

MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

**HEURÍSTICA DE LOCALIZAÇÃO-ALOCAÇÃO PARA
PROBLEMAS DE LOCALIZAÇÃO DE FACILIDADES**

Reinaldo Gen Ichiro Arakaki

Tese de Doutorado em Computação Aplicada, orientada pelo Dr. Luiz Antonio Nogueira
Lorena, submetida para aprovação em 25 de março de 2002

INPE
São José dos Campos
2002

AGRADECIMENTOS

Ao meu orientador Dr. Luiz Antonio Nogueira Lorena, pela compreensão e paciência que teve comigo ao longo deste trabalho.

Aos meus colegas de curso que sempre se fizeram presentes, especialmente ao amigo Anibal Fernandes.

Ao Marcos Pereira pelas discussões feitas durante este trabalho que para mim foram bastante proveitosas.

Ao amigo José Marinaldo Gleriani pela ajuda na preparação das transparências.

Ao Instituto Nacional de Pesquisas Espaciais pelo apoio e ao Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pela bolsa concedida.

E, em especial, aos meus pais (Gensho - in memoriam e Yuko), pelo apoio que me deram em todos os momentos.

RESUMO

Neste trabalho foi desenvolvida uma nova heurística de localização-alocação (HLA) para problemas de localização de facilidades. Em tais problemas a questão central é localizar um objeto (ou objetos) que são chamados de facilidades e minimizar o custo de localizar esta facilidade. A HLA foi aplicada a dois problemas: o Problema de Localização de Máxima Cobertura (PLMC) e o Problema das P-Medianas Capacitado (PMC) com o intuito de uma possível integração a um Sistema de Informações Geográficas (SIG). A HLA baseia-se na formação de agrupamentos (clusters) e na possibilidade de melhorá-los (em relação a algum objetivo). Uma bateria de problemas testes foi escolhida para validar a HLA. Bons resultados foram encontrados tanto para problemas pequenos como para problemas grandes (para o PLMC) e para os problemas pequenos (para o PMC). Ainda foi feita uma aplicação da HLA como processo de mutação dentro do Algoritmo Genético Construtivo, para os mesmos problemas. Conclui-se que a HLA, sendo uma heurística de simples implementação é rápida e bastante eficiente portanto indicada para ser integrada aos SIG.

HEURISTIC LOCATION-ALLOCATION FOR FACILITIES LOCATION PROBLEMS

ABSTRACT

New location-allocation heuristic (LAH) algorithms applied in facility location problems are presented in this thesis. Such algorithms approach is based on clustering and its main objective is to find out a facility (object) in a space by minimizing a function. The LAH developed throughout this work was employed in two problems: the first problem is the Maximal Covering Location Problem (MCLP) and the second one is the Capacitated p-Median Problems (CPMP) with the purpose of a possible integration to Geographic Information Systems (GIS). A set of test problems was chosen to validate this LAH. Good results were obtained for small and large-scale problems in MCLP cases. Good results were also obtained for small-scale problems in CPMP cases. The LAH were also applied as a mutation process in Constructive Genetic Algorithms for the same problems. The good results demonstrate that LAH, being quick and fast, may be usefully applicable to GIS.

SUMÁRIO

	<u>Pág.</u>
LISTA DE FIGURAS	
LISTA DE TABELAS	
CAPÍTULO 1 – INTRODUÇÃO	17
CAPÍTULO 2 – O PROBLEMA DE LOCALIZAÇÃO DE MÁXIMA COBERTURA	21
2.1 – Descrição do Problema	22
2.2 – Formulação Matemática do <i>PLMC</i>	22
2.3 – Aplicações do <i>PLMC</i>	24
2.4 – Métodos de Solução do <i>PLMC</i>	24
2.4.1 – Métodos Heurísticos	24
2.4.2 – Métodos Exatos	25
CAPÍTULO 3 – O PROBLEMA DAS P-MEDIANAS CAPACITADO	27
3.1 – O Problema das P-Medias	27
3.1.1 – Formulação Matemática do <i>PM</i>	28
3.2 – O Problema das P-Medias Capacitado	29
3.2.1 – Formulação Matemática do <i>PMC</i>	32
CAPÍTULO 4 – HEURÍSTICA DE LOCALIZAÇÃO-ALOCAÇÃO	35
4.1 – O Algoritmo	35
4.2 – <i>HLA</i> e <i>PLMC</i>	36
4.3 – <i>HLA</i> e <i>PMC</i>	44
4.3.1 – O Algoritmo MTHG	44
4.3.2 – A <i>HLA</i> para o <i>PMC</i>	45
4.3.3 – Testes Computacionais para o <i>PMC</i>	48
CAPÍTULO 5 – ALGORITMO GENÉTICO CONSTRUTIVO	53
5.1 – Modelagem do AGC	53

5.1.1 – Representação de uma Estrutura e Esquema	53
5.1.2 – Problema de Otimização com Dupla Função Objetivo	54
5.1.3 – O Processo Evolutivo	55
5.1.4 – População Inicial	56
5.1.5 – Seleção e Recombinação	56
5.1.6 – A Mutação	58
5.2 – PLMC e o AGC	59
5.2.1 – As Funções de Avaliação	60
5.3 – Testes Computacionais	61
5.4 – PMC e o AGC	65
5.4.1 – As Funções de Avaliação	66
5.5 – Testes Computacionais	67
CAPÍTULO 6 – CONCLUSÕES E RECOMENDAÇÕES	71
REFERÊNCIAS BIBLIOGRÁFICAS	75

LISTA DE FIGURAS

	<u>Pág.</u>
2.1 Exemplo de uma configuração de um <i>PLMC</i>	22
3.1 Exemplo de uma configuração de um <i>PM</i>	28
3.2 Exemplo de uma configuração de um <i>PMC</i>	30
4.1 Exemplo da <i>HLA</i> para um caso do <i>PLMC</i>	38
4.2 Exemplo da aplicação da <i>HLA</i> para um caso do <i>PMC</i>	47
5.1 Pseudo-código do <i>AGC</i>	58
5.2 Representação de $s_k=(0,0,\#,0,1,0,\#,0,\#,0,\#,0,1)$	60
5.3 Representação de $s_k=(0,1,0,0,1,0,\#,1,\#,0,\#,0)$	66

LISTA DE TABELAS

	<u>Pág.</u>
4.1 Sumário de problemas testes	39
4.2 Resultados computacionais para os dados de GR100	40
4.3 Resultados computacionais para os dados de GR150	41
4.4 Resultados computacionais para os dados de LP324	42
4.5 Resultados computacionais para os dados de LP402	42
4.6 Resultados computacionais para os dados de LP500	43
4.7 Resultados computacionais para os dados de <i>OC</i>	50
4.8 Resultados computacionais para os dados de São José dos Campos	51
5.1 Resultados computacionais para os dados de LP324	61
5.2 Resultados computacionais para os dados de LP402	62
5.3 Resultados computacionais para os dados de LP500	63
5.4 Resultados computacionais para os dados de GR100	64
5.5 Resultados computacionais para os dados de GR150	65
5.6 Resultados computacionais para os dados de <i>OC</i>	69
5.7 Resultados computacionais para os dados de São José dos Campos	70

LISTA DE SÍMBOLOS

S	distância de serviço
p	número de facilidades, medianas
N	conjunto de pontos de demanda
M	conjunto de possíveis facilidades
D_i	a demanda de população da área i
d_{ij}	matriz de distâncias
q_i	demanda de cada entidade
Q_j	capacidade total de cada mediana
C_k	agrupamentos (<i>clusters</i>)
$ C_k $	número de elementos do agrupamento C_k
J	conjunto dos vértices facilidades
p_{ij}	custo de atribuir vértice i a mediana j
g	função de avaliação do AGC
f	função de avaliação do AGC
g_{max}	limite superior de avaliação do AGC
s_k	esquema ou estrutura
s_i	estrutura aleatória
α	parâmetro de evolução
$\Delta\alpha$	incremento do parâmetro de evolução
$\delta(s_k)$	”ranking” de s_k
X	conjunto de todos os esquemas de uma representação
$C_{n,m}$	combinação de n , m a m
d	desvio percentual global admitido
V	conjunto de vértices de um grafo

CAPÍTULO 1

INTRODUÇÃO

Uma importante linha de pesquisa da pesquisa operacional envolve as aplicações dos problemas de localização de facilidades. Estes problemas tratam a questão de onde localizar um objeto (ou objetos) que é chamado facilidade. A facilidade deverá interagir com outros objetos que possuirão localizações fixas. O conceito de distância entre a facilidade a ser localizada e os outros objetos contribui para identificar uma função objetivo que será utilizada para avaliar possíveis localizações da facilidade. Além deste conceito alguns problemas combinam informações de demandas e capacidades de serviços.

Estes problemas podem ser identificados em redes que são formadas por pontos (nós ou vértices) e linhas (arcos ou arestas) e podem descrever de forma natural vias públicas, conexões de água, telefonia, etc. O termo facilidade pode ser substituído por fábricas, depósitos, escolas, antenas, etc. Em geral, várias facilidades serão localizadas, que por sua vez, serão alocados aos seus clientes. Desta forma tais problemas são também conhecidos como problemas de localização-alocação. A maioria dos problemas de localização de facilidades é considerada de difícil solução, alguns desses problemas pertencem à classe NP-difícil.

Duas classes de problemas de localização são importantes e podem ser classificadas como a classe dos problemas de cobertura e a classe dos problemas de localização de medianas. Em ambas, decisões são tomadas sobre onde localizar facilidades (centros), considerando os outros centros como clientes que devem ser servidos, de forma a otimizar um dado critério. Dentre estes problemas podemos citar o Problema de Localização de Máxima Cobertura (*PLMC*) que tem como objetivo localizar p facilidades de modo que a máxima população possível seja coberta dentro da distância de serviço. Uma área (ponto) de demanda é considerada coberta se está dentro da distância de serviço de pelo menos uma facilidade. Outro problema interessante é o Problema das p -medianas Capacitado (*PMC*) que dado um conjunto de objetos com diferentes pesos, deseja-se particionar este conjunto em p agrupamentos, de tal forma que o peso total dos objetos em cada agrupamento seja menor ou igual a um dado valor e ainda minimizar a dispersão total dos objetos em relação a uma mediana definida como centro do agrupamento.

Contribuindo para a importância dos problemas de localização, nos países de

grande dimensão existe uma carência de informações adequadas para tomada de decisões sobre problemas urbanos e ambientais. Para preencher tal lacuna, os instrumentos computacionais do Geoprocessamento, chamados de Sistemas de Informações Geográficas (*SIG*), permitem a realização de análises complexas ao integrar dados de diversas fontes e criar bancos de dados georreferenciados (Assad e Sano, 1998).

Em particular, existem *SIG* que se preocupam com a resolução de problemas de localização e já implementam alguns algoritmos, como por exemplo: o *ARC/INFO* (ESRI, 1996) já possui implementado heurísticas para resolver o *PLMC* e o Problema das *p*-medianas (*PPM*). Estas heurísticas foram desenvolvidas por Densham e Rushton (1992) e por Teitz e Bart (1968).

Dentro desta perspectiva, propomos neste trabalho uma nova heurística de localização-alocação para problemas de localização de facilidades, pois é sabido que a combinação das funções de visualização e análise espacial de um *SIG* e um modelo de localização-alocação fornecerão uma poderosa ferramenta para suporte de decisão espacial.

A Heurística de Localização-Alocação (*HLA*) foi baseada em trabalhos de Cooper (1963) e Taillard (1996), e supõe que inicialmente haja agrupamentos (*cluster*). A idéia básica é identificar *p* centros abertos (medianas ou centros para cobertura), que por sua vez identificam *p* agrupamentos, formados pelos centros abertos e os alocados a estes (ou cobertos por estes). Pode-se desta forma tentar melhorar a qualidade das localizações e alocações realizando troca dentro dos agrupamentos, realocando e formando novos agrupamentos.

A *HLA* é aplicada ao *PLMC* e ao *PMC* e comparada com resultados de outros autores.

Algoritmo Genético Construtivo (*AGC*) foi desenvolvido por Furtado (1998) e aprimorado recentemente por Ribeiro Filho (2000). Neste trabalho, implementou-se a *HLA* como processo de mutação para o *AGC* e desta forma aplicou-se esta heurística aos mesmos problemas (*PLMC* e *PMC*), uma vez que o *AGC* utiliza heurísticas dependentes do problema como processo de mutação (Ribeiro Filho, 2000), a *HLA* está sendo avaliada neste contexto.

Nos capítulos 2 e 3 são fornecidos detalhes sobre os dois problemas de localização estudados. No capítulo 2 apresentamos o *PLMC*, juntamente com a sua formulação matemática e algumas heurísticas e métodos exatos para o problema. No capítulo 3

apresentamos inicialmente o *PPM* que é um problema já clássico e a seguir o *PMC* na qual foi aplicado a *HLA*. No capítulo 4 descrevemos a nova heurística e sua aplicação aos dois problemas de localização *PLMC* e *PMC*. No capítulo 5 apresentamos o *AGC* e sua formulação para o *PLMC* e o *PMC*. No capítulo 6 são apresentadas as conclusões.

CAPÍTULO 2

O PROBLEMA DE LOCALIZAÇÃO DE MÁXIMA COBERTURA

O Problema de Localização de Máxima Cobertura (*PLMC*) consiste em escolher locais para instalar facilidades de forma que o maior número de clientes (pontos de demanda) sejam cobertos, e saber que cliente será atendido por qual facilidade. Podemos exemplificar este problema citando a aplicação feita por Adenso-Diaz e Rodriguez (1997). Dado um número limitado de bases de ambulâncias de atendimento médico, como determinar onde elas devem ser localizadas de forma que a maioria da população seja atendida prontamente em um dado tempo. E ainda minimizar os custos de implantação de tal rede de ambulâncias.

No *PLMC* também podemos ter como restrições da função objetivo: a distância ou tempo total da viagem, a partir do ponto de demanda até a facilidade mais próxima ou ainda a distância ou tempo máximo que o usuário mais distante de uma facilidade pode percorrer até alcançar a facilidade mais próxima, isto é, a distância crítica de serviço (S). Este conceito foi largamente discutido por Toregas e Reville (1972) que mostra que é uma importante medida alternativa para uma dada configuração de localização.

A abordagem do *PLMC* surgiu num artigo de Church e Reville (1974) como uma terceira alternativa já que tanto o *SCLP* (*Set Covering Location Problem*) - que consiste em identificar e localizar o número mínimo de facilidades, de modo que nenhuma das distâncias entre um ponto de demanda e a facilidade mais próxima seja maior que a distância crítica - quanto o problema dos *p*-centros *PCP* (*p-Center Problem*) - procura localizar p facilidades de maneira que a medida de dissimilaridade máxima de qualquer ponto de demanda até sua facilidade mais próxima seja mínima - exigem que todas as áreas de demanda sejam cobertas, porém nem sempre os recursos são suficientes para garantir a cobertura total à demanda, dado que na prática pode existir um número insuficiente de facilidades. Desta forma esta formulação (*PLMC*) se aproxima mais da realidade: Maximizar a cobertura (população coberta - demanda atendida) dentro de uma desejada distância S localizando um número fixo de facilidades.

