

UNIVERSIDADE CANDIDO MENDES
MESTRADO EM PESQUISA OPERACIONAL E INTELIGÊNCIA COMPUTACIONAL

TARCÍSIO BARROSO MARQUES

**HEURÍSTICAS PARA O PROBLEMA DE LOCALIZAÇÃO/ALOCAÇÃO
DE ANTENAS DE TRANSMISSÃO.**

CAMPOS DOS GOYTACAZES

Abril de 2007

UNIVERSIDADE CANDIDO MENDES
MESTRADO EM PESQUISA OPERACIONAL E INTELIGÊNCIA COMPUTACIONAL

TARCÍSIO BARROSO MARQUES

HEURÍSTICAS PARA O PROBLEMA DE LOCALIZAÇÃO/ALOCAÇÃO DE
ANTENAS DE TRANSMISSÃO.

Dissertação apresentada ao Programa de Pós – Graduação em Pesquisa Operacional e Inteligência Computacional, da Universidade Cândido Mendes – Campos/RJ, como requisito parcial para a obtenção do grau de MESTRE EM PESQUISA OPERACIONAL E INTELIGÊNCIA COMPUTACIONAL. Área de concentração: Otimização combinatória.

Orientador: Prof. José Elias Cláudio Arroyo, D.Sc.
Co-orientador: Prof. Dalessandro Soares Vianna, D.Sc.

CAMPOS DOS GOYTACAZES
Abril de 2007

TARCÍSIO BARROSO MARQUES

HEURÍSTICAS PARA O PROBLEMA DE LOCALIZAÇÃO/ALOCAÇÃO
DE ANTENAS DE TRANSMISSÃO.

Dissertação apresentada ao Programa de Pós – Graduação em Pesquisa Operacional e Inteligência Computacional, da Universidade Candido Mendes – Campos/RJ, como requisito parcial para a obtenção do grau de MESTRE EM PESQUISA OPERACIONAL E INTELIGÊNCIA COMPUTACIONAL. Área de concentração: Otimização combinatória.

Aprovada em _____

BANCA EXAMINADORA

José Elias Cláudio Arroyo, D.Sc.
Universidade Candido Mendes.

Dalessandro Soares Vianna, D.Sc.
Universidade Candido Mendes.

Jacqueline Magalhães Rangel Cortes, D.Sc.
Universidade Estadual do Noroeste Fluminense.

Ricardo Silveira Sousa, D.Sc.
Faculdades Redentor.

CAMPOS DOS GOYTACAZES, RJ
2007

Ao meu pai Edson, que ao longo de seus 85 anos continua sendo um exemplo de educação e perseverança.

À minha mãe Zuleika pelo seu amor incondicional e dedicação a todos da família.

À minha esposa Valquíria, pela força que tem, simplicidade e incentivo aos estudos. Dedico este trabalho também ao meu filho Tarcísio que mudou minha vida completamente trazendo-me muitas alegrias ao nascer no período que desenvolvia a escrita desta dissertação.

AGRADECIMENTOS

A Deus, que sempre esteve ao meu lado possibilitando o término deste trabalho e colocando pessoas importantes para me acompanhar por toda esta jornada.

Ao meu orientador, Prof. José Elias pelos ensinamentos, inteligência e rigidez que possibilitaram o desenvolvimento deste trabalho, de uma forma muito completa.

Ao meu co-orientador, Prof. Dalessandro pela amizade e indiscutível sabedoria em lidar com situações extremas.

A todos os meus queridos irmãos.

“[...] as teorias, em geral, não são capazes de ser estabelecidas ou justificadas; e embora possam ser sustentadas por argumentos críticos, esta sustentação nunca é conclusiva.”

Sir Karl R. Popper

RESUMO

Este trabalho aborda o problema de posicionamento de antenas de telecomunicações (por exemplo, antenas de transmissão de sinais de rádio-difusão, sinais de TV, Internet via rádio, etc) em pontos específicos de uma região. O objetivo é atender a uma maior quantidade de pontos de demanda usando um número mínimo de antenas, minimizando a distancia dos pontos de demanda às antenas mais próximas. São consideradas restrições tais como, alcance de transmissão dos equipamentos e obstáculos interferentes.

Para resolver o problema foram desenvolvidos os algoritmos baseados nas metaheurísticas *GRASP* e Algoritmo Genético. O algoritmo *GRASP* usa a heurística gulosa *ADD* na construção de soluções juntamente com técnicas híbridas, que usam estratégias de recombinação para melhorar a qualidade das soluções obtidas a cada iteração. O Algoritmo Genético usa a população de soluções iniciais geradas de forma aleatória e através do uso de uma heurística construtiva gulosa-aleatorizada. O cruzamento de soluções é baseado na união de soluções seguida de um método guloso (Drop) que remove algumas facilidades melhorando o valor da função objetivo.

O bom desempenho das heurísticas desenvolvidas é verificado em diversas instâncias de problemas de médio e grande porte, comprovando a viabilidade dos métodos propostos.

PALAVRAS-CHAVE: Heurísticas, Metaheurísticas, Localização de facilidades, *GRASP*, Algoritmo Genético, Técnicas híbridas, *ADD*.

ABSTRACT

This work approaches the positioning problem of telecommunication antennas (e.g., transmission antennas of broadcasting signs, internet through radio, etc.) in specific points of an area. The goal is to diminish the large amount of demanding points using a minimum number of antennas, minimizing its distances of the demand points to the closest antennas. Restrictions are then considered such as transmission range of the equipments and interfering obstacles.

To solve the problem the algorithms based on the metaheuristics *GRASP* and Genetic Algorithm were developed. The *GRASP* algorithm uses the greedy heuristic *ADD* in the construction of solutions together with hybrid techniques, that use combination strategies to improve the quality of the solutions obtained to each iteration. The Genetic Algorithm uses the population of initial solutions created at random and through the use of a greedy constructive heuristic. The crossing of solutions is based on the union of solutions followed by a greedy *DROP* method that removes some means (facilities) improving the value of the objective function.

The good performance of developed heuristics is tested in great and medium situation problems, confirming the viability of the proposed methods.

KEYWORDS: Heuristics, Metaheuristics, Location of facilities, *GRASP*, Genetic Algorithm, Hybrid Techniques, *ADD*.

LISTA DE FIGURAS

FIGURA 1:	Propagação de um campo eletromagnético.....	25
FIGURA 2:	Densidade de potência em função da distância.....	27
FIGURA 3:	Teoria dos raios.....	28
FIGURA 4:	Teoria de frente de onda.....	28
FIGURA 5:	Refração por Snell.....	29
FIGURA 6:	Custos x Número de facilidades.....	34
FIGURA 7:	Resultados gráficos obtidos pelo Algoritmo de Acréscimo Guloso (AAG).	36
FIGURA 8:	Resultados gráficos obtidos pela heurística <i>GRASP-ADD</i>	37
FIGURA 9:	Formulação matemática do problema de localização de antenas	41
FIGURA 10:	Exemplo da formação do Conjunto N em relação a um ponto de Demanda	43
FIGURA 11:	Exemplo gráfico de uma melhor solução viável para uma Instância hipotética de problema.....	44
FIGURA 12:	Exemplo gráfico da procura por uma boa solução.....	45
FIGURA 13:	Exemplo gráfico de uma solução viável que não minimiza a distância entre as antenas e os pontos de demanda.....	46
FIGURA 14:	Obstáculo em sua forma real aproximado para o formato de um paralelepípedo.....	47
FIGURA 15:	Análise da interferência pelas coordenadas x e y	48
FIGURA 16:	Visada direta entre uma antena e um ponto de demanda, passando por cima de um obstáculo.....	49
FIGURA 17:	Pseudocódigo do procedimento que detecta a interferência considerando as alturas envolvidas.....	50
FIGURA 18:	Visão superior de uma maquete montada para o problema de alocação-localização de facilidades.....	51
FIGURA 19:	Instância da maquete resolvida pela heurística <i>GRASP-ADD</i>	52
FIGURA 20:	Pseudocódigo para o cálculo da matriz de custos.....	53
FIGURA 21:	Pseudocódigo da metaheurística <i>GRASP</i>	55

