

ELON SANTOS CORRÊA



ALGORITMOS GENÉTICOS E BUSCA TABU APLICADOS AO PROBLEMA DAS P-MEDIANAS

Dissertação apresentada como requisito parcial
para a obtenção do grau de Mestre em Ciências,
Curso de Pós-Graduação em Métodos Numéricos
em Engenharia, Setores de Tecnologia e de
Ciências Exatas, Universidade Federal do Paraná.

Orientadora:

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

Co-orientadores:

Prof. Dr. Celso Carnieri

Prof. Ph.D. Alex A. Freitas

CURITIBA

2000

ELON SANTOS CORRÊA

ALGORITMOS GENÉTICOS E BUSCA TABU APLICADOS AO PROBLEMA DAS P-MEDIANAS

Dissertação apresentada como requisito parcial
para a obtenção do grau de Mestre em Ciências,
Curso de Pós-Graduação em Métodos Numéricos
em Engenharia, Setores de Tecnologia e de
Ciências Exatas, Universidade Federal do Paraná.

Orientadora:

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

Co-orientadores:

Prof. Dr. Celso Carnieri

Prof. Ph.D. Alex A. Freitas

CURITIBA

2000


TERMO DE APROVAÇÃO

ELON SANTOS CORRÊA


ALGORITMOS GENÉTICOS E BUSCA TABU APLICADOS AO PROBLEMA DAS P-MEDIANAS

Dissertação aprovada como requisito parcial para a obtenção do grau de Mestre em Ciências no Curso de Pós-Graduação em Métodos Numéricos em Engenharia da Universidade Federal do Paraná, pela comissão formada pelos professores:

Orientadora:


Prof.ª Dr.ª Maria Teresinha Arns Steiner
Departamento de Matemática, UFPR


Prof. Dr. Celso Carnieri
Departamento de Matemática, UFPR


Prof. Ph.D. Alex Alves Freitas
Departamento de Informática, PUC-PR

Curitiba, 18 de dezembro de 2000

What is jazz, Mr. Armstrong?

*My dear lady, as long as you have to ask that
question, you will never know it.*

AGRADECIMENTOS

À minha família, em especial à minha mãe Nilza Santos Corrêa e à minha irmã Kelly Cristina Corrêa Brigido pelo incentivo e apoio durante a realização de todo o curso. À Lucimar Dione Fernandes pela compreensão e carinho sempre presentes.

Ao meu pai Airton Bulhões Corrêa (*in memoriam*) pelo bom exemplo que sempre foi para mim.

Muitas pessoas me ajudaram a realizar as pesquisas descritas nesta dissertação. Espero que aqueles que eu tenha acidentalmente esquecido de mencionar aqui possam me perdoar. Gostaria de expressar minha gratidão às seguintes pessoas:

À professora Maria Teresinha Arns Steiner, pela orientação durante a realização das pesquisas e que com paciência invulgar, indicou sempre a melhor forma de conduzir as pesquisas.

Ao professor Celso Carnieri, pela co-orientação e valiosas sugestões.

Ao professor Alex Alves Freitas, pela co-orientação, disposição em ajudar e pela revisão do texto o que contribuiu para torná-lo mais claro em alguns pontos, e livrá-lo de erros em outros.

Aos professores Celso Carnieri, Maria Teresinha Arns Steiner, Rubens R. Ortega e Anselmo Chaves Neto, pelos ensinamentos que transmitiram.

Aos colegas, pela amizade formada durante o curso, em especial aos amigos Danielle Durski Figueiredo, Sérgio Augusto Fleischfresser e Raimundo Ronilson Leal do Rosário.

Aos funcionários do Centro de Estudos de Engenharia Civil da Universidade Federal do Paraná, em especial à Maristela Bandil e Elizeu dos Santos, que souberam dar sua parcela de contribuição nessa jornada e procuraram na amizade o termo comum de diálogo.

A Administração do Colégio Militar de Curitiba pelas dispensas concedidas para a realização do curso e aos colegas de trabalho pela força que deram, muitas vezes tendo que suprir a minha ausência.

A todo o pessoal da Comissão Central do Concurso Vestibular da Universidade Federal do Paraná em especial aos professores Dartagnan Baggio Emerenciano e Mildred Ballin Hecke e aos funcionários Maria Cleusa de Souza Henklein e Marcus Vinicius Lemos do Prado pelo incentivo e ajuda com os dados do vestibular 2001.

A Liliana Luisa Pizzolato da Biblioteca Central da Universidade Federal do Paraná pela gentileza em colaborar com sugestões e valiosos conselhos sobre a elaboração da presente dissertação.

Ao técnico responsável pela localização dos endereços no mapa digitalizado de Curitiba, Vladimir Biermann Silveira.

A todos o meu sincero obrigado!

SUMÁRIO

RESUMO	xi
ABSTRACT	xii
1 INTRODUÇÃO	1
1.1 ESTRUTURA DO TRABALHO	2
2 DESCRIÇÃO DO PROBLEMA DAS P-MEDIANAS	3
2.1 PROBLEMA DAS P-MEDIANAS	3
2.1.1 Definição do Problema das p-medianas	3
2.1.2 Problema das p-medianas Capacitado	5
2.3 ALGORITMO DE TEITZ E BART	5
2.3.1 Procedimentos Básicos do Algoritmo das p-medianas de Teitz e Bart	7
2.4 UM PROBLEMA REAL	8
2.4.1 Obtenção dos Dados do Problema Real	12
3 MÉTODOS UTILIZADOS	14
3.1 ALGORITMO DE GILLET E JOHNSON	14
3.2 ALGORITMOS GENÉTICOS	15
3.2.1 O que são Algoritmos Genéticos?	15
3.2.2 Qual a Origem dos AGs?	16
3.2.3 Conceitos Fundamentais e Terminologia	17
3.2.4 Em que os AGs Diferem de outras Heurísticas de Busca?	19
3.2.5 Inicialização de um Algoritmo Genético	19
3.2.6 Avaliação da Adaptabilidade (<i>fitness</i>)	20
3.2.7 Seleção	20
3.2.8 Operadores Genéticos (OGs)	21
3.2.8.1 Reprodução	21
3.2.8.2 Cruzamento	21
3.2.8.3 Mutação	23
3.2.9 Condição de Término	24
3.2.10 Convergência Prematura	24
3.2.11 Parâmetros dos Operadores Genéticos	24
3.2.11.1 Tamanho P da população	25
3.2.11.2 Probabilidades de cruzamento e de mutação	25
3.2.12 Critérios de Seleção	25

3.2.12.1	Roleta simples	26
3.2.12.2	<i>Ranking</i>	26
3.2.13	Método do Estado Estável	27
3.3	BUSCA TABU	28
3.3.1	O que é Busca Tabu?	28
3.3.2	Origem da Busca Tabu	28
3.3.3	Como funciona a Heurística Busca Tabu?	28
3.3.4	Busca através das Soluções Vizinhas	29
3.3.5	Critérios de Aspiração	29
3.3.6	Estratégia da Oscilação	30
3.3.7	Algumas Iterações de um Algoritmo BT Elementar	30
3.3.7.1	Apresentando o problema fictício	30
3.3.7.2	As iterações do problema fictício	32
4	REVISÃO DA LITERATURA	36
4.1	LOCALIZAÇÃO DE INSTALAÇÕES	36
4.2	ALGORITMOS GENÉTICOS	36
4.3	BUSCA TABU	39
5	IMPLEMENTAÇÃO DOS MÉTODOS UTILIZADOS	42
5.1	ALGORITMO PARA DESIGNAÇÃO DE CANDIDATOS AO VESTIBULAR	42
5.1.1	Procedimentos Básicos do DeCan	45
5.1.2	Melhorias no Resultado Final do DeCan	46
5.1.2.1	Análise pós-processamento	47
5.1.2.1.1	Procedimentos básicos da análise pós-processamento	48
5.2	ALGORITMO GENÉTICO PROPOSTO (AGpMcap)	49
5.2.1	Representação Cromossômica	50
5.2.2	Tamanho da População	50
5.2.3	Avaliação do Indivíduo (<i>fitness</i>)	51
5.2.3.1	Procedimentos básicos do DeCan simplificado	52
5.2.4	Seleção dos Cromossomos Pais	53
5.2.5	Vetores de Troca	54
5.2.6	Operador Genético de Cruzamento	54
5.2.7	Operador Genético de Reprodução	55
5.2.8	Operador de Mutação	55
5.2.9	Hipermutação Direcionada	56
5.2.10	Condição de Término	57
5.2.11	Procedimentos Básicos do AGpMcap	57

5.2.11.1	Notação utilizada	57
5.2.11.2	O AGpMcap passo a passo	58
5.3	HEURÍSTICA BUSCA TABU PROPOSTA (BTpMcap)	60
5.3.1	Solução Inicial	60
5.3.2	Movimentos e Restrições Tabu	61
5.3.2.1	Procedimento Escolha_Movimento passo a passo	62
5.3.3	Restrições Tabu	63
5.3.4	Critérios de Aspiração (Critérios para cancelamento das restrições tabu)	64
5.3.5	Diversificação e Memória de Longo Prazo	64
5.3.6	Estratégia da Oscilação	66
5.3.7	Procedimentos Básicos da Heurística BTpMcap.....	67
5.3.7.1	Notação utilizada	67
5.3.7.2	A heurística BTpMcap passo a passo	67
6	TESTES COMPUTACIONAIS E RESULTADOS	69
6.1	RESULTADOS COMPUTACIONAIS OBTIDOS PELO DeCan	69
6.2	TESTES COMPUTACIONAIS DAS HEURÍSTICAS AGpMcap E BTpMcap	73
6.2.1	Resultados Computacionais Obtidos pelo AGpMcap	75
6.2.2	Resultados Computacionais Obtidos pela BTpMcap	79
7	ANÁLISE DOS RESULTADOS E CONCLUSÕES	82
7.1	ANÁLISE DOS RESULTADOS OBTIDOS PELO AGpMcap	82
7.2	ANÁLISE DOS RESULTADOS OBTIDOS PELA BTpMcap	83
7.3	COMPARANDO AGpMcap E BTpMcap	84
7.4	CONCLUSÕES	85
7.5	SUGESTÕES PARA TRABALHOS FUTUROS	85
	REFERÊNCIAS	87

LISTA DE TABELAS

TABELA 2.1 - CIDADES ONDE SERÃO APLICADAS AS PROVAS DO VESTIBULAR 2001 DA UFPR	9
TABELA 2.2 - OS 26 LOCAIS UTILIZADOS PARA A DESIGNAÇÃO DE 38.711 CANDIDATOS AO VESTIBULAR 2001 DA UFPR	11
TABELA 2.3 - MATRIZ DE DISTÂNCIAS RESIDÊNCIA X LOCAIS DE PROVAS	12
TABELA 3.1 - RELAÇÃO DA TERMINOLOGIA DOS AGs COM A BIOLOGIA	18
TABELA 3.2 - DADOS PARA ROLETA SIMPLES	26
TABELA 3.3 - DADOS PARA <i>RANKING</i>	27
TABELA 6.1 - RESULTADOS COMPUTACIONAIS OBTIDOS PELO DeCan E UTILIZADOS PELA CCCV – UFPR NO VESTIBULAR 2001	70
TABELA 6.2 - RESULTADOS COMPUTACIONAIS OBTIDOS PELO DeCan ALTERADO PARA O CÁLCULO DA RAZÃO	70
TABELA 6.3 - RESULTADOS COMPUTACIONAIS OBTIDOS PELO DeCan EXCLUINDO-SE 4 LOCAIS COM POUCAS VAGAS	71
TABELA 6.4 - OS 26 LOCAIS UTILIZADOS PARA A DESIGNAÇÃO DOS 19.710 CANDIDATOS ATRAVÉS DO DeCan COM SUAS RESPECTIVAS CAPACIDADES TOTAIS	72
TABELA 6.5 - RESULTADOS COMPUTACIONAIS OBTIDOS PELO DeCan PARA 26 LOCAIS COM AS CAPACIDADES TOTAIS	73
TABELA 6.6 - RELAÇÃO DOS 43 LOCAIS DE PROVAS UTILIZADOS NAS SIMULAÇÕES	74
TABELA 6.7 - RESULTADOS COMPUTACIONAIS OBTIDOS PELO AGpMcap NA PRIMEIRA SIMULAÇÃO (COM HIPERMUTAÇÃO DIREC.)	76
TABELA 6.8 - RELAÇÃO DOS 26 LOCAIS DE PROVAS SELECIONADOS PELO AGpMcap NA PRIMEIRA SIMULAÇÃO REALIZADA	77
TABELA 6.9 - RESULTADOS COMPUTACIONAIS OBTIDOS PELO AGpMcap NA SEGUNDA SIMULAÇÃO (SEM HIPERMUTAÇÃO DIREC.)	78
TABELA 6.10 - RELAÇÃO DOS 26 LOCAIS DE PROVAS SELECIONADOS PELO AGpMcap NA SEGUNDA SIMULAÇÃO	78
TABELA 6.11 - RESULTADOS COMPUTACIONAIS OBTIDOS PELA BTpMcap NA SIMULAÇÃO REALIZADA	80
TABELA 6.12 - RELAÇÃO DOS 26 LOCAIS DE PROVAS SELECIONADOS PELA BTpMcap NA SIMULAÇÃO REALIZADA	80
TABELA 7.1 - RESULTADOS OBTIDOS NAS SIMULAÇÕES COMPUTACIONAIS REALIZADAS COM O AGpMcap E A BTpMcap	84

LISTA DE ILUSTRAÇÕES

FIGURA 3.1 - REPRESENTAÇÃO DE UM CROMOSSOMO NA BIOLOGIA E EM AGs	18
FIGURA 3.2 - EXEMPLO DE CRUZAMENTO UNIFORME	22
FIGURA 3.3 - EXEMPLO DE CRUZAMENTO COM 1 - PARTIÇÃO	22
FIGURA 3.4 - EXEMPLO DE CRUZAMENTO COM 2 - PARTIÇÕES	23
FIGURA 3.5 - EXEMPLO DE MUTAÇÃO POR TROCA SIMPLES	23
GRÁFICOS 3.1 - COMPARAÇÃO ROLETA SIMPLES X <i>RANKING</i>	27
FIGURA 3.6 - UMA SOLUÇÃO INICIAL	31
FIGURA 3.7 - EXECUÇÃO DO MOVIMENTO [5, 6] GERANDO UMA NOVA SOLUÇÃO	31
ESQUEMA 3.1 - ITERAÇÃO 0	32
ESQUEMA 3.2 - ITERAÇÃO 1	33
ESQUEMA 3.3 - ITERAÇÃO 2	33
ESQUEMA 3.4 - ITERAÇÃO 3	34
ESQUEMA 3.5 - ITERAÇÃO 4	35

RESUMO

Os problemas de localização de instalações (*facility location*) possuem várias aplicações como em telecomunicações, distribuição e transporte industrial. Um dos problemas de localização de instalações mais conhecido é o problema das p-medianas (HAKIMI, 1965) e (REVELLE, 1970). Neste trabalho é apresentada uma aplicação do problema das p-medianas capacitado a um problema real. É proposto um algoritmo que otimiza a designação de candidatos ao vestibular para os locais de provas mais próximos de suas residências. Para resolver o problema das p-medianas capacitado são propostas duas heurísticas modernas adaptadas ao problema. A primeira é baseada em um algoritmo genético simples que utiliza os operadores genéticos usuais e um operador heurístico chamado “hipermutação direcionada”. A segunda heurística proposta é baseada em busca tabu e usa memória de curto e de longo prazo para controlar a busca. Também utiliza uma estratégia de oscilação e tempo tabu aleatório para tentar evitar a repetição de soluções. As duas heurísticas propostas são utilizadas para resolver o problema real mencionado anteriormente, caracterizado como das p-medianas capacitado.

Palavras-chave: localização de instalações, p-medianas, designação, algoritmos genéticos, busca tabu.

ABSTRACT

Facility location problems have several applications in telecommunications, distribution and industrial transportation. One of the most well known facility location problems is the p-median problem (HAKIMI, 1965), (REVELLE, 1970). This work presents an application of the capacitated p-median problem to a real-world problem. This work proposes an algorithm that optimizes the designation of candidate students (who have to pass a university admission exam) to exam facilities closer to their residences. In order to solve the capacitated p-median problem we propose two modern heuristics adapted to the problem. The first one is based on a simple genetic algorithm that uses both conventional genetic operators and a new heuristic operator called “directed hipermutation”. The second one is based on tabu search and uses both short-term and long-term memory to control the search. It also uses an oscillation strategy and random tabu tenure in an attempt to avoid the generation of repeated solutions. The two proposed heuristics are used to solve the above-mentioned real-world problem, cast as a capacitated p-median problem.

Keywords: facility location, p-median, designation, genetic algorithms, tabu search.

1 INTRODUÇÃO

Os problemas de localização de instalações (*facility location*) possuem várias aplicações como em telecomunicações, distribuição e transporte industrial. Um dos problemas de localização de instalações mais conhecido é o problema das p -medianas (HAKIMI, 1965) e (REVELLE, 1970). Este problema consiste em localizar p instalações em um espaço considerado (espaço euclidiano, por exemplo) que devem atender a n pontos de demanda de tal forma que a soma das distâncias percorridas de cada ponto de demanda até a instalação mais próxima seja minimizada.

No caso do problema das p -medianas não-capacitado, considera-se que cada instalação candidata a mediana pode atender a um número ilimitado de pontos de demanda. No caso do problema das p -medianas capacitado, cada instalação candidata a mediana possui uma capacidade limite fixa e não pode atender a mais pontos de demanda do que sua capacidade permite (TRAGANTALERNGSAK et al., 1999). O problema das p -medianas pertence a uma classe de problemas chamados *NP-hard* (KARIV e HAKIMI, 1979) (até o momento, não foi encontrado um algoritmo que os resolva de forma exata cujo número de operações seja limitado por um polinômio no número de dados do problema). Consequentemente, até mesmo as heurísticas especializadas em resolver estes problemas requerem considerável esforço computacional. Uma das heurísticas mais conhecidas para resolver o problema das p -medianas é o algoritmo de TEITZ e BART (1968). Esta heurística é baseada na substituição de medianas na solução e seu objetivo é, a partir de uma solução inicial, melhorar o valor da função objetivo a cada iteração. Esta heurística é fácil de ser implementada e produz boas soluções para problemas pequenos, principalmente quando aplicada várias vezes ao mesmo problema com diferentes soluções iniciais.

Recentemente muitos pesquisadores têm focalizado suas pesquisas em desenvolver heurísticas modernas para resolver o problema das p -medianas. Em geral, estas heurísticas podem gerar resultados melhores do que uma heurística que

simplesmente substitui medianas na solução. Algumas heurísticas modernas como os algoritmos genéticos e busca tabu têm demonstrado que são adequadas para o problema das p -medianas (ERKUT, em fase de publicação). Os algoritmos genéticos são algoritmos computacionais de busca baseados nos mecanismos de evolução natural e na genética e podem ser aplicados com sucesso a inúmeros problemas combinatórios (GOLDBERG, 1994). A busca tabu é uma heurística computacional de busca e utiliza uma “memória” que tenta impedir que a busca fique restrita a uma determinada área do espaço de busca considerado (GLOVER, 1995).

1.1 ESTRUTURA DO TRABALHO

Na seção 2 o problema das p -medianas e p -medianas capacitado são formalmente apresentados. Na seção 3 são apresentados os métodos utilizados para a implementação de dois algoritmos propostos no presente trabalho para resolver o problema das p -medianas capacitado. Na seção 4 são feitas referências a alguns trabalhos anteriormente publicados sobre localização de instalações, algoritmos genéticos e busca tabu aplicados ao problema das p -medianas. Na seção 5 são propostos os algoritmos utilizados no presente trabalho para resolver o problema das p -medianas capacitado. Na seção 6 os algoritmos desenvolvidos no presente trabalho são submetidos a testes computacionais. Na seção 7 são feitas a análise dos resultados obtidos nos testes computacionais da seção 6 e as conclusões do presente trabalho.

2 DESCRIÇÃO DO PROBLEMA DAS P-MEDIANAS

Esta seção é o ponto de partida desse trabalho. Nela, o problema de localização de instalações chamado problema das p-medianas é definido. Esta definição é estendida a um caso particular do problema das p-medianas chamado problema das p-medianas capacitado. É feita também a descrição de um problema real que pode ser interpretado como um modelo real do problema das p-medianas capacitado.

2.1 PROBLEMA DAS P-MEDIANAS

O objetivo do problema das p-medianas é determinar p instalações em um conjunto predefinido com n ($n > p$) instalações candidatas que deverão atender a um conjunto existente de demandas de forma que, a soma total das distâncias percorridas de cada ponto de demanda até a instalação mais próxima seja a mínima possível. As p instalações que pertencerem a uma solução qualquer para o problema são chamadas de medianas (por isso, p-medianas).

2.1.1 Definição do Problema das p-medianas

Considerando-se todos os vértices de um grafo dado como potenciais medianas, o problema das p-medianas pode ser definido como segue: seja $G = (V, A)$ um grafo não direcionado onde V são os vértices e A as arestas. Deve-se encontrar um conjunto de vértices $V_p \subset V$ (conjunto de medianas) com cardinalidade p , tal que a soma das distâncias de cada vértice restante em $\{V - V_p\}$ (conjunto das demandas) até seu vértice mais próximo em V_p seja a mínima possível.

