170.885,34	00:00:031
50 x 10	130.452,54	00:00:093	131.712,50	00:00:031	132.826,01	00:05:203	153.814,11	00:00:031
100 x 1	<b>1.968.186,60</b>	00:00:094	<b>1.968.186,60</b>	00:00:031	<b>1.968.186,60</b>	00:01:062	<b>1.968.186,60</b>	00:00:031
100 x 2	<b>903.984,21</b>	00:00:078	<b>903.984,21</b>	00:00:031	904.135,35	00:01:478	<b>903.984,21</b>	00:00:015
100 x 3	<b>950.734,18</b>	00:00:094	<b>950.734,18</b>	00:00:062	<b>950.734,18</b>	00:02:125	<b>950.734,18</b>	00:00:047
100 x 4	<b>771.362,84</b>	00:00:156	<b>771.362,84</b>	00:00:078	<b>771.362,84</b>	00:02:586	<b>771.362,84</b>	00:00:078
100 x 5	492.716,26	00:00:172	<b>488.154,90</b>	00:00:062	492.287,41	00:04:188	496.355,63	00:00:047
100 x 6	<b>639.058,97</b>	00:00:156	<b>639.058,97</b>	00:00:206	<b>639.058,97</b>	00:06:187	<b>639.058,97</b>	00:00:078
100 x 7	<b>547.366,04</b>	00:00:203	<b>547.366,04</b>	00:00:253	548.760,02	00:07:250	<b>547.366,04</b>	00:00:078
100 x 8	<b>448.572,20</b>	00:00:219	<b>448.572,20</b>	00:00:297	448.913,00	00:08:265	<b>448.572,20</b>	00:00:079
100 x 9	464.299,81	00:00:234	<b>459.688,78</b>	00:00:267	464.299,81	00:08:343	462.753,55	00:00:093
100 x 10	<b>325.186,66</b>	00:00:235	<b>325.186,66</b>	00:00:313	333.023,77	00:8:500	354.187,95	00:00:109
100 x 15	<b>225.388,29</b>	00:00:219	<b>225.388,29</b>	00:00:250	226.814,16	00:10:281	260.861,99	00:00:094
100 x 20	<b>143.381,44</b>	00:00:360	<b>143.381,44</b>	00:00:391	144.113,15	00:13:453	187.589,88	00:00:062
100 x 25	<b>106.870,29</b>	00:00:453	<b>106.870,29</b>	00:00:500	109.214,42	00:14:406	164.260,03	00:00:078
100 x 30	102.712,04	00:00:594	<b>102.656,59</b>	00:00:766	105.452,71	00:14:515	149.119,19	00:00:047



## APÊNDICE C

## C. RESULTADOS OBTIDOS E RESPECTIVOS TEMPOS DE EXECUÇÃO EM REDES MÉDIAS

Tabela C.1: Resultados obtidos e respectivos tempos de execução em redes médias.