FIGURA 22:	Pseudocódigo da fase de construção da <i>GRASP</i>	56
FIGURA 23:	Algoritmo básico de busca local <i>GRASP</i>	56
FIGURA 24:	Exemplo de uma roleta de seleção do Algoritmo Genético	60
FIGURA 25:	Representação gráfica do Algoritmo Genético.....	61
FIGURA 26:	Pseudocódigo da fase de construção <i>GRASP-ADD</i>	66
FIGURA 27:	Lista de candidatos e lista de candidatos restrita da <i>GRASP</i>	66
FIGURA 28:	Evolução gráfica da fase de construção da <i>GRASP-ADD</i>	67
FIGURA 29:	Solução gráfica já melhorada pela fase de busca local para uma instância hipotética de problema.....	68
FIGURA 30:	Pseudocódigo da <i>GRASP-ADD</i>	69
FIGURA 31:	Pseudocódigo da busca local que adiciona duas facilidades e remove uma (<i>Vizinhança1</i>).....	70
FIGURA 32:	Pseudocódigo da busca local que remove duas facilidades e adiciona uma (<i>Vizinhança2</i>).....	71
FIGURA 33:	Pseudocódigo da busca local <i>Vizinhança3</i>	72
FIGURA 34:	Comparação gráfica entre a fase de construção <i>ADD</i> e a fase de busca local.....	72
FIGURA 35:	Comparativo entre as soluções X, Y e Z.....	75
FIGURA 36:	Pseudocódigo do procedimento União-DROP da <i>GRASP-ADD-HÍBRIDA</i> ..	76
FIGURA 37:	Pseudocódigo da heurística <i>GRASP-ADD- HÍBRIDA</i>	77
FIGURA 38:	Comparativo para diversos valores de α da <i>GRASP</i>	81
FIGURA 39:	Tempo médio computacional gasto pelas implementações <i>GRASP</i> para instâncias de problemas do conjunto1 entre 100_100 e 1000_1000	84
FIGURA 40:	Tempo médio computacional gasto pelas implementações <i>GRASP</i> para instâncias de problemas do conjunto1 entre 100_100 e 1000_1000.....	85
FIGURA 41:	Tamanho da população de um A.G. em função de n para valores de p	87
FIGURA 42:	Tamanho da população de um A.G. em função de p para valores de n	88
FIGURA 43:	Pseudocódigo de um algoritmo construtivo que gera uma população inicial para o A.G.....	91
FIGURA 44:	Pseudocódigo que gera uma população inicial de forma aleatória.....	93

FIGURA 45:	Exemplificação da seleção e cruzamento dos cromossomos de um A.G..	94
FIGURA 46:	Pseudocódigo do procedimento <i>DROP</i> aplicado ao cromossomo filho do Algoritmo Genético.....	95
FIGURA 47:	Pseudocódigo do A.G., adaptado ao problema.....	96
FIGURA 48:	Tempo computacional médio do A.G. para as instâncias de problemas do conjunto 1.....	99
FIGURA 49:	A.G. com população construtiva x <i>GRASP-ADD-HÍBRIDA</i> nas instâncias de problemas do conjunto 1.....	103
FIGURA 50:	Tempo computacional médio de todas as metaheurísticas para as instâncias de problemas do conjunto 1.....	103
FIGURA 51:	Gráfico comparativo para a instância 3 do conjunto 1 de problemas	106
FIGURA 52:	Gráfico comparativo para a instância 5 do conjunto 1 de problemas.....	106
FIGURA 53:	Gráfico comparativo para a instância 7 do conjunto 1 de problemas.....	107
FIGURA 54:	Gráfico comparativo para a instância 10 do conjunto 1 de problemas.....	107
FIGURA 55:	Gráfico comparativo para a instância 13 do conjunto 1 de problemas.....	108
FIGURA 56:	Gráfico comparativo para a instância 16 do conjunto 1 de problemas.....	108
FIGURA 57:	Gráfico comparativo para a instância 20 do conjunto 1 de problemas.....	109
FIGURA 58:	Passagem do sinal por cima dos obstáculos na instância de problema 1 do conjunto 2.....	110
FIGURA 59:	Passagem do sinal por cima dos obstáculos na instância de problema 20 do conjunto 2.....	110

LISTA DE TABELAS

TABELA 1:	Tipos de transmissões eletromagnéticas.....	23
TABELA 2:	Valores médios da função objetivo em relação ao parâmetro α (conjunto1)	80
TABELA 3:	Valores médios da função objetivo em relação ao parâmetro α (conjunto2)	81
TABELA 4:	Testes computacionais realizados com o uso da heurística <i>GRASP</i> nas instâncias de problemas de médio porte do conjunto 1.....	82
TABELA 5:	Testes computacionais realizados com o uso da heurística <i>GRASP</i> nas instâncias de problemas de grande porte do conjunto 1.....	83
TABELA 6:	Testes computacionais realizados com o uso da heurística <i>GRASP</i> nas instâncias de problemas de médio porte do conjunto 2	84
TABELA 7:	Testes computacionais realizados com o uso da heurística <i>GRASP</i> nas instâncias de problemas de grande porte do conjunto 2.....	85
TABELA 8:	Valores da função objetivo em relação ao número estimado de facilidades nas instâncias de problemas de médio porte do conjunto 1.....	90
TABELA 9:	Valores da função objetivo em relação ao número estimado de facilidades nas instâncias de problemas de grande porte do conjunto 1.....	90
TABELA 10:	Número de vezes (nv) que f_{estim} apresentou melhores resultados.....	90
TABELA 11:	Valores de q em função do número de locais candidatos (m) e do número estimado de facilidades (f_{estim}).....	90
TABELA 12:	Resultados obtidos para diferentes quantidades de genes dos cromossomos da população inicial gerada de forma aleatória.....	92
TABELA 13:	Número médio de vezes que a função objetivo foi avaliada nas instâncias de problemas dos conjuntos 1 e 2.....	98
TABELA 14:	Testes computacionais realizados com o uso do Algoritmo Genético nas instâncias de problemas de médio porte do conjunto 1.....	98
TABELA 15:	Testes computacionais realizados com o uso do Algoritmo Genético nas instâncias de problemas de grande porte do conjunto 1.....	98
TABELA 16:	Testes computacionais realizados com o uso do Algoritmo Genético nas instâncias de problemas de médio porte do conjunto 2.....	100
TABELA 17:	Testes computacionais realizados com o uso do Algoritmo Genético nas instâncias de problemas de grande porte do conjunto 2.....	100
TABELA 18:	Valores da função objetivo e tempo computacional requerido em instâncias de problemas de médio porte do conjunto 1.....	102

TABELA 19: Valores da função objetivo e tempo computacional requerido em instâncias de problemas de grande porte do conjunto 1.....	102
TABELA 20: Valores da função objetivo e tempo computacional requerido em instâncias de problemas de médio porte do conjunto 2.....	104
TABELA 21: Valores da função objetivo e tempo computacional requerido em instâncias de problemas de grande porte do conjunto 2.....	104

SUMÁRIO

1 INTRODUÇÃO.....	17
1.1 QUESTÃO HISTÓRICA.....	18
1.2 MOTIVAÇÃO.....	19
1.3 OBJETIVO DA DISSERTAÇÃO.....	20
2 SISTEMAS DE TELECOMUNICAÇÕES.....	22
2.1 CARACTERÍSTICAS DOS SISTEMAS DE TELECOMUNICAÇÕES.....	22
2.1.1 Propagação de ondas eletromagnéticas.....	24
2.1.2 Teoria dos Raios.....	25
2.1.3 Obstáculos interferentes às ondas de visada direta.....	27
3 ALOCAÇÃO-LOCALIZAÇÃO DE FACILIDADES.....	30
3.1 MODELOS DE ALOCAÇÃO-LOCALIZAÇÃO.....	30
3.2 PROBLEMA DE LOCALIZAÇÃO DE MÁXIMA COBERTURA (PLMC).....	32
3.2.1 Surgimento do <i>PLMC</i>	33
3.2.2 Os custos relativos à abertura de novas facilidades.....	33
3.2.3 Aplicações do <i>PLMC</i>	34
3.2.4 Métodos de solução heurística do <i>PLMC</i>	36
3.2.6 Métodos exatos de solução do <i>PLMC</i>	37
3.3 O PROBLEMA DAS <i>p</i> -medianas.....	37
3.3.1 O problema das <i>p</i> -medianas capacitado (<i>PMC</i>).....	39
4 O PROBLEMA DE ALOCAÇÃO-LOCALIZAÇÃO DE ANTENAS.....	40
4.1 O CRITÉRIO PARA ESCOLHA DE LOCAIS CANDIDATOS.....	40
4.2 DEFINIÇÃO DO PROBLEMA.....	41
4.2.1 Definição dos obstáculos.....	46
4.2.1.1 Cálculo da interferência causada pelos obstáculos.....	47
4.3 CÁLCULO DA MATRIZ DE CUSTOS.....	52
5 METAHEURÍSTICAS UTILIZADAS.....	54
5.1 METAHEURÍSTICA <i>GRASP</i>	54

5.2 ALGORITMOS GENÉTICOS.....	57
5.2.1 A função de avaliação.....	59
5.2.2 O processo de seleção.....	59
5.2.3 O processo de cruzamento e mutação.....	60
5.2.4 Parâmetros que influenciam no comportamento dos Algoritmos Genéticos	61
5.2.5 Exemplos de aplicabilidade dos A.G's. em problemas de otimização.....	62
6 METAHEURÍSTICA GRASP APLICADA AO PROBLEMA DE ALOCAÇÃO- LOCALIZAÇÃO DE ANTENAS	64
6.1 INTRODUÇÃO.....	64
6.2 A FASE DE CONSTRUÇÃO DA GRASP-ADD.....	65
6.3 FASE DE BUSCA LOCAL.....	68
6.4 A GRASP HÍBRIDA.....	73
6.4.1 A GRASP-ADD-HÍBRIDA.....	74
6.5 CONSIDERAÇÕES SOBRE A GRASP.....	77
7 TESTES COMPUTACIONAIS COM O USO DA GRASP.....	79
7.1 ANÁLISE DO PARÂMETRO α	80
7.2 COMPARAÇÕES ENTRE AS IMPLEMENTAÇÕES GRASP.....	82
7.2.1 Testes computacionais no conjunto 1 de problemas.....	82
7.2.2 Testes computacionais no conjunto 2 de problemas.....	84
8 ALGORITMO GENÉTICO ADAPTADO AO PROBLEMA DE ALOCAÇÃO- LOCALIZAÇÃO DE ANTENAS	86
8.1 CONSTRUÇÃO DA POPULAÇÃO INICIAL.....	87
8.1.1 Geração da população inicial com a aplicação de um algoritmo construtivo.....	88
8.1.2 Geração da população inicial de forma aleatória.....	92
8.2 SELEÇÃO E CRUZAMENTO.....	93
8.3 ATUALIZAÇÃO DA POPULAÇÃO.....	95
8.4 CRITÉRIO DE PARADA.....	96
9 TESTES E ANÁLISE DOS RESULTADOS COM O USO DO ALGORITMO GENÉTICO	97
9.1 TESTES COMPUTACIONAIS.....	97

