UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

TÉCNICA DE BUSCA BASEADA EM ALGORITMO GENÉTICO PARA LOCALIZAÇÃO DE p-MEDIANAS

DISSERTAÇÃO SUBMETIDA À

UNIVERSIDADE FEDERAL DE SANTA CATARINA

PARA A OBTENÇÃO DO TITULO DE :

MESTRE EM ENGENHARIA DE PRODUÇÃO

NEYZA BIBIANA GUZMÁN MERCADO

FLORIANÓPOLIS
SANTA CATARINA – BRASIL

2001

TÉCNICA DE BUSCA BASEADA EM ALGORITMO GENÉTICO PARA LOCALIZAÇÃO DE p-MEDIANAS

NEYZA BIBIANA GUZMÁN MERCADO

ESTA DISSERTAÇÃO FOI JULGADA ADEQUADA PARA A OBTENÇÃO DO TITULO DE

"MESTRE EM ENGENHARIA"

ESPECIALIDADE ENGENHARIA DE PRODUÇÃO
APROVADA EM SUA FORMA FINAL PELO PROGRAMA DE PÓS-GRADUAÇÃO

Prof. Edson Pacheco Paladini Dr.
Coordenador do Programa

BANCA EXAMINADORA :

Prof. Sérgio Fernando Mayerle, Dr.
Presidente

Prof. Antônio Sérgio Coelho, Dr.

Mirian B Gonçalves

Prof. Mirian Buss Gonçalves, Dra.

Dedico este trabajo a los seres que mas amo... es para ti papito Orlando y para ti mamita Susy, es para ustedes y por ustedes cada uno de mis logros.

AGRADECIMENTOS

- A Deus por estar presente em cada um dos meus dias;

- Aos meus Pais, Orlando e Susy pelo infinito apoio, amor, compreensão e estimulo;

- Aos meus irmãos Silvita, Carito e Alvarito... muito obrigado pelo apoio;

- Ao Professor Sérgio Mayerle pela orientação, paciência e incentivo recebido durante todo o curso.

- Aos Professores Antônio Sérgio Coelho e Mirian Buss Gonçalves pelas contribuições e por terem aceitado o convite para participar da banca examinadora;

- Aos meus amigos das Baias, Lucimara Brandão, João Vollert, Felipe, Miguelito, Edlene, e em especial a minha grande amiga Rosiene;

- A este lindo país que me acolheu durante todo este tempo, a todo o povo brasileiro por ser tão hospedeiro, Muito Obrigada!!

SUMÁRIO

Capítulo III

Capítulo IV

Capítulo V

LISTA DE FIGURAS

LISTA DE TABELAS

RESUMO

Neste trabalho, o problema das p-medianas é apresentado, e é feita uma revisão da literatura sobre o tema.

Para o problema das p-medianas foi desenvolvida e implementada uma heurística baseada em algoritmos genéticos, cujos resultados foram equiparáveis aos obtidos em outros métodos, apontados na literatura como mais indicados para a resolução do problema. Em particular, para efeito de validação dos resultados foi implementado o método de substituição de vértices proposto por Teitz e Bart (1968), citado na literatura como o método mais usado e que melhores resultados oferece para o problema das p-medianas.

ABSTRACT

In this research the p-median problem is shown, and it is performed a literature review about this theme.

For the p-medians problem was developed and implemented a heuristic based on genetics algorithm, which results are equivalents to others methods pointed in the literature as the most indicated for solving the problem. In particular, to validate the results, was implemented a substitution method of vertices proposed by Teitz and Bart (1968), considered on the literature as the most used method and that gives the best results for the p-medians problem.

CAPÍTULO I

1. INTRODUÇÃO

1.1. Considerações Iniciais

De forma geral o problema de localização de p-medianas ou "location of p-medians" consiste em encontrar *p* lugares adequados para localizar instalações, de modo a minimizar a soma total das distâncias de todos os vértices à instalação mais próxima.

A resolução de problema de p-medianas envolve um grande número de operações combinatóriais, o que o torna um problema de difícil solução. Muitos autores têm apresentado métodos de solução, entre os quais se destaca o método de substituição de vértices proposto por Teiz e Bart, que apresenta bons resultados, contudo ainda existem certas dificuldades a serem superadas.

Embora ainda existam algumas críticas aos métodos heurísticos, a característica de obterem boas soluções (e em muitos casos soluções ótimas) em intervalos de tempos reduzidos e compatíveis com as exigências de situações reais, têm contribuído para o grande interesse pelo assunto, sendo que, para diferentes problemas combinatoriais, os melhores resultados que a literatura apresenta foram obtidos através de heurísticas.

O grande desafio do problema de localização de p-medianas consiste em encontrar um método eficiente para a obtenção da melhor solução para o problema.

1.2. Justificativa

O tema de pesquisa justifica-se em razão da necessidade de encontrar a localização considerada ótima para a instalação de medianas em um bairro, cidade ou região. Este tema reveste-se de amplo interesse em muitas aplicações práticas, como o caso da localização de hospitais, estações de bombeiros, ou localizações de escolas, postos de coleta de cartas, cabinas de telefones, subestações de redes de energia elétrica, centros de abastecimento (depósitos), e assim como estas muitas outras aplicações, onde o objetivo é minimizar a distância desde os usuários a instalação mais próxima.

Nesse contexto a importância deste trabalho é inerente às muitas aplicações onde a melhor localização de p-medianas é um fator prioritário, considerando que um bom resultado contribui consideravelmente, para a redução do custo de transporte associado ao problema.

1.3. Objetivos

A presente pesquisa tem como objetivo geral propor, implementar e avaliar uma heurística baseada em algoritmos genéticos para determinar a melhor localização de p-medianas. Com essa visão os objetivos específicos são os seguintes:

- Apresentar uma revisão bibliográfica do problema das p-medianas, citando a evolução do assunto desde as pesquisas iniciais até as mais recentes, enfocando as mais relevantes;

- Propor uma técnica de busca para localizar p-medianas com o intuito de encontrar uma heurística que obtenha uma boa solução para o problema abordado;

- Analisar e avaliar comparativamente os resultados apresentados pela heurística proposta em relação ao método de substituição de vértices proposto por Teitz e Bart com o intuito de validar os resultados obtidos.

1.4. Metodologia

Sob o ponto de vista da abordagem, utilizada para a consolidação dos resultados, este trabalho pode ser classificado como pesquisa quantitativa.

Do ponto de vista dos procedimentos técnicos, caracteriza-se também como pesquisa bibliográfica, uma vez que a partir dela se procurou conhecer as diferentes formas de contribuições científicas que se realizaram sobre o tema em questão.

O desenvolvimento metodológico para atender os objetivos descritos neste estudo, consta de cinco etapas:

Na primeira etapa, foi efetuada a pesquisa bibliográfica, identificando as publicações existentes sobre o tema e os aspectos que já foram abordados. Posteriormente, na segunda etapa, foi realizada uma analise das técnicas disponíveis na literatura com o intuito de obter conhecimento suficiente para propor uma heurística que forneça bons resultados para o problema das p-medianas.

Já na terceira etapa, efetuou-se a escolha da ferramenta que dará suporte à heurística a ser proposta, neste caso foi escolhido algoritmos genéticos, pela simplicidade de implementação e a capacidade de generalização.

Na quarta etapa foi concretizada a implementação computacional, e por último na quinta etapa, aconteceu a análise e validação os resultados computacionais obtidos pela heurística implementada, em relação ao método de substituição de vértices proposto por Teitz e Bart.

1.6. Estrutura da Dissertação

Esta pesquisa subdivide-se em 5 capítulos. Neste primeiro capítulo, é apresentada uma introdução, na qual constam a justificativa para escolha do tema, os objetivos do trabalho, metodologia de pesquisa utilizada para atingir os objetivos e finalmente a estrutura da dissertação.

No segundo capítulo, são abordados algoritmos de busca de caminhos mínimos e fundamentos teóricos indispensáveis para posteriormente dar continuidade à revisão literária.

No terceiro capítulo, é apresentada uma revisão da literatura sobre localização de p-medianas, desde os estudos iniciais até os mais recentes, descrevendo as contribuições mais relevantes.

No quarto capítulo, aborda-se o modelo proposto, apresentando a técnica de busca baseada em algoritmos genéticos proposta neste trabalho. Ainda neste capitulo, efetua-se a análise comparativa e avaliação de resultados da heurística de busca proposta e o método de substituição de vértices proposto por Teitz e Bart., estabelecendo as vantagens de um método em relação ao outro, assim como as limitações e restrições de ambos.

Finalmente no quinto capítulo, são apresentadas as conclusões do trabalho e as recomendações para trabalhos futuros.

CAPÍTULO II

2. FUNDAMENTOS TEÓRICOS

Com o intuito de fornecer conceitos básicos, visando ao entendimento do problema tratado no presente trabalho, serão desenvolvidos os conceitos de sustentação mais relevantes, e apresentados os principais conceitos de grafos, por ser esta a abordagem mais usada para representar uma malha viária em se tratando de problemas de transporte.

Também serão abordados alguns dos métodos empregados para representar grafos na memória do computador, assim como os métodos mais usados de busca de caminhos mínimos em grafos. Visando a classificação do problema de localização de medianas, serão analisados conceitos gerais sobre os problemas de otimização combinatorial e complexidade computacional de um algoritmo. Far-se-á ainda referência aos principais tipos de métodos heurísticos e, por ultimo, a apresentação do conceito de métrica euclidiana a ser aplicado na implementação do algoritmo genético a ser proposto.

2.1. Definições Gerais Sobre Teoria de Grafos

2.1.1. Grafo

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

2.1.2. Arcos e Arestas

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

2.1.3. 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 contém ambos (arcos e arestas).

Exemplos são apresentados nas figuras 2.1. a 2.3.

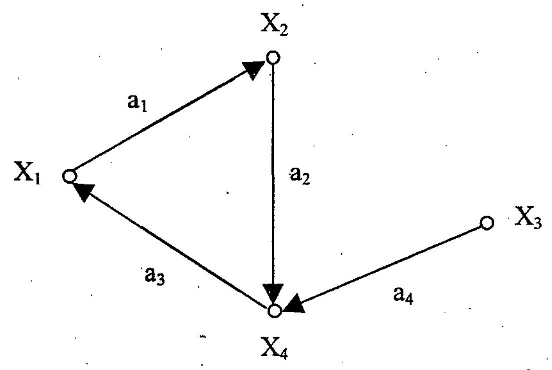


Figura 2.1.: Grafo Orientado

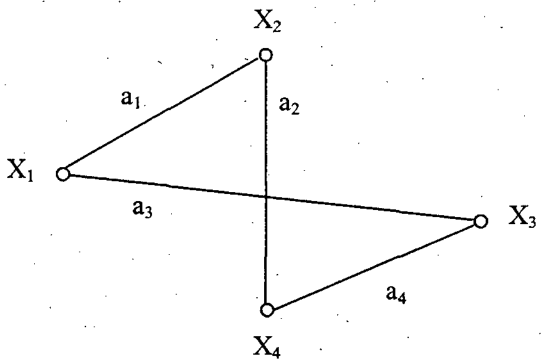


Figura 2.2.: Grafo não Orientado

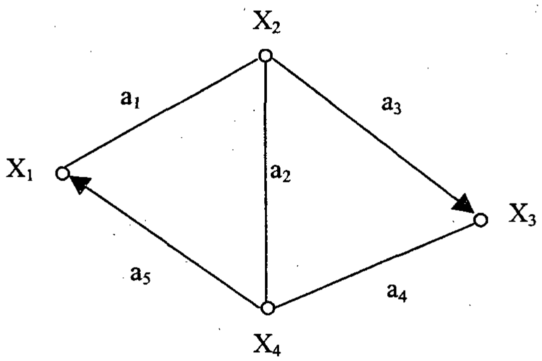


Figura 2.3.: Grafo Misto

Nos grafos orientados, quando se faz referência a um arco, por exemplo o arco a_2 da figura 2.1. , 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 2.2. uma referência à aresta a_1 implica, indiretamente, tanto o arco (x_1, x_2), como o arco (x_2, x_1), ou ambos.

2.1.4. Laço

Laço é um arco cujo vértice inicial coincide com o vértice final (Ver figura 2.4.).

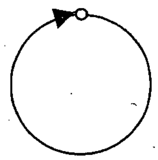


Figura 2.4.: Laço

2.1.5. 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 contém laços.

2.1.6. 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 2.1. os vértices x_1 e x_2 são adjacentes, o que não ocorre em relação a x_1 e x_3.

2.1.7. Grau de um Vértice

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

2.1.8. Grau de Entrada e de Saída de um Vértice

Em um grafo orientado, define-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.

2.1.9. 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. Analogamente, 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 2.1., a seqüência de arcos a_4, a_3, a_1 determina um caminho, e na Figura 2.2. a seqüência de arestas a_1, a_2, a_4 determina uma cadeia.

2.1.10. 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 2.2., a seqüência (a_1, a_2, a_4, a_3) constitui um ciclo.

2.1.11. 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_i a x_j., e outro 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: ou de x_i para x_j ou de x_j para x_i. Exemplos são apresentados nas figuras 2.6. e 2.7.

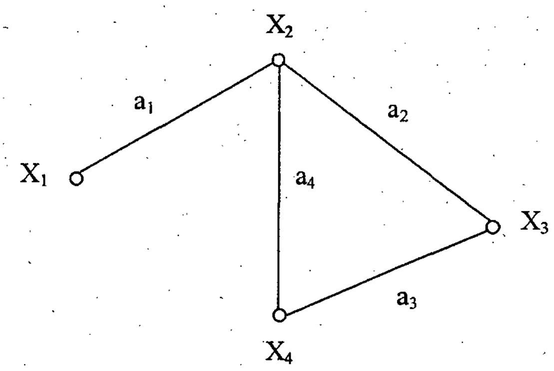


Figura 2.5. : Grafo não Orientado Conexo

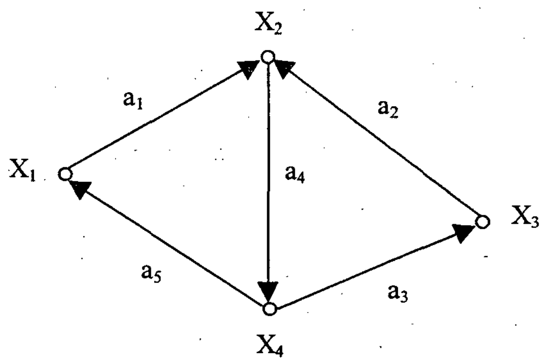


Figura 2.6. : Grafo Orientado Fortemente Conexo

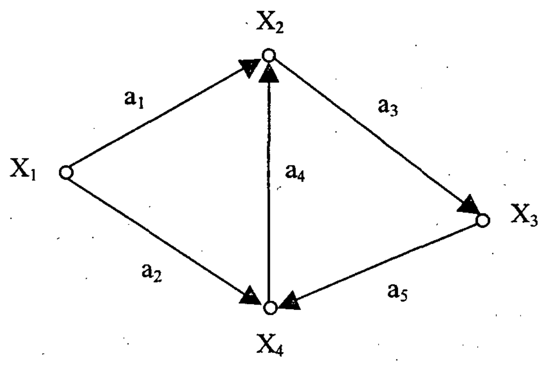


Figura 2.7. : Grafo Orientado unilateralmente Conexo

2.1.12. Árvore

São inúmeros as definições e notações empregadas por diferentes autores a respeito do conceito de árvore, aqui optou-se 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 aresta for retirada, a conexidade fica interrompida;

v) Um grafo acíclico e conexo, porém, se dois vértices quaisquer, não adjacentes, forem ligados por uma aresta, então o grafo passara a ter exatamente um ciclo;

vi) Um grafo conexo que não possui subgrafo (ver definição em 2.1.15.) 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.

2.1.13. 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).

2.1.14. Grafo Parcial

Um grafo parcial de $G = (X, A)$, é o grafo $G = (X, A')$, onde $A' \subset A$. (Ver figuras 2.8. e 2.9.)

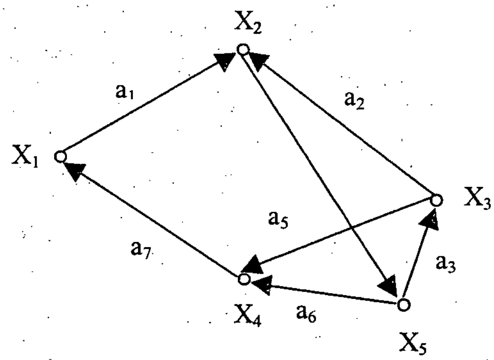


Figura 2.8. : Grafo Proposto

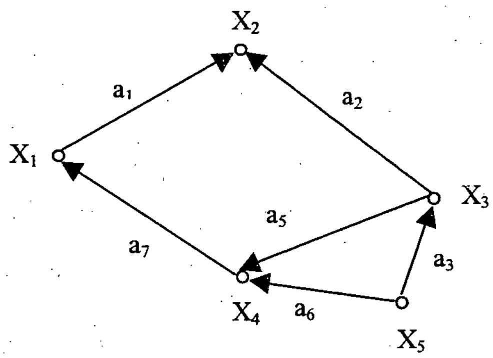


Figura 2.9. : Grafo Parcial

Como se pode observar o grafo da figura 2.9. é um grafo parcial do grafo apresentado na Figura 2.8.