Problema	Teitz e Bart		Busca Tabu		Alg Genético		T&B Modificado	
	Valor	Tempo	Valor	Tempo	Valor	Tempo	Valor	Tempo
200 x 5	<b>931.882,33</b>	00:00:296	<b>931.882,33</b>	00:00:422	941.594,54	00:10:359	931.882,33	00:00:125
200 x 10	<b>677.503,76</b>	00:00:562	<b>677.503,76</b>	00:00:984	700.463,27	00:12:469	703.204,23	00:00:235
200 x 15	<b>511.757,22</b>	00:00:969	<b>511.757,22</b>	00:01:828	516.267,67	00:12:906	540.058,38	00:00:266
200 x 20	426.897,16	00:01:063	<b>420.895,09</b>	00:02:828	429.517,76	00:15:062	435.564,13	00:00:297
200 x 25	357.878,53	00:01:390	<b>357.732,98</b>	00:03:718	361.088,91	00:16:156	412.599,25	00:00:281
200 x 30	318.265,65	00:01:640	318.265,65	00:05:406	320.502,23	00:19:172	349.777,95	00:00:319
200 x 35	270.819,76	00:02:109	<b>270.708,81</b>	00:07:625	286.716,89	00:23:219	326.389,50	00:00:266
200 x 40	223.747,91	00:02:359	<b>223.561,19</b>	00:06:469	236.134,26	00:22:563	312.199,62	00:00:234
200 x 45	<b>217.730,58</b>	00:02:828	218.559,49	00:08:313	226.785,74	00:24:312	300.798,19	00:00:265
200 x 50	<b>195.395,82</b>	00:03:125	195.576,62	00:09:594	203.316,84	00:25:735	261.208,26	00:00:203
200 x 55	165.061,57	00:03:343	<b>164.722,20</b>	00:07:131	182.084,53	00:27:734	192.205,11	00:00:360
200 x 60	<b>124.206,85</b>	00:05:297	<b>124.206,85</b>	00:09:124	136.052,71	00:31:625	144.593,63	00:00:343
300 x 5	1.537.999,93	00:00:344	<b>1.531.149,64</b>	00:01:078	1.566.515,66	00:10:468	1.537.999,93	00:00:172
300 x 10	<b>1.053.844,04</b>	00:01:234	<b>1.053.844,04</b>	00:03:593	1.094.599,39	00:11:094	<b>1.053.844,04</b>	00:00:391
300 x 15	752.773,80	00:01:860	<b>744.217,96</b>	00:06:203	775.726,96	00:13:172	767.394,25	00:00:328
300 x 20	659.075,67	00:02:500	<b>658.626,70</b>	00:09:000	697.787,02	00:12:468	668.681,84	00:00:516
300 x 25	563.384,17	00:03:234	<b>556.121,80</b>	00:13:875	582.958,36	00:16:422	573.063,81	00:00:656
300 x 30	<b>544.296,01</b>	00:02:547	546.181,14	00:14:687	587.371,56	00:19:110	599.476,82	00:00:547
300 x 35	409.872,10	00:02:984	<b>408.420,33</b>	00:15:563	433.158,22	00:20:297	446.381,94	00:00:453
300 x 40	389.929,52	00:06:891	<b>389.801,39</b>	00:20:109	401.032,92	00:26:297	426.650,44	00:00:594
300 x 45	341.866,59	00:06:047	<b>339.926,93</b>	00:19:219	352.396,31	00:25:532	386.829,52	00:00:578
300 x 50	<b>311.913,55</b>	00:09:250	<b>311.913,55</b>	00:27:542	343.823,57	00:34:578	370.086,36	00:00:687
300 x 55	<b>308.760,70</b>	00:08:391	309.505,76	00:41:610	330.679,94	00:46:063	388.278,76	00:00:594
300 x 60	<b>258.597,93</b>	00:16:782	259.952,64	00:48:765	286.098,88	00:55:906	316.419,22	00:00:622

Tabela C.2: Resultados obtidos e respectivos tempos de execução em redes médias.

Problema	Teitz e Bart		Busca Tabu		Alg Genético		T&B Modificado	
400 x 10	1.671.052,75	00:01:532	<b>1.670.968,42</b>	00:02:797	1.711.731,45	00:01:063	1.699.212,94	00:00:531
400 x 20	<b>1.073.476,31</b>	00:03:641	<b>1.073.476,31</b>	00:05:766	1.117.458,35	00:06:735	1.129.680,43	00:00:672
400 x 30	835.094,18	00:06:281	<b>834.219,81</b>	00:10:500	869.967,96	00:13:062	899.332,70	00:00:735
400 x 40	671.922,44	00:09:953	<b>668.300,98</b>	00:22:422	703.674,20	00:23:750	737.465,61	00:00:859
500 x 10	1.902.001,39	00:01:609	<b>1.901.723,91</b>	00:05:812	1.977.157,78	00:06:328	1.904.611,02	00:00:968
500 x 20	1.433.466,43	00:06:047	<b>1.418.395,16</b>	00:18:922	1.480.027,61	00:22:531	1.430.300,95	00:01:312
500 x 30	<b>1.039.266,99</b>	00:13:375	1.041.067,17	00:28:516	1.086.715,96	00:30:047	1.098.096,95	00:01:281
500 x 40	946.419,86	00:16:672	<b>940.412,37</b>	00:39:703	1.012.068,88	00:42:609	969.997,44	00:01:969
500 x 50	757.063,77	00:26:734	<b>752.733,82</b>	01:44:500	781.786,56	02:15:125	819.673,24	00:02:078
600 x 10	<b>2.398.738,83</b>	00:03:141	<b>2.398.738,83</b>	00:06:109	2.506.536,78	00:59:844	<b>2.398.738,83</b>	00:01:563
600 x 20	1.711.609,95	00:08:000	<b>1.704.355,15</b>	00:27:797	1.753.352,85	01:52:297	1.715.621,15	00:01:265
600 x 30	1.254.915,69	00:25:609	<b>1.251.958,15</b>	00:53:281	1.357.564,71	00:49:360	1.265.112,99	00:02:172
600 x 40	1.104.798,91	00:36:219	<b>1.101.675,28</b>	01:15:109	1.201.979,49	01:41:515	1.169.800,47	00:01:922
600 x 50	<b>937.924,64</b>	00:39:953	<b>937.924,64</b>	02:29:250	1.001.904,06	02:56:984	981.922,12	00:02:328
600 x 60	825.917,13	00:59:266	<b>824.008,49</b>	03:00:766	891.852,07	04:48:766	866.455,93	00:01:969
700 x 10	2.865.268,58	00:04:922	<b>2.863.963,14</b>	00:19:750	3.010.707,23	01:02:156	2.878.000,77	00:01:422
700 x 20	<b>1.966.047,12</b>	00:09:984	<b>1.966.047,12</b>	00:32:969	2.067.628,48	01:34:328	1.967.104,86	00:01:766
700 x 30	1.578.579,18	00:19:390	<b>1.575.647,04</b>	01:05:656	1.695.589,43	01:55:656	1.596.133,62	00:02:312
700 x 40	1.349.556,93	00:45:032	<b>1.347.619,78</b>	02:42:953	1.460.351,77	03:05:828	1.374.080,15	00:02:906
700 x 50	1.125.441,38	01:10:750	<b>1.123.640,66</b>	03:09:581	1.209.647,88	03:28:094	1.187.427,34	00:03:375
700 x 60	1.012.989,38	01:44:625	<b>1.011.807,22</b>	04:44:438	1.081.304,36	04:11:203	1.082.569,45	00:03:156
700 x 70	<b>881.483,85</b>	03:33:297	881.545,81	08:54:469	918.896,00	05:13:187	920.344,84	00:03:594