É apresentada a seguir uma formulação do problema das p-medianas como um problema de Programação Inteira, desenvolvida por REVELLE e SWAIN (1970). Esta formulação permite que cada vértice do grafo seja considerado, ao mesmo tempo,

como demanda e instalação (potencial mediana), embora em muitos casos demandas e instalações pertençam a conjuntos disjuntos.

$$\text{Min } \sum_{i=1}^n \sum_{j=1}^n a_i d_{ij} x_{ij} \quad (2.1.1)$$

sujeito a:

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, 2, \dots, n \quad (2.1.2)$$

$$x_{ij} \leq y_j, \quad i, j = 1, 2, \dots, n \quad (2.1.3)$$

$$\sum_{j=1}^n y_j = p \quad (2.1.4)$$

$$x_{ij}, y_j \in \{0, 1\}, \quad i, j = 1, 2, \dots, n \quad (2.1.5)$$

onde,

n = número total de vértices do grafo

a_i = demanda do vértice i .

d_{ij} = distância do vértice i ao vértice j .

p = número de instalações utilizadas como medianas.

$$x_{ij} = \begin{cases} 1, & \text{se o vértice } i \text{ for designado para a instalação } j \\ 0, & \text{caso contrário} \end{cases}$$

$$y_j = \begin{cases} 1, & \text{se o vértice } j \text{ for uma instalação utilizada como mediana} \\ 0, & \text{caso contrário} \end{cases}$$

A função objetivo (2.1.1) minimiza a soma das distâncias (com pesos) dos vértices de demanda até o conjunto de medianas. O conjunto de restrições (2.1.2) garante que todos os vértices demanda serão designados para exatamente uma única mediana. O conjunto de restrições (2.1.3) proíbe que um vértice demanda seja designado para uma instalação que não esteja selecionada como mediana. O número total de vértices medianas (instalações selecionadas) é definido pela restrição (2.1.4) como sendo igual a p . A restrição (2.1.5) garante que os valores das variáveis de decisão x e y sejam binários (0 ou 1).

2.1.2 Problema das p-medianas Capacitado

Considerando-se todos os vértices de um grafo dado como potenciais medianas, o problema das p-medianas capacitado pode ser definido como segue: seja $G = (V, A)$ um grafo não direcionado onde V são os vértices, todos com restrições de capacidade, e A as arestas. Deve-se encontrar um conjunto de vértices $V_p \subset V$ (conjunto de medianas) com cardinalidade p , tal que a soma das distâncias de cada vértice restante em $\{V - V_p\}$ (conjunto das demandas) até seu vértice mais próximo em V_p seja a mínima possível. Além disso, deve-se garantir que todos os pontos de demanda sejam atendidos sem que as restrições de capacidade das instalações medianas sejam violadas.

Em relação ao problema das p-medianas, o problema das p-medianas capacitado possui as seguintes restrições adicionais:

- a) cada instalação pode suprir apenas um número limitado de demandas (restrições de capacidade);
- b) todos os pontos de demanda devem ser atendidos dentro das capacidades das respectivas instalações selecionadas como medianas.

2.3 ALGORITMO DE TEITZ E BART

Uma das heurísticas mais conhecidas para o problema das p-medianas é a desenvolvida por TEITZ e BART (1968) e é conhecida como **algoritmo das p-medianas de TEITZ e BART**. Esta heurística é baseada na substituição de vértices e seu objetivo é, a partir de uma solução inicial, melhorar o valor da função objetivo a cada iteração. Esta heurística é fácil de ser implementada e produz boas soluções para problemas pequenos, principalmente quando aplicada várias vezes ao mesmo problema com diferentes soluções iniciais.

Considerando-se todos os v rtices de um grafo dado como potenciais medianas, o algoritmo de TEITZ e BART para o problema das p-medianas pode ser explicado como segue: seja $G = (V, A)$ um grafo n o direcionado onde V s o os v rtices e A as arestas. Seja v_i um v rtice qualquer pertencente a V . Chama-se **n mero de transmiss o**   soma das menores dist ncias existentes entre o v rtice v_i e todos os outros v rtices do grafo.

Seja n o n mero total de v rtices do grafo, o n mero de transmiss o   dado por:

$$\sigma(v_i) = \sum_{j=1}^n w_j d(v_i, v_j), v_i, v_j \in V \quad (2.3.1)$$

onde,

$d(v_i, v_j)$   a menor dist ncia entre v_i e v_j e w_j   um peso associado ao v rtice v_j .

Assim, v_m   uma mediana se, entre todos os v rtices do grafo,   aquele que produz a menor soma total das dist ncias desde si pr prio at  cada um dos outros v rtices do grafo.

Ou seja:

$$\sigma(v_m) = \text{m nimo}[\sigma(v_i)], \forall v_i \in V. \quad (2.3.2)$$

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

$$d(V_p, v_j) = \text{m nimo}[d(v_i, v_j)], \forall v_i \in V_p, v_j \in V \quad (2.3.3)$$

e

$$\sigma(V_p) = \sum_{j=1}^n w_j d(V_p, v_j), \forall v_j \in V. \quad (2.3.4)$$

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

$$\sigma(V_{p_solução_ótima}) = \text{mínimo}[\sigma(V_p)], \forall V_p \subset V. \quad (2.3.5)$$

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

2.3.1 Procedimentos Básicos do Algoritmo das p -medianas de Teitz e Bart

São descritos a seguir os procedimentos básicos executados pelo algoritmo das p -medianas de TAITZ e BART.

Passo 0

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

Passo 1

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

Passo 2

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

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

$$\Delta_{ij} = \sigma(V_p) - \sigma(V_p \cup \{v_i\} - \{v_j\}), \forall v_j \in V_p. \quad (2.3.6)$$

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

Se $\Delta_{ij_máximo} > 0$ então:

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

Rotule v_j como “analisado”.

Caso contrário continue.

Rotule v_i como “analisado”.

Passo 3

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

Volte ao Passo 2 e continue a execução do algoritmo.

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

Fim.

A seguir é descrito um problema real que pode ser interpretado como um exemplo concreto do problema das p-medianas capacitado.

2.4 UM PROBLEMA REAL

No ano de 1912 foi fundada na cidade de Curitiba (PR) a Universidade Federal do Paraná (UFPR) – a Universidade mais antiga do Brasil. Nestes 88 anos a instituição evoluiu abrindo novos cursos, ganhando reconhecimento nacional e internacional através de suas pesquisas e pesquisadores e estreitando suas relações com a sociedade através da extensão. Atualmente a instituição mantém 61 opções de cursos de graduação, 84 de especialização, 37 de mestrado e 21 de doutorado.

O concurso vestibular é a principal via de ingresso aos cursos de graduação ofertados pela UFPR. A seleção de candidatos para ingresso nos cursos de graduação da UFPR é realizada mediante concurso vestibular, que consiste na aplicação de provas que avaliam conhecimentos do ensino médio ou curso equivalente. Podem candidatar-se às vagas os interessados que estejam cursando a última série do ensino médio ou aqueles que já possuem certificado de conclusão do ensino médio ou curso equivalente.

O setor responsável pela organização do concurso vestibular da UFPR é a Comissão Central do Concurso Vestibular (CCCV) instituída em caráter permanente desde 1973. Desde então, a CCCV vem aprimorando a forma do processo seletivo do vestibular, sempre buscando selecionar o candidato melhor preparado, ofertando serviços ao estudante dentro dos melhores padrões de segurança e qualidade.

Para o concurso vestibular 2001 a aplicação das provas será realizada em oito diferentes cidades conforme a tabela 2.1, ficando a critério do candidato a escolha da cidade onde deseja realizar as provas. A constituição das turmas e dos locais de provas nas respectivas cidades onde as provas serão realizadas são de exclusiva responsabilidade da CCCV e serão conhecidos pelos candidatos mediante edital divulgado na data e nos locais previstos no Guia do Candidato.

TABELA 2.1 - CIDADES ONDE SERÃO APLICADAS AS PROVAS DO VESTIBULAR 2001 DA UFPR

CIDADE	UF
Curitiba	PR
Ponta Grossa	PR
Cascavel	PR
Londrina	PR
Maringá	PR
Palotina	PR
Pato Branco	PR
Joinville	SC

Até o presente ano não existia por parte da CCCV nenhum método específico para a designação dos candidatos ao vestibular aos locais provas; nem para a determinação destes locais dentro da cidade. Em geral, a designação dos candidatos era feita de forma aleatória e a seleção dos locais de provas considerava apenas atender a demanda total de candidatos. Evidentemente, os locais eram selecionados de tal forma que não ocorresse uma grande concentração de locais de provas muito próximos.

Para o concurso vestibular 2001, foi proposta uma otimização na designação dos candidatos ao vestibular aos locais de provas. O objetivo foi designar o maior número possível dos candidatos inscritos para realizar as provas em Curitiba e que

apresentaram endereço de residência no município de Curitiba para os locais de provas mais próximos de suas residências. Desta forma, de um total de 38.711 candidatos inscritos para realizar provas em Curitiba, 19.710 foram designados através de um algoritmo proposto no presente trabalho chamado “algoritmo da designação dos candidatos aos locais de provas” (DeCan) que é apresentado na seção 5. Este algoritmo otimiza a designação dos candidatos ao vestibular aos locais de provas mais próximos de suas residências.

A diferença entre os 38.711 candidatos inscritos e 19.710 submetidos a designados através do DeCan deve-se aos seguintes fatores:

- a) alguns candidatos portadores de deficiência física necessitam de designação especial e foram designados pela própria CCCV;
- b) alguns candidatos apresentaram endereços de outros municípios (por exemplos, região metropolitana, outras cidades ou estados) diferentes de Curitiba, e tiveram que ser designados pela CCCV;
- c) alguns candidatos apresentaram endereços do município de Curitiba mas com dados incorretos o que impossibilitou a localização dos mesmos no mapa digitalizado do município de Curitiba e portanto também foram designados pela CCCV.

Para atender os 38.711 candidatos ao vestibular 2001 que farão provas em Curitiba a CCCV selecionou 26 locais de provas, todos dentro do município de Curitiba entre 43 locais disponíveis para a realização das provas no mesmo município. A seleção dos 26 locais de provas foi realizada somente com base na experiência pessoal dos membros da CCCV sem a utilização de qualquer método computacional a não ser a intuição humana. A tabela 2.2 apresenta os 26 locais de provas selecionados e utilizados para a designação dos 38.711 candidatos ao vestibular 2001 da UFPR. Nas capacidades dos locais apresentados na tabela 2.2 já estão descontados os 19.001 candidatos classificados num dos três casos citados anteriormente e que foram designados para os locais de provas pela própria CCCV.

TABELA 2.2 - OS 26 LOCAIS UTILIZADOS PARA A DESIGNAÇÃO DE 38.711 CANDIDATOS AO VESTIBULAR 2001 DA UFPR

LOCAL	NOME DA INSTITUIÇÃO	CAPACIDADE RESTANTE
1	UFPR - Centro Politécnico	1973
2	PUC – Pontifícia Univ. Católica	7837
3	Col. Estadual Rio Branco	4
4	Col. Estadual do Paraná	123
5	Col. Nossa Senhora Medianeira	1232
6	UFPR - Setor de Ciências Agrárias	1300
7	Col. Estadual Leôncio Correia	1265
8	UFPR – Edifício Dom Pedro I - Reitoria	2
9	UFPR – Edifício Dom Pedro II - Reitoria	4
10	Fund. Estudos Sociais PR - Fesp	290
11	Col. Est. Prof. Loureiro Fernandes	1020
12	Esc. Municipal Pref. Omar Sabbag	960
13	Col. Est. Prof.a. M. Aguiar Teixeira	630
14	UFPR - Depto. de Educação Física	480
15	Col. Padre João Bagozzi	2165
16	Col. Estadual Pedro Macedo	780
17	Instituto de Educação do Paraná	3
18	Col. Estadual Cecília Meireles	1064
19	Escola Social Madre Clélia	1025
20	Col. Estadual Paula Gomes	880
21	Colégio Paranaense	1114
22	UFPR - Setor C. Saúde - Sede Botânico	770
23	Colégio Militar de Curitiba	1348
24	Esc. Est. República O. do Uruguai	710
25	Colégio Estadual Paulo Leminski	1259
26	UFPR - C. Ciências Florestais e Madeira	590
SOMA DAS CAPACIDADES RESTANTES		28828

São apresentados a seguir alguns dos procedimentos referentes a obtenção dos dados necessários para a designação dos 19.710 candidatos ao vestibular 2001 designados, a pedido da CCCV, pelo algoritmo proposto na seção 5. Estes procedimentos antecederam a aplicação dos métodos utilizados no presente trabalho para a real designação dos candidatos apresentada na seção 6 assim como para a realização de todos os testes computacionais descritos também na seção 6.

2.4.1 Obtenção dos Dados do Problema Real

Uma empresa de informática que desenvolveu um mapa digitalizado do município de Curitiba calculou e forneceu as coordenadas geográficas da residência de cada um dos 19.710 candidatos e de cada um dos locais de provas em relação ao mesmo mapa. As coordenadas referentes às residências dos candidatos foram calculadas com base no endereço declarado pelo candidato por ocasião da sua inscrição para o concurso vestibular.

Obtidas as coordenadas das residências de todos os candidatos e dos locais de provas foi calculada a matriz das distâncias retangulares entre a residência de cada candidato e os locais conforme ilustra a tabela 2.3. Foi utilizada a distância retangular e não a euclidiana porque segundo ERKUT (Comunicação pessoal do autor, 06 de setembro de 2000) como dentro da maioria das cidades não se pode viajar sempre em linha reta a distância euclidiana, em geral, avalia de forma imprecisa a distância real percorrida. O ideal seria utilizar a distância real, levando-se em conta o verdadeiro caminho percorrido, mas, estes dados são difíceis de serem obtidos. A distância retangular também é imprecisa na avaliação da distância percorrida mas, na média, tende a se aproximar um pouco mais da distância real percorrida.

TABELA 2.3 - MATRIZ DE DISTÂNCIAS RESIDÊNCIA DOS CANDIDATOS X LOCAIS DE PROVAS

RESIDÊNCIAS/ CANDIDATOS	LOCAIS DE PROVAS				
	1	2	3	...	k
1	distância(1,1)	distância(1,2)	distância(1,3)	...	distância(1,k)
2	distância(2,1)	distância(2,2)	distância(2,3)	...	distância(2,k)
3	distância(3,1)	distância(3,2)	distância(3,3)	...	distância(3,k)
...
n	distância(n,1)	distância(n,2)	distância(n,3)	...	distância(n,k)

Cada local de provas pode atender apenas um número limitado de candidatos e a respectiva limitação de capacidade varia de acordo com cada local. Com base nas distâncias fornecidas pela matriz de distâncias calculada, cada candidato foi designado para o local de provas mais próximo de sua residência que dispunha de vaga. A ordem

em que os candidatos foram designados para os locais de provas foi definida pelo algoritmo DeCan, que é apresentado na seção 5.

Do que foi dito anteriormente, convém observar que: o que caracteriza o problema real descrito como um modelo do problema das p-medianas capacitado são os seguintes fatores:

- a) pode-se considerar o conjunto dos 43 locais disponíveis para a realização das provas do vestibular em Curitiba como sendo o conjunto V (com, $|V| = 43$) de todas as instalações candidatas a mediana (locais de provas);
- b) seja $V_p \subset V$ (com, $|V_p| = 26$) o conjunto formado pelos 26 locais de provas selecionados;
- c) cada um dos 43 possíveis locais de provas pode atender apenas a um número limitado de candidatos (restrições de capacidade);
- d) se o objetivo for selecionar o conjunto $V_p \subset V$ que minimiza a soma das distâncias percorridas por todos os candidatos até os locais de provas mais próximos possíveis de suas residências, então, o problema pode ser resolvido como um problema das p-medianas capacitado.

São apresentados a seguir os métodos utilizados no presente trabalho para a implementação dos algoritmos propostos na seção 5 para a resolução do problema das p-medianas capacitado e do problema real descrito anteriormente.

3 MÉTODOS UTILIZADOS

Esta seção é a base dos estudos desenvolvidos no presente trabalho; é o terreno onde se desenvolve praticamente toda a pesquisa. Primeiramente, o algoritmo que foi utilizado como modelo para a implementação do algoritmo DeCan proposto no presente trabalho é apresentado. Em seguida, são introduzidas as duas heurísticas computacionais de busca implementadas na seção 5 para resolver o problema das p-medianas capacitado.

3.1 ALGORITMO DE GILLET E JOHNSON

Definidos todos os locais de provas que serão utilizados pode-se aplicar para a designação dos candidatos aos respectivos locais de provas o algoritmo proposto por GILLET e JOHNSON (1973) (BODIN et al., 1983) adaptado para esta situação em que as capacidades dos referidos locais devem ser consideradas.

Inicialmente, todos os pontos (residências dos candidatos) encontram-se sem designação. Para cada ponto i seja $L_1(i)$ o local de provas mais próximo a i , e $L_2(i)$ o segundo local mais próximo a i .

Para cada ponto i , a razão: $r(i) = \frac{L_1(i)}{L_2(i)}$ é calculada e todos os pontos são

colocados numa “lista de designação” em ordem crescente pelos valores de $r(i)$. A designação começa pelos primeiros elementos da lista (pontos com menor razão $r(i)$), e é feita obedecendo-se a capacidade dos locais de provas. Durante a designação sempre que um candidato é designado para um local de provas com as vagas esgotadas (evidentemente sem sucesso), a razão $r(i)$ é recalculada para todos os pontos que ainda não foram designados considerando-se apenas os locais de provas cujas vagas não estejam esgotadas. Estes pontos são novamente colocados na “lista de designação” em

ordem crescente pelos valores de $r(i)$ e a designação continua até que todos os pontos sejam designados para algum local de provas.

Através do cálculo da razão $r(i)$ procura-se conhecer a “urgência” de se fazer a designação de um candidato em relação aos demais, pois, quanto menor for o valor de $r(i)$, maior é a urgência já que o segundo local mais próximo fica em posição bem pior para o candidato em relação ao primeiro.

O algoritmo que acabou de ser descrito será implementado na seção 5 com algumas modificações. As modificações realizadas são examinadas com detalhes e visam torná-lo mais eficiente para efetuar a designação de candidatos ao vestibular aos locais de provas.

3.2 ALGORITMOS GENÉTICOS

Nesta seção, é introduzido o conceito de algoritmos genéticos: o que são, qual sua origem, e em que são diferentes de outras heurísticas de busca, são tópicos abordados.

3.2.1 O que são Algoritmos Genéticos?

Algoritmos Genéticos (AGs) são algoritmos computacionais de busca baseados nos mecanismos de evolução natural e na genética. Em AGs, uma população de possíveis soluções para o problema em questão evolui de acordo com operadores probabilísticos concebidos a partir de metáforas biológicas, de modo que há uma tendência de que, na média, os indivíduos representem soluções cada vez melhores à medida que o processo evolutivo continua (GOLDBERG, 1986a).

3.2.2 Qual a Origem dos AGs?

Embora Charles Darwin tenha formulado a Teoria da Evolução no final do século passado, foi só recentemente que se tentou idealizar um modelo matemático do processo evolutivo. Nos anos 60, John Holland, da Universidade de Michigan, começou a definir as bases de algoritmos de otimização de inspiração genética. Seu trabalho culminou na publicação do livro *Adaptation in Natural and Artificial Systems* (HOLLAND, 1975). Livro este, que é hoje muito citado mas pouquíssimo lido, foi pouco divulgado na época, em grande parte devido ao estilo pesado, com notação pouco criteriosa e excessivamente complexa. Felizmente, Holland e seus muitos discípulos, quase todos seus alunos de pós-graduação, continuaram sua linha de investigação, publicando resultados com alguma timidez mas com perseverança (TANOMARU, 1995).

A grande popularidade que os AGs atingiram deve-se a dois importantes fatores: a publicação de um livro tutorial sobre AGs por um dos alunos de doutorado de Holland, David Goldberg, um pesquisador extremamente ativo e com excelente potencial didático, e às Conferências Internacionais sobre algoritmos genéticos. Essas conferências eram realizadas a cada dois anos nos Estados Unidos desde 1985 até 1997, e atualmente são realizadas anualmente como parte de uma conferência mais ampla na área de algoritmos evolucionários. Os AGs pertencem à classe dos algoritmos probabilísticos de busca e otimização, embora não sejam aleatórios. Utiliza-se o conceito de probabilidade, mas AGs não são buscas aleatórias. Pelo contrário, os AGs dirigem a busca para regiões do espaço de busca onde é “provável” que os pontos ótimos estejam.

3.2.3 Conceitos Fundamentais e Terminologia

A forte inspiração dos Algoritmos Genéticos na teoria da evolução das espécies e na genética natural torna importante uma revisão da terminologia comum a esta teoria e aos AGs. A nível biológico, um indivíduo é formado por um conjunto de cromossomos. No entanto, pode-se fazer uma analogia entre indivíduo e cromossomo, tendo em vista que um indivíduo pode ser formado por apenas um cromossomo, o que é comum em AGs. Assim, os dois termos são utilizados indistintamente. O presente trabalho utilizará o termo indivíduo como sinónimo do termo biológico cromossomo. Porém, na literatura é comum aparecer o termo em inglês *string* como sinónimo de cromossomo e indivíduo.

O primeiro passo para aplicação de AGs a um problema é representar cada possível solução no espaço de busca como uma sequência de símbolos (cromossomo) gerados a partir de um alfabeto finito A . No caso mais simples, usa-se o alfabeto binário ou seja $A = \{0, 1\}$, mas, no caso geral, tanto o método de representação quanto o alfabeto genético dependem das características do problema.

Um cromossomo é composto por genes, sendo que cada gene possui um local fixo no cromossomo. Este local é denominado locus. Cada gene pode assumir um valor pertencente a um certo conjunto de valores, os quais são denominados alelos. Em termos de AGs, o cromossomo corresponde ao indivíduo, e este é representado por uma sequência de genes (um *string*) de comprimento finito. O termo alelo refere-se a um valor dentro o conjunto de valores possíveis de serem atribuídos a um determinado gene, ou seja, é um valor (0 ou 1 no caso mais simples). Os conceitos mencionados são ilustrados pela figura 3.1.