2.1.15. 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 2.10. apresentado a seguir é um sub-grafo do grafo da Figura 2.8.

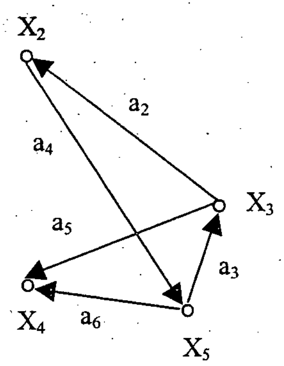


Figura 2.10. : Sub-Grafo

2.1.16. 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 2.11. é um grafo parcial do sub-grafo apresentado na Figura 2.10.

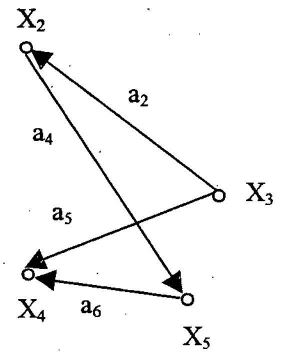


Figura 2.11. : Sub-Grafo Parcial

2.2. Representação de um Grafo no Computador

Um grafo pode ser representado na memória do computador de várias maneiras, até por que há a ocorrência de diversas estruturas que correspondem univocamente a um grafo dado e podem ser manipuladas sem dificuldade com os recursos computacionais existentes atualmente.

Estabelecida a existência destas representações, há necessidade de discorrer sobre as mais usadas: (a) representação pela matriz de custos, (b) representação por lista de arestas e (c) representação no display (graficamente) através de coordenadas cartesianas.

2.2.1. Representação pela Matriz de Custos

A maneira mais simples e provavelmente a mais usada na representação de um grafo no computador é através da " Matriz de Custos " ou também chamada " Matriz de Distâncias " .

A matriz de custos C de um grafo G, com n nós, é uma matriz $C\,[\,c_{ij}\,]$ de ordem $n{\times}n$, onde cada c_{ij} é o custo (ou peso) do arco ou da aresta, ligando os nós x_i e x_j . Ocorrendo mais de uma ligação direta entre x_i e x_j considera-se a de menor custo. Se $x_i = x_j$ então define-se $c_{ij} = 0$. Caso não exista o arco ou a aresta, define-se $c_{ij} = \infty$, que na pratica é representado por um número muito grande.

A representação de grafos por meio de uma "Matriz de Custos", embora simples, apresenta a desvantagem de requerer muita capacidade de memória em se tratando de grafos de grande porte. Dependendo do caso, isto pode não representar um problema sério, pois atualmente dispõe-se de equipamentos muito sofisticados.

Para representar uma matriz de n vértices usando este método, serão necessarias n^2 informações binárias. Se o grafo for não orientado a matriz de custo C será sempre simétrica, bastando portanto só considerar a matriz diagonal superior.

2.2.2. Representação por Lista de Arestas

Uma representação bastante eficiente, principalmente no caso de grafos de grande porte, é através de lista de arestas. Com este fim poderiam ser utilizados 3 vetores, $A\,(\,a_1,\,a_2,\dots,\,a_m\,)$, $B\,(\,b_1,\,b_2,\dots,\,b_m)$ e $C\,(\,c_1,\,c_2,\dots,\,c_m\,)$, onde no vetor A estão os inícios dos arcos, em B as extremidades dos mesmos e em C seus respectivos custos. O espaço de memória requerido neste caso é de 3m informações binárias, em contraste às n^2 da representação de grafos pela " Matriz de Custo". Conseqüentemente, constitui uma maneira mais eficiente de se representar grafos.

2.2.3. Representação no Display (Graficamente)

Nos problemas da vida real são comuns situações em que, em uma determinada área ou zona geográfica, existem pontos de interseção, sejam de vias rodoviárias, ferroviárias, redes de artérias urbanas, etc., onde cada interseção é representada por um vértice de um grafo. A cada vértice pode estar associado ainda uma certa atividade – por exemplo, um centro de abastecimento (depósito). Estes tipos de problemas se vêm bastante favorecidos pela representação do grafo no display (graficamente), tornando o problema de fácil visualização.

Esta representação usa o sistema de coordenadas cartesianas para representar os vértices. Sendo assim pode-se definir a localização de um determinado vértice do grafo por meio de coordenadas X e Y. A representação do grafo no display não dispensa o uso da representação por meio da matriz de custos ou por lista de arestas, estruturas usadas com o intuito de efetuar as operações necessárias à solução de um determinado problema.

2.3. Algoritmos de Busca de Caminhos Mínimos em Grafos

Encontra-se na literatura diversos algoritmos para a obtenção de caminhos mínimos entre pares de vértices de um grafo. Entre os mais conhecidos se destacam o algoritmo de Dijkstra e o algoritmo de Floyd. Além destes algoritmos, é possível usar o método da distância euclidiana para encontrar a menor distância entre dois vértices, que será em realidade a menor distância efetiva. Ambos algoritmos e o método da distancia euclidiana são apresentados a seguir.

2.3.1. Algoritmo de Dijkstra

O algoritmo de Dijkstra foi desenvolvido para determinar o caminho mínimo entre dois vértices. Seja um Grafo *G (V, E)* e uma função distância *L* que associe cada aresta *(v, w)* a um número real não negativo *L (v,w)*e também um vértice fixo v_0 em *V*, chamado fonte. O problema consiste em se determinar os caminhos de v_o para cada vértice *v* de *G*, de tal forma que a somatória das distâncias das arestas envolvidas a cada caminho seja mínima. Isto é equivalente a determinar um caminho $v_0, v_1, \ldots, v_k$ tal que $\sum_{i=0}^{k-1} L(v_i, v_{i+1})$ seja mínimo.

A função *L* que associa cada aresta a um número real não negativo é descrita a seguir:

$$L\,(v_i, v_j) = \begin{cases} \infty & \text{, se não existe a aresta } (v_i, v_j) \\ 0 & \text{, se } v_i = v_j \\ \text{Custo,} & \text{se } v_i \neq v_j \text{ e existe a aresta } (v_i, v_j) \end{cases}$$

Constrói-se um conjunto *S*, que contém os vértices v_i's cujo comprimento mínimo de v_0 a cada v_i seja conhecido. A cada passo se adiciona ao conjunto *S* o vértice *w* pertencente a *V-S* tal que o comprimento do caminho v_o a *w*, seja menor do que o correspondente de qualquer outro vértice de *V-S*. Pode-se garantir que o caminho mínimo de v_0 a qualquer vértice *v* em *S* contém somente vértices pertencentes a *S*.

O algoritmo de Dijkstra basicamente consiste em expandir nós (gerar seus sucessores) começando pelo nó inicial, selecionando sempre aquele que ainda não foi escolhido e que tiver o menor custo acumulado desde a origem.

A seguir é apresentado o Algoritmo de Dijkstra de acordo com RABUSKE (1992).

INICIO

$S \leftarrow \{ V_0 \};$

$D[V_0] \leftarrow 0;$

Para cada v de $V- \{ V_0 \}$ faça $D[v] \leftarrow L(v_0,v);$

Enquanto $S \neq V$ faça

Escolha o vértice $w \in V\text{-}S$ tal que $D[w]$ seja mínimo;

Coloque w em S, isto é, faça $S \leftarrow S \cup \{w\}$;

Para cada $v \in V\text{-}S$ faça

$D[v] \leftarrow Min\,(D[v], D[w] + L(w,v))$

FIM.

Este algoritmo termina ao atingir um nó terminal ou quando não existir nós para serem expandidos.

2.3.2. Algoritmo de Floyd

É um algoritmo matricial e, diferentemente do algoritmo de Dijkstra aceita valores negativos para as arestas. Segundo RABUSKE (1992), o algoritmo de Floyd baseia-se na construção de uma matriz D^0 de custos de arestas, onde os laços possuem custo zero e a não existência de arestas atribui-se o custo infinito.

O algoritmo constrói sucessivamente, n matrizes a partir de D^0 , através de modificações efetuadas de acordo com a seguinte expressão

$$d_{ij}^{k} = Min\,[d_{ij}^{k-1},(d_{ik}^{k-1} + d_{kj}^{k-1})] \qquad (2.1)$$

Para a determinação do caminho, parte-se do final para o inicio, levando-se em conta os vértices intermediários incluídos durante o processo.

Passos do Algoritmo de Floyd

Montar a matriz de custo, onde $d_{ii} = 0$ para todo $i=1,2,...,n$ e que $d_{ij} = \infty$, quando não existe a aresta (x_i, x_j).

Passo 1 Faça $k \leftarrow 0$;

Passo 2 Faça $k \leftarrow k+1$;

Passo 3 Para todo $i \neq k$ tal que $d_{ik} \neq \infty$ e todo $j \neq k$ tal que $d_{kj} \neq \infty$

faça $d_{ij}^{k} = Min\,[d_{ij}^{k-1}, (d_{ik}^{k-1} + d_{kj}^{k-1})]$

Passo 4 [Teste de finalização]

a) Se algum $d_{ii} < 0$, então existe um ciclo de custo negativo contendo o vértice x_i, e não existe solução possível. Pare.

b) Se todo $d_{ii} \geq 0$, e $k = n$, a solução foi achada, e $[d_{ij}]$ fornece os comprimentos de todos os menores caminhos . Pare.

c) Se todo $d_{ii} \geq 0$ mas $k<n$, então retorne a Passo 2.

Este algoritmo tem complexidade da ordem n^3.

2.3.3. Método que Usa a Distância Euclidiana

O método da distância euclidiana consiste em encontrar a distância em linha reta de um ponto ao outro, supondo que os vértices estão unidos por cadeias, o qual significa que é desconsiderada a orientação do grafo. É importante ressaltar que com a aplicação do método de distância euclidiana, torna-se desnecessário usar algoritmos de distância mínima entre dois pontos.

Levando em consideração que, no presente trabalho, foi escolhida a abordagem de distância euclidiana para encontrar a distância mínima entre dois vértices, torna-se necessário buscar o aprofundamento sobre alguns conceitos importantes associados à métrica euclidiana.

2.3.3.1. Métrica Euclidiana

2.3.3.1.1. Redes de transporte

Os sistemas de transporte operam ao longo de vias ou rotas específicas que interligadas formam uma rede, é o caso do transporte urbano, transporte rodoviário, rede ferroviária, e até mesmo o caso do transporte aéreo.

A rede de transporte real representada por interseções de ruas, avenidas ou rodovias, podem ser interpretadas como sendo os nós da rede e algum serviço ou atividade ligando um nó ao outro poderia ser visto como o arco da rede. Uma rede de transportes é formada, então por um conjunto finito de nós e de arcos.

Há situações em que o principal interesse é gerar soluções sem muitos detalhes, mesmo sendo soluções aproximadas. Isso acontece nas fases iniciais de planejamento ou quando se desejam respostas sobre alternativas diversas. Nesses casos é comum lançar mão de uma simplificação. Em lugar de representar todos os nós e arcos da rede, imagina-se que a distância correspondente a uma ligação qualquer entre dois pontos possa ser estimada através da distância geométrica (distância euclidiana) entre esses pontos.

Posteriormente, poderão ser aplicadas técnicas de ajustes de distâncias, baseadas no incremento de um coeficiente de correção apropriado, para assim estimar o valor da distância verdadeira entre esses pontos. Situações desse tipo são bastante comuns em problemas logísticos.

2.3.3.1.2. Distância Euclidiana

Considere dois pontos A e B situados sobre uma rede de transporte. Define-se, a seguir, um sistema de coordenadas cartesianas, arbitrário, com origem num ponto 0 qualquer mostrado na figura 2.12. Ligando os pontos A e B existem vários caminhos, com distâncias quase sempre diferentes. A menor distância possível entre os pontos A e B corresponde à ligação em linha direta, representada pelo segmento tracejado na figura 2.12.

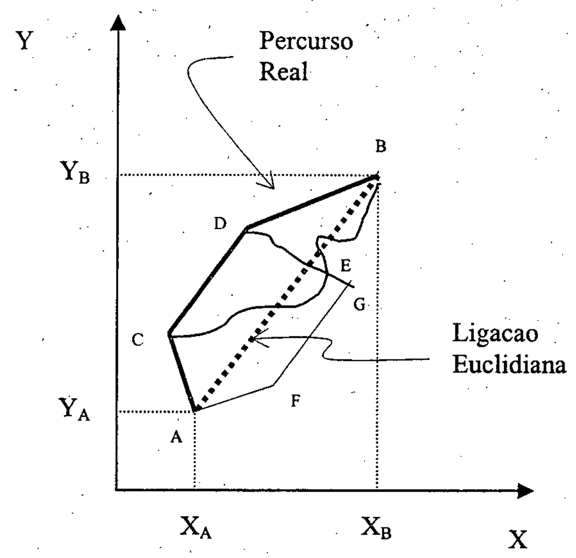


Figura 2.12. : Distancia Euclidiana entre os Pontos A e B. **Fonte: Novaes (1989)**

A distância em linha reta, que é denominada distância euclidiana, constitui, na maioria das aplicações reais de transportes, uma abstração útil para os cálculos e estruturação dos modelos. A razão reside na sua simplicidade de representação analítica e na sua característica de unicidade (isto é, há somente uma ligação euclidiana entre dois pontos). Posteriormente através de coeficientes corretivos médios pode-se ajustar matematicamente a distâncias euclidiana à distância efetiva, possibilitando assim o tratamento mais realista das aplicações.

Considerando as coordenadas X_A , Y_A do ponto A e X_B , Y_B do ponto B, a distância euclidiana entre A e B, na figura 2.12. é dada pela expressão:

$$DE_{AB} = [(X_B - X_A)^2 + (Y_B - Y_A)^2]^{1/2} \qquad (2.2)$$

Seja D_{AB} a distância efetiva entre A e B, podendo corresponder ao caminho viário mais curto entre dois pontos, ou o percurso mais utilizado, etc. dependendo do caso real de aplicação.

Uma vez que a distância euclidiana é a mínima distância entre os dois pontos, concluisse que :

$D_{AB} \geq DE_{AB}$ (2.3)

É comum se calibrar, através de regressão simples, uma relação linear entre D_{AB} e DE_{AB}, do tipo:

$D_{AB} = a + b \cdot DE_{AB}$ (2.4)

2.3.3.1.3. Correspondência entre Distâncias

As redes de transporte não apresentam um padrão geométrico rigoroso. De acordo com Novaes (vide [NOV89]) a distância euclidiana pode ser ajustada à distância efetiva por meio de regressão simples, acrescentando um coeficiente de ajuste. O autor efetuou um estudo, considerando como amostra 110 estradas pavimentadas no estado de São Paulo e determinou as seguintes relações entre distância efetiva e distância euclidiana medida em quilômetros:

$D = 23{,}9 + 1{,}11 \cdot DE$ (valida para $DE \geq 60$ Km)

$D = 1{,}48 \cdot DE$ (valida para $DE < 60$ Km)

Segundo NOVAES (1989), nos estudos ligados a malhas urbanas, é comum se adotar um acréscimo médio de 30% sobre a distância euclidiana, para se obter uma primeira aproximação para a distância efetiva entre dois pontos.

NOVAES (1989) afirma que essa correção, ainda que aproximada, parece representar bem as situações vigentes na maioria das cidades.

2.4. Problemas de Otimização Combinatórial

A otimização combinatorial é um dos campos da programação matemática, cuja aplicação é encontrada numa ampla gama de áreas, tais como, engenharia, economia, ciências sociais e outras. Problemas combinatoriais são freqüentemente encontrados na elaboração de projetos de sistemas de distribuição de energia elétrica, posicionamento de satélites, projetos de computadores e de chips VLSI, roteamento ou programação de veículos, alocação de trabalhadores ou máquinas a tarefas, **problemas de localização**, empacotamento de caixas em containers, corte de barras e placas, seqüenciamento de genes e DNA, classificação de plantas e animais, etc.

De acordo com IBARAKI (1987), os problemas combinatoriais podem ser modelados como problemas de maximização (ou minimização) da função objetivo definida, cujas variáveis devem obedecer (estar sujeitas) a certas restrições.

WAKABAYASHI (1996) afirma que encontrar soluções ótimas, ou mesmo aproximadas, para esses tipos de problemas é um desafio nem sempre fácil de ser vencido. Para alguns desses problemas são conhecidos métodos eficientes (rápidos) para resolvê-los; para outros, métodos de enumeração implícita, relaxação, métodos de planos-de-corte (nem sempre tão rápidos) são alguns dos aplicados com maior sucesso na solução de problemas reais.