## APÊNDICE D

## D. RESULTADOS OBTIDOS E RESPECTIVOS TEMPOS DE EXECUÇÃO EM REDES MÉDIAS

Tabela D.1: Resultados obtidos e respectivos tempos de execução em redes grandes.

Problema	Teitz e Bart		Busca Tabu		Alg Genético		T&B Modificado	
	Valor	Tempo	Valor	Tempo	Valor	Tempo	Valor	Tempo
800 x 10	<b>3.445.034,13</b>	00:04:938	<b>3.445.034,13</b>	00:13:641	3.685.064,88	01:11:687	<b>3.445.034,13</b>	00:02:078
800 x 20	2.391.062,49	00:21:328	<b>2.381.041,65</b>	00:55:625	2.498.041,39	01:13:782	2.410.300,47	00:01:953
800 x 30	1.876.849,80	00:41:890	<b>1.874.394,82</b>	01:20:484	2.052.424,74	02:05:219	1.898.288,35	00:02:547
800 x 40	<b>1.565.662,07</b>	01:20:235	<b>1.565.662,07</b>	03:27:125	1.700.258,58	03:07:969	1.584.437,55	00:03:422
800 x 50	<b>1.373.006,80</b>	1:30:813	1.374.748,14	04:29:765	1.479.585,46	04:08:140	1.395.390,15	00:03:390
800 x 60	1.257.344,47	02:46:687	<b>1.257.050,16</b>	07:35:500	1.359.622,30	06:13:437	1.276.347,63	00:04:312
800 x 70	1.110.122,51	03:54:312	<b>1.109.694,60</b>	09:54:000	1.237.963,70	07:17:719	1.125.248,87	00:04:687
800 x 80	968.335,39	04:09:359	<b>966.600,83</b>	13:12:344	1.061.532,50	09:22:063	990.765,94	00:04:297
900 x 10	3.816.395,94	00:06:890	<b>3.813.106,53</b>	00:25:609	4.001.195,57	01:03:734	3.816.395,94	00:02:312
900 x 20	2.570.471,27	00:17:625	<b>2.569.630,09</b>	01:22:766	2.777.061,10	01:26:531	2.583.777,64	00:02:906
900 x 30	<b>2.162.055,35</b>	00:51:031	<b>2.162.055,35</b>	02:18:391	2.309.870,93	02:08:328	2.176.622,68	00:03:359
900 x 40	1.748.614,59	01:12:938	<b>1.746.374,33</b>	05:42:484	1.878.350,93	04:10:516	1.772.090,78	00:04:625
900 x 50	1.571.091,30	02:25:968	<b>1.570.349,66</b>	08:59:343	1.756.341,18	06:09:593	1.583.991,32	00:04:609
900 x 60	1.388.599,49	02:29:079	<b>1.386.313,79</b>	10:19:969	1.497.066,21	09:17:750	1.411.973,53	00:05:266
900 x 70	<b>1.260.728,75</b>	06:03:297	1.266.208,85	15:06:640	1.388.703,97	12:19:609	1.282.109,26	00:10:266
900 x 80	1.200.258,09	05:20:797	<b>1.195.268,98</b>	21:23:266	1.308.200,76	18:24:156	1.200.811,19	00:16:078
900 x 90	1.092.143,32	08:28:609	<b>1.091.714,97</b>	30:57:453	1.222.417,29	25:29:281	1.092.704,09	00:15:500
1000 x 10	4.616.358,03	00:13:735	<b>4.616.098,06</b>	00:33:688	4.785.734,17	01:42:437	4.616.840,57	00:06:282
1000 x 20	3.294.703,80	00:32:625	<b>3.280.811,35</b>	01:32:122	3.578.500,59	02:04:751	3.282.452,36	00:07:078
1000 x 30	<b>2.539.493,38</b>	00:59:143	2.540.132,67	02:34:282	2.734.977,72	02:47:875	2.564.356,30	00:08:094
1000 x 40	<b>2.112.578,72</b>	01:52:239	2.113.140,27	08:02:265	2.361.226,21	06:08:265	2.127.555,16	00:11:219
1000 x 50	1.936.989,72	02:49:046	<b>1.917.053,19</b>	12:18:922	2.113.096,24	10:14:719	1.942.127,65	00:12:968