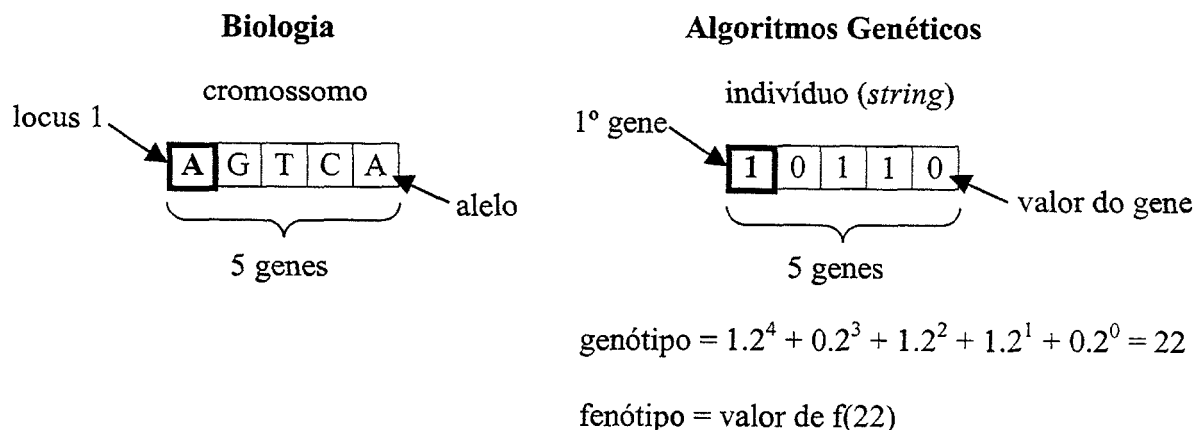


FIGURA 3.1 - REPRESENTAÇÃO DE UM CROMOSSOMO NA BIOLOGIA E EM AGs

Ao conjunto de cromossomo, genes e alelos denomina-se genótipo e às características conferidas por estes, denomina-se fenótipo. No contexto dos AGs, o genótipo é a variável independente x da função objetivo $f(x)$, e o fenótipo é a variável dependente ou valor da função $f(x)$. Um resumo comparativo entre a terminologia utilizada em AGs e na Biologia é apresentada na tabela 3.1.

TABELA 3.1 - RELAÇÃO DA TERMINOLOGIA DOS AGs COM A BIOLOGIA

BIOLOGIA	ALGORITMOS GENÉTICOS
cromossomo	indivíduo ou <i>string</i>
gene	gene, ou bit (no caso binário)
alelo	valor do gene, ou do bit (no caso binário)
locus	posição de um gene específico no indivíduo ou <i>string</i>
genótipo	indivíduo candidato a solução x
fenótipo	valor da função $f(x)$ para um dado indivíduo

Tendo definido a representação cromossômica para o problema, gera-se um conjunto de possíveis soluções, chamadas de soluções-candidatas, um conjunto de soluções codificadas de acordo com a representação selecionada correspondendo a uma população de indivíduos. AGs são algoritmos iterativos e, em geral, após um certo número de iterações (ou gerações) toda a população terá sido modificada, embora nem todos os indivíduos de uma população sejam necessariamente “filhos” de indivíduos da população anterior.

3.2.4 Em que os AGs Diferem de outras Heurísticas de Busca?

Embora possam parecer simplistas do ponto de vista biológico, os AGs são suficientemente complexos para fornecer mecanismos de busca adaptativos poderosos e robustos.

Segundo GOLDBERG (1986a) em seu livro especializado sobre o assunto, de modo geral AGs têm as seguintes características:

- a) operam numa população (conjunto) de pontos no espaço de busca, e não a partir de um único ponto;
- b) operam num espaço de soluções codificadas, e não no espaço de busca diretamente;
- c) necessitam somente de informações sobre o valor de uma função objetivo para cada membro da população, e não requerem que a função seja diferenciável, ou contínua;
- d) usam transições (mudança de um estado para outro) probabilísticas, e não regras determinísticas.

3.2.5 Inicialização de um Algoritmo Genético

Na maior parte das aplicações, uma população de P indivíduos é gerada aleatoriamente ou através de algum processo heurístico.

Como no caso da Biologia, não há evolução sem variedade. Ou seja, a teoria da seleção natural “lei do mais forte”, necessita que os indivíduos tenham diferentes graus de adaptação ao ambiente em que vivem. De acordo com isso, é importante que a população inicial cubra a maior área possível do espaço de busca.

3.2.6 Avaliação da Adaptabilidade (*fitness*)

Os AGs necessitam da informação do valor de uma função objetivo para cada membro da população. Nos casos mais simples, usa-se justamente o valor da função que se quer maximizar (ou minimizar). A função objetivo fornece, para cada indivíduo, uma medida de quão bem adaptado ao ambiente ele está, ou seja, quanto melhor o valor da função objetivo, maiores são as chances do indivíduo sobreviver no ambiente, reproduzir-se e gerar descendentes passando parte de seu material genético às gerações posteriores.

A avaliação de cada indivíduo resulta num valor que, em inglês, é denominado *fitness* (aptidão ou adaptabilidade).

3.2.7 Seleção

A seleção basicamente tem por objetivo fazer com que os indivíduos mais adaptados da geração anterior tenham maior probabilidade de participarem do processo que irá formar a nova população.

Em geral, gera-se uma população temporária de P indivíduos extraídos com probabilidade proporcional à adaptação relativa (*fitness*) de cada indivíduo na população, ou seja, a probabilidade de seleção de um indivíduo, em geral, é dada por:

$$P_{\text{SEL}_X_k} = \frac{f(x_k)}{\sum_{i=1}^P f(x_i)}, \quad (3.2.1)$$

onde $f(x)$ é o valor da função de adaptabilidade (em geral a função objetivo) e x_i, x_k são as soluções candidatas (indivíduos) e: $i \in \{1, 2, \dots, P\}$.

Com base na fórmula (3.2.1) de probabilidade, selecionam-se P indivíduos. Neste processo, indivíduos com baixa adaptação terão alta probabilidade de

desaparecerem da população, ou seja, serem extintos, ao passo que indivíduos mais adaptados terão grandes chances de sobreviverem. Os indivíduos selecionados são então submetidos a operadores genéticos, conforme discutido a seguir.

3.2.8 Operadores Genéticos (OGs)

Basicamente há três tipos principais de operadores genéticos comuns aos AGs. A reprodução, o cruzamento e a mutação que são discutidos a seguir.

3.2.8.1 Reprodução

O processo de reprodução consiste simplesmente em copiar integralmente um indivíduo selecionado para a próxima geração. Em geral um indivíduo selecionado pode ser reproduzido ou sofrer cruzamento, conforme discutido a seguir.

3.2.8.2 Cruzamento

O OG denominado cruzamento, em AGs corresponde a uma generalização do que ocorre na reprodução sexuada. O cruzamento é típico de seres mais evoluídos, e se dá pela aproximação dos cromossomos de dois indivíduos (pais) selecionados, que trocam entre si partes de seus cromossomos. Este processo deve gerar dois novos indivíduos diferentes, mas que ainda guardam influências dos pais. Não é sempre que o cruzamento é efetuado, por isso é definida uma probabilidade de cruzamento, representando a proporção aproximada da população que sofrerá cruzamento. Os pares (pais) que não sofrerem cruzamento são copiados integralmente para a nova população como filhos, aplicando-se o operador de reprodução. O cruzamento é realizado com probabilidade P_{CRUZ} . Assim, dentre n indivíduos selecionados, aproximadamente $P_{\text{CRUZ}} \times n$ sofrerão cruzamento, enquanto $(1 - P_{\text{CRUZ}}) \times n$ serão reproduzidos.

A troca de partes do cromossomo pode ser realizada de várias formas. Basicamente tem-se o cruzamento uniforme, cruzamento com 1- partição, cruzamento com 2 - partições e cruzamento com n - partições.

O cruzamento uniforme consiste no emparelhamento dos dois cromossomos pais, e então os genes em cada locus do cromossomo são trocados de acordo com uma probabilidade predefinida (por exemplo, 50%). Esse processo é ilustrado na figura 3.2.

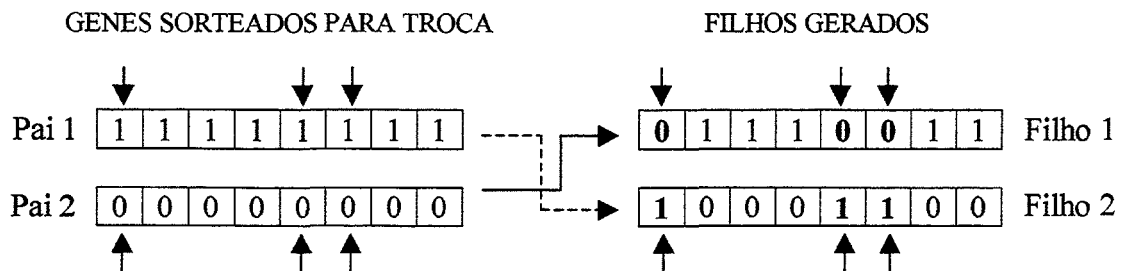


FIGURA 3.2 - EXEMPLO DE CRUZAMENTO UNIFORME

O cruzamento com 1 - partição consiste na escolha aleatória de somente um ponto de corte. Todo o material genético dos pais existente à direita (ou à esquerda) deste ponto é trocado, conforme ilustrado na figura 3.3.

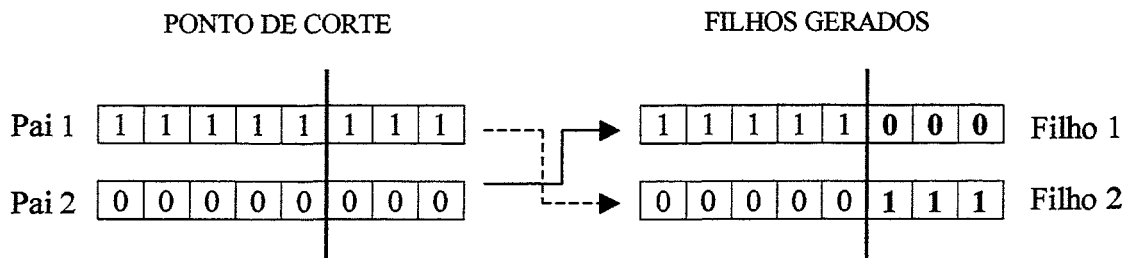


FIGURA 3.3 - EXEMPLO DE CRUZAMENTO COM 1 - PARTIÇÃO

No caso do cruzamento com 2 - partições, há a escolha aleatória de dois pontos de corte. Todo o material genético dos pais existente entre os dois pontos de corte são trocados, e o restante é mantido inalterado, conforme ilustrado na figura 3.4.

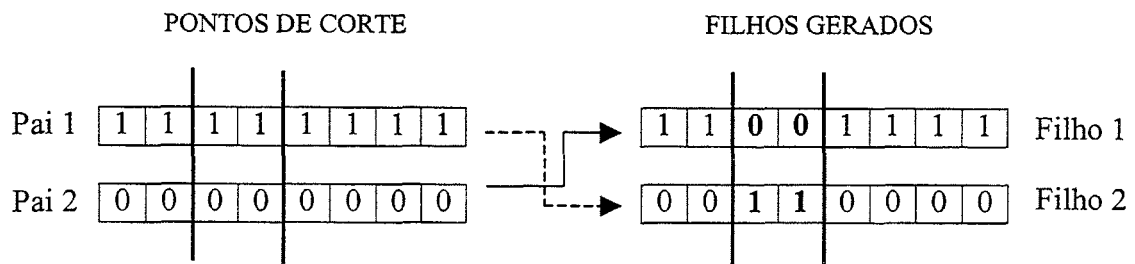


FIGURA 3.4 - EXEMPLO DE CRUZAMENTO COM 2 - PARTIÇÕES

Já o cruzamento com n - partições consiste de n cruzamentos com 2 - partições. Pode-se considerar os cruzamentos com 1 e 2 - partições como casos particulares do cruzamento de n - partições.

3.2.8.3 Mutação

O OG de mutação é necessário para a introdução e manutenção da diversidade genética da população, alterando arbitrariamente um ou mais componentes de um cromossomo. Basicamente, seleciona-se uma posição num cromossomo e muda-se o valor do gene correspondente aleatoriamente para um outro alelo (valor) possível. O processo é geralmente controlado por um parâmetro fixo P_{MUT} que indica a probabilidade de um indivíduo sofrer mutação.

Desta forma, a mutação assegura que a probabilidade de se chegar a qualquer ponto do espaço de busca nunca será zero, além de ajudar a controlar o problema de convergência para máximos (ou mínimos) locais.

Dentre os principais mecanismos de alteração genética que recebem o nome global de mutação, um dos mais utilizados é a troca simples, ou inversão de bit (no caso do alfabeto binário), conforme ilustrado na figura 3.5.

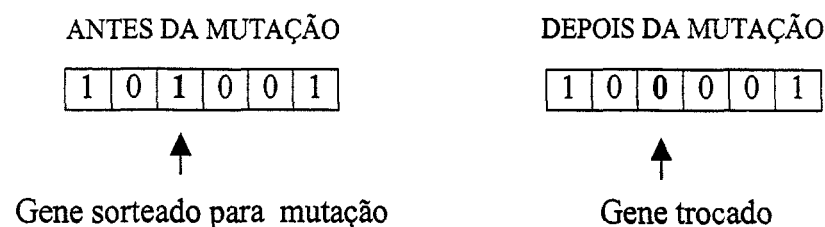


FIGURA 3.5 - EXEMPLO DE MUTAÇÃO POR TROCA SIMPLES

3.2.9 Condição de Término

Para problemas de otimização o ideal seria que o algoritmo terminasse assim que o ponto ótimo fosse descoberto. Já no caso de funções multimodais, um ponto ótimo pode ser o suficiente, mas pode haver situações onde todos ou o maior número possível de pontos ótimos sejam desejados. Um problema prático é que, na maioria dos casos de interesse, não se pode afirmar com certeza se um dado ponto ótimo corresponde a um ótimo global. Como consequência, normalmente utiliza-se o critério do número máximo de gerações ou tempo limite de processamento para um AG. Outro critério plausível é parar o algoritmo usando a idéia de estagnação, ou seja, quando não se observa melhoria da população depois de várias gerações consecutivas, o algoritmo encerra o processamento. Para se avaliar a estagnação um método que pode ser utilizado é comparar o desvio padrão dos valores de adaptação dos indivíduos da população atual com a anterior.

3.2.10 Convergência Prematura

Utilizando-se o modelo de AG simples para a otimização de funções multimodais, um fenômeno que se observa com frequência é que o AG pode convergir muito rapidamente (em algumas dezenas de gerações) para um ponto de alta qualidade, mas não o ótimo global, num fenômeno denominado convergência prematura (embora o OG mutação tente evitar isso).

3.2.11 Parâmetros dos Operadores Genéticos

Num AG básico, o usuário deve definir o tamanho da população, P , além das probabilidades de cruzamento e mutação, respectivamente, P_{CRUZ} e P_{MUT} . Em AGs mais sofisticados, há ainda mais parâmetros, comprometendo parte da robustez dos algoritmos. Infelizmente, não há regras claras para a escolha desses parâmetros (variam de acordo com as características do problema).

3.2.11.1 Tamanho P da população

Quanto ao parâmetro P , a intuição indica que “quanto maior for a população, melhor”, uma vez que, em última análise, com uma população inicial cobrindo todo o espaço de busca, a solução ótima seria obtida na primeira geração. Na prática, é óbvio, deve-se limitar o tamanho da população para reduzir o tempo de processamento. Normalmente, valores da ordem de 50 a 200 cromossomos resolvem a maior parte dos problemas, mas, populações maiores podem ser necessárias para problemas mais complexos (TANOMARU, 1995).

3.2.11.2 Probabilidades de cruzamento e de mutação

Em relação às probabilidades de cruzamento e mutação, estudos empíricos têm mostrado que bons resultados geralmente são obtidos com valor alto para a probabilidade de cruzamento $P_{\text{CRUZ}} \geq 0,7$ (70%) e baixo valor para a probabilidade de mutação $P_{\text{MUT}} \leq 0,01$ (1%). Definida a probabilidade de cruzamento, a probabilidade de reprodução estará automaticamente definida como sendo: $P_{\text{REP}} = (1 - P_{\text{CRUZ}})$ já que, em geral, ou o indivíduo será reproduzido ou sofrerá cruzamento ($P_{\text{CRUZ}} + P_{\text{REP}} = 1$).

3.2.12 Critérios de Seleção

O método de seleção mais conhecido em AGs é o chamado método da roleta, que simula uma roleta fictícia, com base na adaptabilidade (*fitness*) do indivíduo relativamente à população.

Existem basicamente dois métodos empregadas no processo de seleção, a roleta simples e o *ranking*, conforme discutido a seguir.

3.2.12.1 Roleta simples

Cada indivíduo da população anterior terá uma probabilidade de ser sorteado proporcional a sua adaptação, conforme mostrado na tabela 3.2.

TABELA 3.2 - DADOS PARA ROLETA SIMPLES

INDIVÍDUOS	ADAPTAÇÃO	% DO TOTAL
x4	201	91
x2	10	5
x3	7	3
x1	2	1
TOTAL	220	100%

A roleta simples causa uma alta pressão seletiva sobre os indivíduos da população, pois o indivíduo mais adaptado (no exemplo da tabela 3.2) terá uma relação de 91:1 de ser sorteado (selecionado) em relação ao menos adaptado.

A pressão seletiva está implicitamente relacionada com a diversidade da população. Alta pressão seletiva tende a fazer a diversidade cair rapidamente, levando a população a convergir em poucas gerações, o que pode resultar em convergência prematura para um máximo (ou mínimo) local.

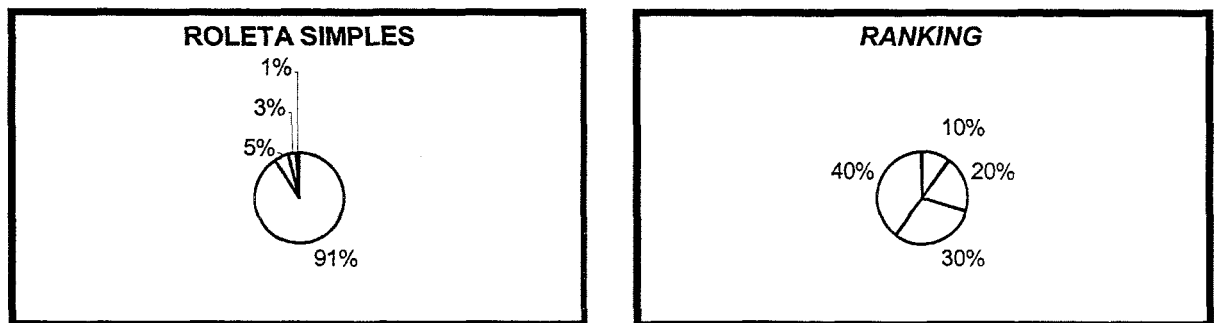
3.2.12.2 *Ranking*

A técnica de *ranking* também apresenta os indivíduos ordenados conforme a adaptação de cada indivíduo. Contudo, em *ranking* cada indivíduo recebe uma nota sendo que o menos adaptado recebe a nota 1 e o mais adaptado fica com nota igual ao tamanho da população (distância de 1 entre cada vizinho). Esse método é uma forma de *ranking* (ordenação), pois a probabilidade de um indivíduo ser selecionado depende apenas de seu *ranking*, e não do valor absoluto de sua adaptabilidade. A tabela 3.3 apresenta os dados de quatro indivíduos ordenados para seleção segundo os critérios de *ranking*.

TABELA 3.3 - DADOS PARA *RANKING*

INDIVÍDUOS	ADAPTAÇÃO	DISTÂNCIA/ %
x4	201	4 (40%)
x2	10	3 (30%)
x3	7	2 (20%)
x1	2	1 (10%)
TOTAL	220	100%

Os gráficos de setor 3.1 comparam a roleta simples x *ranking* com base nos dados das tabelas 3.2 e 3.3.

GRÁFICOS 3.1 - COMPARAÇÃO ROLETA SIMPLES X *RANKING*

3.2.13 Método do Estado Estável

Ao invés de substituir toda a população de uma vez, este modelo considera que somente alguns indivíduos da população devem ser substituídos a cada geração. No caso mais simples, insere-se apenas um indivíduo por vez no lugar do pior indivíduo da população atual.

3.3 BUSCA TABU

Nesta seção, é introduzido o conceito de busca tabu: o que é busca tabu, qual sua origem, e como funciona, são tópicos abordados.

3.3.1 O que é Busca Tabu?

Segundo GLOVER e LAGUNA (1997) em seu livro especializado no assunto: busca tabu é uma heurística computacional de busca conhecida por geralmente superar o problema da convergência local em problemas de otimização.

3.3.2 Origem da Busca Tabu

A heurística busca tabu (BT) é relativamente recente, teve origem em meados da década de 70. A forma moderna de BT é devida aos trabalhos desenvolvidos por GLOVER, embora outros como HANSEN (1986) e LAGUNA (1991) tenham desenvolvido importantes estudos que contribuíram para melhorar o desempenho dessa heurística. BT ainda é pesquisada ativamente e continua evoluindo e melhorando.

3.3.3 Como funciona a Heurística Busca Tabu?

A palavra “tabu” sugere algo proibido, ou pelo menos inibido. A heurística BT básica emprega restrições tabu para inibir certos movimentos e alguns procedimentos denominados critérios de aspiração são utilizados para decidir quando movimentos classificados como tabu podem ser executados. Desta forma, BT conduz a busca para áreas ainda não analisadas do espaço de busca, tendendo a evitar a convergência da solução para um máximo (ou mínimo) local. As restrições tabu são geralmente controladas por uma lista que memoriza os últimos movimentos executados. O tempo que um movimento deve permanecer nesta lista, em geral, está

relacionado com o número de iterações do algoritmo e com o número de movimentos possíveis a partir da solução candidata atual (solução que está sendo analisada). A implementação de um algoritmo BT básico envolve decidir como os movimentos (geração de novas soluções) são realizados, a definição dos critérios de aspiração e como é feito o gerenciamento da memória dos movimentos, além de outros detalhes e técnicas que são brevemente discutidas a seguir.

3.3.4 Busca através das Soluções Vizinhas

BT pode ser convenientemente caracterizada como sendo uma busca através das soluções vizinhas (GLOVER, 1991). Cada solução $x \in X$ tem um conjunto associado de soluções vizinhas $V(x) \subset X$ chamadas soluções vizinhas a x . Toda solução $x' \in V(x)$ pode ser gerada a partir de x por um certo tipo de operação denominada movimento. Normalmente em BT, soluções vizinhas são simétricas, ou seja, x' é solução vizinha a x se, e somente se, x é solução vizinha a x' .