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, tornando os algoritmos intratáveis.

Outro fator interessante de ser analisado é o espaço de memória requerido. Ambos fatores – tempo e memória requerida - são os responsáveis pela complexidade computacional do algoritmo.

2.4.1. Complexidade Computacional de um Algoritmo

Um problema é denominado computável se existe um procedimento que o resolva em um número finito de passos, ou seja, se existe um algoritmo que leve à sua solução. Conforme WESLLEY (1999) um problema considerado "em princípio" computável pode não ser tratável na prática, devido às limitações dos recursos computacionais para executar o algoritmo implementado.

É importante caracterizar quais os recursos que irão ser despendidos na resolução do algoritmo, isto é o tempo que demorará a resolver, assim como a quantidade de memória necessária para tomar a resolução possível. Estas duas variáveis (tempo e espaço) constituem a medida do desempenho de um algoritmo. Ambas deverão ser apresentadas como função de uma métrica do problema, que representará de alguma forma a dimensão do problema que se pretende solucionar com o algoritmo (TING, 1997).

Com o objetivo de distinguir os problemas de fácil e difícil solução classificam-se os problemas de otimização combinatorial em Polinomiais (P) e os Não Polinomiais (NP), que por sua vez subdividem-se em problemas NP-Hard e problemas NP-completo.

2.4.1.1. Dimensão do Problema

A dimensão de um problema é o tamanho da entrada do algoritmo que resolve o problema.

1. A busca em uma lista de N elementos ou a ordenação de uma lista de N elementos requerem mais operações à medida que N cresce;

2. O cálculo do fatorial de N tem o seu número de operações aumentado com o aumento de N;

3. A determinação do valor de F_N na seqüência de Fibonacci F_0, F_1, F_2, F_3, ... envolve uma quantidade de adições proporcional ao valor de *N*; e

4. Determinar se um grafo tem um ciclo hamiltoniano (um ciclo que usa cada vértice do grafo) fica mais complicada, à medida que o número de vértices, *N*, cresce.

É dito que o tamanho destes problemas é *N*. Tipicamente utiliza-se um número inteiro positivo, de forma que as funções, *f(N)* que exprimem a complexidade são funções de números inteiros positivos, ou seja, f: $Z+ \rightarrow R$.

2.4.1.2. Importância da Análise de Complexidade Computacional de um Algoritmo

Como foi dito anteriormente a complexidade computacional de um algoritmo diz respeito aos recursos computacionais - espaço de memória e tempo de máquina - requeridos para solucionar um problema.

Geralmente existe mais de um algoritmo para resolver um problema. A análise de complexidade computacional é, portanto, fundamental no processo de definição de algoritmos mais eficientes para a sua solução. Apesar de parecer contraditório, com o aumento da velocidade dos computadores, torna-se cada vez mais importante desenvolver algoritmos mais eficientes, devido ao aumento constante do "tamanho" dos problemas a serem resolvidos.

De acordo com SIANG (1997), a importância da complexidade pode ser observada pelo seguinte exemplo que mostra 5 algoritmos *A, B, C, D e E* de complexidades diferentes para resolver um mesmo problema. Supõe-se que uma operação leva 1 ms para ser efetuada. A tabela seguinte dá o tempo (T_1 , T_2 , T_3 , T_4 , T_5) necessário por cada um dos algoritmos. (Supondo logaritmos na base 2.)

	A	B	C	D	E
n	$T_1(n) = n$	$T_2(n) = n\log n$	$T_3(n) = n^2$	$T_4(n) = n^3$	$T_5(n) = 2^n$
16	0,016 s	0,064 s	0,256 s	4 s	1 m 4 s
32	0,032 s	0,16 s	1 s	33 s	46 dias
512	0,512 s	9 s	4 m 22 s	1 dia 13 h	10^{137} séculos

Tabela 2.1. : Tabela de Complexidade.**Fonte Siang (1997).**

Ainda de acordo com SIANG (1997), pode-se muitas vezes melhorar o tempo de execução de um programa (que implementa um algoritmo dado) fazendo otimizações locais (e.g. usar *x+x* ao invés de *2x*, em máquinas em que a multiplicação é mais demorada que a adição).

Entretanto, melhorias substanciais são muitas vezes obtidas se é usado um algoritmo diferente, com outra complexidade de tempo, por exemplo obter um algoritmo de O (n log n) ao invés de $O(n^2)$.

Os intervalos de tempo e memória, exigidos pelas heurísticas, podem ser medidas de diferentes maneiras. Uma parte significativa da literatura está dedicada a estudar a complexidade do pior caso, isto é, obter um limite superior para o tempo e espaço (memória) necessários para obter uma solução para o problema. Mesmo sendo uma medida importante, freqüentemente, existe uma grande diferença entre o tempo e espaço do pior caso e a sua média. Outra medida da eficiência de uma heurística baseia-se na qualidade de suas soluções, que pode ser obtida através da comparação com algum limite (inferior ou superior) ou outro método (heurístico ou exato).

Na análise de complexidade de algoritmos é importante a ordem de grandeza das funções. Ao invés de computarmos a função exata do número de operações necessário, f(N), é mais fácil e freqüentemente válido trabalhar com a sua ordem de grandeza. É claro que para uma mesma função f(N) pode existir uma infinidade de funções limitantes superiores; mas o que se procura é a menor função limitante superior para caracterizar a sua ordem de grandeza da complexidade.

A seguir algumas das funções limitantes superiores (Cota superior ou "upper bound") mais conhecidas.

2.4.1.2.1. Funções Limitantes Superiores

Complexidade	*Designação*
O(1)	Constante
O(log n)	Logarítmica
O(n)	Linear
O(n log n)	nlogn
$O(n^2)$	Quadrática
$O(n^3)$	Cúbica
O(n^c), c número real	Polinomial
O(c^n), c número real, maior que 1	Exponencial
O(n!)	Fatorial

Tabela 2.2. : Funções Limitantes Superiores. **Fonte: Ting (1997).**

Seja dado um problema, por exemplo, multiplicação de duas matrizes quadradas de ordem n. É conhecido um algoritmo para resolver este problema (pelo método trivial) de complexidade $O(n^3)$. Sabe-se assim que a complexidade deste problema não deve superar $O(n^3)$, uma vez que existe um algoritmo desta complexidade que o resolve. Uma cota superior (ou "upper bound") deste problema é $O(n^3)$.

A cota superior de um problema pode mudar se alguém descobrir um outro algoritmo melhor. Isso de fato aconteceu com o algoritmo de Strassen que é de $O(n^{\log 7})$. Assim a cota superior do problema de multiplicação de matrizes passou a ser $O(n^{\log 7})$. Outros pesquisadores melhoraram ainda este resultado. Segundo SIANG (1997) atualmente o melhor resultado é o de Coppersmith e Winograd – de $O(n^{2.376})$.

2.4.1.3. Problema Polinomial (P)

A classe de problemas cuja complexidade é polinomial denomina-se P. De acordo com PINTO (1999), são denominados problemas tratáveis, pois é viável construir para estes um algoritmo que encontre uma solução em tempo polinomial, em contraposição a um programa que gere todas as soluções e determine qual a válida (este teria complexidade exponencial).

O problema polinomial tem solução de complexidade até ordem polinomial. Para entender melhor a ordem de grandeza de uma função de complexidade de um problema, será introduzida a seguinte definição:

Sejam $f, g : Z+ \rightarrow R$. Diz-se que g domina f (ou f é dominada por g) se existe m pertencente a $R+$ e k a $Z+$, tais que $|f(N)| <= m|g(N)|$ para todo N pertencente a $Z+$, onde $N >= k$.

Em outras palavras, $f(N)$ é dita dominada por $g(N)$, se o limite de $|f(N)|/|g(N)|$ com N tendendo a infinito é igual a um valor real m. Se f é dominada por g, diz-se ainda que f é da ordem de grandeza g, ou seja f pertence a $O(g)$.

A ordem de grandeza das funções é importante na análise de complexidade de algoritmos.

Um algoritmo com complexidade polinomial é aquele que tem a sua função de complexidade, $f(N)$, limitada por uma função $g(N)$ de ordem polinomial (por exemplo, $g(N) = N^3$).

2.4.1.4. Problema não-Deterministicamente Polinomial (NP)

Um problema não-deterministicamente polinomial (NP- non-deterministic polynomial time algorithm) é um problema computável cujas soluções até então conhecidas são de ordem exponencial e não se sabe se existe uma solução melhor, de complexidade polinomial. Alguns exemplos típicos de ordens associadas a problemas da classe NP são : $O(n^{\log n})$, $O(n!)$, $O(k^n)$, etc.

Observe-se que o termo "não-determinístico" não implica aleatoridade, mas apenas que não se pode afirmar a existência de um algoritmo de complexidade polinomial para o problema em consideração. Uma observação interessante é que a comprovação de uma solução correta de um problema NP tem complexidade da ordem polinomial.

Os problemas não deterministicamente polinomiais (NP), subdividem-se nas subclasses NP-hard e NP-completo. Segundo PINTO (1999), os problemas na classe NP-completo têm complexidade equivalente entre si (daí a classe), mas não foi demonstrada a não-existência de algoritmos polinomiais para os mesmos.

2.4.1.4.1. Problema NP-Hard

Um problema X é chamado um problema NP-Hard se todo problema NP é redutível polinomialmente a X.

A teoria de NP-complexidade demostra que problemas não deterministicos, são problemas supostamente intratáveis. Para isto define-se a classe dos problemas NP-hard como a classe dos problemas tais que se fossem polinomiais se verificaria $P = NP$. Novamente se utiliza a técnica de redutibilidade para demostrar que um problema é NP-hard, pois se um problema NP-completo é redutível a outro problema então o segundo é NP-hard. Aplicando esta técnica se encontram problemas de otimização que são NP-hard. Concretamente existem técnicas para demostrar que estes tipos de problemas são NP-hard, construindo problemas de decisão associados aos mesmos, e demostrando que estes são NP-completos.

2.4.1.4.2. Problema NP-Completo

Um problema *X* é chamado NP-Completo: se (1) *X* pertence a NP, e (2)*X* é NP-Hard. Uma vez encontrado um problema NP-Completo, pode-se provar que outros problemas também são NP-Completo.

Em outras palavras, um problema NP-completo é um problema "representante" de uma classe de problemas NP, de tal sorte que os problemas da classe são redutíveis a ele em tempo polinomial. Por exemplo, o problema do caixeiro-viajante é um problema redutível ao problema de ciclo hamiltoniano. É dito então que o problema de ciclo hamiltoniano é um problema NP-completo.

Foi demonstrado que, se um algoritmo de complexidade polinomial puder ser encontrado para qualquer um dos problemas NP-completos, então todos os problemas NP-completos serão na verdade problemas P. Por outro lado, se for provado que um deles requer um algoritmo de solução que apresente complexidade exponencial, então todos irão requerer complexidade exponencial.

Ullman citado por VACA (1995), conclui que os problemas NP-completos são os mais difíceis problemas NP.

2.5. Heurísticas

Nos últimos anos têm sido aplicadas múltiplas técnicas heurísticas na aproximação à resolução de problemas de otimização combinatórial NP-completos. Com particular destaque estão as que se relacionam com "Simulated Annealing", "Genetic Algorithms" e "Neural Networks". Porém, na maioria dos casos, tem existido uma

grande arbitrariedade na escolha dos parâmetros associados a este tipo de métodos heurísticos que, de um modo geral, variam de problema para problema sem qualquer critério.

As heurísticas de procura podem ser classificadas amplamente em heurísticas deterministicas , heurísticas randômicas e heurísticas de começo aleatório:

Heurísticas Deterministicas

Esta classe de heurísticas é caracterizada pela escolha de caminhos determinísticos de procura. Eles regularmente adotam uma estratégia de procura fixa baseado no conhecimento de domínio disponível. Muitas heurísticas de procura local são exemplos deste categoria.

Heurísticas Randômicas

Esta categoria de heurísticas emprega operadores escolhidos aleatoriamente na estratégia de busca e não são muito dependentes no domínio de conhecimento. Execuções sucessivas destas heurísticas não precisam necessariamente gerar a mesma solução. Simulated Annealing e algoritmos genéticos são exemplos desta classe de heurística.

Heurísticas de Começo Aleatório

Estas heurísticas são caracterizadas por uma escolha fortuita da solução inicial que é então melhorada iterativamente. A maioria das heurísticas de melhoria iterativa corresponde a esta categoria.

2.7. Considerações Finais

Após um primeiro contato com todos os conceitos abordados neste capítulo, será apresentada uma revisão sobre o problema de localização de instalações, comentando brevemente o problema de localização de centros e dedicando especial atenção ao problema de localização de medianas. Sobre este último serão abordados os diferentes métodos de resolução desde as primeiras pesquisas até as mais recentes, descrevendo as contribuições mais relevantes.

CAPÍTULO III

3. PROBLEMAS DE LOCALIZAÇÃO EM GRAFOS

3.1. Localização de Instalações

A necessidade de encontrar uma localização ótima para uma instalação (serviços) em um grafo - ou rede viária como freqüentemente é referido na literatura - torna o problema de localização um assunto de amplo interesse em muitas situações práticas.

De modo geral se um grafo representa uma rede viária com seus vértices representando comunidades, pode-se ter o interesse de localizar um hospital, uma estação de bombeiros, etc. Do mesmo modo, se a rede representa uma cidade, pode-se estar interessado em encontrar as localizações ótima para escolas, postos de coleta de cartas, cabinas de telefones, etc. Todos estes problemas são tratados na literatura como problemas de localização de instalações, cujo objetivo é encontrar a melhor alternativa para a localização de um determinado serviço ou recurso. As definições básicas sobre grafos podem ser encontradas no capitulo 2, seção 2.1.1.

De acordo com IBARAKI (1987), os problemas de localização como é o caso dos problemas dos p-Centros e p-Mediana são catalogados como problemas de otimização combinatorial (ver seção 2.4.).

Na literatura se encontram dois tipos de critérios de otimização para problemas de localização de instalações em um grafo. A seguir serão apresentadas as duas vertentes. A primeira se voltou para a pesquisa de **"Localizações de Centros"** (seção 3.2.) e a segunda vertente se preocupou com a **"Localização de Medianas"**, apresentada em detalhe na seção 3.3.

Conforme foi mencionado no capítulo introdutório, o presente trabalho de pesquisa concentra-se na abordagem de localização de medianas.

3.2. Noções Gerais sobre Localização de Centros

O problema de **"Localização de Centros"** consiste em encontrar lugares adequados para instalações de modo a minimizar o máximo custo (por exemplo a máxima distância) de qualquer vértice à instalações mais próxima. Neste tipo de problema estão incluídos a localização de serviços de emergências.

A localização ótima da instalação que satisfaz a condição acima é chamada de centro do grafo. Existem situações nas quais tem-se o interesse de encontrar mais de um centro, e nesse caso trata-se dos problemas de p-centros, conhecidos na literatura como **"Problemas de Mini-max"** ou **"Problema de Localização de p-Centros "**.

A fórmula matemática que descreve este problema é apresentada a seguir:

$$\underset{x_i \in X}{Min}\ [S_{ot}(x_i)] \qquad (3.1)$$

onde, $$S_{ot}(x_i) = \underset{x_j \in X}{Max}\ \{v_j[d(x_i, x_j) + d(x_j, x_i)]\} \qquad (3.2)$$

Vários algoritmos e procedimentos heurísticos foram desenvolvidos para resolver estes tipos de problema. Segundo SOUZA (1996), uma das heurísticas que apresenta melhores resultados e de mais fácil aplicação é a heurística proposta por Maranzana, que consiste em localizar aleatoriamente os p locais candidatos a centros, a seguir são alocados os vértices a sua instalação mais próxima e em seguida é calculada a função objetivo, posteriormente é avaliado se alguma mudança de posicionamento do centro produz algum ganho na função objetivo, em caso afirmativo realizar a troca de modo a obter ganho, caso contrario manter a solução, por último apresentar a solução que apresente um melhor ganho na função objetivo.

Enquanto que no problema de p-Centros o objetivo é minimizar o pior caso (por exemplo a máxima distância), no caso das p-Medianas o objetivo é minimizar o custo total.

3.3. Localização de Medianas

3.3.1. Definições Gerais

Em alguns problemas de localização o objetivo é minimizar a soma total das distâncias dos vértices do grafo para a instalação central (assumindo que somente uma instalação será localizada). Tais objetivos são, melhor adaptados ao problema de localizar um depósito em uma rede viária, onde os vértices representam os clientes a serem providos pelo depósito. Situações deste tipo geralmente são chamadas de **"Problemas de Localização de Medianas"** ou **"Problemas de Localização de Mini-sum"**, embora a função objetivo não seja em geral simplesmente a soma de distâncias, mas sim a soma de várias funções de distância. A localização da instalação resultante da solução do Problema Mini-sum é chamada de mediana do grafo.