1000 x 60	1.790.177,55	03:09:110	<b>1.775.015,03</b>	18:18:828	1.992.610,41	15:45:969	1.791.268,95	00:12:609
1000 x 70	1.601.703,69	05:59:422	<b>1.598.068,23</b>	23:25:125	1.756.477,56	20:21:047	1.618.197,76	00:13:328
1000 x 80	1.486.011,71	08:46:375	<b>1.484.400,62</b>	40:49:515	1.572.925,66	32:21:437	1.503.473,21	00:15:687
1000 x 90	1.291.388,39	15:14:797	<b>1.290.475,13</b>	34:16:625	1.346.342,83	33:35:234	1.305.072,22	00:17:657
1000x100	1.219.071,54	13:02:906	<b>1.218.017,68</b>	44:10:625	1.378.065,00	36:39:078	1.233.265,41	00:19:254

Tabela D.2: Resultados obtidos e respectivos tempos de execução em redes grandes.

Problema	Teitz e Bart		Busca Tabu		Alg Genético		T&B Modificado	
	Resultado	Tempo	Resultado	Tempo	Resultado	Tempo	Resultado	Tempo
2000 x 3	<b>20.954.560,4</b>	00:04:21	<b>20.954.560,4</b>	00:26:53	21.056.424,0	00:21:52	<b>20.954.560,4</b>	00:17:51
2000 x 5	<b>15.721.059,8</b>	00:13:17	<b>15.721.059,8</b>	00:31:12	15.924.726,2	00:23:98	<b>15.721.059,8</b>	00:16:65
2000 x 10	11.201.925,9	00:42:84	<b>11.142.180,6</b>	02:45:37	11.536.829,8	02:07:81	11.229.746,5	00:10:50
2000 x 20	<b>7.445.776,4</b>	02:03:79	7.453.143,3	09:29:01	8.061.442,5	06:48:92	7.572.761,3	00:15:79
3000 x 3	<b>30.801.991,8</b>	00:37:79	<b>30.801.991,8</b>	01:12:61	30.993.506,6	00:43:29	<b>30.801.991,8</b>	00:26:48
3000 x 5	23.830.354,4	00:41:12	<b>23.700.852,6</b>	01:45:21	24.356.133,0	01:27:42	23.830.354,4	00:28:59
3000 x 10	16.674.074,1	01:45:04	<b>16.502.559,0</b>	05:57:26	17.257.739,1	03:18:37	16.591.377,5	00:31:51
3000 x 20	11.243.051,9	04:28:29	<b>11.241.382,9</b>	32:25:03	12.038.828,8	10:16:82	11.282.250,1	00:30:68
3000 x 30	9.093.899,1	10:37:09	<b>9.035.909,3</b>	58:32:76	9.583.230,4	19:37:71	9.080.092,2	00:34:65
5000 x 3	<b>51.856.455,9</b>	00:36:67	<b>51.856.455,9</b>	07:29:07	52.210.396,0	00:29:42	<b>51.856.455,9</b>	00:41:49
5000 x 5	40.079.986,3	00:36:03	<b>40.038.386,5</b>	15:23:39	41.116.987,1	01:06:93	40.086.237,6	00:48:12
5000 x 7	32.754.326,9	03:10:89	<b>32.579.905,4</b>	12:31:86	33.402.091,9	08:15:68	32.580.297,2	02:15:12
5000 x 10	27.457.408,2	02:52:93	<b>27.147.345,6</b>	45:02:65	27.894.255,0	18:13:92	27.273.845,8	02:15:50
5000 x 20	<b>18.737.483,2</b>	16:21:71	19.255.826,6	1:13:43:5	19.882.720,1	23:45:57	19.063.290,5	02:25:23
5000 x 25	<b>17.005.553,3</b>	17:29:20	17.349.360,6	1:45:29:9	17.818.562,6	31:09:78	<b>17.005.553,3</b>	03:03:87
5000 x 30	15.162.639,8	27:27:62	<b>14.778.845,3</b>	2:52:22:0	16.147.680,9	32:55:91	15.063.192,4	03:13:81
5000 x 40	13.154.948,8	43:27:87	<b>12.983.449,5</b>	4:31:06:6	13.812.475,6	42:17:72	13.156.984,6	03:06:41
5000 x 50	11.753.893,2	1:34:42:82	<b>11.691.750,1</b>	7:57:08:0	12.765.150,8	1:22:18:6	<b>11.691.750,1</b>	03:57:20