10 COMPARAÇÃO E ANÁLISE DOS RESULTADOS ENTRE A GRASP E O ALGORITMO GENÉTICO.....	101
10.1 A GRASP VERSUS ALGORITMO GENÉTICO.....	101
10.1.2 Comparações gráficas.....	105
11 CONCLUSÃO.....	111
12 REFERÊNCIAS BIBLIOGRÁFICAS.....	113

1 INTRODUÇÃO

Problemas de alocação-localização de antenas de transmissão ocorrem com bastante frequência nos diversos segmentos das telecomunicações, dentre os quais pode-se citar: provedores e acesso à internet, redes de computadores sem fio, sistemas de transmissão de sinais de rádio e televisão, etc. Geralmente a correta alocação-localização dos equipamentos de telecomunicações (rádios-transmissores, torres metálicas, antenas, etc.) é considerado de vital importância para o correto atendimento a uma demanda, proporcionando uma considerável redução nos custos.

Neste Capítulo é feita uma introdução ao problema de alocação-localização de antenas de transmissão em cidades, sendo apresentados os problemas que este trabalho propõe-se a solucionar e as abordagens utilizadas para tal empreitada.

O restante deste trabalho está organizado da seguinte forma: no Capítulo 2 são apresentadas as características dos sistemas de telecomunicações como frequências de operação, propagação de ondas e obstáculos interferentes. Na sequência, o Capítulo 3 apresenta os modelos de alocação-localização de facilidades comumente encontrados na literatura, como o modelo de máximo recobrimento (*maximum covering*), o problema de localização de máxima cobertura (*PLMC*) e o problema das p -medianas. A modelagem matemática é apresentada no Capítulo 4 com formulação da função objetivo e definição das restrições de uso. As metaheurísticas *GRASP* e Algoritmo Genético são apresentadas no Capítulo 5, seguidas pelo Capítulo 6, que adapta a *GRASP* ao problema de alocação-localização de antenas usando as técnicas definidas como: *GRASP-ADD* e *GRASP-ADD-HÍBRIDA*, cuja eficiência pode ser comprovada nos testes computacionais mostrados no Capítulo 7. O Algoritmo Genético também está

adaptado ao problema, com populações criadas de forma aleatória e de forma construtiva, sendo mostrado no Capítulo 8. Os testes computacionais do Algoritmo Genético estão apresentados no Capítulo 9. A eficiência das metaheurísticas *GRASP* e Algoritmo Genético são comparadas no Capítulo 10 através dos resultados obtidos em 42 problemas testes. Finalmente, apresenta-se as conclusões e propostas para trabalhos futuros no Capítulo 11.

1.1 QUESTÃO HISTÓRICA

Desde os primórdios da humanidade o homem tem lutado para desenvolver meios de comunicações, cada vez mais sofisticados, na busca de tornar as distâncias cada vez menores. Na pré-história, as comunicações referiam-se a perigos eminentes, busca de caça, etc. Com o advento da escrita, o homem passou comunicar-se por mensagens inscritas em pedras, que eram transportadas pelos primeiros mensageiros. Mais tarde o homem descobriu que codificando as mensagens por sinais visuais ou sonoros poderia aumentar a velocidade da comunicação: o uso de tambores e fogueiras datam desta época.

Os grandes conquistadores Ciro e Dário da Pérsia, Alexandre da Macedônia e Júlio César de Roma, tendo que administrar seus imensos territórios conquistados, foram obrigados a estabelecer um sistema de mensageiros, em que o tempo de transporte da mensagem podia significar vitória ou derrota nas batalhas, (FERRARI, 1991, p. 1) .

As telecomunicações diferenciam-se destes processos visuais pelo uso de sinais processados eletricamente no transporte das informações.

Desde o invento do telégrafo em 1844 por Samuel Morse, até o uso atual dos satélites de comunicação, desenrolou-se uma fantástica sucessão de inventos e aperfeiçoamentos que propiciaram novas modalidades de telecomunicações.

Nos dias de hoje pode-se notar uma franca expansão das telecomunicações em todo o mundo. Sinais de TV, rádio-difusão, telefonia móvel celular, internet a rádio são apenas alguns exemplos do uso das telecomunicações. Flexibilidade, mobilidade, rapidez e redução dos custos, explicam a crescente expansão deste setor.

1.2 MOTIVAÇÃO

Telecomunicação quer dizer transmissão de informação à distância. Para que esta informação seja enviada de sua origem ao destino é necessário que ela seja modulada por um equipamento rádio transmissor e enviada a uma antena que irradia o sinal de telecomunicações. A partir do momento que o sinal é entregue pela antena transmissora ao espaço, ele sofre muitas degradações à medida que afasta-se do local de transmissão (SMIT, 1988, p. 19). Obstáculos densos como montanhas, muito comuns em terrenos com topologia irregular, também degradam o sinal e em vários casos provocam o surgimento das zonas de silêncio (SMIT, 1988, p. 22). Por este motivo, em um sistema de telecomunicações, o melhor caso ocorre quando há uma linha de visada direta entre uma antena que está transmitindo um sinal e outra que está recebendo-o, e esta linha de visada deverá ser a menor possível, de forma que ambas as antenas fiquem próximas.

Prover uma região com serviços de telecomunicações é um problema difícil de ser resolvido, pelo menos otimamente, por ser de natureza combinatória e por envolver uma série de fatores como um grande número de pontos de demanda, topologia irregular com a presença de obstáculos que obstruem a passagem do sinal de rádio, diversos locais candidatos para instalação de facilidades, dentre outros.

A instalação de antenas de transmissão deve levar em consideração a demanda (clientes a serem atendidos), topologia, os possíveis locais de instalação das antenas, o tipo de antena usada, a frequência do sinal empregada, as potências de transmissão dentre outros fatores. Resumidamente, objetiva-se atender a um maior número de clientes, com um mínimo custo (investimento em equipamentos).

Em algumas cidades, por exemplo, pode-se facilmente observar diversos bairros que simplesmente não conseguem captar os sinais de TV, provindos de uma torre de transmissão. Olhando diversas casas, podemos facilmente observar o elevado número de antenas parabólicas para captarem estes sinais, que não chegam pelas antenas de transmissão. Isto se deve não somente à presença de obstáculos, mas também à má distribuição das antenas de transmissão.

A alocação-localização de antenas para sinais de TV é apenas um exemplo em um universo de várias aplicações. Redes de computadores sem fio, sinais de rádio-difusão de pequena potência, comunicação entre radiopatrulhas, são outros

bons exemplos dos benefícios trazidos por uma correta alocação-localização das antenas de transmissão.

1.3 OBJETIVO DA DISSERTAÇÃO

Este trabalho propõe encontrar uma boa solução em um baixo tempo computacional para o problema de localização de facilidades em uma dada região. As facilidades são consideradas como sendo todos os equipamentos e serviços necessários para a implementação de um sistema de telecomunicações. Isto inclui, por exemplo, equipamentos rádios-transmissores, torres metálicas, antenas, mão-de-obra especializada, dentre outros, representando um alto custo de implementação sendo interessante a redução de seu emprego ao máximo.

Um ponto de demanda poderá ser considerado como sendo uma quadra, quarteirão ou bairro a ser atendido. O objetivo é atender a uma maior quantidade de pontos de demanda usando um número mínimo de antenas. Os algoritmos propostos procuram aproximar as facilidades dos pontos de demanda evitando os obstáculos que obstruem a passagem do sinal de telecomunicações conforme descrito por SMIT (1986, p. 10). Em suma, os principais objetivos deste trabalho são:

- Atender a uma maior quantidade possível de pontos de demanda.
- Minimizar o número de facilidades utilizadas.
- Reduzir a distância entre um ponto de demanda e a facilidade que o atende.
- Considerar os obstáculos que interfiram na passagem do sinal.

Como propostas para solução do problema de alocação-localização de antenas de transmissão, foram implementados os algoritmos baseados nas seguintes metaheurísticas:

- *GRASP (Greedy Randomized Adaptive Search Procedure)*, proposto por FEO & RESENDE (1989), estando adaptada ao problema em duas versões diferentes, de agora em diante chamadas de *GRASP-ADD* e *GRASP-ADD-HÍBRIDA*.