Existem situações nas quais pode-se estar interessado em encontrar mais de uma mediana. Neste caso o interesse passa a ser encontrar as p-Medianas da rede, com o objetivo de localizar instalações em uma rede de modo a minimizar a somatória de todas as distâncias de cada ponto de demanda a sua instalação mais próxima. Este critério de otimização tem o enfoque voltado à resolução de problemas cujo objetivo é minimizar a soma dos custos de transporte associados.

Encontrar as p-Medianas de um grafo é o problema central em uma classe geral conhecida na literatura como " localização de instalações ou recursos " ou " localização de depósitos ".

Segundo MINIEKA (1977), a localização de medianas contempla a possibilidade da instalação estar localizada em um vértice e/ou sobre uma aresta. Sendo assim torna-se necessário distinguir dois conceitos: se as instalações devem ser localizadas somente nos vértices, as localizações são chamadas *Medianas*; se as instalações podem ser localizadas sobre as arestas e nos vértices, as localizações são chamadas de *Medianas Absolutas*. Usando **p** para denotar o número de instalações a serem localizadas, tem-se assim, respectivamente, o Problema da Determinação das p-Medianas e o Problema da Determinação das p-Medianas Absolutas.

Segundo Hakimi citado por GALVÃO (1979), a localização de medianas é um problema computacionalmente complexo. O autor afirma que para uma rede de *n* vértices a complexidade é da ordem de $O(\frac{n^p}{p!})$, classificando-o como um problema NP-completo (ver seção 2.4.1.4.2.).

O problema de p-Medianas assim como muitos problemas de otimização combinatória (Ver Seção 2.4.) é afetado pela " explosão combinatorial ". Por exemplo, no caso de uma rede com 88 vértices , há mais de 39 milhões de combinações em se tratando de localizar 5 instalações, e acima de $4{,}5 \times 10^{12}$ combinações para 10 instalações. Supondo que um computador pudesse avaliar um milhão de combinações por segundo, ainda exigiria mais de um mês e meio para avaliar todas as combinações, e assim encontrar a melhor delas. Em essência, um método que dispenda tanto tempo de processamento seria claramente descartado de uma solução viável.

No problema de p-Medianas cabe ainda distinguir a diferenciação entre os problemas de p-Medianas "Puro" e o "Generalizado". No primeiro os custos de localização das instalações são desconsiderados, enquanto que no segundo os custos associados com a localização das instalações fazem parte da função objetivo.

3.3.2. Aplicações onde se torna necessária a Alocação de Medianas

Na pratica o problema freqüentemente aparece em uma variedade de formas:

- Na localização de centros de comutação em redes telefônicas;
- Subestações de redes de energia elétrica;
- Centros de abastecimento (deposito) em uma rede de distribuição rodoviária (onde os vértices representam os clientes);
- Na localização de centros comerciais, lojas, shoppings;
- Localização de provedores de internet;
- Na localização de posto de coleta de cartas;
- Localização de estações de satélites;
- Na localização de escolas em uma cidade de maneira a minimizar a distancia percorrida pelos alunos de cada bairro.

E assim como estas, muitas outras aplicações de localização onde se deseje minimizar uma soma ou uma média.

O problema de encontrar a p-Mediana pode ser feito ligeiramente mais geral associando com cada vértice x_j um peso v_j (representando, tamanho ou importância), de forma que o objetivo é minimizar a soma das distâncias ponderados pelos pesos.

No presente trabalho é abordado o problema de localização de instalações, especificamente, o Problema de p-Medianas puro; isto é o problema de localizar um determinado número (p) ótimo de instalações ao longo de uma rede viária desconsiderando os custos de localização das instalações.

3.3.3. Conceitos Básicos do Problema de Medianas Propriamente Dito

3.3. 3.1. Notação e Definições Elementares

$G = (X, A)$	- grafo;
x_i ou x_j	- vértices, i = 1, 2,...,n; j = 1, 2,...,n;
v_j	- peso (ou importância) associado ao vértice x_j;
$d(x_i, x_j)$	- menor distância do vértice x_i ao vértice x_j;
$\sigma_o(x_i)$	- no. de out-transmissão do vértice x_i;
$\sigma_t(x_i)$	- no. de in-transmissão do vértice x_i;
$\sigma_{o,t}(x_i)$	- no. de in-out-transmissão;
$\bar{x}_o$	- out-mediana;
$\bar{x}_t$	- in-mediana;
$\bar{x}_{o,t}$	- in-out-mediana;
X_p	- subconjunto de vértices do grafo G que contém p vértices;
$D = [\, d(x_i, x_j) \,]$	- matriz das menores distâncias;
$[\, \varepsilon_{ij} \,]$	- matriz dos alocados composta dos vértices alocados aos vértices-medianas.

3.3.3.2. Números de Transmissão de um Vértice: out-transmission, in-transmission e in-out-transmission

Dado o grafo $G = (X, A)$ para cada vértice $x_i \in X$ define-se:

Definição de out-transmission :

$$\sigma_o(x_i) = \sum_{x_j \in X} v_j d(x_i, x_j) \qquad (3.3)$$

O número de out-transmission $\sigma_o(x_i)$ é a soma das entradas da linha x_i de uma matriz obtida pela multiplicação de cada coluna j da matriz de distância $D(G) = [d(x_i, x_j)]$ por v_j. E $d(x_i, x_j)$ é a menor distância do vértice x_i para o vértice x_j.

Definição de in-transmission :

$$\sigma_t(x_i) = \sum_{x_j \in X} v_j d(x_j, x_i) \tag{3.4}$$

O número de in-transmission $\sigma_t(x_i)$ é a soma das entradas da coluna x_i de uma matriz obtida pela multiplicação de cada linha j da matriz de distância $D(G) = [d(x_i, x_j)]$ por v_j.

Definição de in-out-transmission :

O número de in-out-transmission $\sigma_{o,t}(x_i)$ do vértice $x_i \in X$ está definido por :

$$\sigma_{o,t}(x_i) = \sum_{x_j \in X} v_j [d(x_i, x_j) + d(x_j, x_i)] \tag{3.5}$$

3.3.3.3. Medianas do Grafo : out-mediana, in-mediana e in-out-mediana

Definição de out-mediana :

Um vértice $\bar{x}_o$ para o qual

$$\sigma_o(\bar{x}_o) = \min_{x_i \in X} [\sigma_o(x_i)] \tag{3.6}$$

é chamado de out-mediana do grafo G.

Definição de in-mediana :

Um vertice $\bar{x}_t$ para o qual:

$$\sigma_o(\bar{x}_t) = \min_{x_i \in X}[\sigma_t(x_i)] \tag{3.7}$$

é chamado de in-mediana de G.

Definição de in-out-mediana :

Um vertice $\bar{x}_{o,t}$,para o qual:

$$\sigma_{o,t}(\bar{x}_{o,t}) = \min_{x_i \in X}[\sigma_{o,t}(x_i)] \tag{3.8}$$

é chamado de in-out-mediana de G.

3.3.4. Múltiplas Medianas (p-Medianas)

O conceito de mediana de um grafo pode ser generalizado a mais que uma única mediana de forma que um conjunto de p pontos formem uma multi-mediana (p-medianas) como segue:

3.3.4.1. Definições de Distância:

Seja X_p um subconjunto do conjunto X de vértices do grafo $G = (X, A)$, onde X_p contém p vértices.

Distância de um Vértice a um Subconjunto de Vértices :

Neste tópico será apresentado o conceito de distância entre um vértice $x_j \in X$ e um conjunto de vértices (neste caso o subconjunto X_p). A dita distância é definida por :

$$d(x_j, X_p) = \min_{x_i \in X_p} [\, d\,(x_j\,, x_i)] \tag{3.9}$$

Distância de um Subconjunto de Vértices a um Vértice :

A distância entre o subconjunto X_p e o vertice x_j é dada por :

$$d(X_p, x_j) = \min_{x_i \in X_p} [\, d\,(x_i\,, x_j)] \tag{3.10}$$

3.3.4.2. Definições Números de Transmissão de um Subconjunto de Vertices: out-transmission, in-transmission

Definição de out-transmission de um Conjunto de Vértices :

Se $x_i^{'}$ é o vértice de X_p o qual produz o mínimo nas eqns. (3.9) ou (3.10) é dito que o vértice x_j é alocado a $x_i^{'}$. A seguir é definido o número de out-transmissão para o conjunto X_p de vértices.

$$\sigma_o\left(\mathrm{X_p}\right) = \sum_{x_j \in \mathrm{X}} \mathrm{v_j}\; d(X_p, x_j) \tag{3.11}$$

Definição de in-transmission de um Conjunto de Vértices :

Definimos o número de in-transmission de modo análogo.

$$\sigma_t\left(\mathrm{X_p}\right) = \sum_{x_j \in \mathrm{X}} \mathrm{v_j}\; d(x_j\,, X_p) \tag{3.12}$$

3.3.4.3. Definições de p-out-Mediana, p-in-Mediana:

Definição de p-out-Mediana :

Seja um conjunto $\overline{X}_{po}$ para qual

$$\sigma_o(\overline{X}_{po}) = \min_{X_p \subseteq X} [\sigma_o(X_p)] \tag{3.13}$$

é chamado p-out-mediana do grafo *G*.

Definição de p-in-Mediana :

De maneira semelhante, dado $\overline{X}_{pt}$ para o qual

$$\sigma_t(\overline{X}_{pt}) = \min_{X_p \subseteq X} [\sigma_t(X_p)] \tag{3.14}$$

é chamado de p-in-mediana do grafo G.

Não é computacionalmente prático usar as equações (3.9), (3.10), (3.11), (3.12), (3.13) e (3.14) diretamente para encontrar as p-medianas do grafo, mesmo sendo um grafo de tamanho moderado. Algoritmos computacionais adequados serão apresentados na Seção 3.3.8.

3.3.5. p-Medianas Absolutas

Considere um grafo não direcionado *G(X,A)*, onde *A* representa o conjunto de arestas. A questão é saber se sobre alguma destas arestas existe algum ponto *y* (não necessariamente um vértice) onde o número de transmissão

$$\sigma(y) = \sum_{x_j \in X} v_j\, d(y, x_j) \tag{3.15}$$

seja menor que o número de transmissão da mediana do grafo. Chamar-se-ia então aquele ponto $\bar{y}$, com o mínimo $\sigma(y)$, de mediana absoluta de G.

CHRISTOFIDES (1975), demostrou com o Teorema 1 e o Teorema 2, apresentados a seguir, que, ao contrário do caso do centro de um grafo, não há nenhum ponto $\bar{y}$ com $\sigma(\bar{y}) < \sigma(\bar{x})$. Portanto, segundo o autor não há necessidade de pesquisar a solução nas arestas, porque sempre existirá uma solução ótima no conjunto de vértices, conforme apresentação de tais teoremas.

Teorema 1. Existe pelo menos um vértice x de $G = (X, A)$ para qual $\sigma(x) \le \sigma(y)$ para qualquer ponto arbitrário y em G.

Prova :

Dado y um ponto sobre o arco (x_a, x_b) distante ξ de x_a Então:

$$d(y, x_j) = \min\left[\xi + d(x_a, x_j), c_{ab} - \xi + d(x_b, x_j)\right] \tag{3.16}$$

Onde c_{ab} é o comprimento da aresta (x_a, x_b).

Dado X_a o conjunto de vértices x_j para o qual o primeiro termo na eq. (3.16) é menor e dado X_b o conjunto de vértices x_j para o qual o segundo termo é menor. Pode-se escrever :

$$\sigma(y) = \sum_{x_j \in X} v_j d(y, x_j) = \sum_{x_j \in X_a} v_j\left[\xi + d(x_a, x_j)\right] + \sum_{x_j \in X_b} v_j\left[c_{ab} - \xi + d(x_b, x_j)\right] \tag{3.17}$$

Dada a desigualdade triangular para distâncias tem-se :

$$d(x_a, x_j) \le c_{ab} + d(x_b, x_j) \tag{3.18}$$

substituindo $\{ c_{ab} + d(x_b , x_j)\}$ por $d(x_a , x_j)$ no segundo termo da eq. (3.17)

$$\sigma(y) \geq \sum_{x_j \in X_a} v_j \left[\xi + d(x_a, x_j) \right] + \sum_{x_j \in X_b} v_j \left[d(x_a, x_j) - \xi \right] \tag{3.19}$$

Note que $X_a \cup X_b = X$ e reorganizando (3.19) obtém-se:

$$\sigma(y) \geq \sum_{x_j \in X} v_j d(x_a, x_j) + \xi \left[\sum_{x_j \in X_a} v_j - \sum_{x_j \in X_b} v_j \right] \tag{3.20}$$

Desde que se escolha satisfatoriamente qual vértice da aresta (x_a , x_b) chamar x_a e qual chamar x_b, poderá se obter :

$$\sum_{x_j \in X_a} v_j \geq \sum_{x_j \in X_b} v_j , \tag{3.21}$$

note que o primeiro termo no lado da mão direita de (3.20) é $\sigma(x_a)$, a desigualdade (3.20) se torna:

$$\sigma(y) \geq \sigma(x_a) \tag{3.22}$$

Onde, o vértice x_a tem $\sigma(x_a)$ tão baixo quanto $\sigma(y)$. Segundo CHRISTOFIDES (1975), o Teorema 1 pode ser generalizado para o caso das p-medianas absolutas como segue:

Teorema 2. Existe pelo menos um subconjunto $X_p \subset X$ contendo p vértices , tal que $\sigma(X_p) \leq \sigma(Y_p)$ para qualquer Y_p de p pontos sobre as arestas ou vértices do grafo $G = (X, A)$.

Teoremas 1 e 2 acima aplicados quando as transmissões $\sigma(x)$ e $\sigma(x_p)$ são definidas por expressões da forma (3.3) ,(3.4), (3.11) e (3.12).

De acordo com os Teorema 1 e 2, acima apresentados, os conceitos de p-medianas e p-medianas absolutas mencionados nas seções 3.3.4. e 3.3.5. respectivamente, podem ser sintetizados a encontrar a solução para as p-medianas. Os referidos Teoremas garantem que sempre existirá uma solução ótima no conjunto de vértices de p-medianas. Portanto não é preciso buscar a solução nos arcos, bastando procurá-la nos vértices para garantir a melhor solução (ao contrário do caso de centros absolutos, onde se faz necessário procurar a solução nas arestas).

3.3.6. Problema de p-Medianas Puro

O problema acima apresentado (na Seção 3.3.3. até 3.3.5.) pode ser considerado como sendo um problema de p-Medianas "Puro" no sentido de que no mesmo os custos fixos não variam com a localização das medianas (não sendo portanto necessário incluí-los na função objetivo), e possíveis restrições adicionais tais como restrições nas capacidades dos arcos da rede e no tamanho máximo das medianas não são consideradas.

3.3.7. Problema das p-Medianas Generalizado

No problema das p-Medianas generalizado são consideradas a inclusão de diversas variantes, tais como a inclusão de custos fixos na função objetivo, a restrição na capacidade dos arcos da rede, tamanho máximo das medianas entre outras.

O problema das p-medianas generalizado constitui uma classe mais geral que o problema da p-medianas puro. No problema generalizado custos fixos f_i são associados com os vértices x_i. Define-se então o problema p-mediana generalizado como segue.

Dado um grafo $G = (X, A)$, com a matriz de distância mínimas $[d(x_i, x_j)]$, pesos de vértice v_i e custo fixo do vértice f_i, o problema é achar um subconjunto $\overline{X}_p$ que contém p vértices de forma que seja minimizada a seguinte função

$$Z = \sum_{x_i \in \bar{X}_p} f_i + \sigma(\bar{X}_p) \qquad (3.23)$$

Assim, neste caso o objetivo não é somente minimizar a transmissão $\sigma(\bar{X}_p)$ de $\bar{X}_p$ mas a função total Z a qual inclui um custo fixo f_i para todo vértice x_i em $\bar{X}_p$. Na prática f_i representa o custo fixo associado com construir uma instalação (depósito, fábrica, centro de comutação), onde a localização é representada pelo vértice x_i.

O problema de p-mediana corresponde ao caso onde todos os f_i são iguais a um certo valor f de forma que o primeiro termo da eq (3.23) se torna um vetor pf constante sem importar quais vértices fazem parte do subconjunto X_p.

Um problema similar ao das p-medianas generalizadas, definido acima, consiste em encontrar X_p tal que $|X_p| \leq p$, e $\sigma(x_p)$ é máximo. O problema de minimizar a expressão (3.23) sujeito a $|\bar{X}_p| \leq p$ é uma versão do problema das p-medianas generalizado que na pratica é freqüentemente encontrado em diversas aplicações.