3.3.5 Critérios de Aspiração

Os critérios de aspiração são introduzidos em BT para determinar quando uma restrição tabu pode ser quebrada. Ou seja, a restrição é ignorada e o movimento, mesmo classificado como proibido, é executado. Um critério de aspiração bastante utilizado é o de ignorar a restrição tabu sempre que a solução formada por um determinado movimento proibido for melhor que a melhor solução encontrada até o momento. A aplicação adequada desses procedimentos é fundamental para se atingir altos níveis de performance em BT.

3.3.6 Estratégia da Oscilação

Um procedimento bastante utilizado em BT é a estratégia da oscilação. Consiste em admitir a oscilação entre soluções factíveis e infactíveis durante o processo de busca. Considerar soluções infactíveis é um outro meio de tentar escapar de ótimos locais. Além disso, o fato de permitir temporariamente soluções infactíveis tende a dirigir a busca mais rapidamente para soluções factíveis cada vez melhores.

3.3.7 Algumas Iterações de um Algoritmo BT Elementar

Os problemas de permutação formam uma importante classe dos problemas de otimização. Para exemplificar as idéias básicas sobre BT são apresentadas algumas iterações de um algoritmo BT elementar aplicado a um problema de permutação fictício.

3.3.7.1 Apresentando o problema fictício

Considere o problema de se determinar a ordem em que devem ser colocados num módulo de isolamento 7 filtros isolantes, todos com propriedades distintas, de tal forma que o isolamento obtido seja o máximo possível. O isolamento é medido através de uma função de isolamento F , que mede o sinal de saída através do módulo de isolamento. A ordem em que são colocados os filtros isolantes influi no isolamento obtido.

Suponha que uma solução inicial gerada aleatoriamente para o problema proposto esteja representada na figura 3.6, onde 1, 2, ..., 7 representam respectivamente cada um dos filtros isolantes do problema em questão.

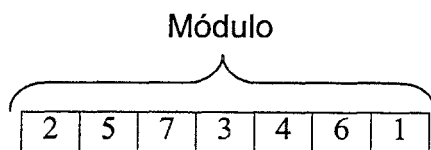


FIGURA 3.6 - UMA SOLUÇÃO INICIAL

Novas soluções são geradas a partir da solução atual por um “movimento” que consiste na troca de posição entre dois filtros no módulo. Os filtros que são trocados num movimento são selecionados ao acaso. Os movimentos são representados através de pares de filtros, como por exemplo, [5, 6], que indica a mudança das posições dos filtros 5 e 6 entre si (ver figura 3.7). Note que por essa definição o movimento é simétrico, no sentido que os pares [5, 6] e [6, 5] representam a mesma mudança.

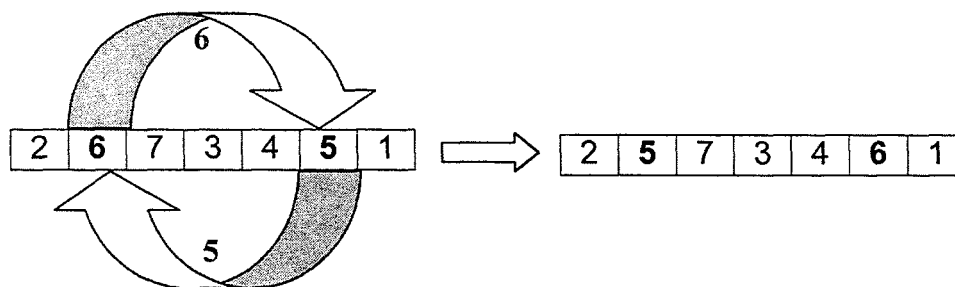


FIGURA 3.7 - EXECUÇÃO DO MOVIMENTO [5, 6] GERANDO UMA NOVA SOLUÇÃO

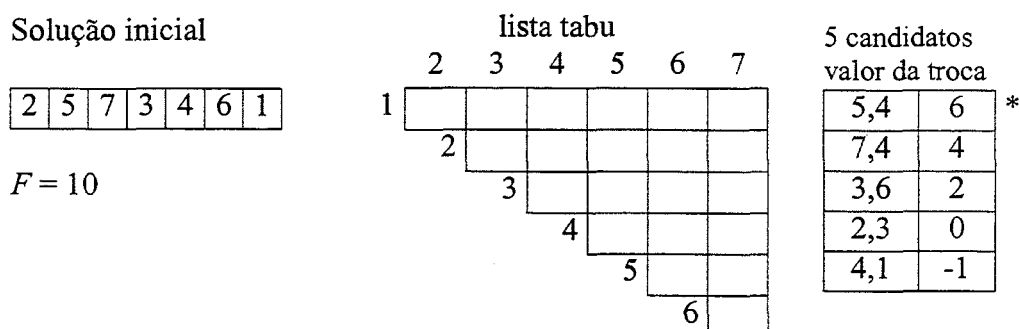
A cada iteração, cinco novas soluções são geradas aleatoriamente dentre todos os movimentos possíveis e ordenadas pelos valores da função de isolamento F da melhor para a pior. Uma lista de memória (lista tabu) armazenará cada movimento executado que deverá permanecer nesta lista durante as três próximas iterações. Enquanto um determinado movimento permanecer na lista tabu será considerado

proibido e só poderá ser executado se este resultar num valor de isolamento melhor que o melhor valor obtido em todas as iterações anteriores (critério de aspiração). Como se deseja maximizar a função F , procura-se sempre o maior valor de troca possível.

São utilizados nos esquemas abaixo um asterisco (*) para representar um movimento que será executado e a letra (T) maiúscula para representar um movimento classificado como tabu.

3.3.7.2 As iterações do problema fictício

Iteração 0



ESQUEMA 3.1 - ITERAÇÃO 0

Na iteração (0), representada pelo esquema 3.1, uma solução inicial para o problema foi gerada aleatoriamente. Em seguida, partindo-se sempre da solução inicial foram executados ao acaso cinco movimentos (troca de posição entre dois filtros). Para cada nova solução gerada pelos movimentos foi calculado o valor da função de isolamento F . Os movimentos foram ordenados do melhor (primeiro da lista de candidatos) para o pior (último da lista de candidatos) de acordo com os valores obtidos no cálculo da função de isolamento. Nesta iteração (0), o movimento [5, 4] do topo da lista de candidatos foi escolhido para gerar a próxima solução porque dentre os 5 movimentos gerados foi o que produziu o melhor valor para a função de isolamento.

Iteração 1

Solução inicial

2	4	7	3	5	6	1
---	---	---	---	---	---	---

$F = 16$

lista tabu						
	2	3	4	5	6	7
1						
2						
3						
4				3		
5						
6						

5 candidatos
valor da troca

3,1	2	*
2,3	1	
3,6	-1	
7,1	-2	
6,1	-4	

ESQUEMA 3.2 - ITERAÇÃO 1

O esquema 3.2 referente a iteração (1) mostra que, o movimento [5, 4] foi executado e incluído na lista tabu. O valor da solução atual é o valor da anterior acrescido do valor da troca executada. Além disso, a célula (4, 5) da lista tabu passa a conter o valor 3, que representa o número de iterações que o movimento [5, 4] permanecerá na lista tabu. Foram executados novamente cinco movimentos ao acaso a partir da solução atual. Os movimentos foram ordenados pelo valor da função de isolamento e novamente o movimento do topo da lista de candidatos [3, 1] foi escolhido para gerar a próxima solução.

Iteração 2

Solução inicial

2	4	7	1	5	6	3
---	---	---	---	---	---	---

$F = 18$

lista tabu						
	2	3	4	5	6	7
1		3				
2						
3						
4				2		
5						
6						

5 candidatos
valor da troca

1,3	-2	T
2,4	-4	
7,6	-6	
4,5	-7	T
5,3	-9	

ESQUEMA 3.3 - ITERAÇÃO 2

Pelo esquema 3.3 pode-se notar que, o valor da célula (4, 5) da lista tabu foi reduzido em uma unidade, já que uma iteração foi completada. Dentre os cinco movimentos gerados na iteração atual (2), o movimento do topo da lista de soluções candidatas é o movimento [1, 3] mas, este está classificado como tabu (foi executado na iteração anterior) e não será executado. Então, o segundo “melhor” movimento da lista de candidatos [2, 4] será o movimento executado para gerar a próxima solução.

Iteração 3

Solução inicial							
4	2	7	1	5	6	3	
$F = 14$							

lista tabu							
	2	3	4	5	6	7	
1		2					
	2		3				
		3					
			4	1			
				5			
					6		

5 candidatos			
valor da troca			
4,5	6	T*	
5,3	2		
7,1	0		
1,3	-3	T	
2,6	-6		

ESQUEMA 3.4 - ITERAÇÃO 3

Conforme mostra o esquema 3.4, a solução atual tem valor inferior ao das duas iterações anteriores. Isto devido a um movimento com valor de troca negativo ter sido executado. Permitir que movimentos que causem prejuízo no valor da função objetivo sejam executados ajuda a evitar a convergência da solução para um ótimo local. Além disso, permite a avaliação de áreas do espaço de busca que de outra forma talvez não fossem avaliadas. A lista tabu tem agora 3 movimentos registrados. Nesta iteração (3), o movimento do topo da lista de candidatos é [4, 5] que está na lista tabu (proibido). Acontece que este movimento produz na função de isolamento um valor melhor que qualquer valor obtido até o momento. Portanto, a proibição será ignorada e o movimento será executado (critério de aspiração).

Iteração 4

Solução inicial

5	2	7	1	4	6	3
---	---	---	---	---	---	---

$$F = 20$$

		lista tabu					
		2	3	4	5	6	7
1	1		1				
	2			2			
	3						
	4				3		
	5						
	6						

5 candidatos

valor da troca

7,1	6	T
4,3	-3	
6,3	-5	
5,4	-6	
2,6	-8	

ESQUEMA 3.5. ITERAÇÃO 4

Note pelo esquema 3.5 que a célula (4, 5) da lista tabu voltou a receber o valor 3, já que o movimento [4, 5] foi executado novamente. A solução da iteração atual (4) tem o melhor valor obtido até o momento, e a lista tabu tem agora 3 dos 21 movimentos possíveis classificados como tabu. As iterações continuam até que um critério de parada seja satisfeito (como por exemplo, um número fixo de iterações ser completado).

A seguir, é feita uma revisão da literatura sobre localização de instalações, algoritmos genéticos e busca tabu aplicados ao problema das p-medianas.

4 REVISÃO DA LITERATURA

Esta seção faz referência a alguns trabalhos anteriormente publicados ou desenvolvidos sobre localização de instalações e principalmente sobre algoritmos genéticos e busca tabu aplicados ao problema das p-medianas.

4.1 LOCALIZAÇÃO DE INSTALAÇÕES

Os problemas de localização de instalações têm recebido considerável atenção de pesquisadores de diversas áreas. RÖNNQVIST et al. (1999) desenvolveram uma heurística para resolução de problemas de localização de instalações em que cada ponto de demanda é atendido por uma única instalação. A heurística é baseada na repetição de um algoritmo *matching* e basicamente resolve uma série de problemas *matching* até que um critério de convergência predefinido seja satisfeito. SYAM (1997) propôs uma resolução para o problema de localização de instalações baseada em Relaxação Lagrangeana. TRAGANTALERNGSAK et al. (1999) desenvolveram um algoritmo *Branch and Bound* baseado em Relaxação Lagrangeana para resolver problemas de localização de instalações onde as instalações possuem diferentes níveis hierárquicos. DREZNER (1995) apresenta uma coleção de problemas de localização de instalações enfocando suas diferentes abordagens e métodos de solução.

4.2 ALGORITMOS GENÉTICOS

HOSAGE e GOODCHILD (1986) (H&G) parecem ter sido os primeiros a desenvolver um algoritmo genético para o problema das p-medianas. Os operadores genéticos utilizados neste algoritmo foram os usuais dos AGs (reprodução, cruzamento e mutação). No AG apresentado por H&G, cada possível solução para o problema (indivíduo) é representada por uma seqüência de números binários (0 ou 1). Cada dígito indica se a instalação correspondente à posição do dígito em questão é uma mediana “1” ou não é uma mediana “0”. Se o número de 1’s nesta seqüência não for

igual a p , a solução é considerada infactível e uma penalidade (proporcional à violação das restrições) é imposta à solução. O algoritmo desenvolvido por H&G foi testado em um problema que seleciona 3 medianas entre 20 possíveis (isto é, $n = 20$ e $p = 3$) e com população inicial $P = 25$ para quatro números diferentes de gerações (120, 150, 180 e 210). Foram utilizados 100 grafos distintos gerados aleatoriamente e a solução ótima foi encontrada 69, 85, 84 e 89 vezes para cada nível de geração. A princípio pode parecer, pelos testes realizados, que o algoritmo de H&G garante relativamente um bom desempenho pois, encontrou a solução ótima entre 70% e 90% dos casos. Porém, o algoritmo trabalha com uma representação binária de tamanho n (número de instalações candidatas a mediana), e esta não é uma boa representação. Não apenas desperdiça memória, como também requer operações desnecessárias para a realização do cruzamento entre os pais e para o cálculo do valor da função objetivo. O problema utilizado para teste possui apenas 1140 soluções possíveis (C_{20}^3). Ainda, o algoritmo gera e avalia 2905 soluções, não necessariamente distintas, para 120 gerações e 5065 para 210 gerações. Embora o problema tenha apenas 1140 soluções, o algoritmo busca pela solução num espaço com 2^{20} possibilidades, definidas por todos os números de 20 dígitos binários. Certamente, a grande maioria (aproximadamente 99,9%) das soluções geradas são infactíveis para o problema das p -medianas e o algoritmo desperdiça tempo analisando-as.

DIBBLE e DENSHAM (D&D) (1993) relataram a aplicação de um AG que desenvolveram para o problema das p -medianas. Neste AG, foi proposta uma representação cromossômica mais adequada para o problema das p -medianas. Cada indivíduo possui exatamente p genes, e cada gene pode representar o valor correspondente ao número da instalação que se deseja inserir na solução. Utilizaram apenas os operadores genéticos usuais dos AGs. O AG desenvolvido por D&D foi aplicado a um problema em que teve que selecionar 9 medianas entre 150 possíveis foi utilizada uma população inicial de $P = 1000$ indivíduos e 150 gerações. Os resultados obtidos nessa aplicação foram comparados com os obtidos para o mesmo problema resolvido pelo algoritmo de Teitz e Bart. No entanto, o tempo de processamento do

AG de D&D foi bem maior que o do algoritmo de Teitz e Bart e as soluções encontradas foram praticamente as mesmas para os dois algoritmos.

MORENO-PEREZ et al. (M-P&al) (1994) também desenvolveram um AG para o problema das p-medianas. Neste AG, a representação cromossômica é idêntica a do algoritmo de D&D descrito anteriormente. Foram utilizados apenas os operadores genéticos usuais dos AGs. O que diferencia o algoritmo proposto por M-P&al dos outros anteriormente apresentados é a existência de múltiplos grupos de populações (colônias) e a troca de soluções candidatas entre estes grupos (migração) colocadas em uma população paralela. Este método ajuda a evitar a convergência da solução para um máximo (ou mínimo) local. Por outro lado, a utilização simultânea de várias populações torna a implementação do algoritmo complicada e seu processamento demorado. Infelizmente, é difícil avaliar a relativa eficiência desse algoritmo em relação a outros, uma vez que os autores não implementaram nenhuma comparação para o algoritmo.

NUNES (1998) e SAMPAIO (1999) utilizaram um AG para o problema das p-medianas que foi proposto por MAYERLE (1994). Neste AG, a representação cromossômica é idêntica a do algoritmo de D&D descrito anteriormente. Quanto aos operadores genéticos, o algoritmo utiliza o operador genético de cruzamento e um operador de correção chamado “operador de mutação”. A ressalva nesse algoritmo é que o cruzamento entre os pais permite a duplicação de genes (medianas repetidas na solução), o que quando acontece, torna a nova solução gerada ineficaz. Para corrigir estas soluções ineficazes o operador genético de correção é acionado. Esse operador troca aleatoriamente o valor de um dos genes duplicados por outro que não faça parte da solução que será corrigida. O problema é que, sempre que o número de uma mediana estiver ocorrendo em quase todos os indivíduos da população, esse número tenderá a ser duplicado em alguns indivíduos como resultado do cruzamento. Portanto, o número de “correções” realizadas aumentará, o que pode fazer a busca ficar um tanto quanto aleatória, desviando-se assim do objetivo dos AGs e, possivelmente da solução ótima procurada.

ERKUT et al. (em fase de publicação) desenvolveram um AG para o problema das p -medianas. Neste algoritmo, cada indivíduo também possui exatamente p genes que representam um conjunto de p medianas selecionadas. O algoritmo traz além dos operadores genéticos usuais outros como, por exemplo, o *String-of-Change Operator* sugerido independentemente por BOOKER (1987) e FAIRLEY (1991). Esse operador utiliza um “*string* de mudança”, que consiste em um vetor binário gerado para cada par de pais selecionados para o cruzamento. Os cromossomos pais são submetidos a um operador chamado ou exclusivo (a expressão “ a ou exclusivo b ” é definida como sendo 1 se $a \neq b$ e 0 caso contrário). Para exemplificar, o vetor binário correspondente aos pais [10, 9, 12, 24, 7, 3] e [10, 9, 7, 8, 12, 3] seria [0, 0, 1, 1, 1, 0]. Ou seja, 0 se os genes de mesma posição nos pais representarem a mesma mediana e 1 caso contrário. Para evitar que os filhos gerados sejam idênticos aos pais, somente os genes entre o primeiro e o último 1 nos respectivos pais devem ser selecionados como pontos de cruzamento. Com base em testes comparativos realizados, o algoritmo demonstrou ser eficiente, embora, segundo o próprio autor, leva muito tempo para obter boas soluções.

4.3 BUSCA TABU

São apresentadas a seguir as idéias gerais resumidas de dois artigos sobre busca tabu aplicada ao problema das p -medianas.

O primeiro, é um artigo escrito por GLOVER (não publicado), onde a estrutura básica de seu algoritmo BT é a seguinte: considere o conjunto V de todas as instalações candidatas para o problema em questão e $V_p \subset V$ com $|V_p| = p$ o conjunto inicial de medianas selecionadas ao acaso. Cada “movimento” é um procedimento que consiste em adicionar, retirar ou trocar em V_p a mediana que resultar no melhor valor para a nova solução. Estes movimentos são executados em sequência e, desta forma, haverá uma variação no número de medianas em V_p , ou seja:

$$p - 1 \leq |V_p| \leq p + 1.$$

Este fenômeno é chamado de “estratégia da oscilação” e além de ajudar a evitar a convergência para uma solução local, conduz com maior rapidez a busca para soluções factíveis cada vez melhores.

A “lista tabu” memoriza a iteração em que cada mediana foi adicionada à solução e proíbe, durante um certo número de iterações subsequentes, que esta possa ser novamente adicionada. O “critério de aspiração” utilizado é permitir a quebra da restrição tabu se a qualidade da nova solução gerada por um movimento proibido superar a qualidade da melhor solução encontrado até o momento. O autor não chegou a implementar o método, apenas descreveu os passos necessários para sua implementação.

O segundo artigo mencionado é de ROLLAND et al. (1996). O funcionamento básico do algoritmo BT descrito nesse artigo é semelhante ao do algoritmo descrito anteriormente, exceto que são considerados apenas dois tipos de movimento: adicionar e retirar medianas em uma solução. Além disso, a ordem em que estes movimentos são executados é aleatória. Desta forma, o algoritmo pode numa iteração retirar uma mediana da solução e na iteração seguinte retirar outra novamente. O mesmo vale para o caso de adicionar. Em outras palavras, haverá uma oscilação maior que $p \pm 1$ no número de medianas da “solução-atual”. Este algoritmo foi aplicada a 100 problemas testes. Os testes foram realizados com 8 diferentes grafos para problemas pequenos (de 13 a 100 vértices) e 4 para problemas grandes (200 a 500 vértices). Cada vértice foi considerado, ao mesmo tempo, como demanda e potencial mediana. Todos os grafos foram gerados aleatoriamente. Os pesos associados a cada vértice foram definidos por números aleatórios gerados entre 0 e 100. Para cada grafo o problema foi resolvido para diversos valores de p . Os resultados foram comparados em termos de qualidade da solução e tempo de processamento, com os obtidos para os mesmos problemas resolvidos por outras duas heurísticas para o problema das p -medianas. Uma desenvolvida por GOODCHILD e NORONHA (1983) chamada de NS e a outra desenvolvida por DENSHAM e RUSHTON (1992) chamada de GRIA. Para os problemas pequenos (de 13 a 100 vértices) a solução ótima também foi calculada via programação linear. Nos experimentos com os problemas pequenos, a

heurística BT encontrou a solução ótima em 66% dos testes contra 48% para GRIA e 39% para NS. Em termos de qualidade da solução, BT foi melhor que ou igual a NS em 97% dos testes e melhor que ou igual a GRIA em 92% dos testes. Em relação ao tempo de processamento BT foi, na média, mais rápida que as outras duas heurísticas. Para os problemas grandes (de 200 a 500 vértices) BT encontrou a melhor solução entre as heurísticas em 91% dos testes e foi mais rápida em todos testes com os problemas grandes.

5 IMPLEMENTAÇÃO DOS MÉTODOS UTILIZADOS

Esta seção trata da implementação proposta no presente trabalho de cada um dos métodos descritos na seção 3, começando com o algoritmo que faz a designação de candidatos ao vestibular aos locais provas.

5.1 ALGORITMO PARA DESIGNAÇÃO DE CANDIDATOS AO VESTIBULAR

A forma básica do DeCan é semelhante a do algoritmo proposto por GILLET e JOHNSON (1973) (BODIN et al., 1983) descrito na seção 3. Difere basicamente no cálculo realizado para a ordenação da “lista de designação”, conforme apresentado a seguir.