- ALGORITMO GENÉTICO, proposto por HOLLAND (1975), também adaptado ao problema com duas versões diferentes. A primeira utilizando-se de populações geradas aleatoriamente e a segunda fazendo o uso de boas populações, geradas previamente por um processo construtivo.

2 SISTEMAS DE TELECOMUNICAÇÕES

Apresenta-se neste Capítulo, as principais características dos sistemas de telecomunicações, sendo abordadas as frequências de operação, propagação das ondas eletromagnéticas, interferências causadas por obstáculos e a qualidade do sinal recebido, de forma a definir a aplicabilidade do modelo proposto.

2.1 CARACTERÍSTICAS DOS SISTEMAS DE TELECOMUNICAÇÕES

Existem diversas formas de estabelecermos comunicações por ondas eletromagnéticas entre dois pontos distintos. A qualidade do sinal recebido, alcance e o tipo de informação a ser transmitida definem basicamente o sistema que deverá ser utilizado. Uma transmissão em ondas curtas de rádio-difusão é ideal para o envio de sinais de voz a milhares de quilômetros, mas é inaplicável na transmissão de informações de uma rede *wireless*, usada no envio de dados entre computadores.

Muitos fatores influenciam no alcance e qualidade do sistema de telecomunicações. Os principais referem-se às frequências empregadas, potência do sinal irradiado e ganhos das antenas transmissoras e receptoras (HOFFMAN & GÓMEZ, 2003, p. 3). A Tabela 1 faz um comparativo entre as frequências comumente utilizadas, mostrando o alcance e aplicabilidade de cada uma.

Tabela 1 - Tipos de transmissões eletromagnéticas.

Comprimento de onda	Frequência	Modo de propagação usual	Alcance	Uso
Kilométricas. <i>VL</i> F (Very Low Frequency.) <i>L</i> F (Low Frequency).	10 kHz a 500 KHz.	Entre o solo e a ionosfera.	Centenas de Km.	Comunicação marítima (rádio-farol).
Hectométricas. <i>M</i> F (Médium Frequency). Ondas médias	500 KHz a 3 MHz.	De dia vinculada ao solo. De noite pela reflexão ionosférica.	Até 500 Km.	Rádio-difusão. Rádio-farol.
Decamétricas. <i>H</i> F (High Frequency) Ondas curtas.	3 MHz a 30 MHz.	Por reflexão ionosférica principalmente à noite.	Milhares de Km.	Rádio-difusão, comunicação.
Métricas. <i>V</i> H <i>F</i> (Very High Frequency).	30 MHz a 300 MHz.	Direta até o horizonte e além, por espalhamento e por cabos.	Entre 0,1 e 400 Km, dependendo do meio e equipamentos.	Rádio-difusão FM. TV.
Decimétricas. <i>U</i> H <i>F</i> (Ultra High Frequency).	300 MHz a 3 GHz.	Direta até o horizonte e além, por espalhamento e por cabos.	Entre 0,1 e 400 Km, dependendo do meio e equipamentos.	Rádio-difusão FM. TV. Internet's a rádio.
Centimétricas Micro-ondas	3 GHz. a 30 GHz.	Direta e em guia de ondas.	De 0,1 Km a um alcance ilimitado, quando do uso de satélites	Satélites. Telefonia em visada direta.
Milimétricas	30 GHz. a 300 GHz.	Direta e em guia de ondas.	Alcance ilimitado, quando do uso de satélites.	Rádio astronomia.
Micrométricas ou óticas	0,1 μ m a 1 mm.	Fibras ópticas	Variável, dependendo das características da fibra.	Comunicações em faixas amplíssimas.

Fonte: SMIT, 1986, p. 24

Segundo SMIT (1988, p. 25) de acordo com a faixa de frequência transmitida, pode-se dividir os tipos de propagação em três grandes grupos, descritos a seguir:

- Ondas terrestres: onde a superfície da terra comporta-se como um condutor para a onda eletromagnética.
- Ondas espaciais: onde o princípio da propagação encontra-se na reflexão da onda nas camadas ionosféricas.
- Ondas em visada direta: onde a propagação dá-se como um fecho de luz, apenas em linha reta, sujeita aos fenômenos de reflexão, difração e absorção por obstáculos.

Transmissões de frequências altas, principalmente situadas na faixa de *VHF*, *UHF* e microondas, dão-se preferencialmente através de visada direta, onde a

antena transmissora e a antena receptora devem encontrar-se na mesma linha de horizonte, sem obstáculos se interpondo entre elas para uma boa inteligibilidade do sinal recebido. Em muitos casos, tais transmissões aproveitam os fenômenos da reflexão e difração da onda transmitida em obstáculos para realizar a recepção dos sinais. Estes efeitos, embora possibilitem a recepção sem a visada direta, degradam o sinal. Conclui-se que, idealmente, deveria haver uma visada direta entre uma facilidade e um ponto de demanda.

Este trabalho aplica-se ao grupo das ondas em visada direta. As heurísticas propostas procuram posicionar as antenas evitando os obstáculos densos que se interponham entre a transmissão e a recepção, não sendo analisados fenômenos de reflexão e difração em obstáculos. O problema é modelado considerando ainda os seguintes pontos:

- Os equipamentos utilizados (rádios-transmissores e antenas) possuem um máximo alcance em visada direta devendo ser considerado.
- A qualidade do sinal é melhor à medida que a distância entre as antenas transmissora e receptora diminui.
- As antenas utilizadas são onidirecionais, irradiando o sinal igualmente em todas as direções.
- Os morros e montanhas presentes em uma cidade são considerados interferentes por absorverem grande quantidade do sinal transmitido, provocando o surgimento de *zonas de silêncio*.
- As edificações em alvenaria não provocam interferências consideráveis no sinal transmitido, não sendo consideradas como obstáculos.

2.1.1 Propagação de ondas eletromagnéticas

As antenas são utilizadas para transferir energia de um circuito (transmissor) ao espaço, e do espaço para outro circuito (receptor). Uma carga elétrica em movimento devido a uma diferença de potencial, provoca um campo elétrico variável nas suas vizinhanças (HOFFMAN & GÓMEZ, 2003, p. 3).

Uma antena quando submetida a cargas elétricas em movimento provoca o surgimento de uma corrente elétrica. A física por sua vez afirma que todo condutor

percorrido por corrente elétrica cria ao seu redor um campo magnético perpendicular ao campo elétrico (Figura 1). O campo magnético induz perfeitamente uma corrente elétrica em um condutor (ponto de demanda) separado fisicamente daquele causador do campo (facilidade). Isto é a propagação de um campo eletromagnético variável no espaço.

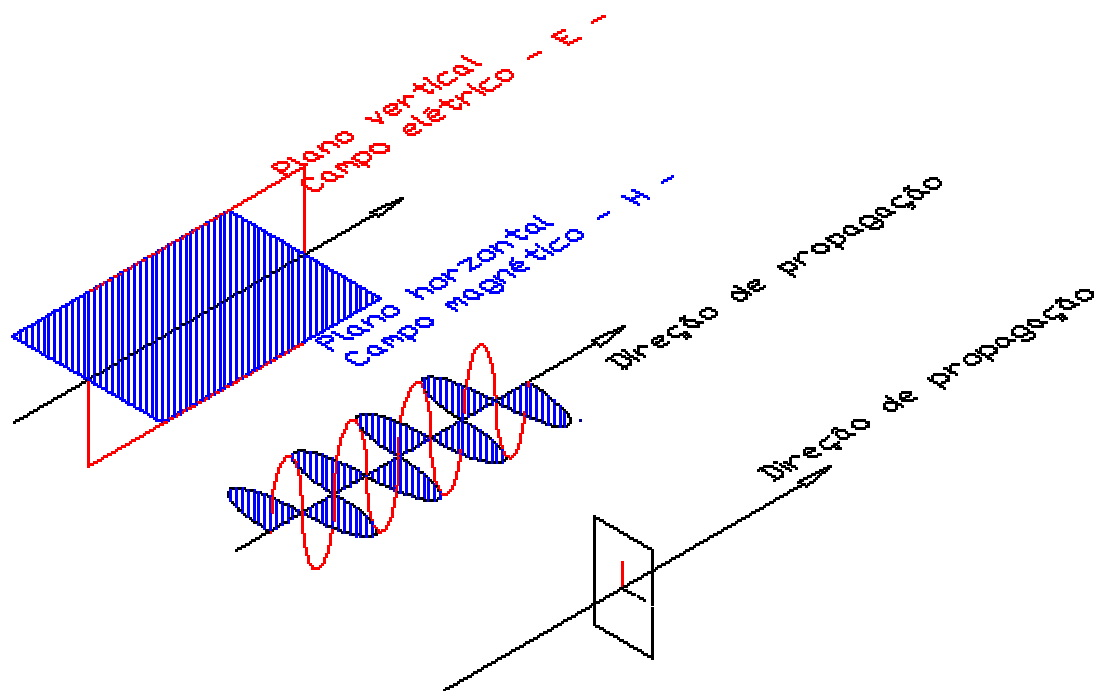


Figura 1. Propagação de um campo eletromagnético.