A figura 2.1 ilustra um caso de *PLMC* com duas facilidades e distância de serviço S . Esta poderia ser uma solução ao *PLMC* não necessitando que todas as demandas sejam atendidas.

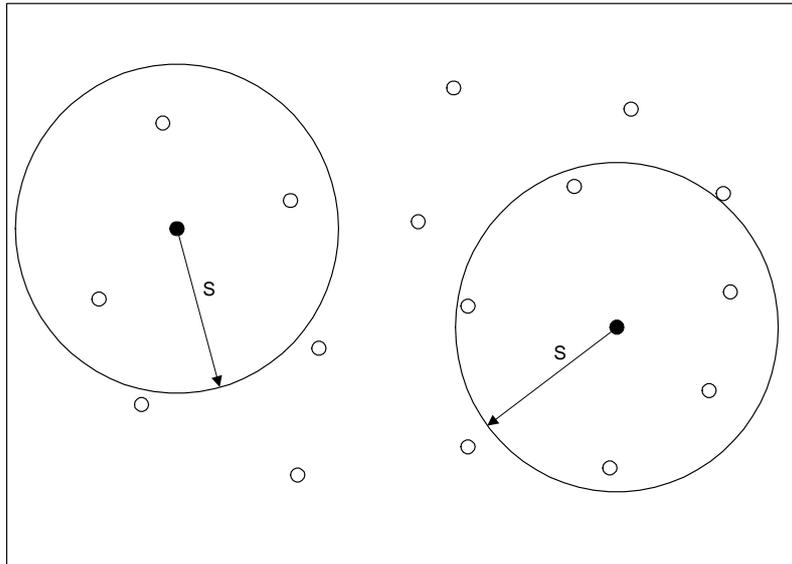


Fig. 2.1 – Exemplo de uma configuração de um *PLMC*

2.1 Descrição do Problema

O *PLMC* procura cobrir as áreas de demanda, tendo como critério para determinar se um ponto de demanda é coberto ou não, é estar a menos de uma distância crítica S determinada a priori, de pelo menos uma facilidade. Uma área de demanda é considerada não coberta caso a facilidade mais próxima a ela esteja localizada a uma distância maior do que S . O *PLMC* não faz restrições de capacidade e não exige que todas as áreas de demanda estejam cobertas; este problema tem como objetivo localizar p facilidades de modo que haja a máxima cobertura possível dentro da distância pré-definida S .

2.2 Formulação Matemática do *PLMC*

Church e Reville (1974) apresentaram a formulação matemática deste problema. O *PLMC* pode ser formulado matematicamente como:

$$v(\text{PLMC}) = \text{Max} \sum_{i \in N} D_i y_i \quad (2.1)$$

$$\text{sujeito a} \quad \sum_{j \in N_i} x_j \geq y_i \text{ para todo } i \in N \quad (2.2)$$

$$\sum_{j \in M} x_j = p, \quad (2.3)$$

$$x_j \in \{0, 1\}, \text{ para todo } j \in M, \quad (2.4)$$

$$y_i \in \{0, 1\}, \text{ para todo } i \in N. \quad (2.5)$$

Onde:

S : distância de serviço - a área de demanda é coberta se está dentro desta distância;

$N = \{1, 2, \dots, n\}$: conjunto de pontos de demanda;

$M = \{1, 2, \dots, n\}$: conjunto de possíveis facilidades;

D_i : a demanda de população da área i ;

p : número de facilidades a serem localizadas;

d_{ij} = a menor distância do nó i ao nó j ;

$N_i = \{j \in J \mid d_{ij} \leq S\}$;

$$y_i = \begin{cases} 1 & \text{se a área de demanda } i \text{ é coberta,} \\ 0 & \text{caso contrário;} \end{cases}$$

$$x_j = \begin{cases} 1 & \text{se a facilidade deve ser localizada em } j, \\ 0 & \text{caso contrário;} \end{cases}$$

Na formulação acima a função objetivo (2.1) procura maximizar a população coberta. A restrição (2.2) diz que a área de demanda $j \in J$ é coberta e se há pelo menos uma facilidade dentro da distância S . A restrição (2.3) limita o número de facilidades a p . E as restrições (2.4) e (2.5) definem as variáveis de decisão a $\{0,1\}$.

2.3 Aplicações do *PLMC*

Podemos encontrar uma revisão das aplicações do *PLMC* em Chung (1986) e Shilling et al. (1993). As aplicações são as mais variadas possíveis envolvendo localização de facilidades (localizar centros de serviços emergenciais, agências bancárias, áreas de conservação e outros).

Temos então :

- Planejamento de uma rede de ambulâncias (Hamon et al., 1979); (Eaton et al., 1986); (Adenso-Diaz e Rodriguez, 1997).
- Localização de sirenes de aviso (Current e O’Kelly, 1992).
- Escolha de agências bancárias para maximizar a base de recolhimento de fundos (Goberna et al., 1990).
- Projeto de uma rede de monitores para controlar a poluição do ar (Houglund e Stephens, 1976).
- Escolha e composição de listas para realizar uma campanha publicitária pelo correio (Dwyer e Evans, 1981).
- Escolha de ferramentas em manufatura flexível (Daskin et al., 1990).
- Seleção de áreas prioritárias para conservação (Woodhouse et al., 2000) (Davis e Goodchild, 1996).
- Identificação de áreas que representam a máxima possível representação de espécies específicas (Church et al., 1996).

2.4 Métodos de Solução do *PLMC*

O *PLMC* é NP-difícil (Garey e Johnson, 1979). A seguir faremos a descrição de algumas heurísticas e métodos exatos para resolvê-lo.

2.4.1 Métodos Heurísticos

Church e Reville (1974) desenvolveram duas heurísticas gulosas para resolver o *PLMC* chamadas de Algoritmo de Acréscimo Guloso (*AAG*) (*Greedy Adding Algorithm*)

e Algoritmo de Acréscimo Guloso com Substituição (AAGS) (*Greedy Adding with Substitution Algorithm*). A primeira heurística faz uma busca gulosa pela primeira facilidade que consiga cobrir a maior parte da população e o mesmo é feito para a segunda facilidade: cobrir a maior parte da população não coberta pela primeira facilidade e assim por diante até p facilidades. O resultado é a máxima cobertura, porém, isto não garante que se encontre a solução ótima. A segunda heurística faz uma melhora da heurística anterior, na medida que, tenta melhorar a solução parcial em cada iteração, pois procura trocar as facilidades por outras que ainda não foram selecionadas. Caso haja melhora na solução a facilidade é trocada. Neste mesmo artigo Church e Reville (1974) apresentam um resolução para o *PLMC* utilizando uma relaxação de programação linear, relaxando as condições de integralidade das restrições (2.4) e (2.5) substituindo-as por

$$y_i \leq 1 \forall i \in I,$$

$$x_j \leq 1 \forall j \in J.$$

Galvão e Reville (1996) propõe uma técnica de relaxação Lagrangeana, de modo a aproveitar a esparsidade da matriz de restrições e ser mais eficiente nos problemas com grandes instâncias que os procedimentos baseados na relaxação linear. O algoritmo interage entre um limite superior dado pela heurística de Church e Reville (1974) (AAGS) e um limite inferior produzido através de um algoritmo de otimização de subgradiente, até obter uma solução ótima ou alcançar uma regra de parada. Cada iteração do algoritmo tenta melhorar os dois limites.

Em Galvão et al. (2000) é feita uma comparação entre duas heurísticas: uma baseada na relaxação Lagrangeana e outra na relaxação *surrogate* para resolver o *PLMC*. A relaxação *surrogate* reduz o *PLMC* ao problema da mochila 0-1, concluindo que para o conjunto de testes utilizado as duas heurísticas não diferem substancialmente.

2.4.2 Métodos Exatos

Dwyer e Evans (1981) propuseram um algoritmo exato para o *PLMC* que resolvia uma problema específico de seleção e composição de listas para realizar campanha publicitária pelo correio. O algoritmo utilizava uma heurística gulosa para encontrar uma solução inicial e depois buscava melhores soluções através de um método *branch and bound*.

Daskin et al. (1990) apresentaram um algoritmo exato de programação dinâmica para uma instância específica do *PLMC*, portanto não generalizável.

Downs e Camm (1996) apresentam o algoritmo *Dual-Based Algorithm (DBA)*, que tem duas etapas e trabalha com o dual da relaxação de programação linear do *PLMC* e o dual da relaxação Lagrangeana do *PLMC*. Foi realizado um estudo computacional extensivo em termos de variedade de aplicações e tamanho das instâncias. Utilizaram instâncias geradas aleatoriamente e conjuntos de dados reais e obtiveram resultados que levam a afirmar que o *DBA* é um algoritmo robusto para obter soluções ótimas para o *PLMC* e ainda mostraram que o uso da relaxação Lagrangeana para encontrar limites para o *PLMC* é um método eficiente tanto para encontrar soluções ótimas como aproximadas.

CAPÍTULO 3

O PROBLEMA DAS P-MEDIANAS CAPACITADO

Antes de descrever o problema das *p*-medianas capacitado (*PMC*) faremos uma breve introdução do problema clássico das *p*-medianas (*PM*).

3.1 O Problema das P-Mediana

A busca de *p*-medianas num grafo tem como objetivo localizar *p* facilidades ou recursos (medianas), de forma a minimizar o custo total (a soma das distâncias de cada vértice à sua mediana mais próxima).

Proposta inicialmente por Hakimi (1964) para uma única mediana, foi em seguida generalizado para múltiplas medianas (Hakimi, 1965), propondo um simples processo de enumeração para o problema.

Reconhecidamente NP-difícil (Garey e Johnson, 1979), diversas heurísticas tem sido desenvolvidas para o problema das *p*-medianas, como a heurística de substituição de vértices de Teitz e Bart (1968) e suas variações (Densham e Rushton, 1992). Outras abordagens utilizam as informações primal/dual do problema (Senne e Lorena, 2000). Hribar e Daskin (1997) propuseram um algoritmo de programação dinâmica e no contexto das metaheurísticas temos o trabalho de Rolland et al. (1996) que utiliza uma busca tabu. Algoritmos exatos exploram uma árvore de busca como as de Christofides e Beasley (1982) e mais recentemente Klose (1998) e outros, como Galvão e Raggi (1989).

Em termos de aplicação do *PM* temos como exemplos: Pizzolato e Silva (1997) na localização de escolas públicas numa região pobre no Rio de Janeiro e a formação de células de manufatura feita por Deutsch et al. (1998).

Na figura 3.1 temos um exemplo de uma solução para o *PM* para o caso de 9 vértices e 3 medianas (vértices 3, 4 e 9).

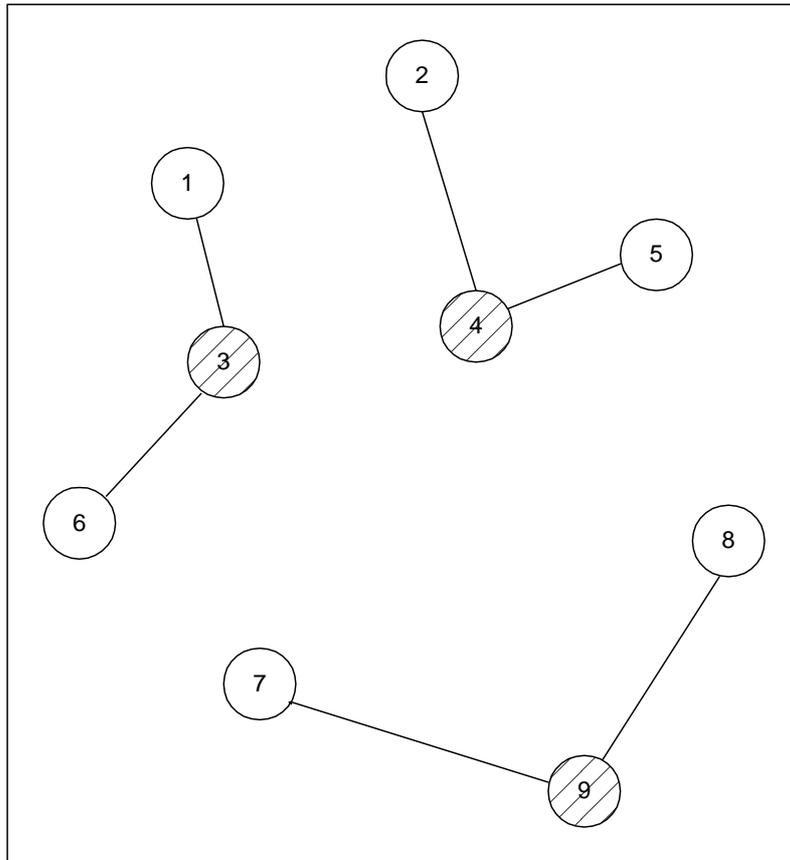


Fig. 3.1 – Exemplo de uma configuração de um *PM*

3.1.1 Formulação Matemática do *PM*

O problema das *p*-medianas pode ser formulado como um problema de programação inteira binária. Dado um grafo descrevendo uma dada instância, obtida através da aplicação do algoritmo de Floyd (Beasley, 1993) e um conjunto resultante de vértices $N = \{1, \dots, n\}$ podemos descrevê-lo da seguinte maneira:

$$v(\text{PM}) = \text{Min} \sum_{i \in N} \sum_{j \in N} d_{ij} y_{ij} \quad (3.1)$$

$$\text{sujeito a} \quad \sum_{j \in N} y_{ij} = 1; \quad i \in N \quad (3.2)$$

$$\sum_{j \in N} y_{jj} = p, \quad (3.3)$$

$$y_{ij} \leq y_{ii} \quad i, j \in N \quad (3.4)$$

$$y_{ij} \in \{0, 1\}, \quad i \in N, j \in N \quad (3.5)$$

Onde:

$N = \{1, \dots, n\}$ é o índice dos objetos a serem alocados;

p : número de medianas;

$[d_{ij}]$ é a matriz de distâncias;

$$y_{ij} = \begin{cases} 1 & \text{se o vértice } i \text{ é atribuído ao vértice } j, \\ 0 & \text{caso contrário.} \end{cases}$$

As restrições 3.2 e 3.4 determinam que cada vértice j seja atribuído somente a um vértice i , que deve ser uma mediana. A restrição 3.3 assegura o número exato de medianas a serem localizadas e a restrição 3.5 indica a condição binária de atribuições.

3.2 O Problema das P-Medianas Capacitado

O *PMC* é um problema de particionamento de conjunto de n objetos ou clientes (vértices), onde cada objeto possui uma demanda (peso), em p agrupamentos disjuntos, de maneira que a dissimilaridade (distância ou tempo) total dentro de cada agrupamento seja minimizada e a restrição de capacidade de cada agrupamento seja respeitada. A dissimilaridade total de cada agrupamento é calculada como a soma das dissimilaridades existentes entre cada vértice e a sua mediana associada no agrupamento. Aparentemente o *PMC* não foi tão estudado quanto o problema clássico da p-medianas.