Inicialmente, todos os pontos (residências dos candidatos) encontram-se sem designação. Para cada ponto i seja $L_1(i)$ o local de provas mais próximo a i , e $L_2(i)$ o segundo local mais próximo a i .

Para cada ponto i , a **diferença**: $d(i) = L_2(i) - L_1(i)$ é calculada e todos os pontos são colocados numa “lista de designação” em ordem **decrescente** pelos valores de $d(i)$. A designação começa pelos primeiros elementos da lista (pontos com maior diferença $d(i)$), e é feita obedecendo-se a capacidade dos locais de provas. Durante a designação sempre que a última vaga restante em um determinado local de provas for preenchida a diferença $d(i)$ será recalculada para todos os pontos que ainda não foram designados considerando-se apenas os locais de provas cujas vagas não estejam esgotadas. Estes pontos são novamente colocados na “lista de designação” em ordem decrescente pelos valores de $d(i)$ e a designação continuará até que todos os pontos tenham sido designados para algum local de provas.

O que justifica ter-se optado pelo cálculo da diferença ao invés da razão entre o primeiro e o segundo locais de provas mais próximos da residência do candidato é o seguinte fato: suponha-se que para dois candidatos quaisquer os dois locais de provas mais próximos de suas residências sejam os locais A e B . A residência do primeiro

candidato está a 100m do local A e a 300m do local B . A residência do segundo candidato está a 500m do local A e 900m do local B . Se o local A só puder aceitar mais um candidato, pelo cálculo da razão o primeiro candidato seria designado para o local A e percorreria 100m enquanto o segundo candidato seria designado para o local B e teria que percorrer 900m o que resultaria num acréscimo de 1000m na soma total das distâncias percorridas. Já pelo cálculo da diferença o primeiro candidato seria designado para o local B e percorreria 300m enquanto o segundo candidato seria designado para o local A e percorreria 500m o que resultaria num acréscimo de 800m na soma total das distâncias percorridas. Percebe-se que, com a utilização do cálculo da diferença a designação dos candidatos aos locais de provas ocorre de forma mais “justa” do que no caso do cálculo da razão.

Quando o objetivo for minimizar a soma total das distâncias percorridas o cálculo da diferença será sempre apropriado, conforme demonstrado a seguir:

Sejam i e j dois candidatos quaisquer “competindo” pelos mesmos locais L_1 e L_2 onde L_1 e L_2 são, nesta ordem, respectivamente o primeiro e o segundo locais de provas mais próximos tanto para i quanto para j . Ainda, L_1 possui apenas mais uma vaga.

Sejam $L_1(i)$, $L_2(i)$, $L_1(j)$ e $L_2(j)$ respectivamente as distâncias dos candidatos i e j aos locais L_1 e L_2 . Consideram-se duas hipóteses (a) e (b):

- a) a menor soma das distância é obtida se o candidato i for designado para o local L_1 e o candidato j para o local L_2 , o que implicaria em:

$$L_1(i) + L_2(j) < L_1(j) + L_2(i) \quad (5.1.1)$$

Através de operações matemáticas a inequação (5.1.1) pode ser transformada em:

$$L_2(i) - L_1(i) > L_2(j) - L_1(j) \quad (5.1.2)$$

A inequação (5.1.2) é exatamente a comparação entre os resultados do cálculo da diferença realizado para os candidatos i e j . Seguindo-se os critérios do algoritmo da designação proposto neste trabalho, como o resultado da diferença foi maior para o candidato i ele será designado primeiro. O que considerando-se apenas os dois candidatos garante que i seja designado para L_1 , atendendo-se a condição proposta em (a).

b) a menor soma das distâncias é obtida se o candidato j for designado para o local L_1 e o candidato i para o local L_2 , o que implicaria em:

$$L_1(j) + L_2(i) < L_1(i) + L_2(j) \quad (5.1.3)$$

Através de operações matemáticas a inequação (5.1.3) pode ser transformada em:

$$L_2(j) - L_1(j) > L_2(i) - L_1(i) \quad (5.1.4)$$

A inequação (5.1.4) é exatamente a comparação entre os resultados do cálculo da diferença realizado para os candidatos i e j . Seguindo-se os critérios do algoritmo da designação proposto neste trabalho, como o resultado da diferença foi maior para o candidato j ele será designado primeiro. O que considerando-se apenas os dois candidatos garante que j seja designado para L_1 , atendendo-se a condição proposta em (b).

5.1.1 Procedimentos Básicos do DeCan

Os procedimentos executados pelo DeCan proposto no presente trabalho são descritos através dos passos a seguir.

Passo 1

Para todo ponto i não designado, faça:

Encontre $L_1(i)$ e $L_2(i)$ respectivamente o primeiro e o segundo locais de provas mais próximos da residência do candidato i cujas capacidades não estejam esgotadas.

Calcule a diferença: $d(i) = L_2(i) - L_1(i)$ e coloque o ponto i na “lista de designação” pela ordem decrescente dos valores de $d(i)$.

Faça $\text{candidato_atual} = \text{primeiro candidato da “lista de designação”}$.

Passo 2

Enquanto existirem pontos sem designação, faça:

Designa o candidato_atual para o local de provas mais próximo de sua residência que dispuser de vaga.

Diminua uma unidade da capacidade do local de provas para o qual o candidato_atual foi designado.

Identifique o candidato_atual como designado.

$\text{candidato_atual} = \text{próximo candidato da “lista de designação”}$.

Se a capacidade do respectivo local de provas que recebeu o candidato_atual ficou esgotada (igual a zero), então:

Volte ao passo 1 e recalcule a “lista de designação”.

Caso contrário, continue.

Fim.

5.1.2 Melhorias no Resultado Final do DeCan

Tanto no algoritmo de GILLET e JOHNSON (1973) (BODIN et al., 1983) (descrito na seção 3) quanto no DeCan descrito anteriormente podem haver erros na designação dos candidatos como por exemplo: se houver empate no cálculo da diferença (ou da razão) para dois ou mais pontos (candidatos), estes serão designados apenas de acordo com a ordem em que os cálculos foram realizados. Ou seja, se forem calculados os valores da diferença para os pontos *i*, *j* ou *k*, nesta ordem, e todos os cálculos resultarem no mesmo valor, os três pontos serão colocados na “lista de designação” na ordem: primeiro *i*, depois *j*, e por último *k*. Mesmo que a ordem de designação dos candidatos seja recalculada quando um candidato ocupar a última vaga de um local de provas, pode acontecer que: o ponto *i*, primeiro que será designado, ocupe a última vaga do primeiro local mais próximo aos pontos *i*, *j*, e *k* e que o ponto *j* ocupe a última vaga do segundo local de provas mais próximo aos pontos *i*, *j*, e *k*. Desta forma, o ponto *k*, na melhor das hipóteses, será designado para o terceiro local de provas mais próximo a *k*. Porém, a distância entre o segundo e o terceiro locais de provas mais próximos ao ponto *k* pode ser muito maior que as mesmas distâncias para os candidatos *i* e *j*, mas, este fato não foi considerado. Talvez, seja justificada até mesmo a troca da designação do ponto *k* com a designação do ponto *i* ou *j* ou até mesmo com algum outro ponto que já tenha sido designado.

O fato de se determinar a ordem de designação dos candidatos levando-se em conta apenas a distância entre os dois primeiros locais de provas mais próximos pode gerar erros de designação, principalmente para aqueles candidatos que forem designados por último. Por outro lado, não há vantagem em se calcular as distâncias entre o segundo e terceiro, terceiro e quarto, etc. locais de provas mais próximos para cada candidato uma vez que, não se pode prever qual será o candidato designado para um determinado local de provas e nem mesmo quando um determinado local de provas terá suas vagas esgotadas.

Para corrigir alguns eventuais erros que podem acontecer na designação dos candidatos aos locais de provas pelo DeCan desenvolveu-se a chamada “análise pós-processamento”, que é apresentada a seguir.

5.1.2.1 Análise pós-processamento

Terminada a designação de todos os candidatos aos locais de provas, será feita uma busca em todos os locais de provas à procura dos candidatos que não foram designados para o primeiro local de provas mais próximo de suas residências. Estes candidatos são então colocados em uma “lista de troca”. Em seguida, para cada um dos candidatos da “lista de troca” é feita uma pesquisa em todos os locais de provas cujas vagas estejam esgotadas considerando-se a possibilidade de trocar cada candidato que estiver num destes locais com o candidato da “lista de troca” analisado. A troca que mais reduzir a soma total das distâncias percorridas é realizada. Caso não haja redução em nenhuma troca analisada o candidato da “lista de troca” permanecerá no local onde está. Um ponto que deve ser esclarecido é que não há necessidade de considerar locais que ainda possuam vagas, pois, caso um destes fosse uma opção melhor, o DeCan já teria se encarregado de fazer a designação do candidato para o local, já que este dispunha de vagas.

Trocar dois candidatos significa simplesmente inverter os locais de provas para os quais ambos foram inicialmente designados, ou seja, o candidato X (candidato da “lista de troca”) passa a ocupar a posição do candidato Y no local de provas onde Y está e vice versa. Para evitar que o candidato Y seja “punido” sendo obrigado a ocupar uma vaga num local que pode não ser uma boa opção para ele, é considerada a hipótese de se transferir Y para cada um dos locais de provas que possua pelo menos uma vaga disponível. Além disso, de qualquer forma o candidato Y é automaticamente inserido na “lista de troca” e terá futuramente outra possibilidade de melhorar sua situação.

5.1.2.1.1 Procedimentos básicos da análise pós-processamento

A análise pós-processamento é executada somente depois que todos os candidatos foram designados para os locais de provas e consiste dos passos apresentados a seguir.

Passo 1

Registre na “lista de troca” todos os candidatos que não foram designados para o primeiro local de provas mais próximo de suas residências.

Faça `candidato_atual` = primeiro candidato da “lista de troca”.

Passo 2

Enquanto existir candidato não analisado na “lista de troca” faça:

Analise a possibilidade de trocar o `candidato_atual` com cada um dos candidatos que estejam em locais de provas cujas vagas estejam esgotadas.

Passo 3

Se existir alguma troca observada no passo 2 que resulte em redução da soma total das distâncias percorridas, então:

Execute a troca que mais reduzir o valor da soma total das distâncias percorridas.

Faça `candidato_atual` = próximo candidato da “lista de troca”.

Vá para o passo 4.

Caso contrário:

Faça `candidato_atual` = próximo candidato da “lista de troca”.

Volte ao passo 2.

Passo 4

Para o candidato que foi trocado com o `candidato_atual`, faça:

Analise a possibilidade de transferi-lo do novo local onde ele está para cada um dos locais de provas que ainda possui pelo menos uma vaga.

Insira o candidato na lista de troca para que este tenha ainda outra chance de melhorar sua designação.

Passo 5

Se existir alguma transferência observada no passo 4 que resulte em redução da soma total das distâncias percorridas, então:

Execute a transferência que mais reduzir o valor total da soma total das distâncias percorridas.

Volte ao passo 2.

Fim.

5.2 ALGORITMO GENÉTICO PROPOSTO (AGpMcap)

Nesta seção, são apresentados os aspectos computacionais do algoritmo genético proposto no presente trabalho para o problema das p -medianas capacitado.

Observe que dado um conjunto $V_p \subset V$ com exatamente p medianas, não se pode garantir que V_p possa atender a todo o conjunto de demandas. Caso a soma das capacidades das instalações medianas contidas em V_p seja inferior ao número total de demandas o conjunto V_p em questão será considerado uma solução infactível para o problema das p -medianas capacitado e não poderá ser utilizado, pois, não atende a todos os pontos de demanda. Portanto, para preservar a clareza dos procedimentos do AGpMcap que são explicados a seguir, deve-se considerar que qualquer conjunto $V_p \subset V$ com exatamente p instalações medianas pode atender a todo o conjunto de demandas. Na prática uma forma simples de garantir que qualquer conjunto $V_p \subset V$ possa atender a todo o conjunto de demandas é determinar p da seguinte forma:

Primeiramente as n instalações candidatas são colocadas em uma lista pela ordem crescente de suas capacidades. Em seguida, considera-se p igual ao menor número de instalações da primeira da lista em diante cuja soma das capacidades seja igual a ou maior que o número total de demandas. Desta forma, qualquer conjunto com p instalações selecionadas entre as n instalações candidatas poderá sempre atender a todo o conjunto de demandas.

5.2.1 Representação Cromossômica

Foi utilizada a representação cromossômica proposta por DIBBLE e DENSHAM (1993). Cada indivíduo (ou cromossomo) possui exatamente p genes, e o alelo de cada gene representa o valor correspondente a uma instalação utilizada como mediana. Por exemplo, considere que para um determinado problema com 15 instalações (possíveis medianas) representadas respectivamente pelos números 1, 2, ..., 15 se deseje selecionar exatamente 5 medianas. Segundo a representação utilizada, o indivíduo [2, 7, 5, 15, 10] representa uma solução para o problema em que as instalações 2, 5, 7, 10 e 15 são instalações selecionadas como medianas. Durante todo o AGpMcap o genoma é interpretado como um conjunto de genes, no sentido matemático de conjuntos, onde não há elementos repetidos e não há ordenação entre os elementos. Definida a representação dos indivíduos, será formada uma população de possíveis soluções para iniciar a resolução do problema, conforme discutido a seguir.

5.2.2 Tamanho da População

O tamanho P da população utilizada em um AG pode variar de acordo com as características de cada problema. Para o AGpMcap o número de indivíduos da população será estimado através da seguinte fórmula proposta por ERKUT (em fase de publicação):

$$P = \left\lfloor k \cdot \frac{n}{p} \right\rfloor, \quad (5.2.1)$$

onde $k > 1$ é um número natural determinado pelo usuário, n é o número total de instalações candidatas, p é o número de medianas que se deseja selecionar e o símbolo $\lfloor b \rfloor$ significa o maior inteiro menor que b . Como os genes dos cromossomos filhos são determinados através da “mistura” entre os genes dos cromossomos pais, se uma das instalações de uma solução ótima não estiver presente nos genes dos cromossomos da

população inicial, o algoritmo terá menor probabilidade de encontrar aquela solução ótima para o problema. Uma população com P indivíduos possui um total de $P \cdot p$ genes. Se este número for igual a $k \cdot n$ cada instalação candidata a mediana, em média, deverá aparecer k vezes entre os $P \cdot p$ genes da população inicial. Certamente, as instalações candidatas que não estiverem presentes entre os genes dos indivíduos da população inicial poderão ser inseridas na população por operadores como por exemplo, mutação. Porém, quanto maior for o valor atribuído a k mais reduzida ficará a probabilidade de uma determinada instalação não estar presente na população inicial.

Segundo ERKUT (em fase de publicação), no caso de se determinar o tamanho da população pela fórmula (5.2.1), a probabilidade de uma instalação não estar presente na população inicial pode ser estimada pela fórmula:

$$P_{\text{INST_NÃO_PRESENTE}} = \left| \frac{n - I}{n} \right|^{k \cdot n}. \quad (5.2.2)$$

Assim, pode-se escolher um valor para k suficientemente alto para que o valor da fórmula (5.2.2) seja pequeno, ou menor que um valor predefinido pelo usuário.

Definido o tamanho P da população, são gerados aleatoriamente P indivíduos para formar a população inicial. Em seguida, o grau de adaptação destes indivíduos é avaliado conforme discutido a seguir.

5.2.3 Avaliação do Indivíduo (*fitness*)

Depois de gerada a população inicial, é calculada a *fitness* de cada indivíduo que é o valor da função objetivo correspondente à solução (conjunto de medianas) que o indivíduo representa. Em seguida, todos os indivíduos são colocados em uma lista R pela ordem crescente do valor da *fitness* de cada indivíduo gerado. Para reduzir o

tempo de processamento, no AGpMcap a *fitness* de cada indivíduo é fornecida pelo algoritmo simplificado da designação dos candidatos aos locais de provas que será apresentado a seguir. Neste algoritmo simplificado, os candidatos ao vestibular são designados para o primeiro local de provas mais próximo que dispuser de vaga e na ordem estabelecida pelo cálculo da diferença. A designação não é interrompida quando um local de provas tem suas vagas esgotadas e a análise pós-processamento não é executada. Ao final do AGpMcap para a melhor solução encontrada será aplicado o DeCan completo.

5.2.3.1 Procedimentos básicos do DeCan simplificado

Os procedimentos básicos executados pelo DeCan simplificado são descritos através dos passos a seguir.

Passo 1

Para todo ponto i não designado, faça:

Encontre $L_1(i)$ e $L_2(i)$ respectivamente o primeiro e o segundo locais de provas mais próximos da residência do candidato i cujas capacidades não estejam esgotadas.

Calcule a diferença: $d(i) = L_2(i) - L_1(i)$ e coloque o ponto i na “lista de designação” pela ordem decrescente dos valores de $d(i)$.

Faça candidato_atual = primeiro candidato da “lista de designação”.

Passo 2

Enquanto existirem pontos sem designação, faça:

Designa o candidato_atual para o local de provas mais próximo de sua residência que dispuser de vaga.

Diminua uma unidade da capacidade do local de provas para o qual o candidato_atual foi designado.

Identifique o candidato_atual como designado.

candidato_atual = próximo candidato da “lista de designação”.

Fim.

Durante a execução do AGpMcap, como o objetivo do problema das p-medianas é minimizar a soma total das distâncias percorridas, são considerados mais adaptados os indivíduos que possuírem os menores valores da *fitness* e estes terão maior probabilidade de serem selecionados para cruzamento ou reprodução, conforme discutido a seguir.

5.2.4 Seleção dos Cromossomos Pais

Depois que os indivíduos estiverem ordenados na população, a seleção de cada cromossomo pai que irá sofrer cruzamento ou reprodução é realizada com base numa fórmula proposta por MAYERLE (1996). Esta fórmula foi ligeiramente modificada mas, assim como a original, privilegia a escolha de indivíduos com melhor *fitness*, isto é:

$$\text{Select}(\mathbf{R}) = \left\{ r_j \in \mathbf{R} / j = P - \left\lfloor \frac{-1 + \sqrt{1 + 4 \cdot \text{rnd}(P^2 + P)}}{2} \right\rfloor \right\}, \quad (5.2.3)$$

onde \mathbf{R} é uma lista $\mathbf{R} = (r_1, r_2, \dots, r_P)$, com P cromossomos colocados em ordem crescente pelo valor da *fitness*, $\text{rnd} \in [0, 1)$ é um número aleatório uniformemente distribuído e o símbolo $\lfloor b \rfloor$ significa o maior inteiro menor que b . Esta fórmula (5.2.3) retorna um número natural aleatório igual a posição na lista \mathbf{R} do cromossomo que será selecionado. Apesar de ser aleatório, o número retornado pela fórmula (5.2.3) tem maior tendência de selecionar os primeiros elementos da lista \mathbf{R} (melhores indivíduos da lista). Depois que os cromossomos pais forem selecionados, são submetidos a uma comparação que resultará em dois vetores de troca, conforme discutido a seguir.

5.2.5 Vetores de Troca

Antes que o operador genético de cruzamento ou a reprodução seja aplicada, para evitar que um gene possa ficar duplicado nos filhos gerados pelos pais selecionados são criados dois vetores de troca (um para cada pai) pelo seguinte critério:

Para todo gene do pai 1 uma consulta (da esquerda para a direita) é feita com o objetivo de saber se no pai 2 existe um gene com o mesmo valor. Se o gene analisado do pai 1 não existir no pai 2, será copiado para o vetor de troca do pai 1. Isto significa que se este gene for transferido para o pai 2 não ficará duplicado, pois, não existe no pai 2. O mesmo procedimento é realizado para o pai 2. Exemplo: sejam os pais [1, 2, 3, 4, 5] e [2, 5, 9, 10, 12] seus respectivos vetores de troca são: $vp_1 = [1, 3, 4]$ e $vp_2 = [9, 10, 12]$. Depois de identificados os genes que podem ser trocados entre os pais, o operador de cruzamento ou o operador de reprodução é aplicado conforme discutido a seguir.

5.2.6 Operador Genético de Cruzamento

Não foi utilizada uma probabilidade fixa para o operador genético de cruzamento. O cruzamento entre os pais ocorre sempre que existir pelo menos um gene para troca nos vetores vp_1 e vp_2 e é realizado da seguinte forma: um número natural aleatório c que pode variar de 1 até $(|vp| - 1)$ é gerado para determinar quantos genes dos vetores de troca vp_1 e vp_2 serão transferidos para os pais. Em seguida, os c primeiros genes do vetor vp_1 são transferidos para o pai 2 colocados da esquerda para a direita e os c primeiros genes do vetor vp_2 são transferidos para o pai 1 também colocados da esquerda para a direita. Desta forma, dois novos indivíduos (filhos) são gerados com a vantagem de que não haverá genes duplicados em nenhum dos filhos. Para cada um dos filhos gerados pelo cruzamento é calculado o valor da *fitness*. Para o processo de evolução da população foi utilizado o “método do estado estável”. Ou seja, o melhor filho gerado é inserido na lista R no lugar do último indivíduo, mas,

somente se possuir uma *fitness* melhor que a do último indivíduo da lista. O outro filho gerado também é inserido no lugar do penúltimo indivíduo da lista R se possuir uma *fitness* melhor que este. Em seguida, a lista R é reordenada pela ordem crescente do valor da *fitness* dos indivíduos da lista. Caso nenhum dos filhos possua uma *fitness* melhor que a do último indivíduo da lista R, nenhum deles será inserido na lista. Se não existirem genes distintos entre os pais selecionados, o operador genético de cruzamento não será aplicado. Neste caso, será aplicado o operador genético de reprodução que é discutido a seguir.

5.2.7 Operador Genético de Reprodução

Em relação ao operador genético de reprodução comum utilizado em AGs, o operador genético de reprodução utilizado no AGpMcap foi ligeiramente modificado. No AGpMcap este operador copia integralmente apenas o primeiro pai selecionado para a lista R e o coloca no lugar do último indivíduo da lista. Em seguida, a lista R é reordenada pela ordem crescente do valor da *fitness* dos indivíduos da lista. Note que se fosse aplicado o cruzamento entre dois pais que possuem exatamente os mesmos genes, os filhos gerados seriam idênticos aos pais. Já que o operador de cruzamento requer um número maior de operações, a aplicação do operador de reprodução neste caso agiliza a execução do algoritmo além de evitar que dois indivíduos iguais sejam eventualmente inseridos na população (lista R).