2.1.2 Teoria dos Raios

Segundo SMIT (1988, p. 19), a teoria dos raios consiste simplesmente em conceituar que o corpo observado, ou emissor, emite raios que percorrem trajetórias retas. Desta teoria, partindo-se do elemento transmissor, com potência W irradiada, a densidade de potência (P) varia inversamente com o quadrado da distância, conforme Eq.(1):

$$P = \frac{W}{4\pi * r^2} \quad (1)$$

Tendo o transmissor uma potência W_t emitida direcionalmente, por uma antena de ganho (G_t) e sendo r a distância entre as antenas transmissora e receptora, o rádio enlace pode ser representado pela Eq.(2) , (SMIT, 1988, p. 20):

$$\frac{W_r}{W_t} = \frac{G_t * A_r}{4\pi * r^2} \quad (2)$$

Onde:

W_r = Potência recebida pelo receptor.

W_t = Potência transmitida.

G_t = Ganho da antena transmissora.

A_r = Área de captação de elemento receptor.

Com base nas Equações (1) e (2), este problema procura aproximar as facilidades de seus pontos de demanda, encurtando as distâncias, o que acarreta em um aumento da densidade de potência recebida pelo receptor, representando uma melhoria da qualidade do sinal.

A Figura 2 ilustra este fenômeno, onde observa-se uma lâmpada emitindo um feixe de luz que ilumina uma superfície (A), a uma distância (L), com determinada intensidade. Em outra superfície (B), que é 2 vezes a distância da fonte ($2L$), a mesma quantidade de energia é distribuída sobre a área (B) que é o dobro do tamanho da área (A). Assim, a intensidade da energia cai conforme mostrado pela Eq. (1). Isto é chamado *lei do quadrado inverso*.

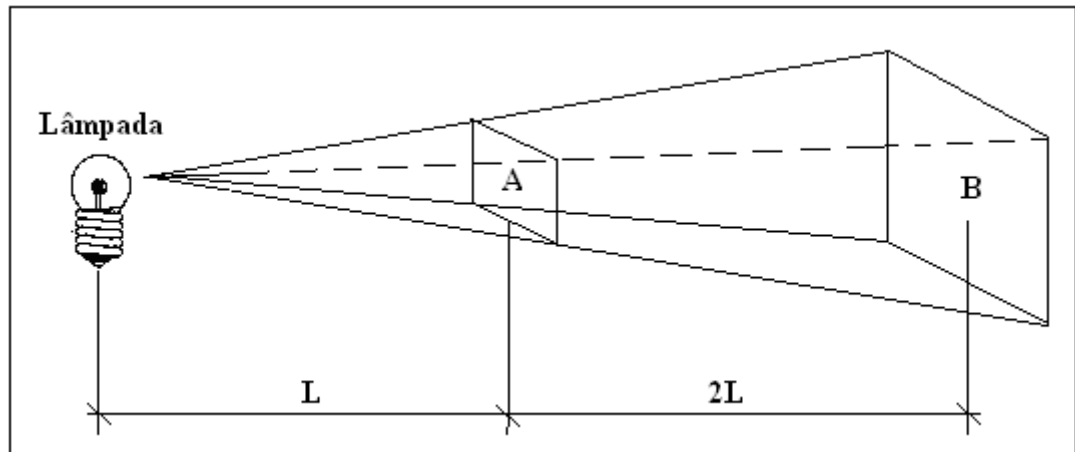


Figura 2. Densidade de potência em função da distância.

2.1.3 Obstáculos interferentes às ondas de visada direta

A propagação em espaço livre foi estudada inicialmente, admitindo-se tão somente que as ondas eletromagnéticas propagam-se em linha reta. Esta teoria, denominada *teoria dos raios*, é válida quando a relação de comprimento de onda (λ) para o tamanho dos obstáculos presentes tende a zero, conforme Eq.(3).

$$\frac{\lambda}{L} \rightarrow 0 \quad (3)$$

Onde:

λ é o comprimento de onda do sinal transmitido.

L é o comprimento do obstáculo.

Desta forma, um obstáculo para ser considerado interferente deverá ser de tamanho muitas vezes superior ao comprimento de onda do sinal (λ) além de ser denso, impossibilitando o fenômeno da refração (SMIT, 1988, p. 22). A interferência causada por um obstáculo provoca em muitos casos o surgimento de *zonas de silêncio* que impossibilitam a recepção do sinal (Figura 3).

Tome-se como exemplo a frequência de transmissão de 98 MHz, típica de uma rádio FM, incidindo em uma montanha com um comprimento aproximado de 300 m. Pode-se facilmente verificar que o comprimento do obstáculo é muitas vezes superior ao comprimento de onda, além de ser denso, logo $\lambda/L \rightarrow 0$.

$$\lambda = \frac{1}{f} \therefore \lambda = \frac{1}{98.000.000} \therefore \lambda = 98 \cdot 10^{-6} \text{ m}$$

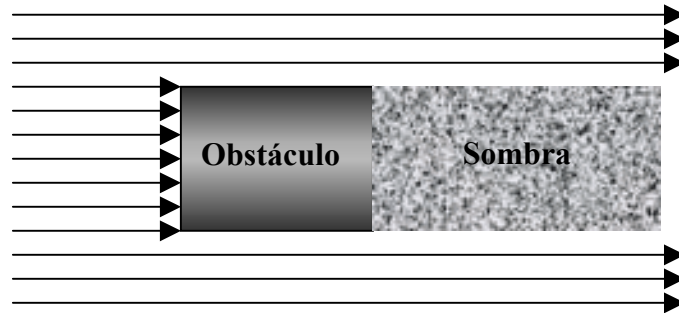


Figura 3. Teoria dos raios (válida quando $\lambda/L \rightarrow 0$).
Fonte: SMIT, 1986, p. 11

O princípio de Huygens, abordado por SMIT (1986, p. 10-13), afirma que cada frente de onda equivale a uma coleção de radiadores infinitesimais, que radia para frente ondas semi-esféricas, conforme ilustrado pela Figura 4. Caso exista um obstáculo no caminho das ondas, a teoria dos raios prevê uma sombra perfeita através do obstáculo, e a teoria de frente de onda com o princípio de Huygens prevê o que realmente ocorre, isto é, uma sombra, porém não total. Isto equivale a dizer que na prática, em alguns casos, poderá ocorrer um atendimento de demanda superior ao previsto na solução do problema.

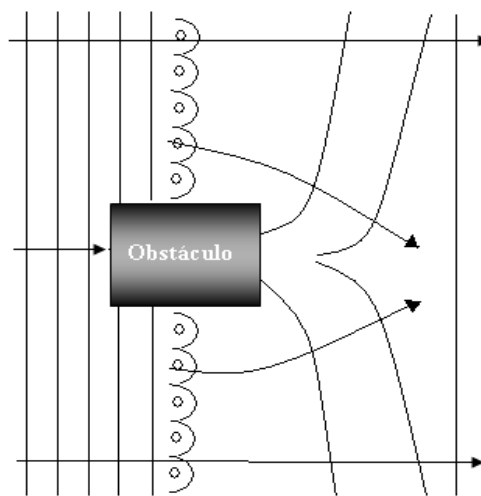


Figura 4. Teoria de frente de onda.
Fonte: SMIT, 1986, p. 11

Objetos pouco densos como construções de alvenaria, provocam refrações no sinal e não a sua total absorção como ocorre no caso das montanhas, Figura 5. Segundo a lei de *Snell*, Eq.(4), discutida por SMIT (1988, p. 28), a relação entre o ângulo de incidência (i) e o ângulo de transmissão (t) de um sinal é dada por:

$$\frac{\sin i}{\sin t} = \frac{v_1}{v_2} \quad (4)$$

Onde v_1 e v_2 são as velocidades de propagação de cada meio.

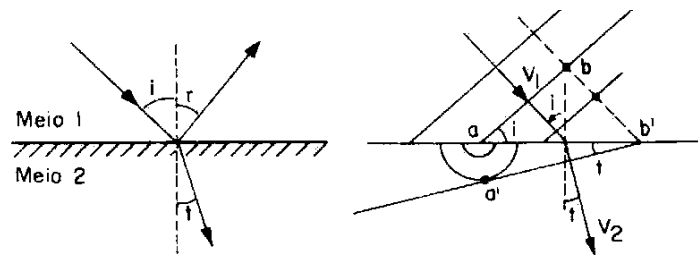


Figura 5. Refração por Snell.
Fonte: SMIT, 1988, p. 28

O fenômeno de refração é muito comum em objetos pouco densos. Isto explica, por exemplo, porque é possível sintonizar um rádio em um quarto totalmente fechado. Sendo assim, as edificações não são consideradas obstáculos na modelagem do problema.

3 ALOCAÇÃO- LOCALIZAÇÃO DE FACILIDADES

Este Capítulo apresenta alguns modelos de localização-alocação de facilidades comumente encontrados na literatura, como o modelo de máximo recobrimento (*maximum covering*), o problema de localização de máxima cobertura (*PLMC*) e o problema das p -medianas, sendo apresentadas algumas heurísticas propostas por diversos autores, para locação-alocação de facilidades.