Problemas práticos de localização de depósito invariavelmente envolvem restrições de capacidade o que não existe no problema de p-mediana puro. Uma restrição bastante habitual se refere ao valor máximo e mínimo da expressão :

$$\sum_{x_j \text{ alocado a } x_i} v_j \qquad (3.24)$$

para qualquer vértice mediana $x_i \in \bar{X}_p$. A expressão (3.24) é uma medida da quantia de material transmitida de x_i (processamento) e é, consequentemente uma medida do tamanho físico do depósito x_i.

As dificuldades para resolver problemas práticos de localizações de depósitos, não está nas variações e restrições adicionais discutidas acima, mas são inerentes ao próprio problema puro de medianas. Devido a isto a seguir serão discutidos alguns dos métodos disponíveis para achar as p-medianas de um grafo. Para simplificar a nomenclatura serão descartados os termos *o* e *t* representando "out" e "in" medianas, respectivamente, desde que os métodos se aplicam igualmente a ambos.

3.3.8. Métodos para Resolver o Problema de Localização de Medianas

Os estudos relativos à localização de medianas foram iniciados por Hakimi no ano 1964-1965. O autor mostrou que pelo menos uma solução, ótima para o problema das p-medianas, estava localizada nos vértices da rede. Segundo Garey e Johnson citado por HRIBAR (1997), não obstante o problema ter sido reduzido a buscar a solução somente nos vértices, continua sendo NP-completo para um número p variável.

Após Hakimi, um grande número de algoritmos foram propostos para o modelo das p-Medianas, inclusive técnicas de *Pesquisa nas Vizinhanças* (Maranzana, 1964), *Algoritmos de substituição* (Teitz e Bart, 1968) e *Algoritmos de Relaxação Lagrangeana* combinado com *Branch and Bound* (Christofides and Beasley, 1982), entre outros.

Entre os algoritmos de resolução das p-Medianas pode-se diferenciar os métodos exatos e os heurísticos.

Entre os métodos exatos pode-se mencionar, em primeiro lugar, a resolução das p-Medianas como um problema de programação inteira, e em segundo o método exaustivo, ambos freqüentemente usados no caso de problemas de pequeno porte, isto é, problemas com um reduzido número de vértices na rede, assim como também com um pequeno número de instalações a serem localizadas.

No que diz respeito aos métodos aproximados existem na literatura uma diversidade de métodos heurísticos. Entre estes cabe ressaltar o **"Método de Substituição"** proposto por Teitz e Bart no ano de 1968. Existem ainda na literatura outros métodos heurísticos tais como o **"Método da Partição"** proposto por Maranzana, o **"Método de Pizzolato"** proposto pelo autor do mesmo nome e por ultimo a

"Heurística de Programação Dinâmica" proposto por Hribar e Daskin no ano de 1997.

É relevante destacar a importância dos métodos heurísticos na resolução de problemas de grande porte, onde a explosão combinatorial impede o tratamento destes problemas através de métodos exatos de resolução.

A seguir serão abordados os métodos citados.

3.3.8.I. O Problema das p-Medianas como um Problema de Programação Inteira.

Seja $\lfloor \xi_{ij} \rfloor$ a matriz de alocação tal que

$$\xi_{ij} = \begin{cases} 1 \text{ , se } x_j \text{ e' alocado a } x_i; \\ 0 \text{ , caso contrario.} \end{cases}$$

Além disso,

$$\xi_{ii} = \begin{cases} 1 \text{ , se } x_i \text{ e' uma mediana;} \\ 0 \text{ , caso contrario.} \end{cases}$$

O problema de p-Medianas pode ser formulado como segue :

$$\text{Minimizar} \quad Z = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}\xi_{ij} \tag{3.25}$$

sujeito a:

$$\sum_{i=1}^{n} \xi_{ij} = 1 \quad \text{para } j = 1,...,n \tag{3.26}$$

$$\sum_{i=1}^{n} \xi_{ii} = p \tag{3.27}$$

$$\xi_{ij} \le \xi_{ii} \quad \text{para todo i,j =1,...,n} \tag{3.28}$$

e

$$\xi_{ij} = 0 \text{ ou } 1 \tag{3.29}$$

onde $[d_{ij}]$ é a matriz peso - distância de ordem $n \times n$, isto é, a matriz distância onde cada coluna j está multiplicada pelo peso v_j.

Significado das restrições:

- A restrição (3.26) assegura que todo vértice x_j é alocado a um e somente um vértice mediana x_i;
- A restrição (3.27) garante que existem exatamente p vértices medianas;
- A restrição (3.28) diz que $\xi_{ij} = 1$ somente se $\xi_{ii} = 1$, isto é, a restrição assegura que as alocações só podem ser feitas a vértices medianas;
- A restrição (3.29) impõe a integridade: ξ_{ij} é variável binária que assume valor zero ou um;

Se $\left[\bar{\xi}_{ij}\right]$ é a solução ótima para o problema proposto acima, então a p-mediana é

$$\bar{X}_p = \{\ x_i / \bar{\xi}_{ii} = 1\ \}$$

Observação: Se a condição (3.29) for substituída por $\xi_{ij} \geq 0$, então o problema resultante é um Problema de Programação Linear cuja solução pode ser obtida facilmente.

As soluções para este problema linear não são necessariamente inteiras e valores fracionários para $\xi_{ij} \leq 1$ podem ocorrer, embora Revelle e Swain, citados por CHRISTOFIDES (1975), observam que tais valores fracionários ocorrem raramente. Então, pode-se usar a formulação de programação linear para obter a p-mediana para a maioria dos problemas.

Em todo caso, se valores fracionários de algum ξ_{ij} acontecerem, então uma resolução destas incertezas pode ser obtida por um procedimento de busca em árvore (Branch and Bound).

Uma aproximação alternativa via programação linear é determinada por Marsten, que mostra que a solução $\lfloor \xi_{ij} \rfloor$ correspondendo à p-mediana de um grafo como descrito pelas equações de (3.25) até (3.29) é um ponto extremo de um certo poliedro H, e que toda outra p-mediana para $1 \leq p \leq n$ também são pontos extremos de H. Usando *multiplicador de lagrange* e *programação linear paramétrica*, Marsten dá um método de percorrer um caminho entre alguns dos pontos extremos de H. Este caminho gera sucessores de p-medianas do grafo G em ordem descendente de p, embora certas p-medianas (para alguns valores de p) podem ser perdidas e nunca serem geradas, ou, reciprocamente, podem ser gerados pontos extremos de H que não correspondem a p-medianas de G, isto é, contenha valores fracionários de ξ_{ij}. Assim, embora este método é teórica e computacionalmente atraente, pode não produzir a p-mediana do grafo para um valor específico de p requerido. Christofides relata o caso de um grafo de 33 vértices onde as p-medianas foram geradas com êxito para p = 33, 32,...,10, mas a 9-mediana e 8-mediana não puderam ser obtidas por este método.

3.3.8.2. Método da Enumeração Exaustiva

O método de enumeração exaustiva ou direta, consiste em avaliar, uma a uma, cada possibilidade de localização da mediana, até que seja encontrada a solução ótima Supondo que, se *n* é o conjunto dos vértices da rede e *p* o número das instalações a serem localizadas, então o número de possibilidades a serem testadas é $\binom{n}{p}$.

Este método requer o cálculo da função objetivo para cada combinação possível de *p* nos *n* locais possíveis para a implantação das instalações, até que seja encontrada a solução ótima.

Porém o número de combinações de *p* em *n* locais pode ser muito elevado. Portanto o método da enumeração exaustiva pode requerer um volume de cálculos excessivo. É evidente que o processo de busca exaustiva garante o melhor esquema de localização das instalações, porém em sistemas com elevado número de vértices e várias instalações a distribuir, o número de combinações possíveis se torna muito grande o que levaria a um tempo de processamento muito elevado, motivo que inviabilizaria sua utilização.

No caso de sistemas de pequeno porte as p-Medianas podem ser encontradas de forma exata através do método da enumeração exaustiva. Este método foi usado por HAKIMI (1985), para encontrar as 3-medianas de um grafo com 10 vértices. Porém, mesmo com o recurso do computador o tempo computacional cresce muito rapidamente, o que limita o uso do método em grafos com poucos vértices.

3.3.8.3. Método da Substituição de Vértices

É um método heurístico baseado na substituição de vértices descrito por TEITZ&BART (1968).

O método consiste basicamente em escolher quaisquer p vértices de forma aleatória como sendo o conjunto inicial S, que supõe-se ser uma aproximação do conjunto $\bar{X}_p$ das p-medianas.

A seguir é testado se algum vértice $x_j \in X - S$ pode substituir um vértice $x_i \in S$ como vértice mediana e então produzir um novo conjunto $S' = S \cup \{x_j\} - \{x_i\}$ para o qual o número de transmissão $\sigma(S') < \sigma(S)$. Se for, então é feita a substituição de x_i por x_j e obtém-se S' que é uma melhor aproximação para o conjunto $\bar{X}_p$. Segue-se este procedimento, produzindo aproximações para $\bar{X}_p$, até chegar-se a um conjunto $\bar{S}$, onde nenhuma substituição de vértice de $\bar{S}$ por outro em X - $\bar{S}$ produza um $\sigma(S)$ menor que $\sigma(\bar{S})$ - e neste caso é a aproximação final para o conjunto $\bar{X}_p$.

Descrição do algoritmo

Passo 1. Selecionar um conjunto S de p vértices para formar a aproximação inicial das p-medianas. Rotular todos os vértices $x_j \notin S$ como *"não testados"*.

Passo 2. Selecione algum vértice *"não testado"* $x_j \notin S$ e para cada vértice $x_i \in$ S, compute a *"redução"* Δ_{ij} na transmissão, se x_j é substituído por x_i, isto é, compute:

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

Passo 3. Encontre $\Delta_{i_o j} = \max_{x_i \in S} [\Delta_{ij}]$

(i) Se $\Delta_{i_o j} \leq 0$ rotule o vértice x_j como "testado" e volte ao passo 2

(ii) Se $\Delta_{i_o j} > 0$ efetuar $S \leftarrow S \cup \{x_j\} - \{x_i\}$, rotular x_j como "testado" e volte para o passo 2.

Passo 4. Repetir os passos 2 e 3 até que todos os vértices em X-S estejam rotulados como testados. Este procedimento é referido como ciclo. Se, durante o último ciclo nenhuma substituição de vértice foi feita no passo 3(i), vá para o passo 5. Caso contrário, se foi feita alguma substituição de vértice, rotule todos os vértice como não testados e retome ao passo 2.

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

O algoritmo descrito acima é de fato da família de algoritmos baseados em otimização local, onde a primeira idéia de λ-otimal foi introduzida por Lin para o problema do caxeiro viajante, e subseqüentemente estendido e usado por uma variedade de problemas combinatoriais (TEITZ&BART, 1968).

3.3.8.4. Método da Partição

Este método também heurístico, deve-se a Maranzana. Em sua essência, são buscados sucessivos vértices únicos de m subconjuntos "destinos", cada um associado com uma origem. Estes conjuntos vão sendo modificados e o processo é repetido.

Maranzana introduziu o conceito de "centro de gravidade" de uma rede ou grafo, análogo ao de mediana (BINFARÉ,1993).

Um vértice p_j é um centro de gravidade de $Q \subset P$ se

$$\sum_{p_k \in Q} D_{jk} w_k \leq \sum_{p_k \in Q} D_{ik} w_k \qquad , \forall i$$

onde

P é o conjunto de nós do grafo;

D_{jk} é o custo do caminho de mínimo custo de p_j a p_k ;

w_k é o peso de p_k

O método inicia com uma seleção arbitrária de origens e particiona-se a rede em subconjuntos a serem "servidos" por estas origens, associando-se cada ponto a sua origem mais próxima. Após o centro de gravidade de cada conjunto da partição é calculado e as origens iniciais são substituídas por estes pontos. O processo é repetido até os "pontos origens" não serem mais mudados.

3.3.8.5. Método de Pizzolato

As contribuições para as soluções heurísticas realizadas por Maranzana, Teitz e Bart foram fundamentais para a formulação de extensões posteriores.

De acordo com Pizzolato, é importante notar que estes métodos requeriam o uso de mainframes e a complexidade dos mesmos reduzia suas aplicações a pequenas redes. No caso dos três primeiros trabalhos, os problemas limitavam-se em cerca de 30 vértices e, para o ultimo, em 100 vértices, embora com alguma perda de qualidade (BINFARÉ, 1993).

O método heurístico proposto por Pizzolato, basicamente consiste em efetuar a construção de p árvores disjuntas. As árvores vão modificando suas formas, progressivamente, conforme sucessivos testes efetuados sobre suas raízes e vértices, podendo ocorrer a eliminação de uma árvore e a partição de outra em duas, bem como modificações nas raízes e transferência de vértices de uma para outra árvore.

BINFARÉ (1993), afirma que o método de Pizzolato oferece, não só uma solução de boa qualidade, freqüentemente ótima, mas também presta-se a grandes redes (cerca de 500 nós), pode ser utilizado por um micro comum e apresenta um baixo tempo de execução inversamente proporcional a *p*.

3.3.8.6. Heurística de Programação Dinâmica

HRIBAR e DASKIN (1997), propuseram uma heurística baseada em programação dinâmica. Este método difere dos apresentados acima desde que identifica um grande número de soluções muito boas; diferentemente dos métodos mencionados anteriormente onde é obtida uma única solução.

O método heurístico de programação dinâmica tem semelhança com o conceito do algoritmo genético, no sentido de obter múltiplas soluções ao longo da execução do algoritmo, além, de permitir ao usuário determinar a freqüência com que candidatos locais aparecem no grupo de soluções boas. Como tal, o método identifica localizações que deveriam ser considerados mais importantes e as que provavelmente podem ser excluídas de qualquer plano de localização. Esta informação é semelhante ao usado em algoritmos de busca tabu para diversificar o processo de procura.

A heurística de programação dinâmica é um método que resulta da mistura do algoritmo guloso com a abordagem de programação dinâmica. A seguir serão abordados brevemente os dois algoritmos antes de apresentar o algoritmo híbrido.

Algoritmos gulosos encontram a melhor situação dado o estado atual de solução. "Melhor" é definido por alguma medida apropriada para o problema. Para o problema das p-medianas a " melhor ligação" é a mínima ligação que não crie ciclo. Em geral, algoritmos gulosos tendem a não ser os mais eficientes, entretanto eles são fáceis de entender e fáceis de implementar.

Para alguns problemas o critério no qual o algoritmo deveria ser guloso não é sempre a priori. Geralmente é bastante fácil mudar o modo de avaliação em algoritmos gulosos. Como tal, a eficácia de medidas novas pode ser testada prontamente.

O algoritmo guloso para o problema das p-Medianas é mostrado a seguir :

```
S ← φ
For K = 1 to P do
    best_node_to_add ← 0
    best_obj ← ∞
    For J = 1 to N do
        If g (S ∪ {J}) < best_obj then
            best_obj ← g (S ∪ {J})
            best_node_to_add ← J
        endif
    endfor
    S ← S ∪ { best_node_to_add }
endfor.
```

Neste algoritmo, S é o conjunto com a localização de instalações candidatas, e N é o número de vértices do problema.

A função $g(X)$ é a função objetivo das p-Medianas quando as instalações são localizadas nos nós definidos pelo conjunto X. Para $P = 1$, o algoritmo guloso apresentará a solução ótima. Para elevados valores de P, o algoritmo pode dar soluções bastante pobres conforme observação realizada por Daskin (HRIBAR , 1997).

A seguir é abordado o método de programação dinâmica para resolver o problema das p-Medianas onde a variável estágio é iniciada com o número de instalações adicionadas até o momento, e a variável estado iniciada com o conjunto de locais selecionados.

Fazendo S_i denotar o conjunto de todos os possíveis estados, com i instalações selecionadas, a seguir é apresentado o algoritmo de programação dinâmica para o problema das p-Medianas.

```
S_o ← φ
For I = 1 to P do
    For S in S_{i-1} do
        For J = 1 to N do
            If {S ∪ {J}} ∉ S_i then S_i ← S_i ∪ {S ∪ {J}}
            endif
        endfor
    endfor
endfor
Encontrar S̄ = arg min_{S∈S_P} {g(S)}
```

Este algoritmo constrói essencialmente toda combinação possível de i vértices fora dos N possíveis vértices candidatos para todos os valores de i entre 1 e P, o número desejado de instalações.

O algoritmo encontra os melhor conjuntos de vértices entre todas as possíveis combinações de P vértices selecionados desde os N candidatos locais. Porém o algoritmo de programação dinâmica não é prático para elevados (ou até mesmo moderados) valores de P e N. Em essência, o algoritmo de programação dinâmica esboçado acima, não é nada além de um esquema de enumeração total.