**ANEXO A**

## **A. CONCEITOS BÁSICOS DA TEORIA DOS GRAFOS**

Com o intuito de fornecer conceitos básicos, visando o entendimento do problema tratado no presente trabalho, serão apresentados os conceitos básicos da teoria dos grafos. Esta abordagem é a mais utilizada para representar uma malha viária em se tratando de problemas de transporte.

### **A.1. Definições Gerais sobre Teoria de Grafos**

#### **A.1.1. Grafo**

Um grafo é uma estrutura matemática  $G = (X, A)$ , onde  $X = \{x_1, x_2, \dots, x_n\}$  é um conjunto nós (ou vértices) do grafo, e  $A = \{a_1, a_2, \dots, a_m\}$  é o conjunto de arcos do grafo que ligam todos ou alguns desses vértices (Christofides, 1975).

#### **A.1.2. Arcos e Arestas**

Se as linhas  $a_i \in A, i = 1, 2, \dots, m$ , têm a orientação, então são chamadas de arcos. Se não tiverem orientação, são chamadas arestas de acordo com Christofides (1975).

#### **A.1.3. Grafos Orientados, Não Orientados e Mistos**

A denominação, Grafos Orientados, Não Orientados e Mistos, envolve, respectivamente, os grafos cujas linhas  $a_i$  são somente arcos, somente arestas ou contem ambos (arcos e arestas).

Exemplos são apresentados nas figuras A.1 a A.3.