3.1 MODELOS DE ALOCAÇÃO-LOCALIZAÇÃO

Problemas de alocação-localização tratam de decisões sobre a obtenção da melhor configuração (ou ótima) para a instalação de uma ou mais facilidades, visando atender a demanda de uma população, (DASKIN,1995). O termo "facilidades" é utilizado para designar postos de saúde, depósitos, escolas, fábricas, antenas de telecomunicações, etc., enquanto "clientes" referem-se a bairros, unidades de vendas, estudantes, etc.

Alguns problemas de alocação-localização são de natureza combinatória, pois consistem em selecionar de um conjunto discreto e finito de dados o melhor subconjunto que satisfaça determinados critérios, sendo considerados altamente complexos e custosos do ponto de vista computacional. Para resolver tais problemas geralmente são utilizados métodos heurísticos que proporcionam uma "boa" aproximação dos resultados que, em muitos casos, são suficientes para o propósito do usuário.

Na literatura existem muitos métodos heurísticos para o problema de alocação-localização de facilidades. KUEHN & HAMBURGER (1963)

desenvolveram o método heurístico conhecido como método das duas fases. A primeira das fases é um método guloso, chamado *ADD*, que inicia o processo com todas as facilidades fechadas e, gradativamente, adiciona (abre) facilidades que resultem em um máximo acréscimo da função objetivo. Este algoritmo é finalizado se nenhuma facilidade a ser adicionada melhorar a função. A segunda fase é um método de busca local no qual uma facilidade aberta, e uma fechada, podem ser trocadas caso melhorem o custo total.

De forma semelhante ao método *ADD*, um outro método guloso, chamado *DROP* (FELDMAN *et al.*, 1966), também pode ser utilizado como uma primeira fase, que inicia com todas as facilidades abertas e vai fechando-as à medida que a função objetivo melhora.

Várias outras heurísticas podem ser citadas, mas de uma forma geral, elas são modeladas de acordo com cada problema, considerando as restrições de uso, demanda, número de facilidades, distância, dentre outros.

A distância cliente-facilidade é um diferencial importante entre tais modelos. Em alguns problemas de localização conhecidos como problemas de máxima distância, por exemplo, uma distância máxima (distância de recobrimento) é dada a priori (TOREGAS & REVELLE, 1972). Dentro deste contexto estão os problemas de recobrimento de conjuntos (*set covering*), máxima cobertura (*maximum covering*) e máximo recobrimento esperado (*maximum expected covering*), (CHURCH & REVELLE, 1974).

O modelo de máxima cobertura abrange os problemas de localização que consiste em maximizar o número de clientes atendidos através de um número de locais candidatos a facilidades, permitindo que nem toda a demanda seja atendida. Nos problemas de cobertura os clientes são, geralmente, designados às facilidades mais próximas. Desta forma, julga-se adequado atender (cobrir) o cliente, se o mesmo estiver dentro de uma dada distância da facilidade e considera-se o atendimento inadequado, se a distância excede um valor crítico estipulado. Existem diversos modelos. Cada um deve ser cuidadosamente aplicado de acordo com o problema estudado, devendo ser observadas as restrições de uso, requisições de demanda, facilidades, custos operacionais, dentre outros. Na seqüência, apresentam-se os mais comuns da literatura:

- *Modelos de competição*: a distribuição de produtos nos locais já conta com produtos similares, distribuídos por concorrentes. Objetiva-se entrar no mercado capturando a maior quantidade possível de demanda, levando-se em consideração as instalações dos concorrentes.
- *Modelos probabilísticos*: um recurso alocado poderá não estar disponível quando necessário, por exemplo, a ambulância localizada pode estar atendendo um outro chamado quando está sendo necessária em mais de um local ao mesmo tempo. Considera-se a possibilidade de uma ocorrência deste tipo de evento incluindo no modelo medidas de probabilidades. Pode-se considerar também, filas de atendimento, etc.
- *Modelos que combinam localização e roteamento*: deseja-se localizar e ao mesmo tempo sequenciar várias tarefas.
- *Modelos para materiais perigosos*: Localizar, por exemplo, resíduos tóxicos. Neste caso, procura-se aumentar ao máximo a distância dos aglomerados populacionais.

3.2 PROBLEMA DE LOCALIZAÇÃO DE MÁXIMA COBERTURA (*PLMC*)

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 atendidos.

Existem na literatura diversas aplicações do *PLMC*. Dentre elas, pode-se citar a aplicação feita por ADENSO-DIAZ & RODRIGUEZ (1997), que determinam onde localizar um dado número limitado de bases de ambulâncias de atendimento médico, de forma que a maioria da população seja atendida prontamente em um dado tempo, minimizando os custos de implantação da rede de ambulâncias.

No *PLMC* podemos ter diversas restrições como 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 poderá percorrer até alcançar a facilidade mais próxima, comumente chamada de distância crítica de serviço. Este conceito foi amplamente discutido por TOREGAS & REVELLE (1972).

3.2.1 Surgimento do *PLMC*

A abordagem do *PLMC* surgiu no artigo de CHURCH & REVELLE (1974) como uma alternativa a mais para os problemas *SCLP* (*Set Covering Location Problem*) e *PCP* (*p-Center Problem*). *SCLP* 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. *PCP*, por sua vez, é uma classe de problema que procura localizar p facilidades de maneira que a medida de dissimilaridade máxima de qualquer ponto de demanda até a sua facilidade mais próxima seja mínima. Tais abordagens 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.

3.2.2 Os custos relativos à abertura de novas facilidades

DASKIN (1995), mostra um exemplo de como os custos se relacionam com o número de facilidades existentes (Figura 6). Inicialmente, o custo total apresenta um declínio em função da redução do custo de transporte, o que acontece quando novas facilidades são abertas. Por outro lado, à medida que mais facilidades são abertas este custo sofre um acréscimo proporcional ao número de facilidades abertas já que existe um custo fixo associado à abertura de novas facilidades.

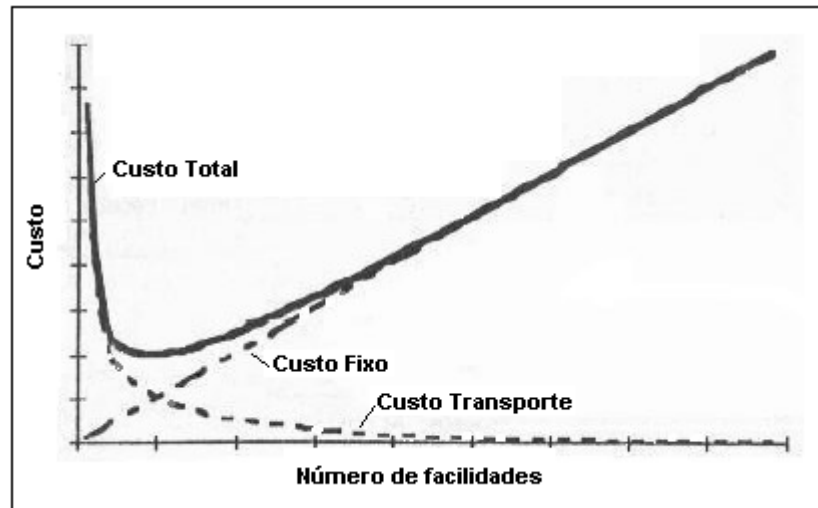


Figura 6: Custos x Número de facilidades.

3.2.3 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 a localização de facilidades como centros de serviços emergenciais, agências bancárias, áreas de conservação e outros. Dentre estes trabalhos podemos citar:

- Escolha e composição de listas para realizar uma campanha publicitária pelo correio (DWYEAR & EVANS, 1981).
- Escolha de ferramentas em manufatura flexível (DASKIN *et al.*, 1990).
- Escolha de agências bancárias para maximizar a base de recolhimento de fundos (GOBERNA *et al.*, 1990).
- Planejamento de uma rede de ambulâncias (HAMON *et al.*, 1979); (EATON *et al.*, 1986); (ADENSO-DIAZ & RODRIGUEZ, 1997).
- Localização de sirenes de aviso (CURRENT & O'KELLY, 1992).
- Projeto de uma rede de monitores para controlar a poluição do ar (HOUGLAND & STEPHENS, 1976).
- Seleção de áreas prioritárias para conservação (WOODHOUSE *et al.*, 2000)
- Identificação de áreas que representam a máxima possível representação de espécies específicas (CHURCH *et al.*, 1996).

Embora a alocação-localização de antenas de transmissão seja um problema antigo, a sua abordagem científica começou a receber uma maior atenção recentemente. Os trabalhos publicados que tratam especificamente deste problema realizam simplificações do modelo real simplesmente fixando um número de antenas e desconsiderando os obstáculos presentes. Isto simplifica bastante o modelo, porém a sua aplicação prática pode comprometer a qualidade final das soluções obtidas em relação à solução ótima do problema original.

Dentre os trabalhos pioneiros que tratam deste problema, podemos citar os métodos de alocação-localização e pesquisa tabu proposto por HOFFMAN & GÓMEZ (2003). Neste trabalho, os autores modelam o problema como *PLMC*, onde dado um número fixo de antenas, procura-se maximizar o número de clientes receptores que podem possuir prioridades distintas de atendimento. Esta é uma solução interessante, embora não procure aproximar as antenas transmissoras dos pontos de demanda e, desta forma, melhorar a qualidade do sinal. Também a demanda a ser atendida é considerada livre de obstáculos que possam causar interferência no sinal, o que, em muitos casos, inviabiliza a aplicação do modelo.