A dificuldade para computar, salvar e avaliar todas as possíveis combinações de P-1 localizações de instalações em N locais candidatos, com o intuito de encontrar a solução ótima de p-medianas, torna esta busca claramente proibitiva. Por outro lado, tentar encontrar uma solução ótima, também é provável que seja desnecessário. Em outras palavras, é improvável que os melhores locais de P irão incluir P-1 locais de baixo desempenho como uma solução ao P-1 problema mediano. É bastante provável que boas soluções de P localizações para o problema medianas venham de soluções boas do P-1. Isto sugere que não é necessário armazenar todas as possíveis combinações de q instalações de $(1 \leq q \leq P-1)$ e que será necessário guardar somente aquelas boas soluções de q-medianas. Este fato conduziu à heurística baseada em programação dinâmica para resolver o problema de p-Medianas mostrada no algoritmo que segue :

$S_o \leftarrow \phi$

For $I = 1$ **to** P **do**

  $h \leftarrow 0$

  For S **in** S_{i-1} **do**

    For J = 1 **to** N **do**

      If h < H **Then**

        $h \leftarrow h + 1$

        $S_i \leftarrow S_i \cup \{S \cup \{J\}\}$

        $G_i^h(S^h) \leftarrow G(s \cup \{J\})$

        SORT the h solution in S_i

      Else

        If $G(S \cup \{J\}) < G_i^H(S^H)$ **Then**

          $S_i \leftarrow S_i \setminus S^H$

          $S_i \leftarrow S_i \cup \{S \cup \{J\}\}$

          $G_i^H(S^H) \leftarrow G(s \cup \{J\})$

          SORT the H solution in S_i

        endif

      endif

    endfor

  endfor

endfor

Neste algoritmo foi usada a seguinte notação adicional:

H = número máximo de soluções para armazenar em qualquer estágio.

h = número de solução realmente armazenadas,

$G_i^m(S^m)$ = mth melhor valor da função encontrada para o problema i-mediano (que é obtido localizando instalações ao longo dos vértices no conjunto S^m).

No pseudo código apresentado acima, se o conjunto { 1, 3, 7 } está entre o topo de soluções H para um problema de 3-medianas, pode-se obter esta solução por qualquer um dos três modos: adicionando o nó 7 para o conjunto { 1, 3 } na solução de 2-medianas; adicionando o nó 3 para o conjunto { 1, 7 } na solução de 2-medianas; ou adicionando o nó 1 para o conjunto { 3 , 7} na solução de 2-medianas. Assim, quando são geradas a segunda e terceiras vezes, será preciso conferir se já está no conjunto S_3.

Esta heurística é equivalente ao algoritmo guloso se $H = 1$ e ao algoritmo de programação dinâmica se H é suficientemente grande, quer dizer, se

$$H > \max_{0 \le j \le P} \binom{N}{j}$$

3.4. Considerações Finais

Ao longo deste capítulo além de efetuar a revisão bibliográfica sobre o problema, verificou-se a importância da eficiente localização das p-medianas como meio de redução de custos de transporte associados à localização das instalações.

Nessa perspectiva cabe ressaltar o método de substituição de Teitz e Bart, indicado pelos autores como a heurística mais usada e a que melhores resultados oferece para problemas de grande porte.

CAPÍTULO IV

4. Técnica de Busca Proposta

4.1. Definição do Problema

O problema de **"Localização de medianas"**, conforme foi visto no terceiro capitulo, consiste em encontrar lugares adequados para instalações de modo a minimizar a soma total das distâncias de todos os vértices à instalação mais próxima.

Nesse contexto a localização ótima da instalação é chamada de mediana do grafo. Existem situações nas quais têm-se o interesse de encontrar mais de uma mediana, nesse caso trata-se do problema de p-Medianas, conhecido na literatura como **"Problemas de Mini-sum"** ou **"Problema de Localização de p-Medianas "**.

O primeiro pesquisador a formular o problema de uma única mediana foi HAKIMI (1964). Pouco tempo depois o problema de uma mediana foi generalizado para múltiplas medianas. O referido autor propôs um procedimento simples de enumeração para o problema. Além disso, a literatura apresenta diversas heurísticas para resolver o problema das p-medianas. Algumas delas são usadas para obter boas soluções iniciais. O método de substituição de TEITZ&BART (1968) é freqüentemente citado na literatura como a heurística mais simples e com melhores resultados. Heurísticas mais complexas exploram uma árvore de busca

(GALVÃO&RAGGI, 1989). Algumas abordagens bem sucedidas usam informações primal/dual do problema (SENNE&LORENA, 1997 e BEASLEY, 1993).

O problema das p-medianas pode ser formulado como um problema de programação inteira binária. A fórmula matemática que o descreve é apresentada a seguir:

$$\text{Minimizar} \quad Z = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}\xi_{ij} \tag{4.1}$$

sujeito a:

$$\sum_{i=1}^{n} \xi_{ij} = 1 \quad \text{para } j = 1,\dots,n \tag{4.2}$$

$$\sum_{i=1}^{n} \xi_{ii} = p \tag{4.3}$$

$$\xi_{ij} \le \xi_{ii} \quad \text{para todo } i,\ j = 1,\dots,n \tag{4.4}$$

e

$$\xi_{ij} = 0 \text{ ou } 1 \tag{4.5}$$

Onde:

$\lfloor \xi_{ij} \rfloor$ é a matriz de alocação tal que

$\xi_{ij} = 1$, se x_j é alocado a x_i;

0, caso contrário.

e,

$\xi_{ii} = 1$, se x_i é um vértice mediana ;

0, caso contrário.

$[d_{ij}]$ é a "matriz peso" que representa o custo de atender a zona j a partir da instalação i, multiplicado pelo peso v_j.

4.2 Técnica de Busca Baseada em Algoritmos Genéticos para Encontrar a Localização de p-Medianas

4.2.1. Origem dos Algoritmos Genéticos

Nos últimos anos a comunidade cientifica internacional vem mostrando um crescente interesse pela nova técnica de busca baseada na teoria da evolução conhecida como algoritmos genéticos. Esta técnica se baseia no mecanismo de seleção existente na natureza, segundo o qual os indivíduos mais aptos de uma população são os que sobrevivem, pois se adaptam mais facilmente às mudanças produzidas em seu entorno.

Hoje em dia sabe-se que as mudanças do indivíduo se efetuam nos genes (unidade básica de codificação de cada um dos atributos de ser vivo) e seus atributos mais desejáveis (isto é, aqueles que permitem a adaptação melhor a seu entorno) são transmitidos a seus descendentes quando estes se reproduzem sexualmente.

Neste sentido, John Holland, pesquisador da Universidade de Michigan, consciente da importância da seleção natural, desenvolveu nos anos 70, uma técnica para poder incorporar estes conceitos em um programa de computador. Seu objetivo era conseguir que os computadores apreendessem por si mesmos. A técnica que Holland inventou foi chamada originalmente de "planos reprodutivos", mas se tornou conhecida com o nome de "Algoritmos Genéticos"

4.2.2. Definições Gerais

Algoritmos Genéticos (AG) constituem uma técnica de busca inspirada no processo de evolução dos seres vivos. São métodos generalizados de busca e otimização que simulam os processos naturais de evolução, aplicando a idéia Darwiniana de seleção. De acordo com a aptidão e a combinação com outros operadores genéticos, são produzidos métodos de grande robustez e aplicabilidade. Tais AG's inicialmente desenvolvidos por Holland, na década de 70, e atualmente vem sendo aplicados com bons resultados nas mais variadas áreas de conhecimento.

Uma definição ampla de algoritmo genético é a proposta por KOZA (1992), segundo o qual um A.G. 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 principio Darwiniano de reprodução e sobrevivência do mais apto.

De modo mais específico, Algoritmo Genético é um procedimento iterativo que mantém uma população de estruturas (chamadas 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 são mantidos, enquanto os outros são descartados (Darwinismo). 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 **(reprodução).**

O algoritmo genético também permite obter soluções para problemas que não podem ser resolvidos por nenhum método exato, ou cuja solução exata não pode ser calculada em um tempo aceitável. É o caso particular dos problemas de otimização combinatorial, tais como: formar equipes de trabalho, planificar rodízios de entrega, localizar instalações e outros.

4.2.3. Estrutura Básica da Técnica de Busca Baseada em Algoritmos Genéticos

Os passos principais da técnica de busca baseada em algoritmos genéticos, para localizar as p-medianas, podem ser descritos como segue :

Início Algoritmo

Representar uma solução na estrutura de um cromossomo;

Construir uma população inicial de cromossomos

Avaliar o Fitness dos indivíduos da população;

Repetir

Selecionar ancestrais da população;

Efetuar o cruzamento (crossover) entre os ancestrais selecionados;

Efetuar uma mutação ou recombinação nos descendentes gerados;

Avaliar o Fitness dos descendentes gerados

Até que uma solução satisfatória seja encontrada.

Fim Algoritmo.

Dentro desta perspectiva, são apresentados: a representação de uma solução na estrutura de um cromossomo, a construção de uma população inicial, a avaliação do fitness e os operadores genéticos (seleção, crossover e mutação).

4.2.4. Representação de uma solução na Estrutura de um Cromossomo

Nos Algoritmos Genéticos (AGs convencionais), a codificação mais comum das soluções se dá através de cadeias binarias (zero – um). E a popularidade desta representação se deve à estrutura originalmente proposta por Holland, e a facilidade para implementar computacionalmente.

Embora a representação por cadeias binarias tenha se mostrado eficiente para vários problemas, observou-se, à medida que foram crescendo as aplicações de AGs, que em diversos problemas com um elevado número de restrições, esta representação pode não ser a mais adequada, surgindo daí alternativas, a exemplo da representação por inteiros, onde um cromossomo é descrito por um vetor de números inteiros.

Na presente pesquisa foi adotada a representação de soluções através de vetores. Cada vetor representando um cromossomo com números inteiros, isto é, cada vetor representando uma solução para o problema de p-medianas, conforme ilustração da figura 4.1.

1	2	3	4	5	6	7	8	9	10
3	3	3	5	5	5	3	8	8	8

Figura 4.1. : Representação do Cromossomo

O exemplo acima é um vetor de 10 posições, significando que o número de vértices (nós) na rede é 10, cada posição do vetor representando um vértice, o conjunto *V* de vértices pode ser dividido em dois subconjuntos, o primeiro subconjunto V_1 dos vértices que são medianas e o segundo subconjunto V_2 dos vértices que não são medianas. No exemplo $v_1 = \{3,5,8\}$ e $v_2 = \{1,2,4,6,7,9,10\}$, cabe ressaltar que os vértices 3, 5 e 8 são medianas, e os demais vértices são atribuídos a mediana mais próxima (de menor custo). Nele, o elemento da posição 1 do vetor é 3, significando que o vértice 1 esta ligado a mediana 3, o elemento da posição 3 é 3, significando que o vértice 3 é mediana, e assim sucessivamente.

Independentemente do tipo de representação selecionada, deve-se sempre verificar se a representação está corretamente associada com as soluções do problema analisado, ou seja, que toda solução tenha um cromossomo associado e reciprocamente que todo cromossomo gerado pelo AG esteja associado a uma solução válida do problema analisado.

4.2.5. Construção de uma População Inicial de Cromossomos

O problema de gerar uma população inicial de cromossomos num AG normalmente não é uma tarefa das mais difíceis. Pelo contrário, em muitos problemas esta tarefa é muito simples. Basicamente consiste em escolher, de forma aleatória ou com fundamentação probabilística, aqueles representantes que julgarem serem possíveis soluções para o problema.

Já nos problemas onde a geração de uma população inicial não é tão imediata, normalmente se costuma usar heurísticas simples e rápidas.

No problema em questão, adotou-se a geração aleatória da população inicial de cromossomos, onde as medianas de cada cromossomo da primeira população são escolhidas aleatóriamente.

4.2.6. Avaliação do Fitness

A avaliação do Fitness de um cromossomo em um AG significa determinar o seu nível de aptidão de sobrevivência, ou seja, em um AG sobrevivem prioritariamente os cromossomos mais aptos. E nos problemas de Otimização, o critério de sobrevivência pode ser determinado pelo valor da sua função objetivo avaliado pelo cromossomo analisado.

Nesse contexto, a avaliação do fitness na heurística proposta está diretamente ligada à função objetivo, isto é o fitness representa a soma da distância percorrida de cada um dos nós a sua instalação mais próxima, ponderada pelos respectivos pesos. A solução que tiver menor distância percorrida das medianas a seus vértices será considerada mais apta, por tanto as soluções com maior fitness terão maior possibilidade de sobreviver.

4.2.7. Operadores Genéticos

Os operadores genéticos têm como função criar novos espaços de busca de soluções. A seguir são apresentados três tipos de operadores: Seleção, Cruzamento e Mutação.

4.2.7.1. Seleção

A **seleção** é um processo no qual os indivíduos da população atual são copiados para a população da geração posterior, em quantidades proporcionais à aptidão de cada um, ou seja, ao valor da função objetivo associado a cada indivíduo (fitness). As cópias produzidas são submetidas à ação de outros operadores genéticos, para a geração de novos indivíduos.

Uma das formas mais simples e mais usada é a seleção pela regra da roleta. Essa seleção simula o procedimento de girar uma roleta viciada, onde cada indivíduo da geração atual tem uma fatia da roleta cujo tamanho é proporcional à sua aptidão. Assim, aos indivíduos com alta aptidão é dada uma porção maior da roleta, enquanto aos de aptidão mais baixa é dada uma porção relativamente menor da roleta. Finalmente, a roleta é girada um determinado número de vezes, dependendo do tamanho da população, e são escolhidos, como indivíduos que participarão da próxima geração, aqueles sorteados na roleta.

Indivíduo S_i		Aptidão $f(S_i)$	Aptidão Relativa
S_1	10110	2.23	0.14
S_2	11000	7.27	0.47
S_3	11110	1.05	0.07
S_4	01001	3.35	0.21
S_5	00110	1.69	0.11

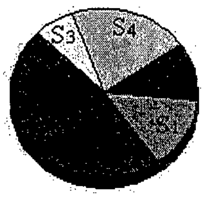


Figura 4.2. : População e sua Correspondente Roleta de Seleção. **Fonte Carvalho (2000)**

Assim, a probabilidade que um indivíduo P_i seja selecionado para reprodução é:

$$P_i = \frac{f(S_i)}{\sum_{j=1}^{n} f(S_i)}$$

onde:

$f(S_i)$ = aptidão do indivíduo i

n = número total de indivíduos na população

P_i = probabilidade do indivíduo i ser selecionado

Na técnica de busca proposta, foi adotada a seguinte estratégia, selecionar aleatoriamente 3 cromossomos da população atual e escolher o melhor cromossomo dentre os três. Guardar este primeiro cromossomo, repetir o processo anterior, isto é escolher novamente mais três cromossomos da população atual e posteriormente efetuar o cruzamento entre os dois cromossomos escolhidos, gerando assim o próximo sucessor da população e assim sucessivamente até obter a nova população.

4.2.7.2. Cruzamento

O **operador de cruzamento** ou crossover é considerado o operador mais importante do algoritmo genético, consiste em recombinar as características dos pais durante a reprodução, permitindo que as próximas gerações herdem essas características. Ele é considerado o operador genético predominante, por isso é aplicado com probabilidade dada pela taxa de crossover ***Pc***, que deve ser maior que a taxa de mutação. O cruzamento pode ser realizado sobre apenas alguns indivíduos.

Os tipos de operadores crossover mais conhecidos para cadeias de bits são o de n-pontos e o uniforme, descritos a seguir :

Crossover em Um-ponto: um ponto de corte é escolhido e a partir dele as informações genéticas dos pais serão trocadas. As informações anteriores a este ponto em um dos pais são ligadas às informações posteriores à este mesmo ponto no outro pai.

Por exemplo, considere-se os indivíduos C_1 e C_2 da figura 4.3.

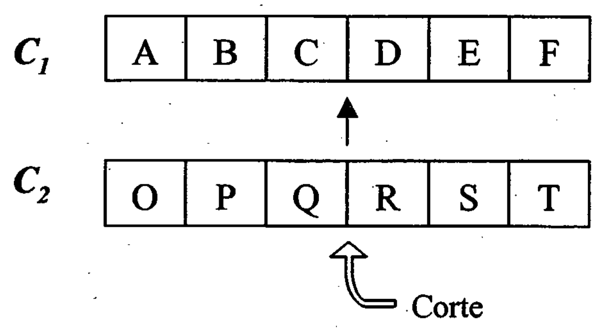


Figura 4.3. : Crossover em um Único Ponto

Escolhido aleatoriamente o ponto de cruzamento *k=3*, o cruzamento gera dois novos indivíduos C_1' e C_2' apresentados na figura 4.4.

Resultado do Cruzamento

C'_1	A	B	C	R	S	T
C'_2	O	P	Q	D	E	F

Figura 4.4. : Resultado do Crossover em um Unico Ponto

Crossover Multi-pontos : é uma generalização do operador anterior, onde se utiliza mais de um ponto de corte na estrutura do cromossomo.