Na figura 3.2 temos um caso que embora a distância euclidiana do vértice 2 à mediana 3 seja maior do que à mediana 4, devido a restrição da capacidade o vértice 2 é alocado à

mediana 3.

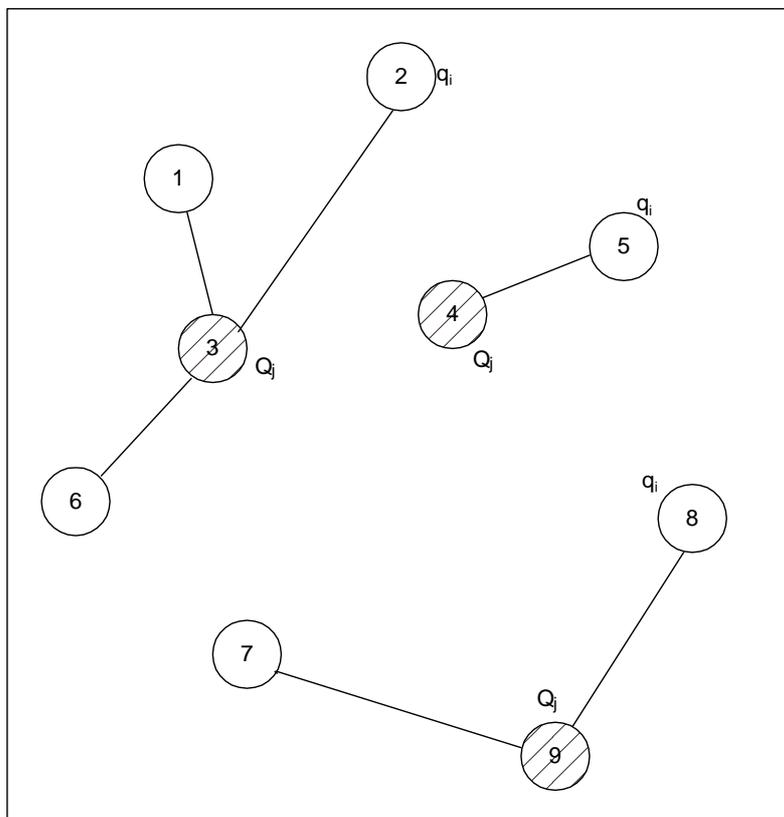


Fig. 3.2 – Exemplo de uma configuração de um *PMC*

O problema é NP-difícil (Garey e Johnson, 1979) e foi estudado em teoria de localização e agrupamento (*clustering*). Devido a sua complexidade, algoritmos exatos encontram dificuldades para resolver problemas das dimensões encontradas no mundo real, e portanto a maioria das soluções são baseadas em heurísticas. Estas podem ser classificadas em heurísticas de construção e heurísticas de melhoramento. Nos métodos construtivos, a solução viável é construída através de acréscimos sucessivos de novos elementos de acordo com algum critério (encaixam nesta classe os algoritmos gulosos). Nas heurísticas de melhoramento, um mecanismo troca iterativamente elementos de uma solução e esta é utilizada para melhorar a função objetivo (encaixam-se nesta classe de algoritmos a busca local ou a busca em vizinhança).

Vários algoritmos exatos e heurísticas já foram propostos na literatura para o *PMC* ou

similares (Problemas de Agrupamento Capacitado). Pirkul (1987) descreve um algoritmo *branch and bound* que utiliza a relaxação Lagrangeana nas restrições de particionamento.

Mulvey e Beck (1984) apresentam dois algoritmos. O primeiro é uma heurística primal que faz uma busca "gulosa". E para minimizar a atribuição de clientes a medianas muito distantes, utilizam um valor de "insatisfação" (*regret*) que é definido como um valor absoluto da diferença em distância entre a primeira e a segunda mais próxima mediana de um cliente. Depois de todas as atribuições e o conjunto final de todas as medianas é encontrado realiza-se troca de medianas entre os agrupamentos para tentar melhorar a solução. O outro algoritmo apresentado é um híbrido da heurística primal e algoritmo de subgradiente. Resultados computacionais de 18 problemas gerados aleatoriamente de até 100 clientes e 20 medianas mostram que o método é capaz de encontrar soluções com desvio relativo do limite inferior do subgradiente variando de 0,37 a 3,6 %.

Golden e Skiscim (1986) aplicam o método "simulated annealing" ao problema do caixeiro viajante e ao problema das p-medianas.

Osman e Christofides (1994) utilizam um algoritmo híbrido que implementa um algoritmo de busca tabu e o método "simulated annealing" que utiliza um procedimento de três estágios para obter a solução inicial, o primeiro estágio garante o espalhamento das sementes iniciais quando estas são determinadas; os dois clientes mais remotos são escolhidos e atribuídos como medianas; a próxima mediana é selecionada de modo a maximizar o produto das distâncias desta mediana as medianas já selecionadas. No segundo estágio os clientes não-medianas são atribuídas às suas mais próximas medianas de forma que se respeite a capacidade. Sempre que uma atribuição produza um estouro de capacidade da mediana, o correspondente cliente é atribuído a sua mais próxima mediana disponível. E no terceiro estágio, a mediana de cada agrupamento é recalculada de acordo com as atribuições finais. Se novas medianas são encontradas, o processo de atribuição é repetido. A partir desta solução aplicam o algoritmo híbrido que emprega uma mudança e uma troca na vizinhança. Os resultados de 20 problemas apresentam desvios de 0,0 a 2,94% das melhores soluções conhecidas.

Maniezzo et al. (1998) apresentam um algoritmo bionômico, uma metaheurística evolutiva que atualiza toda uma população inteira de soluções em cada iteração. O processo de atualização consiste em definir uma solução filha a partir de um conjunto de soluções pai da geração anterior, dentro desta visão o algoritmo bionômico permite um espalhamento evolucionário da população pois possibilita que o conjunto de soluções

tenha tamanho variável e o uso de múltiplos pais. A aplicação desta metaheurística ao *PMC* aos problemas padrões da literatura atestam a sua eficácia.

França et al. (1999) utilizam um algoritmo de busca tabu com um novo esquema adaptativo e quatro heurísticas construtivas para obter a solução inicial, a eficácia do algoritmo foi testada em 20 problemas encontrados na literatura. A redução de custo observada em relação a versão não adaptativa foi de 98,6% em média

Baldacci et al. (2002) utilizam um novo método baseado na formulação de particionamento de conjuntos, a solução alcançada pode não ser uma solução ótima, contudo o novo método permite estimar sua máxima distância da otimalidade.

3.2.1 Formulação Matemática do *PMC*

O *PMC* pode ser modelado como o seguinte problema de programação inteira:

$$v(\text{PMC}) = \text{Min} \sum_{i \in N} \sum_{j \in M} d_{ij} x_{ij} \quad (3.6)$$

$$\text{sujeito a} \quad \sum_{j \in M} x_{ij} = 1; \quad i \in N \quad (3.7)$$

$$\sum_{j \in M} y_j = p, \quad (3.8)$$

$$\sum_{i \in N} q_i x_{ij} \leq Q_j y_j; \quad j \in M \quad (3.9)$$

$$x_i \in \{0, 1\}, \quad i \in N, \quad (3.10)$$

$$y_j \in \{0, 1\}, \quad j \in M. \quad (3.11)$$

Onde:

$N = \{1, \dots, n\}$ é o índice dos objetos a serem alocados;

$M = \{1, \dots, n\}$ é o índice dos objetos das possíveis medianas;

p : número de medianas;

q_i é a demanda de cada objeto;

Q_j é a capacidade total de cada possível mediana;

$[d_{ij}]_{m \times n}$ é a matriz de distâncias;

$[x_{ij}]_{n \times m}$ é a matriz de alocações;

$$x_{ij} = \begin{cases} 1 & \text{se o objeto } i \text{ está alocada à mediana } j, \\ 0 & \text{caso contrário.} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{se a mediana } j \text{ é selecionada,} \\ 0 & \text{caso contrário.} \end{cases}$$

As restrições (3.7) e (3.8) forçam que cada objeto seja alocado a somente uma mediana. A restrição (3.9) impõe que a capacidade total da mediana deve ser respeitada e as restrições (3.10) e (3.11) mantêm a condição de integralidade.

No próximo capítulo descreveremos a Heurística de Localização-Alocação (*HLA*) e sua aplicação aos problemas *PMC* e *PLMC*.

CAPÍTULO 4

HEURÍSTICA DE LOCALIZAÇÃO-ALOCAÇÃO

Neste capítulo, apresentaremos um algoritmo de localização-alocação bastante simples e eficiente. Este algoritmo encontra soluções de qualidade para problemas de *clustering* (agrupamento). Para analisar a sua potencialidade foram feitas aplicações ao *PLMC* e ao *PMC*. Para cada um destes problemas adaptações tiveram que ser feitas. A aplicação desta heurística tem como objetivo a sua possível integração a um *SIG* como mencionado no capítulo 1.

4.1 O Algoritmo

Nesta seção será descrita a Heurística de Localização-Alocação (*HLA*) que foi inicialmente inspirada nos trabalhos de Cooper (1963) e Taillard (1996).

A heurística de Cooper (1963) alternava entre a alocação da população aos centros e localização destes centros numa sequência até que houvesse uma convergência global. Na primeira iteração, uma região era subdividida em subregiões pela alocação dos pontos de demanda aos seus centros mais próximos, que eram escolhidos arbitrariamente. E a seguir, calculava-se a localização ótima de cada centro dentro da subregião. Como exemplo de localização-alocação podemos citar o artigo de Yeh e Chow (1996) que aplica um modelo de *p*-medianas numa área concreta em Hong Kong, integrando a um *SIG*.

Suponha um grafo $G=(V,E)$ e uma instância típica tanto para o *PLMC* e o *PMC*, composto por n pontos de demanda (vértices) $V = \{1, \dots, n\}$. Destes, p vértices são eleitos como sementes, ou seja, vértices que irão iniciar os agrupamentos. Estas sementes serão as medianas para o *PMC* ou facilidades no caso do *PLMC*. E uma vez localizados, teremos portanto, associado a eles os agrupamentos, no caso p agrupamentos (*clusters*) $C_k, k \in \{1, 2, \dots, p\}$ formados por eles próprios e os demais vértices alocados a estes (ou cobertos por estes).

Dada uma solução inicial com os seus respectivos agrupamentos tenta-se melhorar a solução através de uma busca de uma nova semente (localização) dentro de cada agrupamento (*cluster*), troca-se uma semente por um vértice não-semente e recalcula-se as alocações. Este processo se repete até que não seja mais possível obter melhorias no custo total das alocações.

O algoritmo da *HLA* está descrito a seguir em pseudo-código:

```
Enquanto (solução-inicial melhora) faça
  Para  $k = 1, \dots, p$ 
    Troque vértice semente por não-semente do agrupamento  $C_k$ ;
    Calcule o valor  $v$  correspondente à melhor realocação;
    Se  $v$  é melhor que solução-inicial então
      Atualize o vértice semente do agrupamento  $C_k$ ;
      Faça solução-inicial  $\leftarrow v$ ;
    Fim_Se
  Fim_Para
Fim_Enquanto
```

Algoritmo *HLA*

A troca entre vértices semente e não-semente em cada agrupamento C_k , $k = 1, \dots, p$, pode ser executada para:

- Todos os vértices não-ementes do agrupamento C_k , ou
- Apenas para os vértices não-ementes localizados a uma certa distância (ou tempo) do vértice semente do agrupamento C_k .

4.2 *HLA* e *PLMC*

A *HLA* foi aplicada ao *PLMC*, sendo que inicialmente os agrupamentos foram formados escolhendo-se aleatoriamente os vértices sementes (facilidades) e através da distância crítica S foram construídos os agrupamentos. A troca entre vértices semente e não-semente foi feita para todos os vértices não-ementes do agrupamento.

A adaptação da *HLA* para o *PLMC* é apresentada no pseudo-código a seguir. No *PLMC* queremos maximizar a cobertura do pontos de demanda.

```

Dados  $J$  conjunto dos vértices facilidades =  $\{j_1, \dots, j_p\}$  e
 $C_k$  conjunto de vértices do agrupamento  $k = \{v_1, \dots, v_{|C_k|}\}$ 
 $|C_k|$  cardinalidade de  $C_k$ 
Enquanto (solução-inicial melhora) faça
  Para  $k = 1, \dots, p$ 
    Para  $i = 1, \dots, |C_k|$ 
      Troque facilidade  $j_k$  por um vértice  $v_i$ ;
      Calcule o valor  $sol$  correspondente ao novo valor de
      cobertura;
      Se  $sol$  é melhor que solução-inicial então
        Faça  $melhor\_vertice \leftarrow v_i$ ;
        Faça  $solução-inicial \leftarrow sol$ ;
      Fim_Se;
    Fim_Para;
  Se houver  $melhor\_vértice$  então;
  Atualize  $J$ ;
  Fim_Se;
Fim_Para;
Fim_Enquanto;

```

Algoritmo *HLA* para o *PLMC*

Na figura 4.1 é mostrado como a *HLA* funciona. Ilustraremos com um exemplo prático.

Na parte (a) da figura 4.1 temos dois agrupamentos e uma solução inicial. Dentro do primeiro agrupamento faz-se a troca do vértice facilidade por um não-facilidade e obtem-se uma nova solução (parte (b) da figura 4.1). Caso esta nova solução seja melhor que a solução inicial guarda-se esta nova solução e atualiza-se a solução inicial, e continua a busca no agrupamento até varrer todos os seus elementos (parte (c) da figura 4.1). Ao término da busca no agrupamento, atualiza-se a nova facilidade e segue-se a outro agrupamento, repetindo o processo até não encontrar melhora na solução.