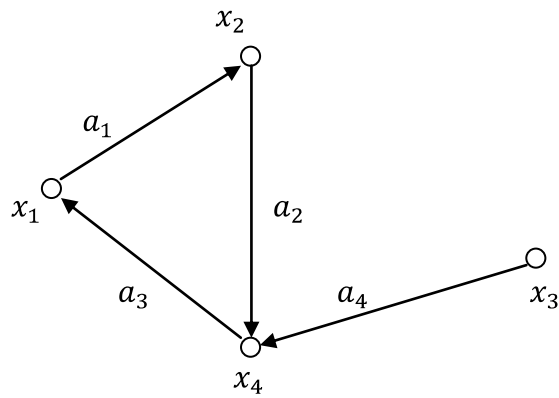


Figura A.1.: Grafo Orientado

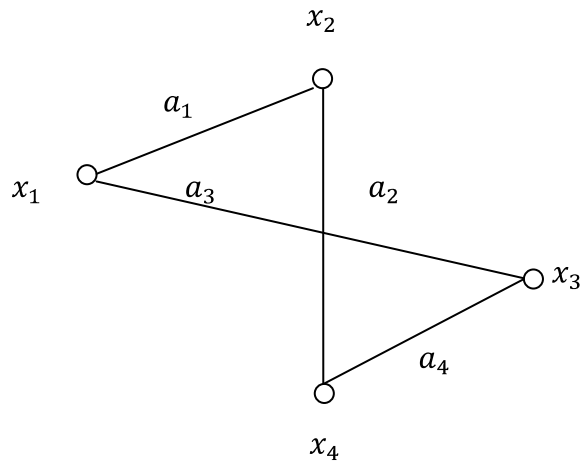


Figura A.2.: Grafo Não Orientado

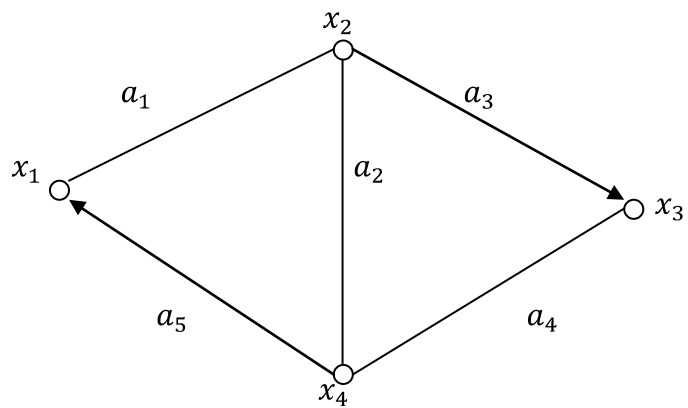


Figura A.3.: Grafo Misto

Nos Grafos Orientados, quando se faz referência a um arco, por exemplo, o arco  $a_2$  da figura A.1., matematicamente denota-se como o par ordenado  $a_2 = (x_2, x_4)$ , significando que o arco  $a_2$  se origina em  $x_2$  e termina em  $x_4$ . Já no grafo não orientado da figura A.2. uma referência à aresta  $a_1$  implica, indiretamente, tanto o arco  $(x_1, x_2)$ , como o arco  $(x_2, x_1)$ , ou ambos.

#### A.1.4. Laço

Laço é um arco cujo vértice inicial coincide com o vértice final (Ver figura A.4.).

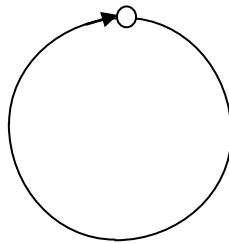


Figura A.4. Laço

#### A.1.5. Rede

Rede é um grafo  $G(X, A)$  formado por um conjunto finito de nós e de arcos, porém, uma rede não modela arcos do tipo  $(x_i, x_i)$ , ou seja, não contem laços.

#### A.1.6. Vértices Adjacentes

Dois vértices  $x_i$  e  $x_j$  de um grafo  $G = (X, A)$  são ditos adjacentes, se algum, dos arcos  $(x_i, x_j)$ ,  $(x_j, x_i)$  ou ambos existem no grafo (Christofides, 1975). Assim, por exemplo, no grafo da figura A.1 os vértices  $x_1$  e  $x_2$  são adjacentes, o que não ocorre em relação à  $x_1$  e  $x_3$ .

### A.1.7. Grau de um Vértice

Grau de um vértice  $x_i$  qualquer é o número de arcos ou arestas nele incidentes.

### A.1.8. Grau de Entrada e de Saída de um Vértice

Em um grafo orientado, defini-se como grau de entrada e grau de saída de um vértice  $x_i$  qualquer o número total de arcos que tem o vértice  $x_i$  como vértice final e inicial respectivamente.

### A.1.9. Caminhos e Cadeias

Em um grafo direcionado, caminho é uma seqüência de arcos, onde o vértice final de um arco é o vértice inicial do próximo arco. Analogicamente, define-se cadeia para um grafo não direcionado, isto é, a extremidade de uma aresta coincide com a extremidade de outra aresta (Binfaré, 1993).

Por exemplo, na Figura A.1., a seqüência de arcos  $a_4, a_3, a_l$  determina um caminho, e na Figura A.2. a seqüência de arestas  $a_1, a_2, a_4$  determina uma cadeia.

### A.1.10. Circuitos e Ciclos

Circuito é um caminho fechado, isto é, o vértice inicial coincide com o vértice final. No caso de uma cadeia fechada tem-se o que é denominado por ciclo. Na figura 2.1. o caminho  $(a_1, a_2, a_3)$  constitui um circuito. Na figura A.2., a seqüência  $(a_1, a_2, a_3)$  constitui um circuito. Na figura A.2., a seqüência  $(a_1, a_2, a_4, a_3)$  constitui um ciclo.