Na literatura, são apresentados diversos operadores de cruzamento multi-pontos, entre eles destacam-se, devido à quantidade de aplicações em que são empregados:

- **Operador OX** (Order Crossover) - Efetua o cruzamento através da escolha aleatória de dois pontos de corte em cada um dos dois indivíduos pais P_1 e P_2 selecionados, ficando assim cada cromossomo dividido em três parcelas, das quais só uma delas será mantida no primeiro filho F_1. As outras duas parcelas serão preenchidas usando as informações não repetidas do segundo indivíduo. O mesmo processo será repetido para a obtenção do segundo filho F_2. Na figura 4.5. são apresentados os cromossomos pais e na figura 4.6. o resultado do crossover aplicando o operador OX.

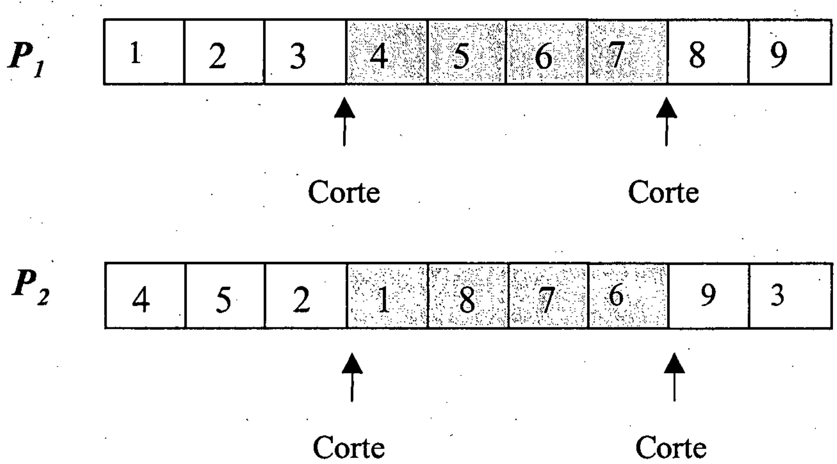


Figura 4.5. : Cromossomos Pais

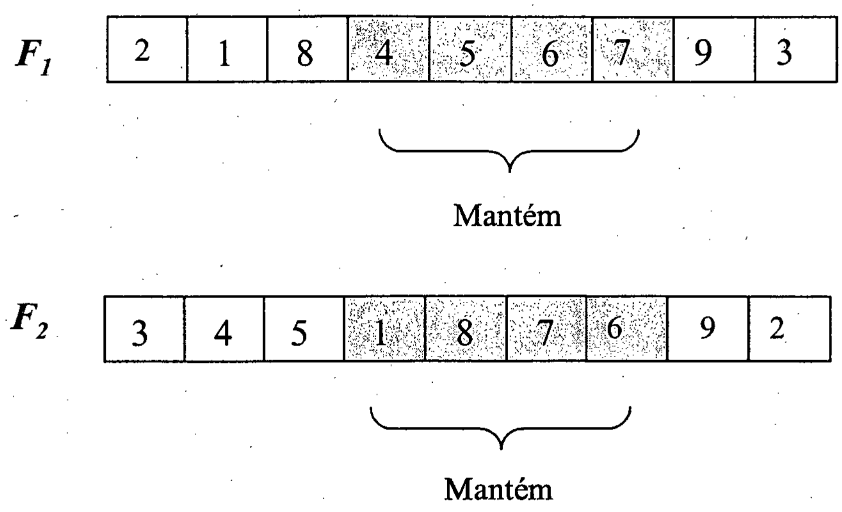


Figura 4.6. : Cromossomos Filhos Aplicando o Operador OX

- **Operador PMX** (Partially Mapped Crossover) - Este operador também é executado escolhendo aleatoriamente dois pontos de corte nos cromossomos pais P_1 e P_2. Os dois cromossomos filhos F_1 e F_2 herdarão integralmente os genes situados entre os dois cortes feitos nos pais P_2 e P_1 respectivamente, preservando a ordem e a posição de cada gene. Em seguida são preenchidos cada componente (gene) ainda não preenchido do filho F_1 pelo componente em P_1 e os de F_2 com P_2 , desde que seja uma solução válida (por exemplo, que não contenha elementos repetidos). Caso contrário, serão preenchidos os componente que faltam em F_1 pelo componente da mesma posição do pai P_2 , analogamente para o F_2. A ilustração deste operador está apresentada nas figuras 4.7. e 4.8.

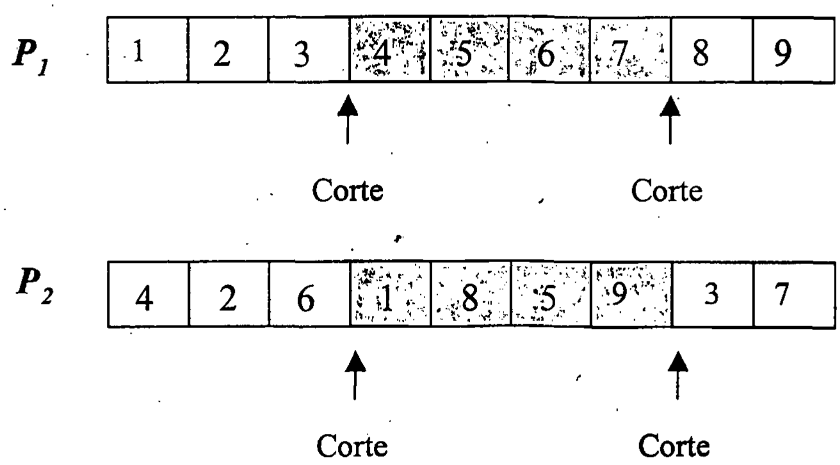


Figura 4.7. : Cromossomos Pais

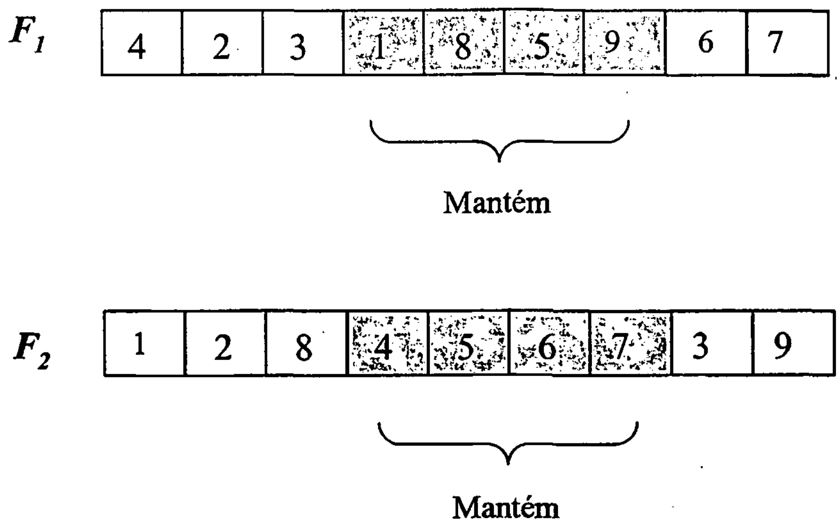


Figura 4.8. : Cromossomos Filhos Aplicando o Operador PMX

Crossover Uniforme : não utiliza pontos de cruzamento, mas determina, através de um parâmetro global, qual a probabilidade de cada variável ser trocada entre os pais. O crossover uniforme acha-se na Figura 4.9. Para cada par de pais é gerada uma máscara de bits aleatórios. Se o primeiro bit da máscara possui o valor 1, então o primeiro bit do pai1 é copiado para o primeiro bit do filho1, caso contrário o primeiro bit do pai2 é copiado para o primeiro bit do filho1. O processo se repete para os bits restantes do filho1, como ilustrado na figura 4.9. Na geração do filho2 o procedimento é invertido, ou seja, se o bit da máscara é 1, então será copiado o bit do pai2. Se o bit for igual a 0, então será copiado o bit do pai1.

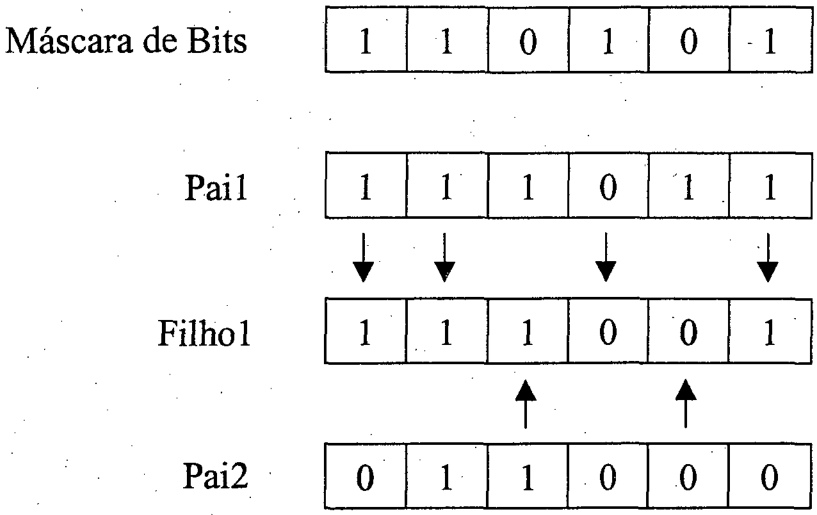


Figura 4.9. : Crossover uniforme

Na técnica de busca proposta, adotou-se uma estratégia diferente de crossover, onde são mantidas algumas características dos dois cromossomos escolhidos para o cruzamento, os genes que se repetem tanto no primeiro cromossomo como no segundo são mantidos no cromossomo filho para fazer parte da próxima população. Isto é, se é mediana no primeiro cromossomo, e também é mediana no segundo cromossomo, então será mediana no filho. Os genes que são diferentes em ambos cromossomos, ou seja, se não é mediana em nenhum dos dois cromossomos ou somente é mediana em um deles, a respectiva posição do gene do cromossomo filho recebe o valor zero. Caso seja mediana em algum dos dois cromossomos, então ficará armazenado em uma lista de possíveis medianas, podendo ser aproveitado posteriormente de forma aleatória para completar as medianas faltantes para o filho gerado.

4.2.7.3. Mutação

O **operador de mutação** modifica o valor de algum gene (escolhido aleatoriamente) de um indivíduo, fornecendo assim meios para introdução de novos elementos na população. Desta forma, a mutação assegura que a probabilidade de se chegar a qualquer ponto do espaço de busca nunca será zero. O operador de mutação é aplicado aos indivíduos com uma probabilidade dada pela taxa de mutação Pm.

Geralmente se utiliza uma taxa de mutação pequena, na ordem de uma mutação a cada mil genes.

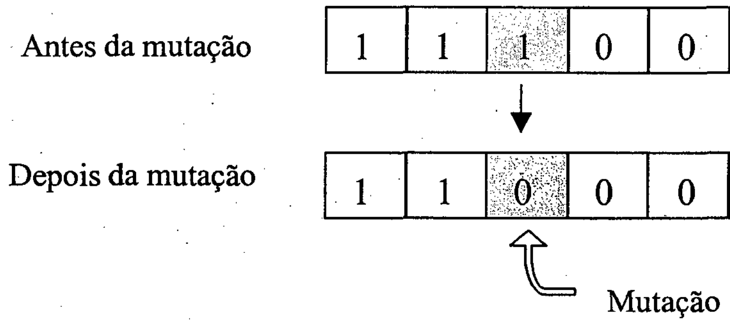


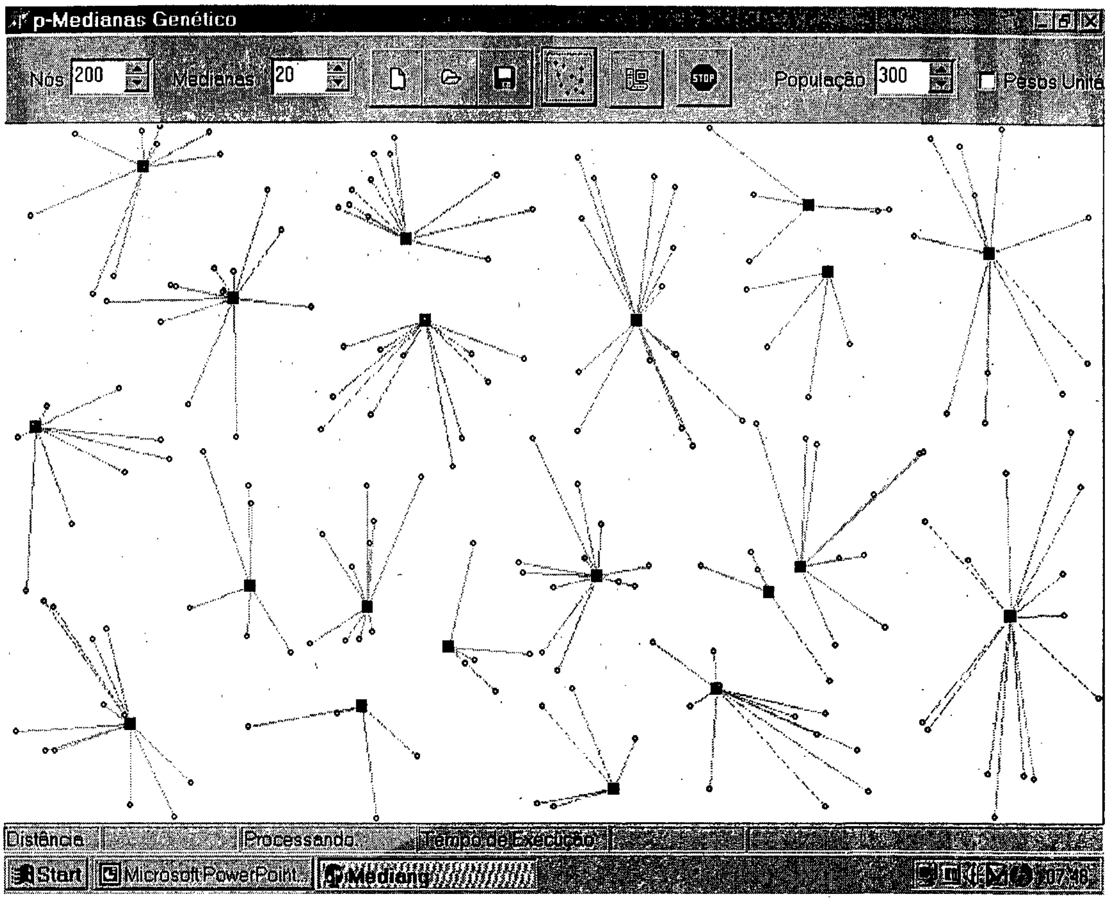
Figura 4.10. : Mutação

Com relação à mutação, a heurística proposta acrescenta na lista de possíveis medianas alguns pontos escolhidos aleatoriamente para assim ter possibilidade de gerar soluções diferentes, quanto maior for a taxa de mutação maior serão as chances de gerar soluções diferentes.

Na heurística proposta após efetuar a mutação é calculado o fitness dos cromossomos gerados, e comparados com o fitness dos cromossomo geradores (pai), com o intuito de substituir o cromossomo do pior pai pelo cromossomo do filho.

4.2.8. Interface do Sistema Implementado

O sistema de localização de medianas implementado possui a seguinte interface



O sistema deverá ter como entrada o número de nós (vértices), cujo limite máximo é 1000. A segunda entrada necessária é o número de medianas a serem encontradas, que não poderá ultrapassar de 100. O terceiro dado de entrada é o número da população a ser gerada, e por último o quarto dado de entrada diz respeito aos pesos associados aos vértices, em se tratando do problema de p-medianas puro é associado um peso unitário.

O primeiro botão da barra de menu gera os vértices que foram determinados no passo anterior. O segundo botão da barra de menu abre um arquivo de p-medianas que tenha sido rodado e salvo no sistema. O terceiro botão da barra de menu salva o arquivo de medianas gerado no sistema. O quarto botão gera uma solução inicial aleatória com o número de nós e número de medianas determinados no primeiro passo. O quinto botão da barra de menu aplica a técnica de busca baseada em algoritmos genéticos proposta, e apresenta a solução encontrada para o problema. O sexto botão da barra de menu interrompe a execução do programa em um determinado momento.

Na barra de status é apresentada a distância total percorrida desde cada um dos vértices a sua mediana associada e o tempo de execução do programa.

4.2.9. Teste Computacional e Análise de Resultados

Esta pesquisa teve como foco, buscar uma solução para o problema de p-medianas. Para tanto foram gerados e testados vários exemplos. A heurística proposta mostrou um melhor desempenho para problemas de pequeno porte. Também se obteve bons resultados para problemas de grande porte, porém com custo computacional mais elevado.

Além de que, para efeito de análise e avaliação dos resultados obtidos foi implementada ao lado da heurística proposta, o método de substituição de vértices proposto por Teitz&Bart.

Os resultados computacionais obtidos com a aplicação do Algoritmo Genético, proposto para resolver o problema de localização de p-medianas acham-se elencados no quadro abaixo.