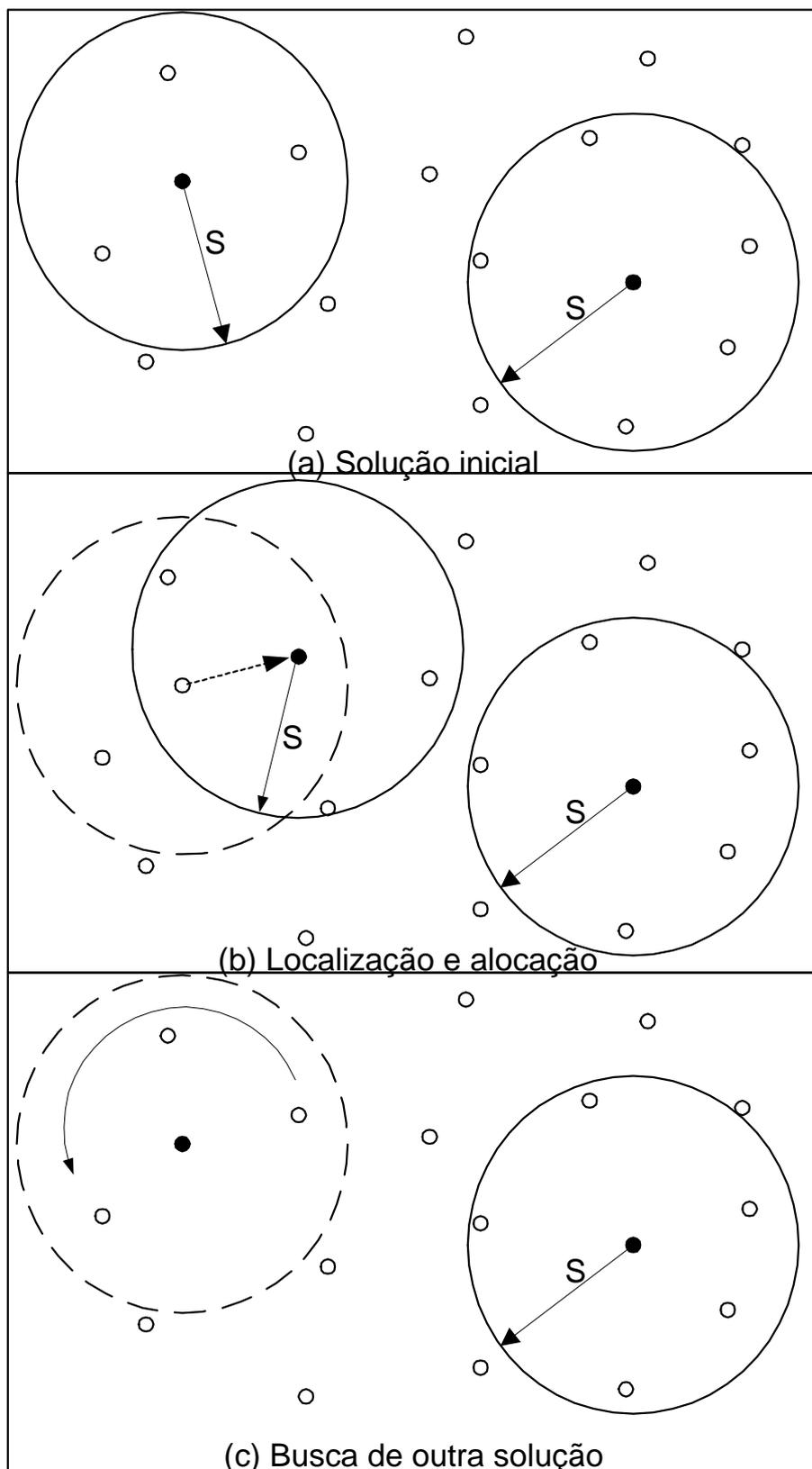


Fig. 4.1 – Exemplo da *HLA* para um caso do *PLMC*

Para testar a *HLA* para o *PLMC*, em termos de eficiência, foram organizados um conjunto de instâncias reais coletados na área central da cidade de São José dos Campos (SP) através de um Sistema de Informações Geográficas (*SIG*) chamado *ArcView*, dados do projeto *ARSIG2* (<http://www.lac.inpe.br/~lorena/projetos.html>).

As instâncias foram denominadas LP324, LP402, LP500, onde os seus números representam o respectivo número de vértices. Cada ponto é localizado sobre um quarteirão que representa uma demanda de população e é também um possível lugar para posicionar as facilidades. Foi simulado a instalação de antenas de rádio para uso de Internet com alcance de 800 m, 1200 m e 1600 m. O número de facilidades variou para cada uma destas instâncias de 1 até que fosse completado 100% de cobertura.

Outro conjunto de testes utilizado são as matrizes de distância usadas por Galvão e Revelle (1996) e Galvão et al. (2000) para uma rede de 100 e 150 vértices. Os valores de demanda utilizados não são idênticos, mas gerados da mesma maneira: a demanda de cada nó (vértice) foi gerada a partir de uma distribuição uniforme em um intervalo [20,30] para a rede de 100 vértices (distância de serviço igual a 50, 65 e 80 m) e a partir de uma distribuição normal com média igual a 80 e desvio padrão igual a 15 para a rede de 150 vértices (distância de serviço igual a 70, 75, 80, 85 e 90 m). O algoritmo descrito está codificado em C e os testes foram feitos num PC Pentium II MMX 233 MHz e 128 MB de RAM.

Um sumário dos problemas testes utilizados são apresentados na Tabela 4.1.

Problema	No vértices	Valores de p	Valores de S
GR100	100	8,10,12	50,65,80
GR150	150	8,10,12,14,16,18,20	70,75,80,85,90
LP324	324	1 a 5	800,1200,1600
LP402	402	1 a 6	800,1200,1600
LP500	500	1 a 8	800,1200,1600

TABELA 4.1 – Sumário de problemas testes

A *HLA* foi aplicada em todas estas instâncias e para cada uma foram feitas 100 rodadas, e avaliadas a melhor solução encontrada (cobertura em %), a frequência da melhor solução,

a média das soluções encontradas e o tempo total das rodadas.

As tabelas 4.2 e 4.3 mostram-se os resultados computacionais para problemas com 100 e 150 vértices. A melhor solução encontrada para os problemas foram obtidas de Galvão et al. (2000) e Galvão e Reville (1996) que chamaremos este conjunto de dados de *GR* e rodadas em estações de trabalho Digital Alpha 3000/300. A última linha apresenta a média de tempos da *HLA* e *GR* para cada grupo de problemas.

As tabelas 4.4, 4.5, 4.6 apresentam-se resultados obtidos para os problemas LP324, LP402, LP500. Estas mostram a população máxima atendida, a cobertura (%), a média da população atendida, a frequência da melhor solução, o tempo das cem rodadas.

Os resultados se encontram nas tabelas a seguir.

<i>n</i>	<i>p</i>	<i>S</i>	Mel. sol. <i>GR</i> (%)	Mel. sol. <i>HLA</i> (%)	Média sol. <i>HLA</i> (%)	Freq. mel. sol. <i>HLA</i>	Tempo(s) <i>HLA</i>	Tempo(s) <i>GR</i>	
100	8	50	69,43	67,72	59,08	1	2	51	
100	10	50	76,23	74,69	65,83	1	2	62	
100	12	50	81,61	78,75	70,91	1	2	64	
100	8	65	87,36	86,22	79,59	1	4	53	
100	10	65	94,33	93,31	86,99	1	4	20	
100	12	65	99,81	97,00	91,75	1	4	22	
100	8	80	88,46	87,99	81,74	2	4	43	
100	10	80	96,21	94,04	88,59	1	4	20	
100	12	80	100,00	98,28	92,96	1	4	7	
média dos tempos								3,3	38

TABELA 4.2 – Resultados computacionais para os dados de GR100

Na tabela 4.2 podemos notar que embora os problemas não sejam totalmente idênticos (as demandas foram geradas da mesma forma), a melhor solução *HLA* se aproxima bem da solução *GR* tendo em conta os tempos de processamento e o tipo de processador empregado nos dois casos (um Pentium II x Digital Alpha 3000/30 , fazendo que a comparação entre os dois tempos não seja algo substancial devido a falta de informação a respeito dos processadores) .

n	p	S	Melhor sol. conhecida (%)	Melhor sol. <i>HLA</i> (%)	Média sol. <i>HLA</i> (%)	Freq. melhor sol. <i>HLA</i>	Tempo (s)	Tempo (s) <i>GR</i>
150	10	70	68,86	69,09	63,66	1	8	9
150	12	70	77,09	76,48	69,46	1	8	11
150	14	70	83,34	83,81	77,00	1	11	12
150	16	70	87,75	86,64	80,53	1	12	13
150	18	70	92,39	90,20	84,71	1	11	12
150	20	70	93,95	92,90	88,16	1	13	6
150	8	75	59,14	59,33	54,67	1	6	109
150	10	75	68,86	67,86	63,11	1	7	122
150	12	75	77,34	75,39	69,63	1	9	127
150	8	80	61,49	62,80	57,61	1	6	4
150	10	80	70,91	70,65	65,14	2	8	6
150	12	80	78,14	77,74	71,60	1	8	10
150	8	85	73,94	73,83	69,34	1	7	96
150	10	85	81,56	80,68	75,89	1	9	127
150	12	85	87,95	87,33	82,10	1	10	154
150	8	90	89,79	89,53	84,28	1	9	8
150	10	90	94,04	93,37	88,56	1	10	7
150	12	90	96,93	96,02	91,62	1	12	5
média dos tempos							9,1	46,5

TABELA 4.3 – Resultados computacionais para os dados de GR150

Na tabela 4.3 nota-se que *HLA* apresentou um resultado melhor em relação a *GR* em 4 problemas.

n	p	S	Pop. atend. max.	Melhor Cobert. (%)	Média pop. atend.	Freq. melhor sol.(%)	Tempo (s)
324	1	800	5461	44,94	5461	100	6
324	2	800	8790	72,33	8610	65	15
324	3	800	11604	95,49	11177	8	28
324	4	800	12106	99,62	11559	5	29
324	5	800	12152	100,00	11661	13	24
324	1	1200	9932	81,73	9932	100	10
324	2	1200	11555	95,09	11200	17	21
324	3	1200	12152	100,00	11975	46	18
324	1	1600	12123	99,76	12123	100	13
324	2	1600	12152	100,00	12150	88	10

TABELA 4.4 – Resultados computacionais para os dados de LP324

n	p	S	Pop. atend. max.	Melhor Cobert. (%)	Média pop. atend.	Freq. melhor sol.(%)	Tempo (s)
402	1	800	6555	41,01	6555	100	6
402	2	800	11339	70,94	11271	98	25
402	3	800	14690	91,90	13922	79	36
402	4	800	15658	97,96	14816	6	44
402	5	800	15970	99,91	15408	1	48
402	6	800	15984	100,00	15451	14	47
402	1	1200	10607	66,36	10607	100	13
402	2	1200	14832	92,79	14543	60	36
402	3	1200	15984	100,00	15623	35	30
402	1	1600	15438	96,58	15438	100	19
402	2	1600	15984	100,00	15959	34	23

TABELA 4.5 – Resultados computacionais para os dados de LP402

n	p	S	Pop. atend. max.	Mel. Cobert. (%)	Média pop. atend.	Freq. mel. sol.(%)	Tempo (s)
500	1	800	7944	40,31	7944	100	8
500	2	800	12454	63,20	11953	43	23
500	3	800	15730	79,82	14939	4	41
500	4	800	17794	90,29	16944	8	59
500	5	800	18859	95,70	17748	4	75
500	6	800	19525	99,08	18738	2	94
500	7	800	19692	99,92	18771	1	90
500	8	800	19707	100,00	19176	3	92
500	1	1200	10726	54,43	10726	100	15
500	2	1200	18070	91,69	17238	49	42
500	3	1200	19393	98,41	18537	4	57
500	4	1200	19707	100,00	19320	40	56
500	1	1600	14804	75,12	14804	100	21
500	2	1600	19668	99,80	19327	51	46
500	3	1600	19707	100,00	19480	35	41

TABELA 4.6 – Resultados computacionais para os dados de LP500

Os resultados computados nas tabelas 4.4, 4.5, 4.6 foram obtidos variando o número de facilidades até alcançarmos cobertura total e serão comparados com os dados obtidos pelo *AGC* (a ser apresentado no capítulo 5), avaliando desta forma os possíveis benefícios de utilizar a *HLA* como processo de mutação.

Como podemos ver que a *HLA* na tabela 4.2 para o grupo de problemas *GR100*, não conseguiu chegar nos valores de *GR*. No grupo de problemas *GR150* da tabela 4.3, em quatro instâncias a *HLA* gerou resultados melhores do que os apresentados por *GR*. Resultado bastante bom considerando sua simplicidade em relação a heurística Lagrangeana e *surrogate* utilizado por *GR* mesmo tendo em conta que em cem rodadas a *HLA* encontrou apenas uma vez esta solução, porém a encontra num tempo bastante bom para uma possível implementação em *SIG*. O que nos leva a crer que para instâncias pequenas (100 a 150 vértices) a *HLA* é bastante eficiente. A confrontação com os dados gerados pelo *AGC* nos indicará como a *HLA* se comporta para instâncias maiores (324, 402, 500 vértices).

4.3 HLA e PMC

No caso do *PMC*, o problema de alocação é mais difícil devido às capacidades dos *clusters* (agrupamentos). Inicialmente, a escolha das medianas é feita de forma aleatória. Tendo o conjunto de medianas $J = \{1, \dots, p\}$ sido selecionado, o problema de alocar os vértices não-medianas aos vértices medianas torna-se um Problema Generalizado de Atribuição (*PGA*), que pode ser formulado da seguinte forma:

$$v(\text{PGA}) = \text{Max} \sum_{i \in N} \sum_{j \in J} p_{ij} x_{ij} \quad (4.1)$$

$$\text{sujeito a} \quad \sum_{i \in N} q_i x_{ij} \leq Q_j, \quad j \in J \quad (4.2)$$

$$\sum_{j \in J} x_{ij} = 1; \quad i \in N \quad (4.3)$$

$$x_{ij} \in \{0, 1\}, \quad i \in N, \quad (4.4)$$

onde $p_{ij} = -d_{ij}$, $i \in N$; $j \in J$, é o custo de atribuir o vértice i a mediana j e

$[d_{ij}]$ é a matriz de distâncias.

Utilizou-se o algoritmo MTHG de Martello e Toth (1990) para encontrar uma solução aproximada ao *PGA* proposto, construindo desta forma os agrupamentos. Uma vez tendo os agrupamentos utilizou-se a *HLA*.

4.3.1 O Algoritmo MTHG

Martello e Toth (1990) descreveram o *PGA* utilizando a terminologia do problema da mochila. Na formulação do *PGA* as sementes são as mochilas e os demais vértices são os itens a serem atribuídos às mochilas.

O algoritmo MTHG fornece soluções aproximadas ao *PGA*, criando uma função f_{ij} que mede a "desejabilidade" de se atribuir o item i à mochila j . Considera-se iterativamente todos os itens não atribuídos e determina-se o item i^* que tenha a máxima diferença entre o maior e o segundo maior f_{ij} ; i^* é então atribuído à mochila para qual f_{i^*j} é máximo. Martello e Toth (1990) propuseram quatro escolhas para a função f_{ij} que é definida de acordo com a função objetivo do problema, neste trabalho foi utilizado $f_{ij} = p_{ij}$. Na segunda parte do algoritmo a solução corrente é melhorada através de uma troca local.

Este algoritmo foi implementado em C e é utilizado tanto na *HLA* como no *AGC*, que será apresentado no próximo capítulo.