5.2.8 Operador de Mutação

Em relação ao operador genético de mutação comum utilizado em AGs, o operador genético de mutação utilizado no AGpMcap foi ligeiramente modificado. No AGpMcap este operador poderá ser aplicado logo após a seleção dos pais e antes do cruzamento ou reprodução. A probabilidade de ocorrer mutação é definida por um número fixo (por exemplo 1%). Para exemplificar, considere que a probabilidade de mutação seja de 0,01. Será sorteado um número aleatório pertencente ao intervalo

numérico $[0, 1]$, se este número for menor que ou igual a 0,01, a mutação será aplicada a cada um dos pais selecionados da seguinte forma: um gene do primeiro pai é escolhido ao acaso e seu valor é substituído por outro também escolhido ao acaso que não esteja presente no pai que está sofrendo mutação. O mesmo procedimento é aplicado ao segundo pai. Em seguida os pais são submetidos ao cruzamento ou reprodução. Note que o operador genético de mutação altera o código genético dos pais antes que estes sejam cruzados ou reproduzidos. Caso o número aleatório sorteado determine que a mutação não deva ser aplicada os pais selecionados não sofrerão nenhuma modificação e participarão intactos do processo de cruzamento ou reprodução.

5.2.9 Hipermutação Direcionada

A hipermutação direcionada é um operador heurístico proposto neste trabalho, e é baseado no domínio (conhecimento) do problema. Seu objetivo principal é fornecer de maneira rápida características que possam contribuir para a evolução da população. Este operador é executado uma vez logo após a geração da população inicial e depois disso terá uma probabilidade fixa de ocorrer (por exemplo 0,5%) a cada iteração (geração de filhos) do processo evolutivo pelo “método do estado estável”. Consiste em selecionar aleatoriamente uma porcentagem (por exemplo 10%) dos indivíduos da população atual e tentar melhorar a *fitness* de cada um dos indivíduos selecionados experimentando trocar o valor de cada um dos genes do indivíduo por valores (instalações candidatas) que não estejam presentes no indivíduo. As trocas que mais contribuírem para a melhoria da qualidade (valor da *fitness*) do indivíduo são realizadas. Em seguida, a lista R é reordenada pela ordem crescente do valor da *fitness* dos indivíduos da lista.

O AGpMcap prossegue por algumas iterações executando os procedimentos descritos anteriormente até que uma condição de término seja verificada, conforme discutido a seguir.

5.2.10 Condição de Término

No AGpMcap a condição de término utilizada é a de um número fixo de iterações ser completado. Uma iteração completa corresponde a seleção dos pais e a aplicação dos operadores (não necessariamente todos) descritos nos itens anteriores. A iteração termina quando um ou os dois novos indivíduos depois de avaliados são inseridos (ou não) na lista R. Quando dois novos pais forem selecionados, terá se iniciado uma nova iteração. Os procedimentos básicos do AGpMcap são apresentados a seguir.

5.2.11 Procedimentos Básicos do AGpMcap

Primeiramente é definida a notação utilizada para os parâmetros do AGpMcap e em seguida cada etapa da implementação do algoritmo é explicada.

5.2.11.1 Notação utilizada

P = Tamanho da população.

T = Número da iteração atual.

$Máx_iter$ = Número máximo de iterações.

P_{MUT} = Probabilidade de ocorrer mutação.

$P_{HIPERMUT_DIREC}$ = Probabilidade de ocorrer a hipermutação direcionada.

H = Número de indivíduos da população que serão selecionados a cada vez que a hipermutação direcionada for aplicada.

Lista R = População atual.

5.2.11.2 O AGpMcap passo a passo

Passo 0 (Inicialização do algoritmo)

Defina os valores de Máx_iter, P_{MUT} , $P_{HIPERMUT_DIREC}$ e H .

Atribua um valor inteiro positivo para k e determine o número P de indivíduos da população através da fórmula (5.2.1).

Gere aleatoriamente uma lista R com P indivíduos.

Avalie os P indivíduos e ordene a lista R pela ordem crescente do valor da *fitness* dos indivíduos da lista.

Faça $T = 0$.

Através da fórmula (5.2.3) selecione H indivíduos da lista R para aplicar a hipermutação direcionada.

Para cada um dos H indivíduos selecionados faça:

Experimente trocar o valor de cada gene do indivíduo por cada um dos valores possíveis que não estejam presentes no indivíduo.

Realize as trocas que mais contribuïrem para a melhoria da *fitness* do indivíduo (se existirem).

Se alguma troca foi realizada, então:

Reavalie o indivíduo calculando sua *fitness*.

Depois que todos os indivíduos selecionados sofreram a hipermutação direcionada, reordene a lista R pela ordem crescente do valor da *fitness* dos indivíduos da lista.

Passo 1 (Seleção dos indivíduos pais)

Faça $T = T + 1$.

Enquanto $T \leq \text{Máx_iter}$, faça:

Selecione dois indivíduos da lista R através da fórmula (5.2.3).

Caso contrário, apresente o primeiro indivíduo da lista R como uma solução aproximada para o problema das p -medianas capacitado.

Passo 2 (Operador de mutação)

Sorteie um número aleatório $\text{rnd} \in [0, 1]$.

Se $\text{rnd} \leq P_{MUT}$, então:

Escolha aleatoriamente um gene do pai 1 e aplique mutação ao gene.

Escolha aleatoriamente um gene do pai 2 e aplique mutação ao gene.

Caso contrário, vá direto ao passo 3.

Passo 3 (Vetor de troca e operador de cruzamento)

Gere os vetores de troca vp_1 e vp_2 respectivamente para o primeiro e o segundo pai selecionados.

Se existe em vp_1 pelo menos um gene para troca, então:

Sorteie um número natural c entre 1 e $(|vp_1| - 1)$.

Transfira os c primeiros genes do vetor vp_1 para o pai 2 da esquerda para a direita e transfira os c primeiros genes do vetor vp_2 para o pai 1 também da esquerda para a direita.

Avalie os dois filhos gerados através do cálculo da *fitness* de cada um deles.

Se o melhor filho gerado possuir uma *fitness* melhor que a do último indivíduo da lista R, então:

Insira o melhor filho gerado na lista R no lugar do último indivíduo da lista R.

Se o outro filho gerado possuir uma *fitness* melhor que a do penúltimo indivíduo da lista R, então:

Insira este filho na lista R no lugar do penúltimo indivíduo da lista.

Caso contrário, continue.

Reordene a lista R pela ordem crescente do valor da *fitness* dos indivíduos da lista.

Vá direto ao passo 5.

Caso contrário, vá ao passo 5.

Caso contrário, vá ao passo 4.

Passo 4 (Operador de reprodução)

Copie integralmente apenas o primeiro pai selecionado para a lista R no lugar do último indivíduo da lista.

Reordene a lista R pela ordem crescente do valor da *fitness* dos indivíduos da lista.

Passo 5 (hipermutação direcionada)

Sorteie um número aleatório $rnd \in [0, 1]$.

Se $\text{rnd} \leq P_{\text{HIPERMUT_DIREC}}$, então:

Através da fórmula (5.2.3) selecione H indivíduos da lista R para aplicar a hipermutação direcionada.

Para cada um dos indivíduos selecionados faça:

Experimente trocar o valor de cada gene do indivíduo por cada um dos valores possíveis que não estejam presentes no indivíduo.

Realize as trocas que mais contribuïrem para a melhoria da *fitness* do indivíduo (se existirem).

Reavalie o indivíduo calculando seu valor da *fitness*.

Depois que todos os indivíduos selecionados sofreram a hipermutação direcionada, reordene a lista R pela ordem crescente do valor da *fitness* dos indivíduos da lista.

Volte ao passo 1.

Fim.

5.3 HEURÍSTICA BUSCA TABU PROPOSTA (BTpMcap)

Nesta seção, são apresentados os aspectos computacionais da heurística busca tabu proposta no presente trabalho para o problema das p -medianas capacitado.

Para preservar a clareza dos procedimentos da heurística BTpMcap que são explicados a seguir, assim como na seção anterior, deve-se considerar que qualquer conjunto $V_p \subset V$ com exatamente p instalações medianas pode sempre atender a todo o conjunto de demandas.

5.3.1 Solução Inicial

Seja V o conjunto de todas as instalações medianas possíveis e $V_p \subset V$ de tal forma que $(p - 1 \leq |V_p| \leq p + 1)$. Inicialmente um conjunto $V_p \subset V$ é gerado aleatoriamente com exatamente p instalações medianas. Este conjunto é uma solução inicial para o problema das p -medianas. Em seguida, novas soluções são geradas conforme discutido a seguir.

5.3.2 Movimentos e Restrições Tabu

Na heurística BTpMcap, a geração de novas soluções tem seu foco principal em analisar a cada iteração uma solução vizinha à solução corrente. As soluções vizinhas são geradas basicamente por três tipos de movimentos: adicionar (ADD), retirar (DROP) ou trocar (SWAP) uma instalação candidata na solução corrente.

Os movimentos ADD, DROP e SWAP responsáveis por gerar novas soluções funcionam da seguinte forma:

- a) ADD: O movimento ADD chamado “construtivo” consiste basicamente em selecionar uma instalação candidata do conjunto $\{V - V_p\}$ e colocá-la em V_p de tal forma que o valor da nova solução obtida seja o melhor possível;
- b) DROP: O movimento DROP chamado “destrutivo” seleciona a instalação mediana que retirada de V_p não torne V_p infactível (sem capacidade de suprir todo o conjunto de demandas) e que resulte no menor prejuízo possível para a solução e “devolve” esta instalação para o conjunto $\{V - V_p\}$, retirando-a de V_p ;
- c) SWAP: O movimento SWAP consiste em trocar a instalação de V_p que substituída por outra do conjunto $\{V - V_p\}$ resulte no melhor resultado possível para a solução atual e que não torne V_p infactível. Este movimento é executado sempre que se tiver $|V_p| = p$ entre a alternância da execução dos movimentos ADD e DROP.

5.3.2.1 Procedimento Escolha_Movimento passo a passo

A execução dos movimentos é iniciada logo após uma solução inicial aleatória V_p com exatamente p medianas ter sido gerada e a seqüência da execução dos movimentos será descrita a seguir.

Passo 1 (movimento DROP)

Retire de V_p e devolva a $\{V-V_p\}$ a instalação mediana que resultar na melhor solução possível (obs.: V_p deve continuar atendendo todo o conjunto de demandas).

Passo 2 (movimento ADD)

Adicione a V_p a instalação mediana de $\{V-V_p\}$ que resultar na melhor solução possível.

Passo 3 (movimento SWAP)

Troque a instalação mediana de V_p que substituída por outra de $\{V-V_p\}$ resulte na melhor solução possível (obs.: V_p deve continuar atendendo todo o conjunto de demandas).

Passo 4 (movimento ADD)

Adicione a V_p a instalação mediana de $\{V-V_p\}$ que resultar na melhor solução possível.

Passo 5 (movimento DROP)

Retire de V_p e devolva a $\{V-V_p\}$ a instalação mediana que resultar na melhor solução possível (obs.: V_p deve continuar atendendo todo o conjunto de demandas).

Passo 6 (movimento SWAP)

Troque a instalação mediana de V_p que substituída por outra de $\{V-V_p\}$ resulte na melhor solução possível (obs.: V_p deve continuar atendendo todo o conjunto de demandas).

Volte ao passo 1.

Fim.

Para reduzir o tempo de processamento, na BTPMcap a qualidade (valor) de cada possível solução analisada será fornecida pelo mesmo algoritmo simplificado da designação dos candidatos aos locais de provas utilizado para calcular a *fitness* nos AGs, já que cada possível solução para o BTPMcap também representa um conjunto de instalações candidatas a locais de provas. Ao final da BTPMcap para a melhor solução encontrada será aplicado o DeCan completo.

5.3.3 Restrições Tabu

As restrições tabu são impostas somente aos movimentos ADD e SWAP. Isto é, quando uma instalação candidata for adicionada ou trocada em Vp será classificada como tabu. O tempo que uma instalação candidata permanecerá classificada como tabu, mensurado em termo de número de iterações, será calculado da seguinte forma:

Seja $Add_tempo(v_i)$ o número da última iteração em que a instalação v_i foi adicionada ou trocada em Vp no lugar de outra, $Iteração_atual$ o número da iteração atual e $Tempo_tabu$ o número de iterações que uma instalação candidata permanecerá classificada como *tabu*, tem-se que para toda instalação v_i :

- a) se $Add_tempo(v_i) \geq Iteração_atual - Tempo_tabu$, a instalação v_i é considerada tabu;
- b) caso contrário a instalação v_i não é considerada tabu.

Neste trabalho, o $Tempo_tabu$ não é fixo (varia em uma unidade para mais ou para menos sempre que não houver melhoria na solução corrente durante por exemplo, 20% do número total de iterações do algoritmo). Testes empíricos mostraram que, quando diferentes valores para $Tempo_tabu$ podem ser determinados de forma aleatória a eficiência da heurística BTPMcap aumenta.

5.3.4 Critérios de Aspiração (Critérios para cancelamento das restrições tabu)

Em BT os critérios de aspiração são regras que determinam quando uma restrição tabu pode ser “quebrada” e um movimento classificado como tabu pode ser realizado. Em geral, isto acontece quando o movimento considerado, de alguma forma, conduz a busca para uma solução de alta qualidade. Na heurística BTpMcap é utilizado apenas um critério de aspiração baseado em duas condições.

Ou seja, um movimento classificado como tabu pode ser realizado se:

- a) a solução obtida com o movimento for uma solução factível ($|Vp| = p$), e;
- b) a solução gerada pelo movimento for melhor que a melhor solução encontrada pela busca até o momento.

Se as duas condições do critério de aspiração forem verdadeiras qualquer movimento classificado como tabu pode ser realizado. Com a intenção de diversificar a busca, além das restrições tabu é considerada ainda a frequência com que as instalações foram adicionadas ou trocadas em Vp durante o processo de busca, conforme discutido a seguir.

5.3.5 Diversificação e Memória de Longo Prazo

A diversificação é utilizada na heurística BTpMcap para escapar de soluções ótimas locais e é ativada pela chamada “memória de longo prazo”. No presente trabalho, a memória de longo prazo é uma lista que armazena a frequência (número de vezes) que cada instalação candidata foi adicionada ou trocada em Vp . Esta frequência é representada por $\text{Freq}(v_i)$.

É utilizada uma função de penalização denotada por Π , onde:

$$\Pi(v_i) = k \cdot \text{Freq}(v_i), \quad (5.3.1)$$

e k é uma constante arbitrária. A penalização (5.3.1) é somada à distância de v_i até todos os pontos de demanda sempre que a instalação v_i é adicionado ou inserida em V_p . Desta forma, procura-se inibir a escolha de instalações freqüentemente selecionadas. Esta estratégia promove uma diversificação na busca e tende a investigar áreas pouco exploradas do espaço de busca.

O parâmetro k é escolhido de tal forma que a penalização tenha um impacto significativo sobre a função objetivo.

Na heurística aqui apresentada foi utilizado $k = \text{Máximo}\{d(V, v_j)\}$ fixo durante todo o algoritmo e para todo ponto v_j pertencente ao conjunto de demandas onde $d(V, v_j)$ é a distância do ponto de demanda v_j até sua instalação mais próxima.

De forma resumida, a ação da diversificação nos procedimentos ADD e SWAP é a seguinte:

- a) para cada instalação candidata $v_i \in \{V - V_p\}$ adicionado ou trocada em V_p calcule a função de penalização (5.3.1);
- b) para todo ponto de demanda v_j que for designado para a instalação v_i , faça:
 $d(v_i, v_j) = d(v_i, v_j) + \Pi(v_i)$ e então calcule o valor da solução;
- c) escolha a instalação candidata $v_i \in \{V - V_p\}$ que adicionado ou trocada em V_p resultou na melhor solução possível (considerando-se a penalização imposta), efetue o movimento considerado e calcule o valor da solução atual (agora sem a penalização).

5.3.6 Estratégia da Oscilação

Note que a sequência de movimentos ADD e DROP não geram sempre soluções factíveis, ou seja $|Vp|$ não é necessariamente igual a p . A aceitação temporária destas soluções infactíveis é um outro meio de escapar de ótimos locais. Embora possa parecer o contrário, desta forma, o algoritmo tende a direccionar a busca mais rapidamente para soluções factíveis cada vez melhores. Esta estratégia chamada de “oscilação” no presente trabalho foi implementada da seguinte forma:

Partindo-se da solução inicial aleatória com exactamente p instalações medianas aplica-se o movimento DROP que retira uma instalação mediana deixando Vp com $p - 1$ medianas. Em seguida é aplicado o movimento ADD que adiciona uma mediana fazendo com que Vp volte a conter exactamente p instalações medianas. Como $|Vp| = p$, o próximo movimento executado é o movimento SWAP que troca uma mediana de Vp por outra do conjunto $\{V - Vp\}$ mantendo o conjunto Vp com exactamente p medianas. Em seguida é aplicado o movimento ADD que adiciona uma instalação mediana fazendo agora com que Vp passe a conter $p + 1$ medianas. Novamente é aplicado o movimento DROP que reduzirá novamente o número de instalações medianas em Vp de $p + 1$ para p instalações. O próximo movimento aplicado é o movimento SWAP e em seguida o processo é reiniciado com aplicação do movimento DROP e a sequência de movimentos descrita anteriormente é executada até que uma condição de término seja verificada (por exemplo, um número máximo de iterações ser completado).

A melhor solução factível gerada é armazenada e considerada como a solução factível corrente. Esta solução será apresentada como uma solução aproximada para o problema das p -medianas ao final da execução da heurística BTpMcap. Os procedimentos básicos realizados durante a execução da BTpMcap são apresentados a seguir.

5.3.7 Procedimentos Básicos da Heurística BTpMcap

Primeiramente é definida a notação utilizada para os parâmetros da BTpMcap e em seguida cada etapa da implementação desta heurística é explicada.

5.3.7.1 Notação utilizada

Max_iterações = Número máximo de iterações.

Iteração_atual = Número equivalente a iteração atual.

Tempo_tabu = Número de iterações tabu.

Add_tempo(v_i) = Número da iteração em que a instalação candidata v_i foi adicionada ou inserida em Vp .

Freq(v_i) = Número de vezes que a instalação candidata v_i foi adicionada ou inserida em Vp .

k = Maior distância de um ponto de demanda v_j até sua instalação mais próxima.

Estabilidade = Número máximo de iterações suportadas sem melhoria na solução.

Sem_melhoria = Número de iterações sem melhoria na solução corrente.

5.3.7.2 A heurística BTpMcap passo a passo

Passo 0

Defina Max_iterações, Tempo_tabu, Estabilidade e calcule k .

Gere aleatoriamente um conjunto $Vp \subset V$ com p instalações medianas, avalie Vp e considere o conjunto Vp como a solução corrente para o problema.

Faça Iteração_atual = 0.

Sem_melhoria = 0.

Passo 1

Faça Iteração_atual = Iteração_atual + 1.

Enquanto Iteração_atual \leq Max_iterações, faça:

Execute o procedimento Escolha_Movimento.

Se o movimento executado foi ADD ou SWAP, então:

$\text{Add_tempo}(v_i) = \text{Iteração_atual}.$

$\text{Freq}(v_i) = \text{Freq}(v_i) + 1.$

Vá para o passo 2.

Caso contrário, apresente a solução corrente como uma solução aproximada para o problema das p-medianas.

Passo 2

Calcule o valor da nova solução.

Passo 3

Se $|V_p| = p$ e a solução atual é melhor que a solução corrente, então:

Atualize a solução corrente.

Passo 4

Se não houve melhoria na solução corrente, então:

$\text{Sem_melhoria} = \text{Sem_melhoria} + 1.$

Caso contrário, $\text{Sem_melhoria} = 0.$

Passo 5

Se $\text{Sem_melhoria} = \text{Estabilidade}$, então:

Atribua aleatoriamente um novo valor para $\text{Tempo_tabu}.$

Volte ao passo 1.

Fim.

6 TESTES COMPUTACIONAIS E RESULTADOS

Nesta seção, já de posse da noção geral dos algoritmos propostos neste trabalho, primeiramente são apresentados os resultados reais obtidos com a designação dos 19.710 candidatos ao vestibular 2001 da UFPR pelo algoritmo DeCan e utilizados pela CCCV. Porém, a primazia será concedida às simulações computacionais realizadas com as heurísticas AGpMcap e BTpMcap que focalizam a seleção de 26 locais de provas dentre 43 locais disponíveis de tal forma que a soma das distâncias percorridas pelos mesmos 19.710 candidatos ao vestibular 2001 da UFPR seja minimizada. Todos os dados utilizados nas simulações realizadas são verdadeiros.

6.1 RESULTADOS COMPUTACIONAIS OBTIDOS PELO DeCan

Para o processamento do DeCan foi utilizado um computador Pentium III 550 MHz com 128 megabytes de RAM. O DeCan opera de forma determinística, ou seja, para um mesmo conjunto de locais de provas (não importando a ordem dos locais) cada candidato é designado sempre para o mesmo local de provas e na mesma ordem de designação. Como no problema real abordado os 26 locais de provas já estavam determinados, o DeCan foi aplicado, a pedido da CCCV, para efetuar a designação dos 19.710 candidatos ao vestibular 2001 para os 26 locais de provas mencionados na seção 2. Os resultados reais obtidos e utilizados pela CCCV no concurso vestibular 2001 estão na tabela 6.1.

Uma nova simulação da designação destes 19.710 candidatos ao vestibular 2001 para os mesmos 26 locais foi feita. Nesta nova simulação o algoritmo DeCan foi alterado. Ao invés de utilizar o cálculo da diferença, o DeCan utilizou o cálculo da razão conforme discutido na seção 5. Desta forma, foi possível comparar os resultados obtidos pelo DeCan no caso da utilização do cálculo da diferença proposta no presente trabalho e no caso da utilização do cálculo da razão proposta por GILLET e JOHNSON (1973) apresentado na seção 3. A tabela 6.2 apresenta os resultados obtidos pelo DeCan nesta nova simulação computacional realizada com o DeCan alterado para utilizar o cálculo da razão ao invés da diferença.