### A.1.11. Conexidade

Um grafo  $G = (X, A)$  é dito conexo, se dados quaisquer vértices  $x_i$  e  $x_j$  pertencentes a  $X$ , com  $x_i \neq x_j$ , existe uma cadeia ligando-os. Caso contrário é dito desconexo. Um grafo  $G = (X, A)$  é dito fortemente conexo, se dados qualquer

$x_i, x_j$  pertencentes a  $X$ , com  $x_i \neq x_j$ , existe um caminho ligando  $x_j$  a  $x_i$ . Um grafo é dito unilateralmente conexo, se dados  $x_i, x_j$  pertencentes a  $X$ , existe pelo menos um dos caminhos: o de  $x_i$  para  $x_j$  ou  $x_j$  para  $x_i$ . Exemplos são apresentados nas figuras A.6. e A.7.

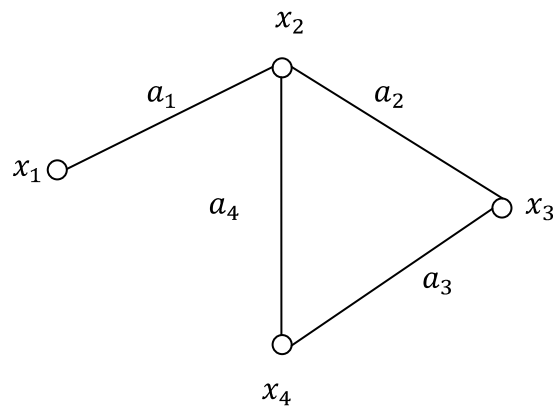


Figura A.5.: Grafo não Orientado Conexo

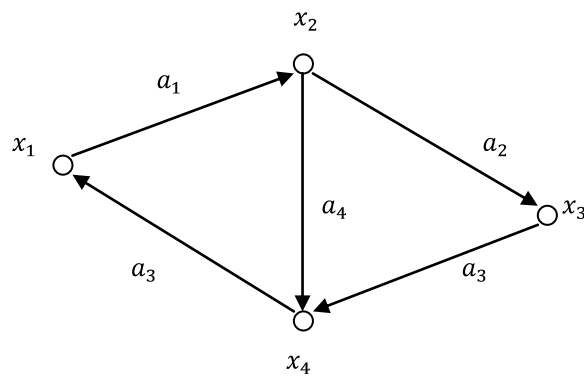


Figura A.6.: Grafo Orientado Fortemente Conexo

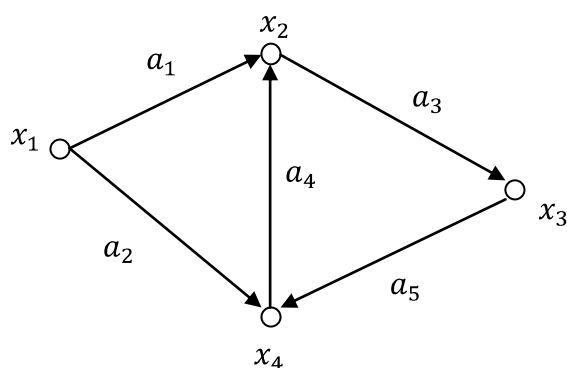


Figura A.7.: Grafo Orientado unilateralmente Conexo

### A.1.12. Árvores

São inúmeras as definições e notações empregadas por diferentes autores a respeito do conceito de árvore, aqui se optou por elencar as definições apresentadas por Rabuske (1992).

- i) Árvore é um grafo conexo de  $n$  vértices e  $(n - 1)$  arestas;
- ii) Um grafo conexo e sem ciclos;
- iii) Um grafo no qual cada par de vértices é ligado por um e somente um caminho simples;
- iv) Um grafo conexo, porém, se qualquer de suas arestas for retirada, a conectividade fica interrompida;
- v) Um grafo acíclico e conexo, porém, se dois vértices quaisquer, não adjacentes, forem ligados por uma aresta, então, o grafo passará a ter exatamente um ciclo;
- vi) Um grafo conexo que não possui sub-grafo  $(K_n)$  para  $n \geq 3$ ;
- vii) Um grafo que não possui  $K_3 \cup K_2$  ou  $K_3 \cup K_1$ , mas tem  $n = m + 1$ , onde  $n$  é o número de vértices e  $m$  o número de arestas.