4.3.2 A *HLA* para o *PMC*

Naturalmente, que a implementação da *HLA* ao *PMC* é um pouco diferente da implementada para o *PLMC*. A sua adaptação está apresentada no pseudo-código que se segue.

```

Dados  $J$  conjunto dos vértices mediana =  $\{j_1, \dots, j_p\}$  e
 $C_k$  conjunto de vértices do agrupamento  $k = \{v_1, \dots, v_{|C_k|}\}$ 
 $\mu_{kj}$  soma das distâncias da mediana  $j$  aos vértices do agrupamento  $k$ 
 $|C_k|$  cardinalidade de  $C_k$ 
Enquanto (solução-inicial melhora) faça
  Para  $k = 1, \dots, p$ 
    Para  $i = 1, \dots, |C_k|$ 
      Troque mediana  $j_k$  por um vértice  $v_i$  se mantiver a capacidade caso contrário
      passe para outro vértice;
      Calcule novo  $\mu_{kj}$ ;
      Se novo  $\mu_{kj}$  for melhor que antigo  $\mu_{kj}$  então
        Atualiza  $\mu_{kj}$ 
        Guarda nova mediana;
      Fim_Se;
    Fim_Para;
  Fim_Para;
  Atualiza  $J$ ;
  Resolve o PGA associado;
  Calcula o valor novasol que corresponde as realocações;
  Se novasol for melhor do que solução-inicial então;
    Faça solução-inicial  $\leftarrow$  novasol;
  Fim_Se;
Fim_Enquanto;

```

Algoritmo *HLA* para o *PMC*

Na figura 4.2 mostramos como a *HLA* procede para um caso do *PMC*. Uma vez construídos os agrupamentos, no caso três agrupamentos, cada um deles possui um custo inicial (a somatória da distância dos vértices não-mediana às medianas) e uma solução inicial (somatória de todos os custos)- parte (a) da figura 4.2. Faz-se a troca do vértice não-mediana pela mediana dentro do agrupamento (1) e calcula-se um novo custo (parte b da figura 4.2) no agrupamento (1). Ou seja dentro do agrupamento (1) procura-se a melhor mediana de modo a minimizar o custo inicial. Guarda-se então esta mediana. Terminado a varredura no agrupamento (1) passa-se para outro agrupamento, e repete-se o procedimento anterior dentro do agrupamento (2). Varrido todos os agrupamentos atualiza-se todas as novas medianas encontradas e a nova solução (parte (c) da figura

4.2), se ela for melhor do que solução inicial, atualiza-a e repete-se o processo inicial até quando não houver mais melhoramentos.

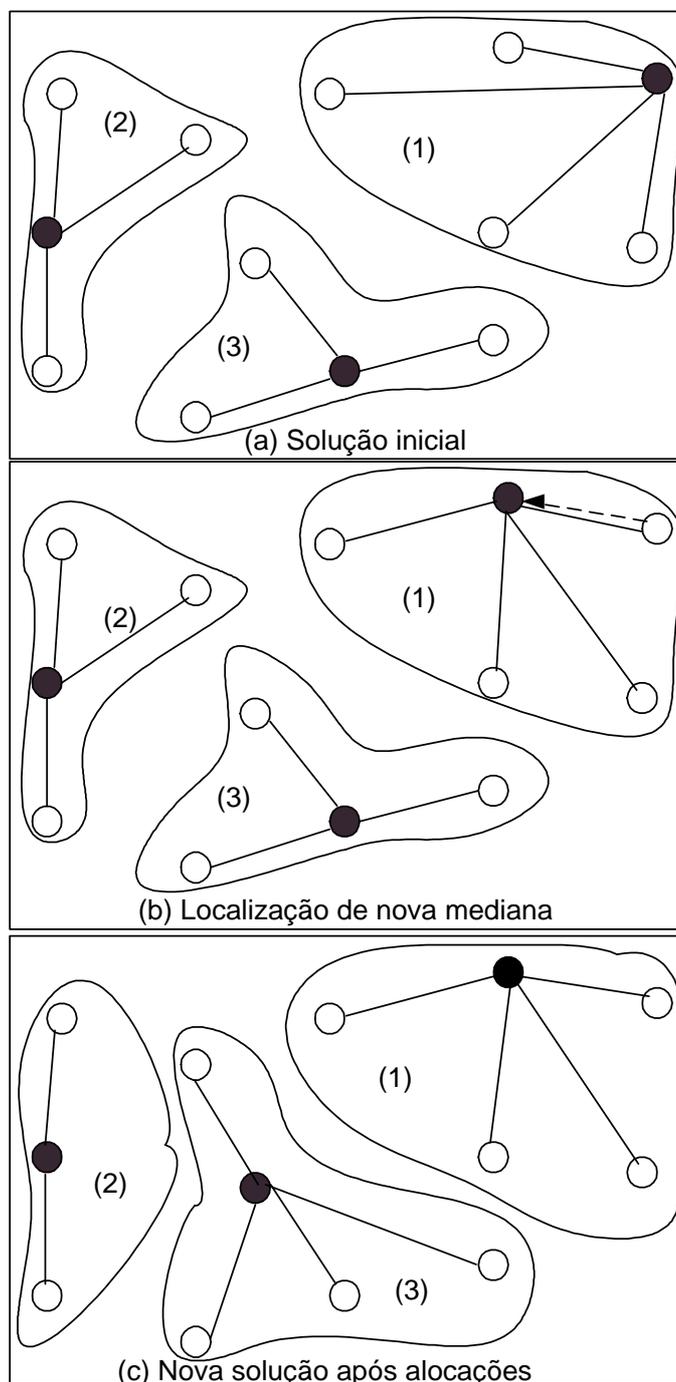


Fig. 4.2 – Exemplo da aplicação da *HLA* para um caso do *PMC*

4.3.3 Testes Computacionais para o PMC

Para testar o *HLA* para o *PMC* foram utilizados as instâncias fornecidas por Osman e Christofides (1994) referenciadas por *pmc1* a *pmc20*. Estes conjuntos de problemas testes foram gerados de forma aleatória sendo que um conjunto possui 10 problemas de dimensão 50 x 5 (50 vértices e 5 medianas) e o outro conjunto possui 10 problemas de dimensão 100 x 10 (100 vértices e 10 medianas). Todos os vértices estão localizados num plano e suas coordenadas foram geradas aleatoriamente dentro de uma distribuição uniforme [1;100] e portanto as distâncias são euclidianas. Os valores de demanda foram gerados a partir de uma distribuição uniforme [1;20]. A capacidade de um dado problema foi obtida através da expressão

$$\text{Capacidade} = \frac{\sum \text{demandas}}{\text{número de medianas}} \times \frac{1}{\tau} \quad (4.5)$$

onde $\tau \in [0, 82; 0, 96]$. E todas as medianas tem capacidade total iguais.

A heurística proposta por Osman e Christofides (1994) para o *PMC* é um híbrido de Busca Tabu (*Tabu Search*) e enrijecimento simulado (*Simulated Annealing*) que implementa um critério de aceitação probabilístico juntamente com programação de esfriamento não-monotônico, uma busca na vizinhança sistemática e um critério de parada que está condicionado a quando a temperatura zera. Os resultados produzidos pela hibridização foram considerados muito bons. A máquina utilizada foi um VAX 8600 e em 85 % dos problemas testes a heurística obteve as melhores soluções conhecidas com uma média para os 20 problemas testes de 180,24 segundos.

Um segundo conjunto de dados reais, fornecidos por Lorena e Senne (2001), foram coletados utilizando um Sistema de Informações Geográficos (*ArcView*) relativo à área central da cidade de São José dos Campos. Seis instâncias foram criadas (100x10), (200x15), (300x25), (300x30), (402x30) e (402x40) referenciadas de *sjc1* a *sjc6* respectivamente. Cada ponto é localizado sobre um bloco que representa uma demanda, que foi estimada considerando o número de casas em cada bloco (um bloco vazio recebe valor 1). A capacidade foi estimada da mesma forma que a anterior, porém τ foi considerado 0,9 ou 0,8.

As heurísticas propostas por Lorena e Senne (2001) para o *PMC* implementam uma busca em *p* agrupamentos em um processo de otimização Lagrangeano/surrogate, utilizando

uma heurística de localização-alocação e uma heurística de troca entre agrupamentos. O enfoque Lagrangeano/surrogate foi capaz de gerar tão boas soluções aproximadas quanto as obtidas por metaheurísticas num tempo computacional menor.

Para cada uma destas instâncias dos dois conjuntos de dados foram feitas cem rodadas. No caso do *PMC* queremos minimizar a distância das entidades (clientes) às medianas mantendo a capacidade do agrupamento. Portanto a média, o mínimo valor encontrado, a frequência do mínimo valor encontrado e o tempo se referem a estas cem rodadas. Há ainda uma coluna que mostra a melhor solução conhecida (não foram encontradas as soluções ótimas) encontrada por Osman e Christofides (1994) (*OC*) na tabela 4.7 e a melhor solução encontrada para os dados de São José dos Campos de Lorena e Senne (2001) (*LS*) na tabela 4.8 (o primeiro dado é o dual e o segundo é o primal que são os limites para a heurística de *LS*) e última linha apresenta a média dos tempos para o grupo de problemas. Na tabela 4.7 a diferença foi calculada da seguinte forma:

$$\text{Diferença} = \frac{\text{Melhor sol. HLA} - \text{Melhor sol. conhecida}}{\text{Melhor sol. conhecida}} \times 100 \quad (4.6)$$

Considerando o tempo de processamento em torno de 2 segundos e a diferença chegando no máximo 5%, concluímos que a *HLA* apresentou bons resultados.

Na tabela 4.8 os dados gerados por *LS* foram realizados numa estação de trabalho SUN ULTRA30 enquanto os dados da *HLA* foram gerados num Pentium II 233MHz e 128 MBytes, a diferença dual foi calculada da seguinte forma:

$$\text{Diferença dual} = \frac{\text{Melhor sol. HLA-solução dual LS}}{\text{solução dual LS}} \times 100 \quad (4.7)$$

A *HLA* mostrou uma diferença em relação aos dados de Lorena e Senne (2001) de no máximo a 5,5%, o desempenho foi considerado bom, considerando a sua simplicidade.

Passaremos a discutir no próximo capítulo a *HLA* como um possível processo de mutação para o Algoritmo Genético Construtivo.

prob.	v	p	Média	Melhor sol. <i>HLA</i>	Melhor sol. conhecida	Dif.	Freq. da melhor sol.(%)	Tempo (s)
pmc1	50	5	769	728	713	2,1	23	2
pmc2	50	5	805	758	740	2,5	21	2
pmc3	50	5	826	768	751	2,2	7	1
pmc4	50	5	679	668	651	2,7	27	1
pmc5	50	5	723	683	664	2,9	71	1
pmc6	50	5	835	797	778	2,4	14	2
pmc7	50	5	864	808	787	2,7	12	2
pmc8	50	5	875	839	820	2,3	7	2
pmc9	50	5	753	733	715	2,5	2	2
pmc10	50	5	887	844	829	1,8	10	1
pmc11	100	10	1139	1038	1006	3,2	2	1
pmc12	100	10	1064	995	966	3,0	1	1
pmc13	100	10	1177	1054	1026	2,7	3	1
pmc14	100	10	1105	1014	982	3,2	1	1
pmc15	100	10	1205	1129	1091	3,5	1	1
pmc16	100	10	1052	990	954	3,8	2	1
pmc17	100	10	1143	1070	1034	3,5	1	1
pmc18	100	10	1153	1073	1043	2,9	1	1
pmc19	100	10	1131	1071	1031	3,9	1	1
pmc20	100	10	1143	1055	1005	5,0	1	1

TABELA 4.7 – Resultados computacionais para os dados de *OC*

prob.	v	p	Média	Melhor sol. <i>HLA</i>	Melhor sol. <i>LS</i>	Dif.dual (%)	Freq. melhor sol. <i>HLA</i> (%)	Tempo (s) <i>HLA</i>	Tempo (s) <i>LS</i>
sjc1	100	10	18989,00	17692,47	17252,12-17288,99	2,55	1	14	68
sjc2	200	15	35917,52	33777,64	33223,66-33395,38	1,67	1	80	2083
sjc3	300	25	50269,28	47313,86	45313,43-45364,30	4,41	1	308	2604
sjc4	300	30	45084,40	42794,49	40634,91-40635,90	5,31	1	344	867
sjc5	402	30	70280,87	64170,73	61842,49-62000,23	3,76	1	647	27717
sjc6	402	40	58690,02	55365,16	52396,54-52641,79	5,67	1	806	4649
média dos tempos								366	6331

TABELA 4.8 – Resultados computacionais para os dados de São José dos Campos

CAPÍTULO 5

ALGORITMO GENÉTICO CONSTRUTIVO

Algoritmos Genéticos (AG) vem sendo considerados poderosas ferramentas para resolver problemas de otimização (Michalewicz, 1996); (Michalewicz, 2000). O embasamento para tais algoritmos são o controle da evolução de uma população estruturada. Holland (1975) deu forma aos AG, juntamente com seus mecanismos de funcionamento (formação de esquemas e a sua propagação de geração em geração).

O *Algoritmo Genético Construtivo (AGC)* foi proposto por Furtado (1998) e aprimorado recentemente por Ribeiro Filho (2000) como uma alternativa ao tradicional enfoque dos AG. Basicamente as diferenças existentes neste novo enfoque são: população dinâmica, dupla função de avaliação, consideração não só das estruturas que representam soluções de um problema mas partes delas chamadas de esquemas e a existência de um parâmetro evolutivo que é utilizado para eliminar os indivíduos ao longo das gerações.

5.1 Modelagem do AGC

Nesta seção descreveremos o enfoque dado pelo AGC. O AGC trabalha com uma população formada por estruturas e esquemas, sendo necessário encontrar uma representação adequada para o problema que se quer resolver.

Também específico para cada problema é a função de avaliação. O problema é formulado com dois objetivos e resolvido aproximadamente através de um processo seletivo.

5.1.1 Representação de uma Estrutura e Esquema

Para o *PLMC* e *PMC* as estruturas e esquemas são representados por um alfabeto composto por três símbolos $\{0,1,\#\}$. Cada cadeia de caracteres é uma solução completa ou parcial para o problema em questão.

Suponha um dado grafo $G=(V,E)$. Normalmente para os problemas apresentados temos n pontos de demanda (vértices) $V = \{1, \dots, n\}$. Para definir a representação, p vértices são eleitos como sementes (medianas ou facilidades), isto é, vértices iniciais dos agrupamentos.

A representação de uma estrutura será uma cadeia de caracteres:

$s_k = (0,0,1,1,0,0,0,0,0,1)$, e a

representação de um esquema será :

$s_k = (0,0,1,1,\#,0,0,\#,1)$,

onde o número 1 indica que é uma semente (mediana ou facilidade) e a sua posição na cadeia de caracteres representa o vértice correspondente. O número 0 indica que o vértice correspondente está ativo no problema e # indica um vértice que está temporariamente fora do problema.

Para cada esquema ou estrutura s_k , exatamente p agrupamentos $C_1(s_k), C_2(s_k), \dots, C_p(s_k)$ são identificados utilizando um algoritmo apropriado para cada problema proposto, considerando que os vértices ativos serão associados a uma semente.

Outro ponto interessante é que o AGC trabalha apenas com uma população formada por esquemas. Apesar de esquemas serem representações incompletas do problema a sua recombinação pode produzir estruturas como uma boa representação da solução do problema.

5.1.2 Problema de Otimização com Dupla Função Objetivo

Seja X o conjunto de todos esquemas que podem ser gerados a partir da representação do alfabeto $\{0, 1, \#\}$ e considere a dupla função de avaliação (*função-fg*) f e g , definido como $f : X \rightarrow \mathbb{R}_+$ e $g : X \rightarrow \mathbb{R}_+$ tal que $f(s_k) \leq g(s_k)$, para todo $s_k \in X$.