TABELA 6.1 - RESULTADOS COMPUTACIONAIS OBTIDOS PELO DeCan E UTILIZADOS PELA CCCV - UFPR NO VESTIBULAR 2001

ITENS AVALIADOS	DADOS NUMERICOS
Total de candidatos designados	19.710
Total de locais de provas utilizados	26
Tempo de processamento do DeCan	00:09:29 (h:m:s)
Distância média percorrida pelos candidatos	3.549,12 m
Soma total das distâncias percorridas	69.953.098,10 m
Menor distância percorrida por um candidato	30,94 m
Distância do local mais próximo para o candidato do item anterior	30,94 m
Distância do local mais distante para o candidato do item anterior	15.668,67 m
Maior distância percorrida por um candidato	19.917,74 m
Distância do local mais próximo para o candidato do item anterior	18.393,04 m
Distância do local mais distante para o candidato do item anterior	27.275,02 m
Total de candidatos designados para o primeiro local mais próximo	10.499 (53%)
Total de candidatos designados para o segundo local mais próximo	3.452 (18%)
Total de candidatos designados para o terceiro local mais próximo	1.154 (6%)
Total de candidatos designados para o (4º, etc.) locais mais próximos	4.605 (23%)
SOMA TOTAL	19.710 (100%)

TABELA 6.2 - RESULTADOS COMPUTACIONAIS OBTIDOS PELO DeCan ALTERADO PARA O CÁLCULO DA RAZÃO

ITENS AVALIADOS	DADOS NUMERICOS
Total de candidatos designados	19.710
Total de locais de provas utilizados	26
Tempo de processamento do DeCan	00:09:29 (h:m:s)
Distância média percorrida pelos candidatos	3.617,00 m
Soma total das distâncias percorridas	71.291.068,33 m
Menor distância percorrida por um candidato	24,82 m
Distância do local mais próximo para o candidato do item anterior	24,82 m
Distância do local mais distante para o candidato do item anterior	12.330,68 m
Maior distância percorrida por um candidato	20.698,52 m
Distância do local mais próximo para o candidato do item anterior	19.173,82 m
Distância do local mais distante para o candidato do item anterior	28.055,80 m
Total de candidatos designados para o primeiro local mais próximo	10.089 (51%)
Total de candidatos designados para o segundo local mais próximo	3.103 (16%)
Total de candidatos designados para o terceiro local mais próximo	1.468 (7%)
Total de candidatos designados para o (4º, etc.) locais mais próximos	5.050 (26%)
SOMA TOTAL	19.710 (100%)

Vale ressaltar que, como os candidatos designados pela CCCV foram designados por primeiro, alguns locais de provas (respectivamente os locais: 3, 8, 9 e 17, ver tabela 2.2 seção 2) ficaram com poucas vagas disponíveis. O fato de haver locais com um número muito reduzido de vagas disponíveis prejudica as estatísticas do resultado final obtido pelo DeCan. Para exemplificar suponha que as residências de 500 candidatos estejam mais próximas do local 8 (UFPR – Edifício Dom Pedro I – Reitoria) que possui apenas 2 vagas disponíveis. Destes 500 candidatos certamente 498 não serão designados para o primeiro local de provas mais próximo, o que diminuirá a porcentagem de candidatos designados para o primeiro local mais próximo e aumentará a porcentagem de candidatos designados para os outros locais. Para evidenciar a real eficiência do DeCan e afirmar o que foi dito, os 4 locais com menor capacidade foram desconsiderados e uma simulação da designação dos 19.710 candidatos ao vestibular 2001 foi refeita considerando-se apenas os 22 locais de provas restantes. Os resultados obtidos estão na tabela 6.3 e podem ser comparados com os resultados da tabela 6.2.

TABELA 6.3 - RESULTADOS COMPUTACIONAIS OBTIDOS PELO DeCan EXCLUINDO-SE 4 LOCAIS COM POUCAS VAGAS

ITENS AVALIADOS	DADOS NUMÉRICOS
Total de candidatos designados	19.710
Total de locais de provas utilizados	22
Tempo de processamento do DeCan	00:09:10 (h:m:s)
Distância média percorrida pelos candidatos	3.550,18 m
Soma total das distâncias percorridas	69.973.959,16 m
Menor distância percorrida por um candidato	30,94 m
Distância do local mais próximo para o candidato do item anterior	30,94 m
Distância do local mais distante para o candidato do item anterior	15.668,67 m
Maior distância percorrida por um candidato	19.917,74 m
Distância do local mais próximo para o candidato do item anterior	18.393,04 m
Distância do local mais distante para o candidato do item anterior	27.275,02 m
Total de candidatos designados para o primeiro local mais próximo	10.828 (55%)
Total de candidatos designados para o segundo local mais próximo	4.003 (20%)
Total de candidatos designados para o terceiro local mais próximo	1.495 (8%)
Total de candidatos designados para o (4º, etc.) locais mais próximos	3.384 (17%)
SOMA TOTAL	19.710 (100%)

Evidentemente, se no caso real os 26 locais de provas utilizados estivessem com todas as vagas originais disponíveis a designação dos 19.710 candidatos ao vestibular 2001 designados pelo DeCan teria sido bem mais satisfatória. Os resultados de uma simulação do DeCan aplicado aos 19.710 candidatos considerando-se os 26 locais com todas as vagas disponíveis são apresentados a seguir. Primeiramente, a tabela 6.4 apresenta a relação dos 26 locais utilizados com suas respectivas capacidades totais (antes das designações feitas pela CCCV).

TABELA 6.4- OS 26 LOCAIS UTILIZADOS PARA A DESIGNAÇÃO DOS 19.710 CANDIDATOS ATRAVÉS DO DeCan COM SUAS RESPECTIVAS CAPACIDADES TOTAIS

LOCAL	NOME DA INSTITUIÇÃO	CAPACIDADE TOTAL
1	UFPR - Centro Politécnico	3829
2	PUC - Pontifícia Univ. Católica	7837
3	Col. Estadual Rio Branco	935
4	Col. Estadual do Paraná	1865
5	Col. Nossa Senhora Medianeira	1240
6	UFPR - Setor de Ciências Agrárias	1300
7	Col. Estadual Leôncio Correia	1265
8	UFPR - Edifício Dom Pedro I - Reitoria	1155
9	UFPR - Edifício Dom Pedro II - Reitoria	1195
10	Fund. Estudos Sociais PR - Fesp	885
11	Col. Est. Prof. Loureiro Fernandes	1020
12	Esc. Municipal Prof. Omar Sabbag	1230
13	Col. Est. Prof.a. M. Aguiar Teixeira	630
14	UFPR - Depto. de Educação Física	485
15	Col. Padre João Bagozzi	2340
16	Col. Estadual Pedro Macedo	1305
17	Instituto de Educação do Paraná	1035
18	Col. Estadual Cecília Meireles	1000
19	Escola Social Madre Clélia	1025
20	Col. Estadual Paula Gomes	880
21	Colégio Paranaense	1114
22	UFPR - Setor C. Saúde - sede Botânico	770
23	Colégio Militar de Curitiba	1348
24	Esc. Est. República O. do Uruguai	725
25	Colégio Estadual Paulo Leminski	1259
26	UFPR - C. Ciências Florestais e Madeira	590
SOMA DAS CAPACIDADES		38262

A tabela 6.5 apresenta os resultados obtidos pelo DeCan aplicado a situação em que foram considerados novamente os mesmos 19.710 candidatos ao vestibular 2001 e os 26 locais com as capacidades totais.

TABELA 6.5 - RESULTADOS COMPUTACIONAIS OBTIDOS PELO DeCan PARA 26 LOCAIS COM AS CAPACIDADES TOTAIS

ITENS AVALIADOS	DADOS NUMERICOS
Total de candidatos designados	19.710
Total de locais de provas utilizados	26
Tempo de processamento do DeCan	00:08:7 (h:m:s)
Distância média percorrida pelos candidatos	2.993,91 m
Soma total das distâncias percorridas	59.009.874,76 m
Menor distância percorrida por um candidato	24,43 m
Distância do local mais próximo para o candidato do item anterior	24,43 m
Distância do local mais distante para o candidato do item anterior	10.389,40 m
Maior distância percorrida por um candidato	19.173,82 m
Distância do local mais próximo para o candidato do item anterior	19.173,82 m
Distância do local mais distante para o candidato do item anterior	28.055,80 m
Total de candidatos designados para o primeiro local mais próximo	14.541 (74%)
Total de candidatos designados para o segundo local mais próximo	4.157 (21%)
Total de candidatos designados para o terceiro local mais próximo	883 (4%)
Total de candidatos designados para o (4º, etc.) locais mais próximos	129 (1%)
SOMA TOTAL	19.710 (100%)

A seguir são realizadas novas simulações computacionais da designação dos 19.710 candidatos ao vestibular 2001 para 26 locais de provas. Porém, os 26 locais utilizados são selecionados dentre os 43 locais de provas disponíveis em Curitiba para a realização das provas do vestibular da UFPR, e para isto são utilizadas as heurísticas AGpMcap e BTpMcap.

6.2 TESTES COMPUTACIONAIS DAS HEURÍSTICAS AGpMcap E BTpMcap

Para testar a eficiência das heurísticas AGpMcap e BTpMcap foram realizadas simulações computacionais utilizando-se os dados do vestibular 2001 da UFPR. O objetivo destas simulações foi selecionar 26 locais de provas entre 43 possíveis ($C_{43}^{26} = 421.171.648.758$ combinações possíveis) e efetuar a designação dos

19.710 candidatos ao vestibular 2001 da UFPR. A tabela 6.6 mostra a relação dos 43 locais de provas que constam na relação da UFPR como locais disponíveis para a realização das provas em Curitiba e que são utilizados nas simulações computacionais seguintes. Evidentemente, os 26 locais que foram fixados como locais de provas para o vestibular 2001 pelos membros da CCCV (conforme discutido na seção 2) estão entre os 43 locais utilizados nestas simulações. Todos os 43 locais foram considerados com suas capacidades totais. Logo, os resultados desta seção não podem ser diretamente comparados com os resultados das tabelas 6.1, 6.2 e 6.3, mas podem ser comparados com os resultados da tabela 6.5. As simulações foram realizadas em um computador Pentium III 550 MHz com 128 megabytes de RAM.

TABELA 6.6 - RELAÇÃO DOS 43 LOCAIS DE PROVAS UTILIZADOS NAS SIMULAÇÕES (continua)

LOCAL	NOME DA INSTITUIÇÃO	CAPACIDADE
1	UFPR - Centro Politécnico	3829
2	PUC - Pontifícia Univ. Católica	7837
3	Col. Est. Hildebrando de Araújo	720
4	Col. Estadual Rio Branco	935
5	Col. da Polícia Militar	595
6	Col. Estadual do Paraná	1865
7	Col. Nossa Senhora Medianeira	1240
8	UFPR - Setor de Ciências Agrárias	1300
9	Col. Estadual Leôncio Correia	1265
10	Esc. Estadual Prof. Brandão	560
11	Esc. Est. Tiradentes	550
12	Col. Marista Santa Maria	1800
13	UFPR – Edifício Dom Pedro I - Reitoria	1155
14	UFPR – Edifício Dom Pedro II - Reitoria	1195
15	Fund. Estudos Sociais PR - Fesp	885
16	Col. Est. Prof. Loureiro Fernandes	1020
17	Esc. Municipal Pref. Omar Sabbag	1230
18	Col. Est. Prof.a. M. Aguiar Teixeira	630
19	UFPR - Depto. de Educação Física	485
20	Col. Est. Pres. Lamenha Lins	500
21	Col. Padre João Bagozzi	2340
22	Col. Estadual Pedro Macedo	1305
23	Instituto de Educação do Paraná	1035
24	Col. Est. Prieto Martinez	480
25	Col. Est. Júlia Wanderley	670

TABELA 6.6 - RELAÇÃO DOS 43 LOCAIS DE PROVAS UTILIZADOS NAS SIMULAÇÕES (conclusão)

LOCAL	NOME DA INSTITUIÇÃO	CAPACIDADE
26	Col. Estadual Cecília Meireles	1000
27	Col. Est. Guaíra	650
28	Esc. Est. Isolda Schmid	450
29	Col. Sagrado Coração de Jesus	800
30	Col. Est. Prof. Elias Abrahão	680
31	Col. Est. Segismundo Falarz	690
32	Col. Est. Prof. Victor do Amaral	690
33	Escola Social Madre Clélia	1025
34	Col. Estadual Paula Gomes	880
35	Instituto Politécnico Estadual	650
36	Esc. Est. Amâncio Moro	660
37	Colégio Paranaense	1114
38	UFPR - Setor C. Saúde - Sede Botânico	770
39	Colégio Militar de Curitiba	1348
40	Esc. Est. República O. do Uruguai	725
41	Colégio Estadual Paulo Leminski	1259
42	União de Curitiba – Campus João Negrão	1430
43	UFPR - C. Ciências Florestais e Madeira	590
SOMA DAS CAPACIDADES DE 1 ATE 43		50837

A seguir são apresentados os resultados obtidos nos testes computacionais realizados com o AGpMcap.

6.2.1 Resultados Computacionais Obtidos pelo AGpMcap

O AGpMcap foi submetido a duas simulações computacionais para selecionar 26 locais de provas dentre os 43 apresentados na tabela 6.6. Na primeira simulação realizada foi utilizado o AGpMcap como descrito na seção 5 (completo). Já na segunda simulação realizada o operador denominado “Hipermutação direcionada” proposto neste trabalho, foi “desligado”. Ou seja, na segunda simulação realizada este operador não foi executado nenhuma vez durante todo o processamento do AGpMcap. O objetivo foi verificar a influência deste operador no AGpMcap. A seguir, são apresentados os parâmetros atribuídos ao AGpMcap para a realização da primeira simulação e em seguida a tabela 6.7 apresenta os resultados obtidos na primeira simulação.

Tamanho da população = 100.

Número máximo de iterações = 1.000.

Probabilidade de ocorrer mutação = 0,01 (1%).

Probabilidade de ocorrer a Hipermutação direcionada = 0,005 (0,5%).

Número de indivíduos que sofrerão hipermutação = 10 (10% da população).

TABELA 6.7 - RESULTADOS COMPUTACIONAIS OBTIDOS PELO AGpMcap NA PRIMEIRA SIMULAÇÃO (COM HIPERMUTAÇÃO DIREC.)

ITENS AVALIADOS	DADOS NUMÉRICOS
DADOS GERAIS	
Total de candidatos designados	19.710
Total de locais de provas utilizados	43
Total de locais de provas selecionados	26
Tempo de processamento do AGpMcap	01:43:34 (h:m:s)
Número de ocorrências da Hipermutação direcionada	05
RESULTADOS OBTIDOS	
Distância média percorrida pelos candidatos	2.333,50 m
Soma total das distâncias percorridas	45.993.335,75 m
Menor distância percorrida por um candidato	13,33 m
Distância do local mais próximo para o candidato do item anterior	13,33 m
Distância do local mais distante para o candidato do item anterior	10.624,54 m
Maior distância percorrida por um candidato	16.400,27 m
Distância do local mais próximo para o candidato do item anterior	16.400,27 m
Distância do local mais distante para o candidato do item anterior	32.430,81 m
Total de candidatos designados para o primeiro local mais próximo	16.264 (83%)
Total de candidatos designados para o segundo local mais próximo	2.057 (9%)
Total de candidatos designados para o terceiro local mais próximo	496 (3%)
Total de candidatos designados para o (4º, etc.) locais mais próximos	893 (5%)
SOMA TOTAL	19.710 (100%)

A tabela 6.8 apresenta a relação dos 26 locais de provas selecionados pelo AGpMcap nesta primeira simulação.

TABELA 6.8 - RELAÇÃO DOS 26 LOCAIS DE PROVAS SELECIONADOS PELO AGpMcap NA PRIMEIRA SIMULAÇÃO REALIZADA

LOCAL	NOME DA INSTITUIÇÃO	CAPACIDADE
1	UFPR - Centro Politécnico	3829
2	Col. Estadual Rio Branco	935
3	Col. da Polícia Militar	595
4	Col. Nossa Senhora Medianeira	1240
5	Col. Estadual Leôncio Correia	1265
6	Esc. Estadual Prof. Brandão	560
7	Col. Marista Santa Maria	1800
8	UFPR- Edifício Dom Pedro II - Reitoria	1195
9	Col. Est. Prof. Loureiro Fernandes	1020
10	Esc. Municipal Prof. Omar Sabbag	1230
11	Col. Padre João Bagozzi	2340
12	Col. Estadual Pedro Macedo	1305
13	Instituto de Educação do Paraná	1035
14	Col. Est. Prieto Martinez	480
15	Col. Est. Júlia Wanderley	670
16	Col. Estadual Cecília Meireles	1000
17	Col. Est. Guaíra	650
18	Esc. Est. Isolda Schmid	450
19	Col. Est. Segismundo Falarz	690
20	Col. Est. Prof. Victor do Amaral	690
21	Escola Social Madre Clélia	1025
22	Col. Estadual Paula Gomes	880
23	Esc. Est. Amâncio Moro	660
24	Colégio Paranaense	1114
25	União de - Campus João Negrão	1430
26	UFPR - C. Ciências Florestais e Madeira	590
SOMA DAS CAPACIDADES		28678

A seguir são apresentados os parâmetros atribuídos ao AGpMcap para a realização da segunda simulação e a tabela 6.9 apresentando os resultados obtidos na segunda simulação realizada com o AGpMcap.

Tamanho da população = 100.

Número máximo de iterações = 12.100.

Probabilidade de ocorrer mutação = 0,01 (1%).

Probabilidade de ocorrer a Hipermutação direcionada = 0 (0%).

Número de indivíduos que sofrerão Hipermutação = 0 (0%).

TABELA 6.9 - RESULTADOS COMPUTACIONAIS OBTIDOS PELO AGpMcap NA SEGUNDA SIMULAÇÃO (SEM HIPERMUTAÇÃO DIREC.)

ITENS AVALIADOS	DADOS NUMÉRICOS
DADOS GERAIS	
Total de candidatos designados	19.710
Total de locais de provas utilizados	43
Total de locais de provas selecionados	26
Tempo de processamento do AGpMcap	01:43:21 (h:m:s)
Número de ocorrências da Hipermutação direcionada	00 (zero)
RESULTADOS OBTIDOS	
Distância média percorrida pelos candidatos	2.400,43 m
Soma total das distâncias percorridas	47.312.540,92 m
Menor distância percorrida por um candidato	57,49 m
Distância do local mais próximo para o candidato do item anterior	57,49 m
Distância do local mais distante para o candidato do item anterior	12.598,50 m
Maior distância percorrida por um candidato	16.400,27 m
Distância do local mais próximo para o candidato do item anterior	16.400,27 m
Distância do local mais distante para o candidato do item anterior	32.430,81 m
Total de candidatos designados para o primeiro local mais próximo	15.595 (79%)
Total de candidatos designados para o segundo local mais próximo	2.486 (13%)
Total de candidatos designados para o terceiro local mais próximo	655 (3%)
Total de candidatos designados para o (4º, etc.) locais mais próximos	974 (5%)
SOMA TOTAL	19.710 (100%)

A tabela 6.10 apresenta a relação dos 26 locais de provas selecionados pelo AGpMcap nesta segunda simulação.

TABELA 6.10 - RELAÇÃO DOS 26 LOCAIS DE PROVAS SELECIONADOS PELO AGpMcap NA SEGUNDA SIMULAÇÃO (continua)

LOCAL	NOME DA INSTITUIÇÃO	CAPACIDADE
1	UFPR - Centro Politécnico	3829
2	Col. Estadual do Paraná	1865
3	Col. Nossa Senhora Medianeira	1240
4	UFPR - Setor de Ciências Agrárias	1300
5	Col. Estadual Leôncio Correia	1265
6	Col. Marista Santa Maria	1800
7	Fund. Estudos Sociais PR - Fesp.	885
8	Esc. Municipal Pref. Omar Sabbag	1230
9	UFPR - Depto. de Educação Física	485
10	Col. Est. Pres. Lamenha Lins	500
11	Col. Padre João Bagozzi	2340
12	Col. Estadual Pedro Macedo	1305
13	Instituto de Educação do Paraná	1035

TABELA 6.10 - RELAÇÃO DOS 26 LOCAIS DE PROVAS SELECIONADOS
PELO AGpMcap NA SEGUNDA SIMULAÇÃO (conclusão)

LOCAL	NOME DA INSTITUIÇÃO	CAPACIDADE
14	Col. Est. Prieto Martinez	480
15	Col. Est. Júlia Wanderley	670
16	Col. Estadual Cecília Meireles	1000
17	Col. Est. Guaíra	650
18	Esc. Est. Isolda Schmid	450
19	Col. Sagrado Coração de Jesus	800
20	Col. Est. Prof. Elias Abrahão	680
21	Col. Est. Prof. Victor do Amaral	690
22	Escola Social Madre Clélia	1025
23	Col. Estadual Paula Gomes	880
24	Instituto Politécnico Estadual	650
25	Esc. Est. Amâncio Moro	660
26	Colégio Paranaense	1114
SOMA DAS CAPACIDADES		28828

A seguir, são apresentados os resultados computacionais obtidos nas simulações realizadas com a BTpMcap.

6.2.2 Resultados Computacionais Obtidos pela BTpMcap

A BTpMcap foi submetida a uma simulação computacional em que selecionou 26 locais de provas entre os mesmos 43 locais utilizados nas simulações do AGpMcap. Primeiramente são apresentados os parâmetros atribuídos à BTpMcap para a realização da simulação e em seguida a tabela 6.11 apresenta os resultados obtidos na simulação.

Número máximo de iterações = 150.

Número de iterações tabu = 10.

Número máximo de iterações suportadas sem melhoria na solução = 30 (20%).