O problema de localização com máxima cobertura também é abordado por LORENA *et al.*, (2001). É utilizado o Modelo Linear Unificado (*MLU*) de HILLSMAN (1984) que transforma os coeficientes de distância de um problema de p -medianas, agregando a estes valores uma informação de demanda. Os autores procuram minimizar as distâncias entre os pontos de demanda e as medianas instaladas ou maximizar o atendimento aos pontos de demanda. Esta abordagem parte do pressuposto que é conhecido o número correto de facilidades que serão instaladas em uma região (medianas). Para a redução dos custos, idealmente, toda a região deveria ser atendida pelo menor número possível de antenas. Desta forma, utilizar a abordagem como sendo o problema das p -medianas poderá significar um excesso de facilidades instaladas, com conseqüente aumento nos custos, ou ainda, uma carência de facilidades com redução do número de pontos de demanda atendidos.

3.2.4 Métodos de solução heurística do *PLMC*

O *PLMC* é NP-difícil, pois não são conhecidos algoritmos de tempo polinomial para encontrar a solução ótima (GAREY & JOHNSON, 1979). CHURCH & REVELLE

(1974) desenvolveram duas metaheurísticas 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, conforme podemos observar na Figura 7, que ilustra a técnica gulosa proposta pelos autores. Neste caso, o algoritmo seleciona a antena (1) por atender a uma maior quantidade de pontos de demanda. Em seguida é escolhida a antena (2). Desta forma consegue-se atender 12 pontos de demanda com duas antenas. Já a Figura 8 mostra uma interessante solução, que não adota esta técnica, escolhendo estrategicamente duas antenas (2) e (3), atendendo a todos os 14 pontos de demanda, com o mesmo número de facilidades. Os obstáculos são considerados interferentes e obstruem a passagem do sinal de telecomunicações, por este motivo busca-se uma linha de visada direta entre os locais candidatos e os pontos de demanda.

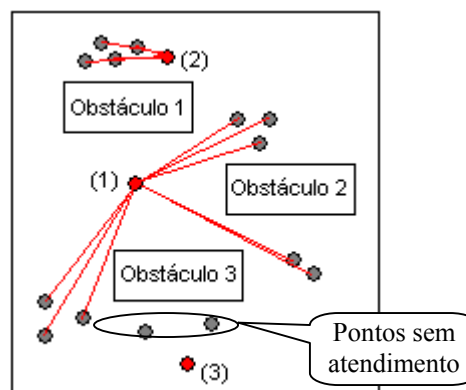


Figura 7: Resultados gráficos obtidos pelo Algoritmo de Acréscimo Guloso (AAG).

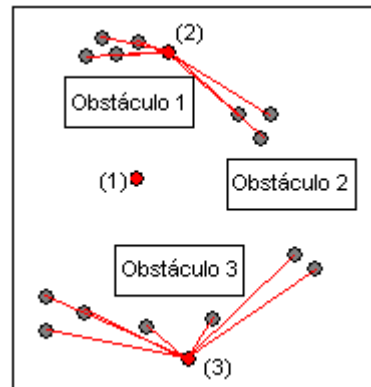


Figura 8: Resultados gráficos obtidos pela heurística *GRASP-ADD*.

3.2.5 Métodos exatos de solução do *PLMC*

DWYEAR & EVANS (1981) propuseram um algoritmo exato para o *PLMC* que resolvia um problema específico de seleção e composição de listas para realizar uma campanha publicitária pelo correio. O algoritmo utiliza 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*, não sendo generalizável.

DOWNS & 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*.

3.3 O PROBLEMA DAS p -medianas

O problema das p -medianas (*PM*) consiste em encontrar a localização de p facilidades em uma rede tal que o custo total (soma dos custos de atendimento dos clientes) seja minimizado. Este modelo é um dos mais populares, tendo sido aplicado por várias vezes na localização de centros nos setores públicos e privados. Conceitualmente, ele é muito simples, entretanto, possui um número muito grande de soluções e não é sempre possível resolvê-lo de forma ótima.

Proposta inicialmente por HAKIMI (1964) e em seguida generalizada para múltiplas medianas (HAKIMI, 1965), *PM* é um problema NP-difícil (GAREY & JOHNSON, 1979). Por isto, diversas heurísticas foram propostas para a solução deste problema, como a heurística de substituição de vértices (TEITZ & BART, 1968) e suas variações (DENSHAM & RUSHTON, 1992). No contexto das metaheurísticas temos o trabalho de ROLLAND & SHILLING (1997), que utiliza a Busca Tabu na solução do *PM*. Algoritmos exatos exploram uma árvore de busca como as de CHRISTOFIDES & BEASLEY (1982) ou GALVÃO & RAGGI (1989).

Existem diversas publicações que abordam o problema das p -medianas, dentre estas pode-se citar o trabalho de PIZZOLATO (1994) na localização de escolas públicas em regiões pobres. O modelo das p -medianas poderá ser aplicado em diversos problemas de alocação-localização. HILLSMAN (1984) formula o problema das p -medianas, conseguindo de forma aproximada tratar outros tipos de problemas de localização. Dependendo do código elaborado pode-se resolver os seguintes problemas:

- p -medianas com restrição de distância máxima: procura-se minimizar a distância total (com pesos) percorrida de cada ponto de demanda a seu centro aberto mais próximo, assegurando que um número máximo de pontos possíveis estão entre uma determinada distância do seu ponto de atendimento.
- Minimização da distância total em potências: onde a distância total percorrida de cada ponto de demanda a seu centro aberto mais próximo é minimizada. As distâncias individuais podem ser elevadas por exemplo ao quadrado, ao cubo, ou outra função de potência.
- Maximização de atendimento: maximiza-se os pontos de demanda atendidos, assumindo que a atribuição de demanda a centros é linearmente proporcional à distância do centro.
- Problema de Máxima Cobertura com restrição de distância máxima: maximiza o número de pontos de demanda atendidos que se encontram a uma determinada distância de seu centro mais próximo. Uma restrição secundária de maior distância é aplicada para assegurar que pontos que estiverem acima da primeira distância sejam atendidos, desde que não excedam a segunda distância.

3.3.1 O problema das p -medianas capacitado (PMC)

O PMC é um problema de particionamento de conjunto de objetos ou clientes (vértices), onde cada objeto possui uma demanda (peso) em 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.

O problema é NP-difícil (GAREY & 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 e metaheurísticas.

Muitos trabalhos foram desenvolvidos nas últimas décadas com o sentido de melhorar os métodos heurísticos, sem, no entanto, prejudicar a sua flexibilidade. Estes trabalhos deram origem as estratégias comumente conhecidas como metaheurísticas. Metaheurísticas são procedimentos destinados a encontrar uma boa solução, eventualmente a ótima, consistindo na aplicação, em cada passo, de uma heurística subordinada, a qual deverá ser modelada para cada problema específico.

Diversas metaheurísticas foram propostas na literatura para o PMC . Dentre estes trabalhos podemos citar o algoritmo *branch and bound*, proposto por PIRKUL (1987), que utiliza a relaxação Lagrangeana nas restrições de particionamento. O método “simulated annealing” foi aplicado por GOLDEN & SKISCIM (1986) para a solução do problema do caixeiro viajante e do problema das p -medianas. OSMAN & CHRISTOFIDES (1994) utilizaram um algoritmo híbrido utilizando-se da busca tabu e do método “simulated annealing”, e uma metaheurística evolutiva, que atualiza uma população inteira de soluções a cada iteração foi apresentada por MANIEZZO *et al.* (1998).

4 O PROBLEMA DE ALOCAÇÃO-LOCALIZAÇÃO DE ANTENAS

Neste Capítulo o problema de alocação-localização de antenas de transmissão é definido e modelado matematicamente, sendo mostrada a formulação da função objetivo que é utilizada pelos modelos propostos. Todas as restrições são tratadas, sendo também apresentados os cálculos matemáticos necessários para detecções de interferências causadas pelos obstáculos. Por último, apresenta-se a formulação da matriz de custos, que é utilizada pelas heurísticas destinadas a procurar boas soluções para este problema.

4.1 O CRITÉRIO PARA ESCOLHA DE LOCAIS CANDIDATOS

O problema é modelado de forma que sejam informados os locais candidatos para a instalação das facilidades. O critério para a escolha dos locais candidatos, consiste em informar em uma dada região que se deseja prover serviços, os pontos mais altos, geralmente correspondentes às montanhas. Desta forma, analisando a topologia do terreno e com base em diversas restrições como dificuldades de acesso, indenizações a propriedades particulares, pontos de fornecimento de energia elétrica, dentre outros, são fornecidas as coordenadas dos locais que candidatam-se à instalação das facilidades. Tais coordenadas são referenciadas como um ponto, que possui informações de latitude, longitude e altura.