O AGC implementa a *função-fg* com dois objetivos:

(*Minimização do intervalo* $\{g(s_k) - f(s_k)\}$)

(*Maximização de* $g(s_k)$)

ou seja, de forma indireta durante o processo evolutivo o AGC persegue estes dois objetivos.

Para alcançar estes objetivos um problema para ser resolvido utilizando AGC é modelado como segue.

Problema de Otimização Bi-objetiva (POB):

$$v(\text{POB}) = \text{Min}\{g(s_k) - f(s_k)\} \quad (5.1)$$

$$\text{Max}\{g(s_k)\} \quad (5.2)$$

$$\text{sujeito a } g(s_k) \geq f(s_k), \forall s_k \in X \quad (5.3)$$

A funções f e g devem ser apropriadamente definidas para representar os objetivos de otimização dos problemas envolvidos.

O primeiro objetivo em *POB* (minimização do intervalo) busca substituir o objetivo de otimização do problema original e o segundo guia o processo de evolução para transformar esquemas em estruturas. Procura-se com a representação de esquemas, que a função de avaliação fg cresça quando o número de sinais $\#$ decresce, e portanto estruturas possuem uma maior função de avaliação fg do que esquemas.

5.1.3 O Processo Evolutivo

O *POB* não é resolvido diretamente pois X não é totalmente conhecido. Alternativamente um processo evolutivo é conduzido de forma a contemplar ambas funções do *POB*. No início do processo dois valores são encontrados. Um número real não negativo $g_{max} > \text{Max}\{g(s_k)\}$ onde $s_k \in X$, que seja um limite superior a $g(s_k)$, para cada $s_k \in X$; e a medida do intervalo dg_{max} , obtido através de g_{max} e utilizando um número real para d entre 0 e 1.

A população inicial sofre evolução controlada por alguns parâmetros. Para o limiar de rejeição é criado o parâmetro $\alpha \geq 0$, onde a expressão:

$$g(s_k) - f(s_k) \geq dg_{max} - \alpha \cdot d[g_{max} - g(s_k)] \quad (5.4)$$

representa uma condição para rejeição de um esquema s_k . O lado direito da expressão 5.4 é o limiar, composto do valor esperado para o intervalo de minimização dg_{max} , e a diferença $[g_{max} - g(s_k)]$. No processo de evolução o parâmetro α está relacionado com o tempo de permanência dos indivíduos na população.

O parâmetro de evolução α começa de 0 e vai aumentando lentamente, em pequenos intervalos ($\Delta\alpha$) de geração a geração. Estes valores (d e $\Delta\alpha$) variam de problema a problema e são determinados de forma empírica. Estes parâmetros são responsáveis pela

dinâmica do tamanho da população que pode ser esvaziada durante o processo.

Ao isolarmos o parâmetro de evolução α na expressão 5.4 temos

$$\alpha \geq \frac{dg_{max} - [g(s_k) - f(s_k)]}{d[g_{max} - g(s_k)]} = \delta(s_k) \quad (5.5)$$

onde o lado direito da desigualdade representa o *rank* associado ao esquema. Os esquemas s_k são removidos da população quando $\alpha \geq \delta(s_k)$. Portanto quanto maior o *rank* mais tempo o esquema terá para sobreviver e recombinar-se.

Para compreendermos a função de d (desvio global permitido) temos, reestruturando a equação 5.5, que:

$$\delta(s_k) = \frac{1 - \frac{g(s_k) - f(s_k)}{dg_{max}}}{1 - \frac{g(s_k)}{g_{max}}} \quad (5.6)$$

Pela equação 5.6 podemos ver que quanto maior d , maior é o *rank* e conseqüentemente a população tenderá a ficar maior. Concluindo, os parâmetros d e $\Delta\alpha$ são importantes para a sobrevivência e permanência da população.

5.1.4 População Inicial

Chamemos de P_0 a população inicial e esta é formada apenas por esquemas e possui para cada problema n indivíduos, onde n é dado pelo número de vértices existente no problema (caso específico para *PLMC* e *PMC*). Cada indivíduo é gerado da seguinte forma:

Se o problema possui n vértices teremos n genes que podem assumir $\{0, 1, \#\}$. Inicialmente são escolhidos p 's símbolos 1 (seriam as medianas ou facilidades) e 20% dos n genes recebem o símbolo 0 e o restante o símbolo #.

5.1.5 Seleção e Recombinação

Cada elemento da população é ordenado de forma não-decrescente numa lista obedecendo a uma chave $\psi(s_k)$

$$\psi(s_k) = \frac{1 + d(s_k)}{\eta(s_k)} \quad (5.7)$$

onde

$$d(s_k) = \frac{g(s_k) - f(s_k)}{g(s_k)}$$

$\eta(s_k)$ é o número de símbolos diferentes de # em s_k .

Esta ordenação como podemos ver privilegia os esquemas mais próximos de estruturas e melhor adaptados, ou seja, maior $\eta(s_k)$ e menor $g(s_k) - f(s_k)$.

Utiliza-se o mesmo processo de recombinação utilizado por Ribeiro Filho (2000) referenciado como *base-guia*. Escolhe-se aleatoriamente um esquema, dentro de um percentual selecionado (os primeiros 30% da população), e chama-o de *base*. E outro esquema é selecionado considerando a população inteira e é chamado de *guia*.

Antes de iniciar o processo de recombinação pega-se o esquema *base* selecionado e transforma-o em uma estrutura (bastando a substituição dos símbolos # pelo símbolo '0'). Esta estrutura sofre uma mutação (processo será apresentado posteriormente) e é comparada com a melhor solução encontrada que é sempre guardada no processo. A complementação do esquema base é realizada porque durante a recombinação embora se mantenha o número de sementes, a sequência de símbolos pode ser alterada. Como o esquema base selecionado a princípio se aproxima de uma boa solução, desta forma não se perde uma possível boa solução.

Com os esquemas *base* e *guia* a recombinação é realizada através do algoritmo abaixo (seguindo esta ordem para que se mantenha o número de sementes), gerando s_{novo} :

Se

$s_{i(base)} = \#$ e $s_{i(guia)} = \#$ então $s_{i(novo)} \leftarrow \#$

$s_{i(base)} = 1$ e $s_{i(guia)} = 1$ então $s_{i(novo)} \leftarrow 1$

$s_{i(base)} = 0$ e $s_{i(guia)} = 0$ então $s_{i(novo)} \leftarrow 0$

$s_{i(base)} = 1$ e $s_{i(guia)} = \#$ então $s_{i(novo)} \leftarrow 1$

$s_{i(base)} = 0$ e $s_{i(guia)} = \#$ então $s_{i(novo)} \leftarrow 0$

$s_{i(base)} = \#$ e $s_{i(guia)} = 0$ então $s_{i(novo)} \leftarrow 0$

$s_{i(base)} = \#$ ou 0 e $s_{i(guia)} = 1$ então

$s_{i(novo)} \leftarrow 1$ e $s_{k(novo)} \leftarrow$ para algum $s_{k(novo)} = 1$

$s_{i(base)} = 1$ e $s_{i(guia)} = 0$ então

$s_{i(novo)} \leftarrow 0$ e $s_{k(novo)} \leftarrow 1$ para algum $s_{k(novo)} = 0$

Caso s_{novo} seja um esquema, ele volta a ser inserido na população, caso contrário, sofre uma mutação e sua solução é novamente comparada com a melhor solução obtida até então.

Na próxima figura 5.1 mostramos o pseudo-código do AGC

```

Dados  $d, \Delta_\alpha, M_{\max}, N_{\text{ind}}$ 
 $\alpha \leftarrow 0$ 
Calcular  $g_{\max}$ 
Inicializar  $P_\alpha$ 
Computar  $\delta(S_i)$  para todo  $S_i \in P_\alpha$ 

Ordenar  $P_\alpha$  com base em  $\delta(S_i)$ 
 $It \leftarrow 0$ 
Inicializar Melhor
Enquanto ( $It < M_{\max}$  e não achou solução satisfatória e população não está vazia)
    Fazer  $N_{\text{ind}}$  vezes
        Selecionar  $S_{\text{base}}$ 
        Instanciar  $S_{\text{base}}$  gerando estrutura  $E$ 
        Mutação em  $E$ 
        Se  $E$  é melhor que Melhor então  $\text{Melhor} \leftarrow E$ 
        Selecionar  $S_{\text{guia}}$ 
        Recombinar  $S_{\text{base}}$  com  $S_{\text{guia}}$  gerando  $S_{\text{novo}}$ 
        Se  $S_{\text{novo}}$  é esquema então
            Calcular  $\delta(S_{\text{novo}})$  e inserir na lista
        Senão
            Mutação em  $S_{\text{novo}}$ 
            Se  $S_{\text{novo}}$  é melhor que Melhor então  $\text{Melhor} \leftarrow S_{\text{novo}}$ 
        Remover todo  $S_i$  tal que  $\alpha \geq \delta(S_i)$ 
         $\alpha \leftarrow \alpha + \Delta_\alpha$ 
         $It \leftarrow It + 1$ 
Exibir Melhor

```

Fig. 5.1 – Pseudo-código do AGC.

FONTE: Ribeiro Filho (2000).

5.1.6 A Mutação

Quando temos problemas de otimização de maior dificuldade, procura-se, nos algoritmos evolutivos, a utilização de uma heurística de busca local aplicada às soluções obtidas por

estes algoritmos.

A utilização de um processo diferenciado de mutação já tem sido realizada por Michalewicz (1996) e Michalewicz (2000) em estrutura binárias para obter melhoramentos nas soluções já encontradas.

Como o *AGC* utiliza heurísticas dependentes do problema como processos de mutação (Ribeiro Filho, 2000) resolvemos utilizar o *HLA* como processo de mutação do *AGC* e com as eventuais adaptações ao *PLMC* e *PMC*.

5.2 PLMC e o AGC

As representações para esquemas e estruturas para o *PLMC* utilizam a representação binária e o curinga #. Para definir a representação, p vértices são eleitos como *sementes* (facilidades) de forma aleatória, estes serão os vértices iniciais que nos agrupamentos irão cobrir os outros vértices que estão situados dentro da distância S . Para cada esquema ou estrutura s_k , exatamente p agrupamentos $C_1(s_k), C_2(s_k), \dots, C_p(s_k)$ são identificados, percorrendo cada cadeia de caracteres de forma que a primeira posição indica o vértice 1, a segunda posição indica o vértice 2 e assim por diante. O símbolo 1 indica que o vértice é uma facilidade, o símbolo 0 indica um vértice que pode ser coberto ou não por alguma facilidade, pois é necessário que ele esteja dentro da distância de serviço S e o símbolo # indica que o vértice no momento está fora do problema.

Na figura 5.2 temos um exemplo de um caso do *PLMC*, com 14 pontos de demanda e duas facilidades. A representação de um esquema para esta configuração seria:

$$s_k=(0,0,\#,0,1,0,\#,0,\#,0,\#,0,1)$$

Repare que por exemplo os vértices 4 e 8 embora ativos (0) não estão sendo cobertos pela facilidade e o vértice 7 embora esteja coberto não está ativo (#).

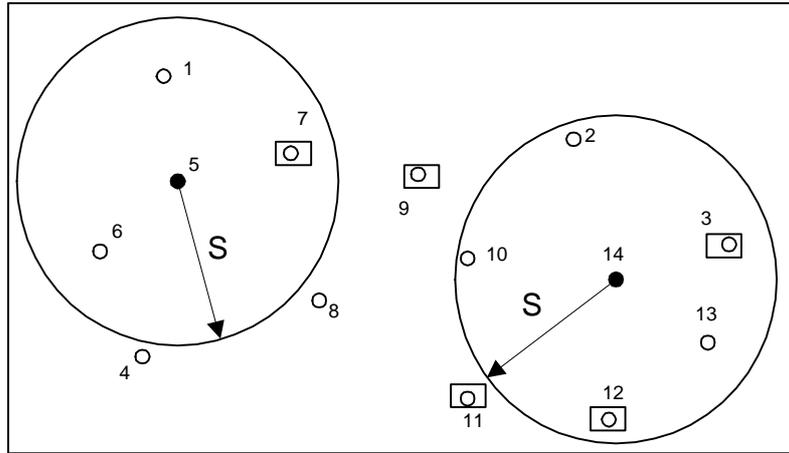


Fig. 5.2 – Representação de $s_k=(0,0,\#,0,1,0,\#,0,\#,0,\#,0,1)$

Os agrupamentos $C_1(s_k), C_2(s_k), \dots, C_p(s_k)$ são necessários para a aplicação da *HLA*.

5.2.1 As Funções de Avaliação

As funções de avaliação f e g e o valor de g_{max} são particularizadas para o *PLMC*. Antes da formação dos agrupamentos $C_1(s_k), C_2(s_k), \dots, C_p(s_k)$ as funções são dadas por:

$$f(s_k) = \sum_{i \in N} D_i y_j \text{ onde } \begin{cases} y_j = 1 & \text{se } d_{ij} \leq S \\ y_j = 0 & \text{se } d_{ij} > S \end{cases} \quad j \in \{\text{vértices sementes}\} \quad (5.8)$$

$$g(s_k) = \sum_{i \in N} D_i \quad (5.9)$$

$$g_{max} = 2 * g(s_i) \quad (5.10)$$

onde D_i é demanda de população da área i ,

$N = \{\text{conjunto de pontos de demanda}\}$,

s_k pode ser uma estrutura ou esquema e as sementes são as facilidades.

A $f(s_k)$ é a própria função objetivo do *PLMC* e a $g(s_k)$ é a soma de todas as demandas

de todos os vértices que já foram atribuídos e g_{max} é o limite superior, que no caso é duas vezes g para uma estrutura s_i gerada aleatoriamente (Esta relação foi utilizada por Furtado (1998) apresentando bons resultados).

5.3 Testes Computacionais

Para os testes computacionais foram utilizadas as mesmas instâncias aplicadas a *HLA* no capítulo 4.

Os resultados são apresentados nas tabelas 5.1, 5.2, 5.3. Para cada instância foram feitas 5 rodadas e os valores encontrados são a média destes resultados. Nas colunas temos o número de vértices, o número de facilidades, a distância de serviço, população atendida, porcentagem de cobertura de *AGC* (média), porcentagem de cobertura da *HLA* (máximo) e os seus respectivos tempos (para o *AGC* a média das 5 rodadas e para a *HLA* o tempo para 100 rodadas) e a última linha apresenta a média dos tempos do *AGC* e *HLA* para o grupo de problemas. Os parâmetros evolutivos utilizados foram $d = 0, 1$ e $\Delta\alpha = 0, 01$.

n	p	S	Pop. atendida	Cobertura <i>AGC</i> (%)	Cobertura <i>HLA</i> (%)	Tempo (s) <i>AGC</i>	Tempo (s) <i>HLA</i>
324	1	800	5461	44,94	44,94	718	6
324	2	800	8790	72,33	72,33	899	15
324	3	800	11604	95,49	95,49	1704	28
324	4	800	12099	99,56	99,62	2643	29
324	5	800	12152	100,00	100,00	3170	24
324	1	1200	9932	81,73	81,73	795	10
324	2	1200	11555	95,09	95,09	2513	21
324	3	1200	12152	100,00	100,00	6219	18
324	1	1600	12123	99,76	99,76	4566	13
324	2	1600	12152	100,00	100,00	1591	10
média dos tempos						2481	17

TABELA 5.1 – Resultados computacionais para os dados de LP324

Na tabela 5.1 vemos que para esta instância (distância de serviço 800, 1200 e 1600) tanto o *AGC* como a *HLA* alcançaram o 100% de cobertura com o mesmo número de facilidades.