### A.1.13. Grafo Ponderado

Dado um grafo  $G = (X, A)$ , se é associado a cada arco (ou aresta)  $a_{ij} \in A$  um número  $c_{ij}$  (chamado custo, distância ou peso do arco ou da aresta) e a cada

vértice  $x_k \in X$  um número  $w_k$  (chamado peso do vértice) o grafo  $G$  é dito grafo ponderado (Christofides, 1975).

#### A.1.14. Grafo Parcial

Um grafo parcial de  $G = (X, A)$ , é o grafo  $G = (X, A')$ , onde  $A' \subset A$ . (Ver figuras A.8 e A.9.).

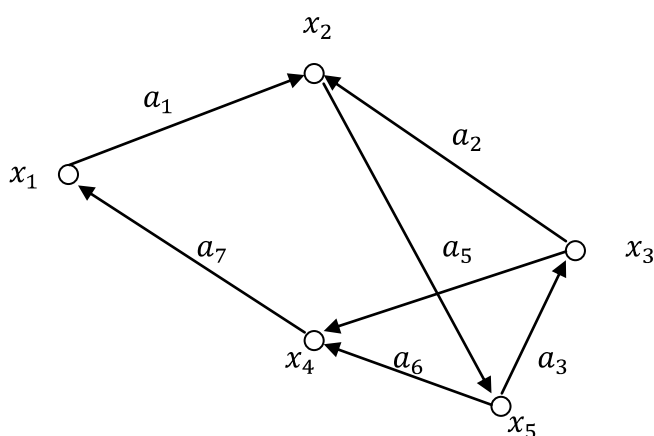


Figura A.8.: Grafo Proposto

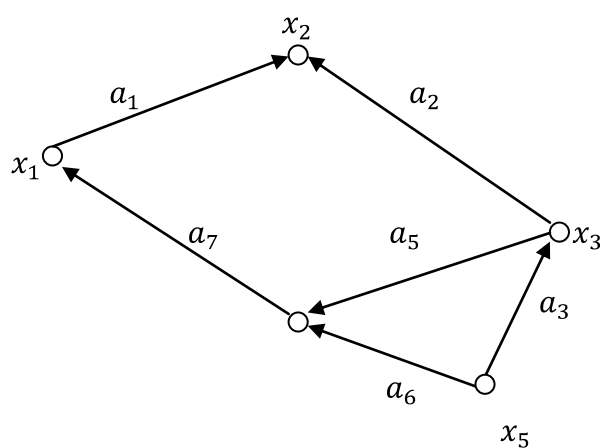


Figura A.9.: Grafo Parcial

Como se pode observar o grafo da figura A.9. é um grafo parcial do grafo apresentado na Figura A.8.

### A.1.15. Sub-Grafo

Sub-Grafo de  $G = (X, A)$  é o grafo  $G = (N, A_N)$  onde  $N \subset X$  e  $A_N$  é a família de arcos de  $A$  que estão contidos no produto cartesiano  $N \times N$ . Por exemplo, o grafo da Figura A.10. apresentado a seguir é um sub-grafo do grafo da Figura A.8.

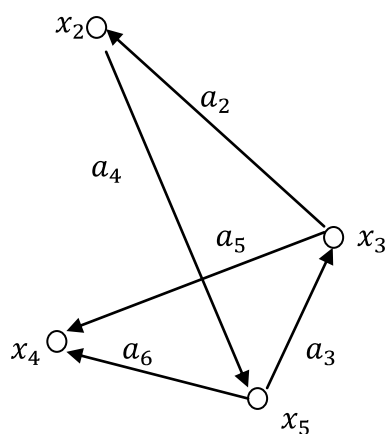


Figura A.10.: Sub-Grafo

### A.1.16. Sub-Grafo Parcial

Um Sub-Grafo Parcial de  $G (X, A)$ , é um grafo parcial de um sub-grafo de  $G (X, A)$ . Por exemplo, o grafo da Figura A.11. é um grafo parcial do sub-grafo apresentado na Figura A.10.

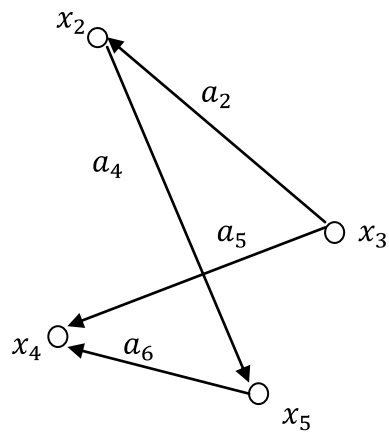


Figura A.11.: Sub-Grafo Parcial