Ressalta-se que a tabela está organizada como um demonstrativo da investigação realizada. O número de vértices e o número de medianas (ver colunas 1 e 2) possibilitaram a geração de dois resultados. O primeiro, obtido através da execução do método proposto por Teitz & Bart. O segundo é o resultado da implementação da técnica de busca baseada em algoritmos genéticos, alvo da pesquisa proposta.

Observa-se que com 200 vértices e 40 medianas o primeiro método obteve melhor resultado. Possivelmente, tal resultado deva-se à quantidade de medianas, pois, à medida que o número de medianas a serem localizadas, é reduzido, o método proposto nesta pesquisa atinge resultados melhores e com um custo computacional equiparável ou menor que o proposto por Teitz&Bart.

Isto permite concluir que, reduzindo o número de medianas, o método genético revela-se mais eficiente em resultados e custos.

Outro fator importante, no desempenho do método proposto envolve o tamanho da população , pois quanto maior o número de indivíduos da população (soluções), melhor desempenho do algoritmo. Uma pequena população implica em resultados regulares.

Este fato remete para a necessidade de se estabelecer o número ideal de indivíduos da população para a execução do método proposto, que pode variar de acordo com o número de vértices e de medianas.

A tabela acima é representativa dos resultados resumidos da bateria de testes efetuada, pode-se observar que são mostrados resultados para problemas de pequeno porte, onde o tamanho das instancias são de 200, 150, 100, 80 e 50 nós (vértices), o número de medianas variando entre 5 e 40 medianas, e uma população de 3000 e 2000 cromossomos. Os resultados obtidos foram equiparáveis ao método de Teitz&Bart.

		Custo da Soluç ão (Distância)		
Nro de Vértices	Nro de Medianas	Teitz&Bart	Genéticos	Nro da População
200	40	7003,47	7078,93	3000
200	15	13987,95	14005,44	2000
200	10	17887,27	17876,96	2000
200	5	26886,34	26906,34	1000
150	15	10628,16	10576,88	2000
150	12	12298,32	12222,25	2000
150	10	13759,04	13697,27	2000
150	8	15556,52	15556,52	2000
150	5	20842,56	20573,1	2000
150	4	**22965,13**	**22965,13**	2000
150	3	27925,58	27674,95	2000
100	10	9098,18	9098,11	2000
100	8	10601,48	10452,48	2000
100	5	14083,44	13689,87	2000
80	15	4975,66	4982,89	2000
80	12	**5836,2**	**5836,2**	2000
80	7	8543,46	8453,46	2000
80	5	10975,82	10922,75	2000
50	15	**2676,79**	**2676,79**	2000
50	12	**3294,98**	**3294,98**	2000
50	10	3856,34	3798,69	2000
50	9	4097,37	4079,91	2000
50	8	4478,67	4474,08	2000
50	7	**4872,84**	**4872,84**	2000
50	5	**6462,29**	**6462,29**	2000

Tabela 4.1. : Resultados Computacionais Comparativos para Problemas com Custo Unitario

Para problemas de pequeno porte a heurística proposta obtém, na maioria dos casos, melhores resultados que o método de Teitz e Bart, com um ganho aproximado de 1%. Já para problemas de grande porte, embora ofereça bons resultados, o custo computacional é um pouco mais elevado. Cabe ressaltar que a heurística proposta no presente trabalho se comporta de modo semelhante tanto para o problema puro das p-medianas, quanto para o problema generalizado, corroborando-se, portanto, a viabilidade da heurística em ambos casos.

Na heurística proposta a busca da melhor solução para o problema é feita sobre uma população de pontos e não sobre um único ponto, reduzindo sensivelmente o risco da solução recair sobre um máximo ou mínimo local ao fazer a busca em diferentes áreas do espaço de solução.

Pode-se dizer que o algoritmo proposto, neste trabalho, realiza uma busca cega. A única exigência é o conhecimento do valor da função objetivo de cada indivíduo. Não há necessidade de qualquer outra informação ou heurística dependente do problema.

Outra característica importante da heurística proposta é a utilização de regras de transição probabilísticas e não determinísticas, para guiar uma busca altamente exploratória e estruturada, onde informações acumuladas nas iterações (gerações) anteriores são usadas para direcionar essa busca.

A heurística proposta foi implementada em Delphi 5.0. Todos os testes foram realizados em um computador com processador Intel II de 300 MHz 64 de Memória Ram.

O desempenho da heurística proposta é, em grande medida, o reflexo da boa construção dos seguintes fatores :

- Adequada representação dos cromossomos.
- Determinação do Fitness (função de avaliação) apropriado ao problema
- Adequado número de gerações e tamanho da população.

É importante também analisar de que maneira alguns parâmetros influenciam no comportamento da heurística proposta . Privilegia-se aqui parâmetros como: tamanho da população, taxa de cruzamento, taxa de mutação e intervalo de geração.

Tamanho da População : O tamanho da população afeta o desempenho global e a eficiência da heurística. Com uma população pequena o desempenho pode cair, pois deste modo a população fornece uma pequena cobertura do espaço de busca do problema. Enquanto uma grande população geralmente fornece uma cobertura representativa do domínio do problema, além de prevenir convergências prematuras para soluções locais ao invés de globais.

Taxa de Cruzamento : quanto maior for esta taxa, mais rapidamente novas estruturas serão introduzidas na população. Caso seja muito alta, estruturas com boas aptidões poderão ser retiradas mais rapidamente, e a maior parte da população será substituída. Valores muito altos provocam a perda de estruturas de alta aptidão, já valores baixos podem tornar o algoritmo muito lento.

Taxa de Mutação : uma baixa taxa de mutação previne que uma dada posição fique estagnada em um valor, além de possibilitar que se chegue em qualquer ponto do espaço de busca. Com uma taxa muito alta a busca se torna essencialmente aleatória. Na heurística proposta nenhum ponto do espaço de busca tem probabilidade zero de ser examinado.

Intervalo de Geração : controla a porcentagem da população que será substituída durante a próxima geração. Com um valor alto, a maior parte da população será substituída, com valores muito altos pode ocorrer perda de estruturas de alta aptidão. O valor baixo torna o algoritmo moroso.

Assim, é possível concluir que os processos que mais contribuem para a evolução são o crossover e a adaptação baseada na seleção/reprodução. Ao lado deles a mutação também desempenha um papel significativo, no entanto, segundo a literatura, seu grau de importância continua sendo assunto de debate.

4.3. Considerações Finais

Durante a pesquisa, foi possível constatar que a heurística proposta apresentou bons resultados para problemas de pequeno a médio porte. Faz necessário também ressaltar que a heurística pode conduzir a melhores resultados, se a população alvo for representada por um numero maior de indivíduos. Há, porém, que se atentar para o fato de que na ação junto a grandes populações, é preciso lançar mão de maiores recursos computacionais, ou que o algoritmo trabalhe por um período de tempo prolongado, de forma a atender as especificidades que um número maior de indivíduos gera.

CAPÍTULO V

5. CONCLUSÃO E RECOMENDAÇÕES

5.1. Considerações Finais

Esta pesquisa buscar enfocar as técnicas exatas e heurísticas mais utilizadas na resolução do problema de localização de p-medianas na atualidade. O enfoque se concentrou principalmente, nas chamadas técnicas heurísticas, por apresentarem bons resultados em tempos de processamento curtos. Em contraposição aos métodos exatos cujo tempo de processamento torna inviável o tratamento exato de problemas combinatoriais, como é o caso do problema de localização de p-medianas.

Nesse contexto foi proposta uma heurística, a fim de encontrar a melhor localização para o problema de p-medianas, baseada em algoritmos genéticos.

Após a aplicação dos programas computacionais, com base na analise de resultados, conclui-se que a heurística, proposta nesta dissertação, é viável para a resolução de problemas de p-medianas, apresentando resultados equiparáveis aos obtidos com o método de substituição de vértices, apontado na literatura como o método que melhor resultado oferece para o problema das p-medianas.

A heurística proposta mostrou-se adequada na resolução de problemas de localização de p-medianas, principalmente, para o caso de pequeno porte, onde o ganho é de 1% em relação ao método de Teitz&Bart.

Cabe ressaltar que a heurística proposta, comporta-se de modo semelhante tanto para o problema puro das p-medianas quanto para o problema generalizado, corroborando-se, portanto, a viabilidade da heurística em ambos casos.

Por último conclui-se, que se aplicada a heurística proposta, certamente ocorrerá uma redução no custo de transporte associado ao deslocamento de cada vértice a sua instalação mais próxima, reduzindo consideravelmente os custos totais.

Cabe ressaltar que a heurística proposta, mesmo utilizando o procedimento heurístico, fornece resultados sempre consistentes e, excepcionalmente, quando não atingem o ponto ótimo da função objetivo, ficam muito próximo dele, comprovando que é robusto e confiável além de ser de fácil aplicabilidade.

Isto posto, é preciso ainda atentar para o fato de que em termos de desenvolvimento cientifico, a contribuição desta dissertação encontra-se na implementação de uma técnica de busca para a localização de p-medianas, pois a implementação da heurística proposta permite encontrar a melhor localização em poucos segundos, o que pela busca exaustiva seria praticamente impossível.

5.2. Recomendações

Existem algumas modificações que se incorporadas à heurística proposta, poderão aumentar a sua performance. Uma primeira sugestão seria testar outros operadores de crossover e, paralelamente, poderiam ser testadas novas formas de mutação, inclusive aumentando a taxa da mutação para permitir a geração de novas soluções.

Outro ponto que poderia ser abordado é a implementação de algoritmos genéticos paralelos, usando o modelo celular, no qual os indivíduos estariam representados em uma matriz, onde cada individuo só poderá se reproduzir com os que estejam ao seu redor (mais perto da casa) em uma escolha aleatória ou por melhor adaptação o descendente passa a ocupar uma posição vizinha.

E finalmente, nesta pesquisa, aponta-se como outra questão a ser explorada implementação de estratégias híbridas, onde se pode obter ganho no número de iterações, o que implica a redução de tempo de processamento para problemas de grande porte.

REFERÊNCIAS BIBLIOGRAFICAS

ABDEL L.L. Optimum positioning of moving service facility. **Computers and Operations Research**, v.12, p. 437 – 444, 1985.

ADLAKHA, V and MERSHA, T. Two heuristic techniques for service location: An evaluation and comparasion. **Computers and Operations Research**, v.16, p. 533 – 540, 1989.

AHN, S.H. Structural study of the k–median problem. **Journal of the korean Operations Research society**, v. 20, p.101–113, 1995.

AHN,S.H. et all. Probabilistic analysis of a relaxation for the k–median problem. **Mathematics of Operation Research**, v.13, 1988.

BATTA, R. and LEIFER, L. A. On the accuracy of demand point solutions to the planar Manhattan metric p–median problem, with and without barriers to travel. **Computers and Operations Research**, v.15, p.253–262, 1988.

BAXTER, J. Depot location: A technique for the avoidance of local optima. **European Journal of Operations Research**, v.18, p. 208–214, 1984.

BERMAN, O., INGCO, D. I. and ODONI, A. R. Improving the location of minisum facilities through network modification. **Annals of Operations Research**, v. 40, p. 1–16, 1993.

BERMAN, O., LARSON, R. C. and PARKAN, C. The stochastic queue p–median problem. **Transportation Science**, v. 21, p. 207–216, 1987.

BERMAN, O. and YANG E. K. Medi–centre location problems. **Journal of the Operational Research Society**, v. 42, p. 313 – 322, 1991.

BEASLEY, J.E. A note on solving large p-median problems. **European Journal of Operational Research**. v.21, p.270, 1985.

BEASLEY, J.E. OR-Library: distribution test problems by electronic mail. **Journal of Operational Research Society**. v.41, p.1069, 1990.

BEASLEY, J.E. Lagrangean heuristic for Location problems. **European Journal of Operational Research**. v.65, p.383, 1993.

BINFARÉ, N. J. **Método para Localização de uma Sede Móvel na Realização de Inventario Forestal** . 1993. Dissertação de Mestrado da UFSC.

BOFFEY, T. B. and KARKAZIS, J. p–medians and multi–medians. **Journal of the Operational Research Society**, v. 35, p. 57 – 64, 1984.

CARVALHO, A. Algoritmos Genéticos. Disponível em: < http://www.icmsc.sc.usp.br/~andre/gene1.html>. Acesso em: maio de 2000.

CHRISTOFIDES, N. **Graph Theory : An Algorithm Approach.** Academic Press, New York, 1975.

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

DREZNER, Z. On the conditional p-median problem. **Computers and Operation Research**, 1995.

GALVÃO, R.D. O problema das p-medianas generalizado. **X Simpósio Brasileiro de Pesquisa Operacional**, São Paulo, v. 1, p.177-191, 1979.

GALVÃO, R.D.; RAGGI,L.A. A method for solving to optimality uncapacitated location problems. **Annals of Operation Research**. v.18, p. 225-244, 1989.

Genetic Algorithms Group. George Mason University, Fairfax, Virginia. Disponível em:< http://www.cs.gmu.edu/research/gag/>. Acesso em: 2000.

Genetic Algorithms Research and Applications Group. Michigan State University. Disponível em:< http://isl.msu.edu//GA/>. Acessado em: 2000.

GOLBERG, D. E.; MILMAN, K.; TIDD, C. **Genetic Algorithms : A Bibliography**, ILLIGAL, Urbana, 1992.

GONÇALVES, Miriam Buss. Modelos para localização de serviços emergenciais em rodovias. **XXVI Simpósio Brasileiro de Pesquisa Operacional**, Florianópolis, 1994.

HAKIMI S. L. Optimum distribution of switching center and the absolute centers and the medians of a graph. **Operations. Research.**, v. 12, p. 450, 1964.

HAKIMI S. L. Optimum distribution of switching center in a communication network and some related graph theoric problems, **Opns. Res.**, v. 13, p. 462, 1965.

HILLSMAN, E. L. The p-median structure as a unfilled linear model for location-allocation analysis. **Environment planning**, v. 16, p.305-318, 1984.

HODGSON, Jhon. The location of public facilities intermediate to journey to work. **European Journal of Operational Research**, v. 6, p. 199 – 204, 1981.

HRIBAR M. and DASKIN, M. A dynamic programming heuristic for the p-median problem, **European Journal of Operational Research**, v. 101, Number 3, setember 16, 1997.

IBARAKI, T. Combinatorial Optimization problems and their complexity, In: Enumerative Approaches to Combinatorial Optimization – Part I. **Annals of Operation Reserch**, v. 10, 1987.

KOZA, J. R. **Genetic Programming : On the Programming of Computers by Means of Natural Selection** , The Mit Press, 1992.

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

MAYERLE, S. F. **Um Sistema de Apoio à Decisão para o Planejamento Operacional de Empresas de Transporte Rodoviario Urbano de Passageiros. 1996.** Tese de Doutorado da UFSC.

MINIEKA, E. The Centers and Median of a graph. **Ops. Res.,** vol 25, no 4, p. 641-650, July-August, 1977.

NOVAES, A. G. **Sistemas Logísticos, Armazenagem e Distribuição Física de Produtos** , Editora Edgar Blucher Ltda, 1996.

PINTO, S. **Analise e Desenho de Algoritmos.** Disponível em : < http://home .dmat.uevora.pt/~spa/aulas/1998-99/s2/Ada/prog-ada-3.html>. Acesso 1999.

RABUSKE, M. A. **Introdução à Teoria dos Grafos**, Editora da UFSC, Florianópolis-SC, 1992.

SENNE, E.L.F.; LORENA, L.A.N. A lagrangean/surrogate approach to p-median problems. **European Journal of Operational Research**-submitted.1997.

SIANG. **Eficiência ou Complexidade de Algoritmos.** <Disponível em : < http://www.ime.usp.br/~song/cursos/complex.html>. Acesso em dezembro de 1997.

SOUZA, J. C. **Dimensionamento, Localização e Escalonamento de Serviços de Atendimento Emergencial** . 1996. Tese de Doutorado da UFSC.

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

TING, W.S. **Complexidade de Algoritmos**. Disponível em : < http://www.dca.fee.unicamp.br/~ting/Courses/ea869/faq1.html>. Acesso em: maio de 1997.

TOREGAS, C. et all. The location of emergency service facilities. **Operations Research**, v.19, p. 1363 – 1373, 1971.

VIEIRA V. et al, **Sistemas Inteligentes**, Editora da universidade, 1999.

WAKABAYASHI, Y. **Otimização Combinatória.** Disponível em : http://www.ime.usp.br/mac/otimizacao.comb.html . Acesso em 1996.

WESLEY. **Problemas Combinatoriais**. Disponivel em: <http://www.ufpi.br/~weslley/probsNP.htm>. Acesso em : 19 de julho 1999.

ZHAO, P. and BATTA, R. Analysis of centroid aggregation for the Euclidian distance p-median problem,. **European Journal of Operational Research**, v. 113, number 1, February 16, p. 147-168, 1999.