Ainda para o caso de 4 facilidades e distância de serviço 800, a *HLA* foi ligeiramente melhor. O tempo da *HLA* chega em alguns casos a ser 400 vezes mais rápido.

<i>n</i>	<i>p</i>	<i>S</i>	Pop. atendida	Cobertura AGC(%)	Cobertura HLA(%)	Tempo (s) AGC	Tempo(s) HLA
402	1	800	6555	41,01	41,01	790	6
402	2	800	11339	70,94	70,94	984	25
402	3	800	14690	91,90	91,90	2271	36
402	4	800	15641	97,85	97,96	3410	44
402	5	800	15955	99,82	99,91	4217	48
402	6	800	15984	100,00	100,00	4815	47
402	1	1200	10607	66,36	66,36	1069	13
402	2	1200	14832	92,79	92,79	2274	36
402	3	1200	15984	100,00	100,00	3318	30
402	1	1600	15438	96,58	96,58	2045	19
402	2	1600	15984	100,00	100,00	2635	23
média dos tempos						2529	30

TABELA 5.2 – Resultados computacionais para os dados de LP402

Na tabela 5.2 vemos para todas as distâncias de serviço 800, 1200 e 1600 m a *HLA* e *AGC* alcançaram o mesmo número de facilidades para 100% de cobertura. Notemos que nos casos intermediários para 800 m (4 e 5 facilidades) a *HLA* consegue ser ligeiramente superior, sem considerar o tempo que chega a ser 100 vezes mais rápido.

n	p	S	Pop. atendida	Cobertura AGC(%)	Cobertura HLA(%)	Tempo (s) AGC	Tempo (s) HLA
500	1	800	7944	40,31	40,31	646	8
500	2	800	12454	63,20	63,20	1651	23
500	3	800	15730	79,82	79,82	2871	41
500	4	800	17782	90,23	90,29	4529	59
500	5	800	18859	95,70	95,70	5784	75
500	6	800	19420	98,54	99,08	6952	94
500	7	800	19663	99,78	99,92	9735	90
500	8	800	19707	100,00	100,00	13424	92
500	1	1200	10726	54,43	54,43	1209	15
500	2	1200	18070	91,69	91,69	3325	42
500	3	1200	19384	98,41	98,41	5125	57
500	4	1200	19707	100,00	100,00	7386	56
500	1	1600	14804	75,12	75,12	2867	21
500	2	1600	19668	99,80	99,80	6521	46
500	3	1600	19707	100,00	100,00	7830	41
média dos tempos						3655	51

TABELA 5.3 – Resultados computacionais para os dados de LP500

Novamente na tabela 5.3 tanto a *HLA* e o *AGC* conseguiram para 100% de cobertura o mesmo número de facilidades. Ainda nas facilidades intermediárias para 800 m (4, 6 e 7 facilidades) *HLA* conseguir superar o *AGC* tendo tempos até 200 vezes menor.

Os dados reportados nas tabelas 5.4 e 5.5 foram comparados com os produzidos por Galvão e Revelle (1996) e Galvão et al. (2000) ou seja o conjunto de testes GR100 e GR150 e a *HLA* (valores máximos obtidos). Para cada instância o *AGC* foi rodado 5 vezes, o valor de cobertura (%) corresponde ao máximo encontrado e tempos são valores médios.

n	p	S	Cobertura <i>AGC</i> (%)	Cobertura <i>GR</i> (%)	Cobertura <i>HLA</i> (%)	Tempo (s) <i>AGC</i>	Tempo (s) <i>GR</i>
100	8	50	66,42	69,43	67,72	93	51,69
100	10	50	72,27	76,23	74,69	129	62,90
100	12	50	75,93	81,61	78,75	165	64,26
100	8	65	83,43	87,36	86,22	149	53,81
100	10	65	92,41	94,33	93,31	222	20,40
100	12	65	96,89	99,18	97,00	332	22,03
100	8	80	87,71	88,46	87,99	153	43,56
100	10	80	93,34	96,21	94,04	231	20,54
100	12	80	96,13	100,00	98,28	354	7,41

TABELA 5.4 – Resultados computacionais para os dados de GR100

Na tabela 5.4 os valores encontrados pelo *AGC* se encontram distânciados de 1% a 7% das soluções conhecidas.

n	p	S	Cob.(%) <i>AGC</i>	Cob. (%) <i>GR</i> (%)	Cob. <i>HLA</i>	Tempo (s) <i>AGC</i>	Tempo (s) <i>GR</i>
150	10	70	68,39	68,86	69,09	387	9
150	12	70	77,18	77,09	76,48	314	11
150	14	70	83,55	83,34	83,81	264	12
150	16	70	88,47	87,75	86,64	387	13
150	18	70	91,83	92,39	90,20	314	12
150	20	70	94,39	93,95	92,90	264	6
150	8	75	58,54	59,14	59,33	387	109,71
150	10	75	67,95	68,86	67,86	314	122,35
150	12	75	75,15	77,34	75,39	264	127,28
150	8	80	61,73	61,49	62,80	222	125,45
150	10	80	68,20	70,91	70,65	212	118,43
150	12	80	75,81	78,20	77,74	400	122,89
150	8	85	73,29	73,94	73,83	463	96,39
150	10	85	81,83	81,56	80,68	356	127,59
150	12	85	85,45	87,95	87,33	755	154,12
150	8	90	87,69	89,79	363	89,53	144,40
150	10	90	92,24	94,04	591	93,37	147,92
150	12	90	96,66	96,93	466	96,02	46,89

TABELA 5.5 – Resultados computacionais para os dados de GR150

Na tabela 5.5 o *AGC* obteve melhores resultados que os de *GR* para a maioria dos valores cuja distância de serviço é igual 70 m, além de obter melhores resultados em alguns casos quando a distância de serviço é igual a 80 e 85 m.

5.4 PMC e o AGC

Para o *PMC*, a representação de uma estrutura ou esquema utiliza o alfabeto descrito nas seções anteriores. Cada indivíduo da população é gerado de forma aleatória de forma a manter o número de medianas e o número de vértices já alocados para cada problema (20% do número de vértices do problema) . Na cadeia de caracteres que representa um indivíduo, sua respectiva posição indica o correspondente vértice. E desta forma o símbolo 1 indica que o vértice é uma mediana, o símbolo 0 indica que o vértice está alocado e o

símbolo # indica que o vértice está no momento fora do problema.

Na figura 5.3 temos um caso do *PMC* com 13 pontos de demanda e devemos localizar 3 medianas com uma capacidade Q_j . A representação de um esquema para esta conformação seria $s_k=(0,1,0,0,1,0,\#, \#, 1, \#, 0, \#, 0)$.

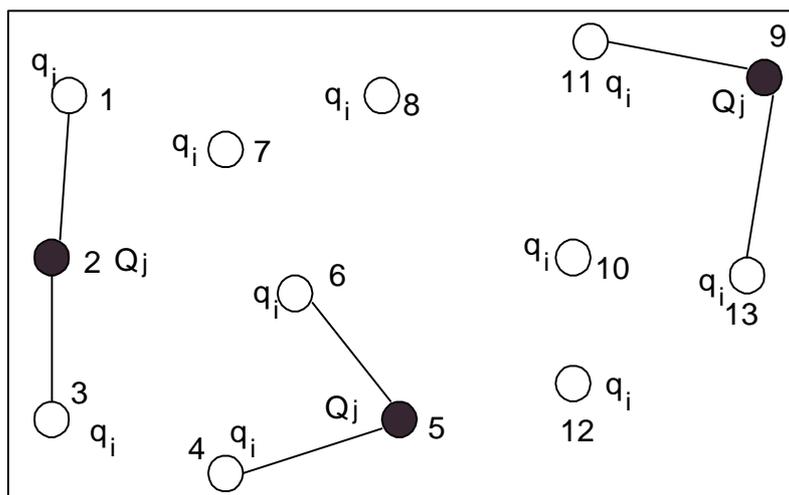


Fig. 5.3 – Representação de $s_k=(0,1,0,0,1,0,\#, \#, 1, \#, 0, \#, 0)$

A heurística que faz a alocação é a mesma utilizado na seção 4.3.1. Novamente, supondo que as medianas estão definidas o *PMC* pode ser resolvido com um enfoque de *PGA*.

5.4.1 As Funções de Avaliação

Diferentemente do *PLMC*, o *PMC* é um problema de minimização. De maneira que as funções serão descritas da seguinte forma:

Para cada estrutura ou esquema são formados p agrupamentos $C_1(s_k), C_2(s_k), \dots, C_p(s_k)$ então teremos

$$f(s_k) = \sum_{j=1}^p (|C_j| - 1) \times \min\{d_{ij}\}, j \in J, i \in C_j \quad (5.11)$$

$$g(s_k) = \sum_{j=1}^p d_{ij} \quad j \in J, i \in C_j \quad (5.12)$$

onde J é o conjunto dos vértices medianas, C_i é o conjunto dos vértices alocados a mediana i e $|C_i|$

A função g_{max} é uma solução completa s_i gerada de forma aleatória que certamente será um limiar superior.

A função $f(s_k)$ é a somatória da menor distância entre um vértice e a mediana vezes o número de arestas do agrupamento, para cada agrupamento.

A função $g(s_k)$ é a própria função objetivo, particularizada quando as medianas já estão definidas.

5.5 Testes Computacionais

Para os testes computacionais foram utilizados as mesmas instâncias aplicadas a *HLA* no capítulo 4, as instâncias de Osman e Christofides (1994) (*OC*) e de Lorena e Senne (2001) (*LS*). Os parâmetros evolutivos foram $d = 0,1$ e $\Delta\alpha = 0,01$. Para cada instância foram realizadas 5 rodadas. Apresentamos na tabela 5.6 a comparação dos resultados obtidos pela *HLA* e *AGC*. A média da *HLA* é sobre 100 rodadas e o tempo corresponde a estas. A média do *AGC* é sobre 5 rodadas para os problemas de *OC* e de *LS* e o tempo é a média destas 5 rodadas.

Na tabela 5.6, os resultados da média do *AGC* foram melhores do que da *HLA*, mas o que surpreende é que as melhores soluções do *AGC* são um pouco melhores do que *HLA*, esperava-se que o *AGC* conseguisse chegar na melhor solução conhecida por *OC*, sendo que a partir das instâncias (100×10) a diferença de tempo chega a ser 270 vezes maior.

Na próxima tabela 5.7 apresentamos os resultados do *AGC* utilizando os dados de Lorena e Senne (2001) (*LS*). As colunas mostram o problema, o número de vértices, o número de medianas, a melhor solução do *AGC* para 5 rodadas, a melhor solução da *HLA*, a melhor solução *LS* (o primeiro dado é o dual e o segundo é o primal que são os limites para a heurística de *LS*), a diferença *LS-AGC*, e os respectivos tempos.

A diferença dual em relação ao *AGC* foi calculada da seguinte forma.

$$\text{Diferença } LS/AGC = \frac{\text{Melhor sol. } AGC\text{-solução dual } LS}{\text{solução dual } LS} \times 100 \quad (5.13)$$

Como podemos ver na tabela 5.7 o *AGC* conseguiu bons resultados em relação as duas primeiras instâncias (*sjc1* e *sjc2*), alcançando o primal de *LS* no conjunto de dados *sjc1* e superando-o no conjunto de dados *sjc2*. Quanto as outras instâncias (*sjc3*, *sjc4* e *sjc5*) o *AGC* ficou obteve uma diferença de no máximo 5,5%.

Para as instâncias de *OC* o *AGC* obteve um desempenho abaixo do esperado e para as instâncias *LS* foi bom para instâncias menores e para as instâncias maiores o tempo computacional se torna bastante oneroso.

prob.	v	p	Média <i>HLA</i>	Média <i>AGC</i>	Melhor sol. <i>HLA</i>	Melhor sol. <i>AGC</i>	Melhor sol. conhecida	Tempo (s) <i>HLA</i>	Tempo (s) <i>AGC</i>
pmc1	50	5	769	728	728	728	713	2	8,7
pmc2	50	5	805	764	758	758	740	2	8,2
pmc3	50	5	826	769	768	767	751	1	8,3
pmc4	50	5	679	668	668	668	651	1	8,5
pmc5	50	5	723	683	683	683	664	1	8,5
pmc6	50	5	835	797	797	796	778	2	8,7
pmc7	50	5	864	810	808	808	787	2	8,5
pmc8	50	5	875	841	839	838	820	2	8,5
pmc9	50	5	753	734	733	732	715	2	8,5
pmc10	50	5	887	843	844	843	829	1	8,6
pmc11	100	10	1139	1064	1038	1038	1006	1	244,1
pmc12	100	10	1064	1003	995	994	966	1	272,5
pmc13	100	10	1177	1082	1054	1053	1026	1	272,5
pmc14	100	10	1105	1037	1014	1013	982	1	272,5
pmc15	100	10	1205	1152	1129	1125	1091	1	274,5
pmc16	100	10	1052	1005	990	987	954	1	271,9
pmc17	100	10	1143	1099	1070	1069	1034	1	260,7
pmc18	100	10	1153	1101	1073	1073	1043	1	259,3
pmc19	100	10	1131	1087	1071	1066	1031	1	259,4
pmc20	100	10	1143	1078	1055	1055	1005	1	272,3

TABELA 5.6 – Resultados computacionais para os dados de *OC*

prob.	v	p	Melhor sol. <i>AGC</i>	Melhor sol. <i>HLA</i>	Melhor sol. <i>LS</i>	Dif. <i>LS/AGC</i> (%)	Tempo (s) <i>AGC</i>	Tempo (s) <i>HLA</i>	Tempo (s) <i>LS</i>
sjc1	100	10	17288,29	17692,47	17252,12-17288,29	0,21	221	14	68
sjc2	200	15	33354.91	33777,64	33223,66-33395,38	0,40	3348	80	2083
sjc3	300	25	47074,12	47313,86	45313,43-45364,30	3,89	18481	308	2604
sjc4	300	30	42849,77	42794,49	40634,91-40635,90	5,45	22614	344	867
sjc5	402	30	64792,51	64170,73	61842,49-62000,23	4,77	26430	647	27717
sjc6	402	40	55281,25	55365,16	52396,54-52641,79	5,51	87448	806	4649

TABELA 5.7 – Resultados computacionais para os dados de São José dos Campos

CAPÍTULO 6

CONCLUSÕES E RECOMENDAÇÕES

A localização de facilidades é um aspecto crítico de planejamento estratégico para um grande espectro de empresas públicas e privadas. Seja um empresário que quer construir um novo shopping, um industrial que deseja situar uma nova fábrica ou um administrador público selecionar locais para estabelecer corpos de bombeiros são frequentemente desafiados pela dificuldade de decidir por um local adequado, pois antes de uma facilidade ser adquirida ou comprada, boas localizações devem ser identificadas, especificações da capacidade da facilidade devem ser determinadas e grande quantidade de capital deve ser alocada. Por isso a determinação das melhores localizações para as novas facilidades é um importante desafio estratégico.