TABELA 6.11 - RESULTADOS COMPUTACIONAIS OBTIDOS PELA BTpMcap NA SIMULAÇÃO REALIZADA

ITENS AVALIADOS	DADOS NUMERICOS
DADOS GERAIS	
Total de candidatos designados	19.710
Total de locais de provas utilizados	43
Total de locais de provas selecionados	26
Tempo de processamento da BTpMcap	01:23:37 (h:m:s)
Total de iterações completadas	150
RESULTADOS OBTIDOS	
Distância média percorrida pelos candidatos	2367,34 m
Soma total das distâncias percorridas	46.660.204,66 m
Menor distância percorrida por um candidato	139,75 m
Distância do local mais próximo para o candidato do item anterior	139,75 m
Distância do local mais distante para o candidato do item anterior	16.458,98 m
Maior distância percorrida por um candidato	16.400,27 m
Distância do local mais próximo para o candidato do item anterior	16.400,27 m
Distância do local mais distante para o candidato do item anterior	32.430,81 m
Total de candidatos designados para o primeiro local mais próximo	16.064 (82%)
Total de candidatos designados para o segundo local mais próximo	1.652 (8%)
Total de candidatos designados para o terceiro local mais próximo	802 (4%)
Total de candidatos designados para o (4º, etc.) locais mais próximos	1.192 (6%)
SOMA TOTAL	19.710 (100%)

A tabela 6.12 apresenta a relação dos 26 locais de provas selecionados pela BTpMcap na simulação computacional realizada.

TABELA 6.12 - RELAÇÃO DOS 26 LOCAIS DE PROVAS SELECIONADOS PELA BTpMcap NA SIMULAÇÃO REALIZADA (continua)

LOCAL	NOME DA INSTITUIÇÃO	CAPACIDADE
1	Col. Est. Hildebrando de Araújo	720
2	Col. Estadual Rio Branco	935
3	Col. Nossa Senhora Medianeira	1240
4	Col. Estadual Leôncio Correia	1265
5	Esc. Estadual Prof. Brandão	560
6	Col. Marista Santa Maria	1800
7	UFPR - Edifício Dom Pedro II - Reitoria	1195
8	Col. Est. Prof. Loureiro Fernandes	1020
9	Esc. Municipal Prof. Omar Sabbag	1230
10	Col. Est. Pres. Lamenha Lins	500
11	Col. Padre João Bagozzi	2340
12	Col. Estadual Pedro Macedo	1305
13	Instituto de Educação do Paraná	1035
14	Col. Est. Prieto Martinez	480

TABELA 6.12 - RELAÇÃO DOS 26 LOCAIS DE PROVAS SELECIONADOS
PELA BTpMcap NA SIMULAÇÃO REALIZADA (conclusão)

LOCAL	NOME DA INSTITUIÇÃO	CAPACIDADE
15	Col. Est. Júlia Wanderley	670
16	Col. Estadual Cecília Meireles	1000
17	Esc. Est. Isolda Schmid	450
18	Col. Sagrado Coração de Jesus	800
19	Col. Est. Prof. Elias Abrahão	680
20	Col. Est. Segismundo Falarz	690
21	Col. Est. Prof. Victor do Amaral	690
22	Escola Social Madre Clélia	1025
23	Col. Estadual Paula Gomes	880
24	Esc. Est. Amâncio Moro	660
25	Esc. Est. República O. do Uruguai	725
26	Colégio Estadual Paulo Leminski	1259
SOMA DAS CAPACIDADES		25154

A seguir, é feita uma breve análise dos resultados obtidos pelas heurísticas AGpMcap e BTpMcap nas simulações computacionais descritas anteriormente. São apresentadas também as conclusões do presente trabalho e algumas sugestões para trabalhos futuros.

7 ANÁLISE DOS RESULTADOS E CONCLUSÕES

Esta seção tem por objetivo fornecer a análise dos resultados apresentados nas seções anteriores destacando o desempenho das técnicas apresentadas no presente trabalho. Sugestões para aprimorar a resolução do problema das p-medianas e do problema real descrito também são fornecidas.

7.1 ANÁLISE DOS RESULTADOS OBTIDOS PELO AGpMcap

O objetivo das simulações computacionais realizadas com o AGpMcap foi demonstrar o poder da heurística denominada algoritmos genéticos. Desta forma, não se teve a preocupação de realizar dezenas de simulações nem de submeter o algoritmo a um longo tempo de processamento para forçar o surgimento de boas soluções. Considerando-se $P = 100$, ao gerar a população inicial o AGpMcap analisa um total de 100 soluções. Em 1.000 iterações são gerados aproximadamente 2.000 novos indivíduos. A cada vez que o operador de hipermutação direcionada foi acionado, considerando-se os parâmetros utilizados na primeira simulação realizada, foram analisadas 4.420 soluções. Como durante todo o processamento da primeira simulação o operador de hipermutação direcionada foi acionado 5 vezes, foram analisadas mais 22.100 soluções. Portanto, na primeira simulação realizada durante todo o processamento o AGpMcap analisou um total de aproximadamente 24.200 soluções (não necessariamente distintas) entre as 421.171.648.758 soluções distintas possíveis para o problema considerado. Como o problema real considerado só é resolvido uma vez por ano, não haveria problema algum se o AGpMcap fosse executado durante algumas horas ou até mesmo dias para resolvê-lo. Na solução encontrada pelo AGpMcap na primeira simulação, 83% dos candidatos foram designados para o primeiro local mais próximo de suas residências. Evidentemente, dificilmente este número seria igual a 100% a não ser que as capacidades dos locais de provas fossem ilimitadas. Ainda, 92% dos candidatos foram designados para o primeiro ou segundo locais mais próximos de suas residências.

Na segunda simulação realizada, o objetivo foi verificar o efeito do operador de hipermutação direcionada no AGpMcap. Como durante o processamento desta segunda simulação este operador não foi acionado nenhuma vez, o número total de iterações teve que ser aumentado para que o algoritmo analisasse o mesmo número de soluções analisadas na primeira simulação. O tempo total de processamento foi praticamente igual ao tempo de processamento da primeira simulação realizada. A solução obtida foi inferior à obtida na primeira simulação. Considerando-se os resultados obtidos pela segunda simulação, 79% dos candidatos seriam designados para o primeiro local de provas mais próximo de suas residências e 92% para o primeiro ou segundo locais mais próximos. O fato da obtenção do bom resultado na segunda simulação deve-se exclusivamente ao poder dos algoritmos genéticos, já que, em última análise, sem o operador de hipermutação direcionada o AGpMcap opera basicamente como um AG convencional. Com base nos resultados obtidos nas duas simulações computacionais a utilização do operador de hipermutação direcionada proposto neste trabalho, é bastante vantajosa. Este operador tem a capacidade de analisar e melhorar um grande número de soluções durante a execução do AGpMcap, o que tende a direcionar a população de soluções-candidatas para níveis cada vez mais elevados de adaptação ao problema. Um fato importante é que a média das distâncias percorridas pelos candidatos até os locais de provas foi menor na primeira simulação computacional realizada com o AGpMcap, o que indica a vantagem da utilização do operador de hipermutação direcionada.

7.2 ANÁLISE DOS RESULTADOS OBTIDOS PELA BTpMcap

O objetivo da simulação computacional realizada com a BTpMcap foi demonstrar o poder da heurística busca tabu. Desta forma, não se teve a preocupação de realizar dezenas de simulações nem de submeter o algoritmo a um longo tempo de processamento para forçar o surgimento de boas soluções. Considerando-se os parâmetros utilizados e os dados da simulação realizada, a BTpMcap analisou 1 solução inicial, 26 ou 27 soluções a cada movimento DROP realizado, 17 ou 18 soluções a cada movimento ADD e 442 soluções a cada movimento SWAP num total de aproximadamente 24.301 soluções (não necessariamente distintas) durante o processamento das 150 iterações

completadas. Em relação ao tempo de processamento a BTpMcap demorou aproximadamente 1h20min para finalizar o processamento. Na solução encontrada pela BTpMcap 82% dos candidatos foram designados para o primeiro local mais próximo de suas residências e 90% para o primeiro ou segundo locais mais próximos.

7.3 COMPARANDO AGpMcap E BTpMcap

Em relação a qualidade da solução nas simulações computacionais realizadas o AGpMcap com o operador de hipermutação direcionada obteve uma solução melhor que a BTpMcap. Durante todo o processamento o AGpMcap analisou um total de 24.200 soluções contra 24.300 soluções analisadas pela BTpMcap. Com relação ao tempo de processamento a BTpMcap foi aproximadamente vinte minutos mais rápida que o AGpMcap e analisou 100 soluções a mais. Os resultados das simulações computacionais realizadas no presente trabalho são apresentados na tabela 7.1.

TABELA 7.1 - RESULTADOS OBTIDOS NAS SIMULAÇÕES COMPUTACIONAIS REALIZADAS COM O AGpMcap E A BTpMcap

DADOS ANALISADOS	AGpMcap com hipermutação direcionada	AGpMcap sem hipermutação direcionada	BTpMcap
Total de candidatos designados	19.710	19.710	19.710
Total de locais de provas utilizados	43	43	43
Total de locais de provas selecionados	26	26	26
Tempo de processamento	01:43:34 (h:m:s)	01:43:21 (h:m:s)	01:23:37 (h:m:s)
Número de soluções analisadas	24.200	24.200	24.300
Distância média percorrida pelos candidatos	2.333,50 m	2.400,43 m	2367,34 m
Soma total das distâncias percorridas	45.993.335,75 m	47.312.540,92 m	46.660.204,66 m
Menor distância percorrida por um candidato	13,33 m	57,49 m	139,75 m
Distância do local mais próximo para o candidato do item anterior	13,33 m	57,49 m	139,75 m
Distância do local mais distante para o candidato do item anterior	10.624,54 m	12.598,50 m	16458,98 m
Maior distância percorrida por um candidato	16.400,27 m	16.400,27 m	16.400,27 m
Distância do local mais próximo para o candidato do item anterior	16.400,27 m	16.400,27 m	16.400,27 m
Distância do local mais distante para o candidato do item anterior	32.430,81 m	32.430,81 m	32.430,81 m
Total de candidatos designados para o primeiro local mais próximo	16.264 (83%)	15.595 (79%)	16.064 (82%)
Total de candidatos designados para o segundo local mais próximo	2.057 (9%)	2.486 (13%)	1.652 (8%)
Total de candidatos designados para o terceiro local mais próximo	496 (3%)	655 (3%)	802 (4%)
Total de candidatos designados para o (4º, etc.) locais mais próximos	893 (5%)	974 (5%)	1.192 (6%)

7.4 CONCLUSÕES

No presente trabalho, foram aplicados os princípios básicos das heurísticas busca tabu e algoritmos genéticos para desenvolver duas heurísticas capazes de resolver o problema de localização de instalações denominado problema das p-medianas capacitado. O objetivo principal foi explorar as características de cada uma das heurísticas utilizadas e demonstrar que ambas podem gerar boas soluções quando bem aplicadas ao problema das p-medianas capacitado. Foi também desenvolvido um algoritmo capaz de otimizar a designação de candidatos ao vestibular aos locais de provas mais próximos de suas residências. Foram realizados testes computacionais para verificar a eficiência das heurísticas desenvolvidas. Nos testes computacionais realizados, com relação à qualidade da solução encontrada a heurística AGpMcap, aperfeiçoada com o uso do operador de hipermutação direcionada proposto neste trabalho, obteve maior êxito que a heurística BTpMcap, pois, conseguiu encontrar uma solução de melhor qualidade. Com base nos resultados obtidos nos testes computacionais realizados, as duas heurísticas desenvolvidas demonstraram que é possível otimizar de forma simples e bem sucedida a seleção dos locais de provas onde os candidatos ao vestibular deverão realizar provas. Esta otimização pode propiciar maior comodidade para os candidatos que terão oportunidade de realizar as provas em locais próximos às suas residências.

7.5 SUGESTÕES PARA TRABALHOS FUTUROS

Com o propósito de aprimorar os resultados e métodos apresentados neste trabalho, seguem algumas sugestões que poderão servir para trabalhos futuros.

- a) com referência ao problema das p-medianas, uma opção seria a de pesquisar novas heurísticas que se adaptem ao problema e aplicá-las para resolvê-lo. Uma heurística recente que pertence aos algoritmos evolucionários chamada *Scatter*

Search e sua forma generalizada chamada *Path Relinking* poderia ser uma das heurísticas pesquisada. Segundo GLOVER (1999) para alguns problemas combinatórios esta heurística tem demonstrado resultados superiores aos dos algoritmos genéticos e busca tabu;

- b) os resultados obtidos pelo algoritmo da designação dos candidatos ao vestibular aos locais de provas desenvolvido no presente trabalho (DeCan) foram bastante satisfatórios mas, por uma simples falta de tempo, não foi possível aprimorar mais este algoritmo. No entanto, a intuição indica que o potencial do mesmo ainda não foi totalmente explorado. Uma sugestão seria a de aperfeiçoar este algoritmo que poderia servir não só para a designação de candidatos aos locais de provas mas que fosse capaz de otimizar, por exemplo, a distribuição de produtos industriais a pontos de demanda considerando-se múltiplos depósitos;
- c) com relação ao problema real resolvido no presente trabalho, seleção de locais de provas e designação dos candidatos ao vestibular para os locais de provas, o problema poderia ser analisado de forma mais completa. Poderia ser prevista, por exemplo, a otimização do percurso seguido pelos veículos que realizam a entrega das provas nos locais selecionados. Outro fator que poderia ser considerado é o de encontrar soluções de boa qualidade para os candidatos mas com o menor número possível de locais de provas utilizados.

REFERÊNCIAS

- ANTUNES, Antonio; PEETERS, Dominique. A dynamic optimization model for school network. **Socio-Economic Planning Sciences**, Amsterdam, NL: Elsevier Science Publishers, n. 34, p. 101-120, 2000.
- BODIN, Lawrence; GOLDEN, Bruce; ASSAD, Arjang et al. Routing and scheduling of vehicles and crews: the state of art. Pergamon Press, England, 1983, Special issue.
- BOOKER, L. B. Improving search in genetic algorithms. In: Davis, L (Ed.), Genetic algorithms and simulated annealing. Morgan Kauffmann, Los Altos, p. 61-73, 1987.
- BOWERMAN, Robert L.; CALAMAI, Paul H.; HALL, G. Brent. The demand partitioning method for reducing aggregation errors in p-median problems. **Computers & Operations Research**, London, v. 26, p. 1097-1111, 1999.
- CANÓS, M. J.; IVORRA, C.; LIERN, V. An exact algorithm for the fuzzy p-median problem. **European Journal of Operational Research**, Amsterdam, Elsevier Science Publishers, n. 116, p. 80-86, 1999.
- DENSHAM, P. J.; RUSHTON, G. A more efficient heuristic for solving large p-median problems. Papers in Regional Science: **The Journal of the RSAI**, London, v. 71, n. 3, p. 307-329, 1992.
- DIBBLE, C.; DENSHAM, P. J. Generating interesting alternatives in GIS and SDSS using genetic algorithms. GIS/LIS symposium, University of Nebraska, Lincoln, 1993.
- DODIN, B.; ELIMAM, A. A.; ROLLAND, Erik. Tabu search in audit scheduling. **European Journal of Operational Research**, Amsterdam, Elsevier Science Publishers, n. 106, p. 373-392, 1998.
- DREZNER, Zvi. Facility location: a survey of applications and Methods. New York: Springer-Verlag, 1995.
- ERKUT, Erhan; BOZKAYA, Burçin. Analysis of aggregation errors for the p-median problems. **Computers & Operations Research**, London, v. 26, p. 1075-1096, 1999.
- ERKUT, Erhan; BOZKAYA, Burçin; ZHANG, Jianjun. An effective genetic algorithm for the p-median problem. Em fase de publicação.

FARLEY, A. Comparison of choosing the crossover point in the genetic crossover operation. Liverpool, Dept. of Computer Science, University of Liverpool, 1991.

GLOVER, Fred, LAGUNA, Manuel. Tabu search. Kluwer, Boulder, Academic Publishers, University of Colorado, 1997.

GLOVER, Fred. Future paths for integer programming and links to artificial intelligence. **Computers & Operations Research**, London, v. 13, p. 533-549, 1986.

GLOVER, Fred. Genetic algorithms and scatter search: unsuspected potentials. **Statistics and Computing**, London, Chapman & Hall, v. 4, p. 131-140, 1994.

GLOVER, Fred. Scatter search and path relinking. Graduate School of Business, Boulder, University of Colorado, 1999. Material Interno.

GLOVER, Fred. Tabu search for the p-median problem. Artigo não publicado.

GLOVER, Fred. Tabu search. In: REEVES, Colin R. Modern heuristic techniques for combinatorial problems. London: McGraw-Hill, 1995.

GOLDBERG, David E. Genetic algorithms in search, optimization and machine learning. Menlo Park, Addison-Wesley, CA: 1986a.

GOLDBERG, David E. Genetic and Evolutionary Algorithms: Come of Age. *Communications of the ACM*, v. 37, n. 3, p. 113-119, 1994.

GOLDBERG, David E. The genetic algorithm approach: why, how, and what next? In: NARENDRA, K. S. (Ed.), *Adaptive and learning systems. Theory and application*. Plenum Press, p. 247-253, 1986b.

GOODCHILD, M. F.; NORONHA, V. Location-allocation for small computers. University of Iowa, Iowa, 1983. (Monograph, 8)

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

HALL, Lawrence O.; ÖZYURT, Ibrahim B.; BEZDEK, James C. Clustering with a genetically optimized approach. **Transactions on Evolutionary Computation**, New York, v. 3, 2, 103-111, 1999.

HALVORSON, Michael. Microsoft visual basic 5 passo a passo. São Paulo: Makron Books, 1997.

HANSEN, P. The steepest ascent mildest descent heuristic for combinatorial programming. In: Congress on Numerical Methods in Combinatorial Optimization, Capri, Italy, 1986. p. 70-145.

HOLLAND, John H. Adaptation in natural and artificial systems. Ann Arbor: University of Michigan Press, 1975.

HOLLAND, John H. Genetic algorithms. **Scientific American**, New York, v. 267, n. 1, p. 66-72, 1992.

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

JOHNSON, Richard A.; WICHERN, Dean W. Applied multivariate statistical analysis. 2. Ed. Prentice-Hall, Englewood Cliffs, New Jersey, 1988.

KARIV, O. HAKIMI, S. L. The p-median problems. In: An Algorithmic Approach to Network Location Problems. **SIAM Journal on applied mathematics**, Philadelphia, v. 37, p. 539-560, 1979.

LAGUNA, Manuel. Tabu search for a single machine-scheduling problem. **Journal of Intelligent Manufacturing**, Norwell, Massachusetts, v. 2, p. 253-260, 1991.

MAYERLE, S. F. Um algoritmo genético para o problema do caixeiro viajante. Florianópolis: UFSC, 1996. Trabalho interno.

MORENO-PEREZ, J. A.; MORENO-VEGA, J. M.; MLADENOVIC, N. Tabu search and simulated annealing in p-median problems. Montreal: Palestra apresentada na Canadian Operational Research Society Conference, 1994.

NAGY, G.; SALHI, S. Nested heuristic methods for the location-routing problem. **Journal of the Operational Research Society**, Pergamon Press, London, v. 47, p. 1166-1174, 1996.

NARULA, S. C.; OGBU, U. I.; SAMUELSSON, H. M. An algorithm for the p-median problem. **Operational Research**, Operational Research Society, London, v. 25, p. 709-712.

NUNES, Luiz Fernando. Algoritmos genéticos aplicados na abordagem de um problema real de roteirização de veículos. Curitiba, 1998. Dissertação, (Mestrado em Métodos Numéricos em Engenharia) UFPR.

OHLEMÜLLER, M. Tabu search for large location-allocation problems. **Journal of Operational Research Society**, Amsterdam, v. 48, p. 745-750, 1997.

PIRKUL, Hasan; GUPTA, Rakesh; ROLLAND, Erik. VisOpt: a visual interactive optimization tool for p-median problems. **Decision Support Systems**, Amsterdam, v. 26, p. 209-223, 1999.

REEVES, Colin R. Genetic algorithms. In: REEVES, Colin R. **Modern heuristic techniques for combinatorial problems**. London: McGraw-Hill, 1995.

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

ROLLAND, Erik; SCHILLING, David A.; CURRENT, John R. An efficient tabu search procedure for the p-median problem. **European Journal of Operational Research**, Amsterdam, v. 96, p. 329-342, 1996.

RÖNNQVIST, Mikael; TRAGANTALERNGSAK, Suda; HOLT, John. A repeated matching heuristic for the single-source capacitated facility location problem. **European Journal of Operational Research**, Amsterdam, v. 116, p. 51-68, 1999.

SAMPAIO, Maria Eugênia de Carvalho e Silva. Aplicação de metaheurísticas ao problema de localização de escolas de Ensino Fundamental. Curitiba, 1999. Dissertação, (Mestrado em Métodos Numéricos em Engenharia) UFPR.

SYAM, Siddhartha S. A model for the capacitated p-facility location problem in global environments. **Computers & Operations Research**, London, v. 24, p. 1005-1016, 1997.

TAITZ, M. B.; BART, P. Heuristic concentration: Two-stage solution construction. **Operational Research**, Operational Research Society, London, v. 16, p. 955-961, 1968.

TANOMARU, Júlio. Motivação, fundamentação e aplicações de algoritmos genéticos. In: CONGRESSO BRASILEIRO DE REDES NEURAIIS, 2; ESCOLA DE REDES NEURAIIS, 3, 1995, Curitiba.

TANSEL, B. C.; FRANCIS, R. L.; LOWE, T. J. Location on networks: a survey, Part I: The p-center and p-median problems. **Management Science**, Baltimore, v. 29, p. 4, 1983.

TRAGANTALERNGSAK, Suda; HOLT, John; RÖNNQVIST, Mikael. An exact method for the two-echelon, single-source, capacited facility location problem. **European Journal of Operational Research**, Amsterdam, v. 123, p. 473-489, 1999.

ZIONTS, Stanley. Linear and integer programming. Englewood Cliffs, New Jersey: Prentice-Hall, 1974.