Neste trabalho apresentamos uma heurística de localização-alocação para problemas de localização de facilidades. A Heurística de Localização-Alocação (*HLA*) apresenta-se como um método promissor na busca de soluções a problemas combinatoriais de localização de facilidades, e em particular para o *PLMC* e o *PMC*. Para cada problema específico uma adaptação à *HLA* deve ser feita, trabalhando-se com a formação de agrupamentos (*clusters*). A extrema simplicidade da *HLA* e sua enorme rapidez e eficiência foram comprovados para as instâncias aplicadas. Para um trabalho futuro seria interessante aplicar a *HLA* à outros problemas de agrupamento (*clustering*).

Para o *PLMC*, a *HLA* apresentou os seguintes resultados: para o grupo de problemas *GR100* mostrou-se boa, apresentando uma diferença em relação as melhores soluções conhecidas, de no máximo 3% e um tempo em média dez vezes mais rápido (tendo em conta que as máquinas são diferentes - Pentium II x Digital Alpha 300/30); para o grupo de problemas *GR150* a *HLA* surpreendeu e obteve melhores resultados do que *GR* em quatro instâncias com um tempo em média 5 vezes mais rápido. O que leva a crer que para instâncias pequenas a *HLA* é rápida e eficiente conseguindo se aproximar e algumas vezes superar a heurística Lagrangeada utilizada por *GR* que é mais complexa.

Para dados reais, a comparação da *HLA* com o *AGC* para as instâncias LP324, LP402 e LP500 mostrou que além de encontrar a cobertura total com o mesmo número de facilidades, em alguns casos a *HLA* foi melhor do que o *AGC* (para o mesmo número de facilidades obteve uma maior cobertura). Resultado considerado surpreendente pois se pensarmos que a solução encontrada pela *HLA* começa com a montagem de um

agrupamento aleatório e do *AGC* tenta a partir de uma construção de agrupamentos bons encontrar a solução do problema, além do fato de que a mutação do *AGC* é a própria *HLA*, leva-nos a concluir que talvez as funções f e g e/ou os parâmetros evolutivos foram mal definidos. Seria interessante para trabalhos futuros investir mais tempo nesta análise. Independentemente disto podemos ver que pela média dos tempos para o grupo de problemas (17 (LP324), 30 (LP402) e 51 (LP500) segundos) e pelos seus resultados para este tamanho de instância (até 500 vértices) a *HLA* continua rápida e eficiente.

Para o *PMC* a *HLA* produziu resultados diferentes para os dois grupos de problemas escolhidos. Para as instâncias de *OC* de 50 e 100 vértices e 5 e 10 medianas os resultados foram bons e a diferença em relação a melhor solução conhecida foi de no máximo 5% em um tempo não superior a 2 segundos.

Em recente artigo Schilling et al. (2000) utilizaram uma geração randômica para produzir as distâncias no *PM* e concluíram que estes são geralmente mais difíceis de resolver do que aquelas baseadas em distância euclidiana ou de trajetórias. Como os dados utilizados por *OC* também foram gerados de forma aleatória, embora estejamos trabalhando com o *PMC* leva a intuir que estes problemas também sejam mais difíceis de se resolver. Um estudo comparativo com estas instâncias e com instâncias reais de mesma dimensão utilizando a *HLA* seria um interessante estudo para um trabalho futuro.

Para as instâncias de *LS* que são problemas maiores (100 a 402 vértices e 10 a 40 medianas) a *HLA* mostrou bons resultados apresentando uma diferença de no máximo 6% em relação aos de *LS* porém em uma média de tempos 20 vezes inferior (366 s x 6331 s) ou seja a *HLA* continua sendo rápida e mais simples que a implementação de *LS* que utiliza uma heurística Lagrangeana/surrogate.

A aplicação do *AGC* ao *PMC* nas instâncias de *OC* gerou resultados ligeiramente superiores a *HLA* (máximo 0,4%). No processo de mutação foi utilizada a *HLA*. Novamente uma melhor definição das funções f e g aliada a uma escolha melhor dos parâmetros adaptativos pudessem ter produzidos resultados melhores. Isto poderia ser feito num trabalho futuro.

Para as instâncias de *LS* o *AGC* produziu bons resultados para as instâncias *sjc1* e *sjc2* sendo que esta última apresentou um resultado melhor do que *LS*.

Embora Ribeiro Filho (2000), Furtado (1998) e Oliveira e Lorena (2001) tenham obtido excelentes resultados na aplicação do *AGC*, este trabalho não obteve de modo

geral bons resultados. A utilização da *HLA* como processo de mutação para o *AGC* não apresentou os resultados esperados (houve melhora mas esperava-se uma melhora substancial). Acredita-se que os fatos mencionados anteriormente possam ter prejudicado o desempenho do *AGC*.

Avaliar o grau de melhoramento produzido pela *HLA* nos problemas citados a partir de soluções iniciais aleatórias seria uma avaliação interessante para medir o grau de eficiência da *HLA*.

Naturalmente, não é uma heurística para grandes instâncias pois se tivéssemos um problema com n vértices e m facilidades a serem localizadas teremos $m \times C_{n,m}$ de agrupamentos possíveis e certamente quando o número de vértices cresce juntamente com o número de facilidades a *HLA* começará a ter soluções inferiores.

Concluimos que a *HLA* é bastante eficiente, de simples implementação e rápida, portanto indicada para ser integrada aos *SIG*.

REFERÊNCIAS BIBLIOGRÁFICAS

- Adenso-Diaz, B.; Rodriguez, F. A simple search heuristic for the MCLP: Application to the location of ambulance bases in a rural region. **Omega Int. J. Mgmt Sci.**, v. 25, n. 2, p. 181–187, 1997.
- Assad, E. D.; Sano, E. E. **Sistema de informações geográficas - aplicações na agricultura**. Brasília: Embrapa-SPI/Embrapa-CPAC, 1998. 434 p.
- Baldacci, R.; Hadjiconstantinou, E.; Maniezzo, V.; Mingozzi, A. A new method for solving capacitated location problems based on a set partitioning approach. **Computer and Operations Research**, v. 29, n. 3, p. 365–386, 2002.
- Beasley, J. E. Lagrangean heuristics for location problems. **European Journal of Operational Research**, v. 65, p. 383–399, 1993.
- Christofides, N.; Beasley, J. A tree search algorithm for the p-median problems. **European Journal of Operational Research**, v. 10, p. 196–204, 1982.
- Chung, C. H. Recent application of the maximal covering location planning (MCLP) model. **Journal of the Operational Research Society**, v. 37, n. 8, p. 735–746, 1986.
- Church, R.; Reville, C. The maximal covering location problem. **Papers of the Regional Science Association**, v. 32, p. 101–118, 1974.
- Church, R.; Stoms, D.; Davis, F. W. Reserve selection as a maximal covering location problem. **Biological Conservation**, v. 76, p. 105–112, 1996.
- Cooper, L. Location-allocation problems. **Operations Research**, v. 11, p. 331–343, 1963.
- Current, J.; O’Kelly, M. Locating emergency warnings sirens. **Decision Sciences**, v. 23, n. 1, p. 221–234, 1992.
- Daskin, M.; Jones, P.; Lowe, J. Rationalizing tool selection in a flexible manufacturing system for sheet metal products. **Operations Research**, v. 38, n. 6, p. 1104–1115, 1990.
- Davis, F. W.; Goodchild, M. F. **A spatial modeling and decision support system for conservation of biological diversity**. [programa de computador on line]. <http://www.biogeog.ucsb.edu/projects/ibm/reports/report96.html>, 1996. IBM Environmental Research Program.

- Densham, P.; Rushton, G. A more efficient heuristic for solving large p-median problems. **Papers of the Regional Science Association**, v. 71, p. 307–329, 1992.
- Deutsch, S. J.; Freeman, S. F.; Helander, M. Manufacturing cell formation using an improved p-median model. **Computers and Industrial Engineering**, v. 34, n. 1, p. 135–146, 1998.
- Downs, B.; Camm, J. An exact algorithm for the maximal covering location problem. **Naval Research Logistics**, v. 43, n. 3, p. 435–461, 1996.
- Dwyer, F.; Evans, J. A branch and bound algorithm for the list selection problem in direct mail advertising. **Management Science**, v. 27, n. 6, p. 658–667, 1981.
- Eaton, D.; Hector, M.; Sanchez, V. Determining ambulance deployment in Santo Domingo, Dominican Republic. **Journal of the Operational Research Society**, v. 37, n. 2, p. 113–126, 1986.
- ESRI. **Avenue customization and application development for arcview**. Environmental System Research Institute, Inc., 1996.
- França, P. M.; Sosa, N. M.; Pureza, V. An adaptive tabu search algorithm for the capacitated clustering problem. **Intl. Trans. in op. res.**, v. 1, n. 6, p. 665–678, 1999.
- Furtado, J. C. **Algoritmo genético construtivo na otimização de problemas combinatoriais de agrupamentos**. São José dos Campos. 112 p. Doutorado em Computação Aplicada – Instituto Nacional de Pesquisas Espaciais, 1998.
- Galvão, R. D.; Espejo, L. G. A.; Boffey, B. A comparison of langrangean and surrogate relaxations for the maximal covering location problem. **European Journal of Operational Research**, v. 124, n. 2, p. 377–389, 2000.
- Galvão, R. D.; Raggi, L. A method for solving to optimality uncapacitated location problems. **Annals of Operations Research**, v. 8, p. 225–244, 1989.
- Galvão, R. D.; Reville, C. A lagrangean heuristic for the maximal covering location problem. **European Journal of Operational Research**, v. 88, n. 1, p. 114–123, 1996.
- Garey, M.; Johnson, D. **Computers and intractability: a guide to the theory of NP-completeness**. San Francisco: W.H. Freeman and Co, 1979. 338 p.

- Goberna, M.; Lopez, M. A.; Pastor, J. **Performance and location of bank branches: a methodological approach.** IEF, Bangor, 1990. Research Papers in Banking and Finance - RP 90/15.
- Golden, B.; Skiscim, C. Using simulated annealing to solve routing and location problems. **Naval Research Logistic Quarterly**, v. 33, p. 261–279, 1986.
- Hakimi, S. Optimum location of switching centers and the absolute centers and the medians of a graph. **Operations Research**, v. 12, p. 450–459, 1964.
- . Optimum distribution of switching centers in a communication network and some related graph theoretic problems. **Operations Research**, v. 13, p. 462–475, 1965.
- Hamon, B.; Eaton, D.; Church, R. Development of a multi-purpose ambulance system. **Modelling and Simulation**, v. 10, p. 1437–1445, 1979.
- Holland, J. **Adaptation in natural and artificial systems.** MIT Press, 1975. 11–147 p.
- Houglund, E. S.; Stephens, N. T. Air pollutant monitor siting by analytical techniques. **Journal of Air Pollution Control Association**, v. 26, p. 52–53, 1976.
- Hribar, M.; Daskin, M. A dynamic programming heuristic for the p-median problem. **European Journal of Operational Research**, v. 101, n. 3, p. 499–508, 1997.
- Klose, A. A branch and bound algorithm for an uncapacitated facility location problem with a side constraint. **Int. Trans. Opl. Res.**, v. 5, n. 2, p. 155–168, 1998.
- Lorena, L.; Senne, E. Local search heuristics for capacitated p-median problems. **Networks and Spatial Economics**, 2001. Submetido.
- Maniezzo, V.; Mingozzi, A.; Baldacci, R. A bionomic approach to the capacitated p-median problem. **Journal of Heuristics**, v. 4, n. 3, p. 263–280, 1998.
- Martello, S.; Toth, P. **Knapsack problems, algorithms and computer implementations.** John Wiley, 1990. 296 p.
- Michalewicz, Z. and Fogel, D. **How to solve it: modern heuristics.** Berlin: Springer Verlag, 2000. 467 p.
- Michalewicz, Z. **Genetic algorithms + data structures=evolution programs.** Berlin: Springer Verlag, 1996. 387 p.

- Mulvey, J.; Beck, M. Solving capacitated clustering problems. **European Journal of Operational Research**, v. 18, p. 339–348, 1984.
- Oliveira, A. C. M. O.; Lorena, L. A. N. A constructive genetic algorithm for gate matrix layout problems. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, 2001. Aceito para publicação.
- Osman, I. H.; Christofides, N. Capacitated clustering problems by hybrid simulated annealing and tabu search. **International Transactions in Operational Research**, v. 1, n. 3, p. 317–336, 1994.
- Pirkul, H. Efficient algorithms for the capacitated concentrator location problem. **Computers and Operations Research**, v. 14, n. 3, p. 197–208, 1987.
- Pizzolato, N. D.; Silva, H. B. F. The location of public schools: Evaluation of practical experiences. **Int. Trans. Opl. Res.**, v. 4, n. 1, p. 13–22, 1997.
- Ribeiro Filho, G. **Melhoramentos no algoritmo genético construtivo e novas aplicações em problemas de agrupamento**. São José dos Campos. 121 p. Doutorado em Computação Aplicada – Instituto Nacional de Pesquisas Espaciais, 2000.
- Rolland, E.; Schilling, D.; Current, J. An efficient tabu search procedure for the p-median problem. **European Journal of Operational Research**, v. 96, n. 2, p. 329–342, 1996.
- Schilling, D.; Rosing, K.; ReVelle, C. S. Network distance characteristics that affect computational effort in p-median location problems. **European Journal of Operational Research**, v. 127, n. 3, p. 525–536, 2000.
- Senne, E.; Lorena, L. Lagrangean/surrogate heuristics for p-median problems. In: M. Laguna and J.L. Gonzalez-Velarde (eds.). **Computing Tools for Modeling, Optimization and Simulation: Interfaces in Computer Science and Operations Research**. New York: Kluwer Academic Publishers, 2000. p. 115–130.
- Shilling, D. A.; Jayaraman, V.; Barkhi, R. A review of covering problems in facility location. **Location Science**, v. 1, n. 1, p. 25–55, Aug. 1993.
- Taillard, E. **Heuristic methods for large centroid clustering problems**. , 1996. IDSIA96-96.
- Teitz, M.; Bart, P. Heuristics methods for estimating the generalized vertex median of a weighted graph. **Operations Research**, v. 16, p. 955–961, 1968.

Toregas, C.; Reville, C. Optimal location under time or distance constraints.

Papers of the Regional Science Association, v. 28, 1972.

Woodhouse, S.; Lovett, A.; Dolman, P.; Fuller, R. Using a GIS to select priority areas for conservation. **Computers, Environment and Urban Systems**, v. 24, n. 3, p. 79–93, 2000.

Yeh, A. G.; Chow, M. H. An integrated GIS and location-allocation approach to public facilities planning-an example of open space planning.

Comput., Environ. and Urban Systems, v. 20, n. 4/5, p. 339–350, 1996.