4.2 DEFINIÇÃO DO PROBLEMA

O objetivo do problema de localização de antenas é determinar os locais para a instalação de um número mínimo de antenas (facilidades) em uma rede de n locais candidatos, de modo a atender a um conjunto de pontos de demanda, procurando minimizar a distância entre cada ponto de demanda e a antena mais próxima. O problema pode ser modelado matematicamente considerando as seguintes notações:

- $B = \{1, \dots, n\}$, um conjunto de pontos de demanda (um ponto de demanda pode ser um bairro, uma quadra ou quarteirão de um bairro);
- $b_i = 1$ se o ponto de demanda $i \in B$ é atendido por pelo menos uma antena, caso contrário, $b_i = 0$;
- $A = \{1, \dots, m\}$, um conjunto de pontos potenciais (candidatos) para instalação de facilidades;
- $a_j = 1$ se a antena $j \in A$ é utilizada para atender à pelo menos um ponto de demanda, caso contrário $a_j = 0$;
- C_j : representa o custo de instalação da facilidade j (antena j);
- d_{ij} : é a distância entre um ponto de demanda i e a facilidade j .
- D : é o alcance máximo (raio de ação) de uma facilidade;
- $N_i = \{j \mid d_{ij} \leq D\}$ é o conjunto de facilidades que cobre ou atende o ponto de demanda i (conjunto de facilidades que podem atender ao ponto i).

O problema de alocação-localização de antenas de transmissão pode ser modelado conforme mostrado na Figura 9.

Função Objetivo	Minimizar $\sum_{j=1}^m C_j a_j + \sum_{i=1}^n \min\{d_{ij} \mid j \in A, a_j = 1\}$	(1)
Restrições	Sujeito a $\sum_{j \in N_i} a_j \geq b_i, \quad i = 1, \dots, n$	(2)
	$\sum_{j=1}^m a_j \geq 1$	(3)
	$b_i \in \{0, 1\}, \quad i = 1, \dots, n$	(4)
	$a_j \in \{0, 1\}, \quad j = 1, \dots, m$	

Figura 9. Formulação matemática do problema de localização de antenas.

A função objetivo é definida como a soma dos custos das antenas utilizadas e a soma das distâncias mínimas entre cada ponto de demanda e a antena instalada. Note que uma solução melhora à medida que menos facilidades são empregadas estando estas, o mais próximo possível dos pontos de demanda.

As restrições (2) indicam que os pontos de demanda serão considerados como cobertos (ou atendidos), se estiverem dentro do raio de cobertura de alguma antena instalada.

A restrição (3) garante que no mínimo deve ser usada uma antena. Na prática isto é muito comum, quando é desconhecida a quantidade de facilidades. Por último, as restrições (4) indicam a escolha binária para b_i e a_j . Onde $b_i = 1$ indica o atendimento ao ponto de demanda i e $b_i = 0$ indica o não atendimento. Um ponto de demanda é considerado atendido, se existir pelo menos uma linha de visada direta entre ele e alguma facilidade, com distância menor ou igual ao máximo alcance do sistema. Também, $a_j = 1$ indica que a facilidade foi utilizada e $a_j = 0$, indica o seu não uso.

Considera-se que as antenas irradiam igualmente o sinal em todas as direções (antenas onidirecionais), onde o alcance de um sistema de telecomunicações está limitado basicamente à potência do sinal empregado, ganho das antenas, degradações sofridas pelo sinal quando ele propaga-se no ar (teoria dos raios) e presença de obstáculos. Estes fatores associados fornecem um valor que representa o máximo alcance (D) com boa qualidade de recepção do sinal. Na Figura 10 mostra-se um exemplo na qual um ponto 1 de demanda está dentro do raio de cobertura de 3 facilidades, ou seja $N_1 = \{1,2,3\}$.

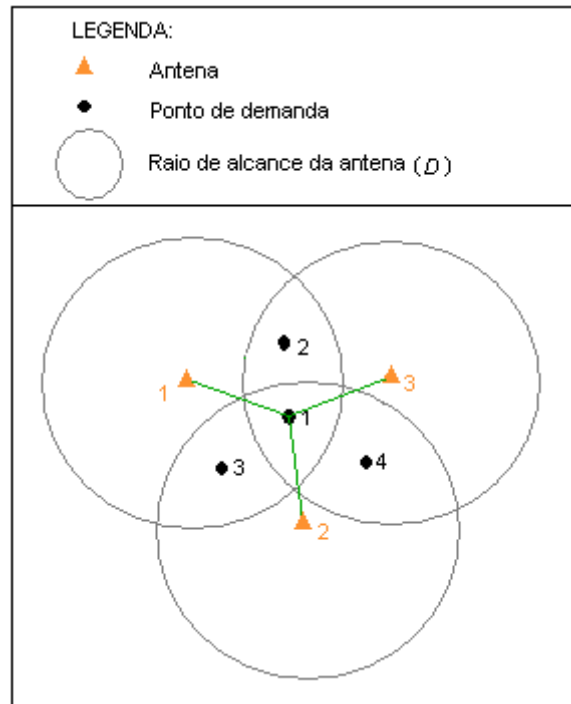


Figura 10: Exemplo da formação do Conjunto N em relação a um ponto de demanda.

Neste trabalho considera-se o mesmo custo para todas as antenas instaladas. O custo simbólico de implantação de uma facilidade influencia no valor da função objetivo. Se for muito elevado, o modelo matemático poderá diminuir as facilidades utilizadas. Como consequência, alguns pontos de demanda poderão ficar sem atendimento. Por outro lado, com um baixo custo de implantação, o sistema irá aproximar ao máximo as antenas dos clientes e com isto, tenderá a utilizar mais facilidades. Desta maneira, em cada problema analisado deve-se procurar um ponto de equilíbrio, que permita o atendimento máximo aos clientes, usando o conjunto transmissor (rádios-transmissores e antenas) dentro de seus limites de alcance máximo.

A Figura 11 representa a melhor solução viável encontrada pela heurística *GRASP-ADD*, que será apresentada no Capítulo 6. Foi gerada uma instância de problema contendo 12 locais candidatos, 8 obstáculos e 125 pontos de demanda propositalmente localizados em torno de três locais candidatos. Como resultado, vê-se três facilidades instaladas, atendendo a todos os pontos de demanda. Nesta figura pode-se observar também, a aproximação dos clientes às facilidades.

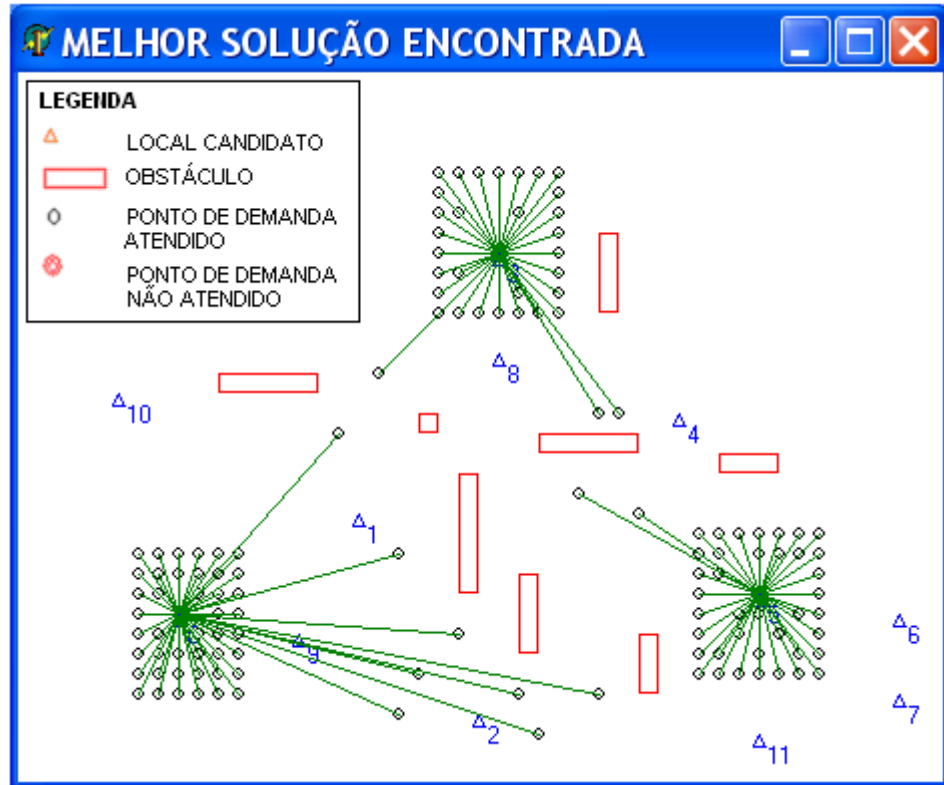


Figura 11. Exemplo gráfico de uma melhor solução viável para uma instância hipotética de problema.

O ponto de equilíbrio (busca pela menor quantidade de antenas e aumento do atendimento aos pontos de demanda) pode ser visualizado, na Figura 12. As Figuras 12 (a) e (b) mostram a falta de facilidades e em consequência a redução no atendimento de pontos de demanda. As Figuras 12 (c) e (d) mostram o excesso de facilidades usadas aumentando o custo de implantação.

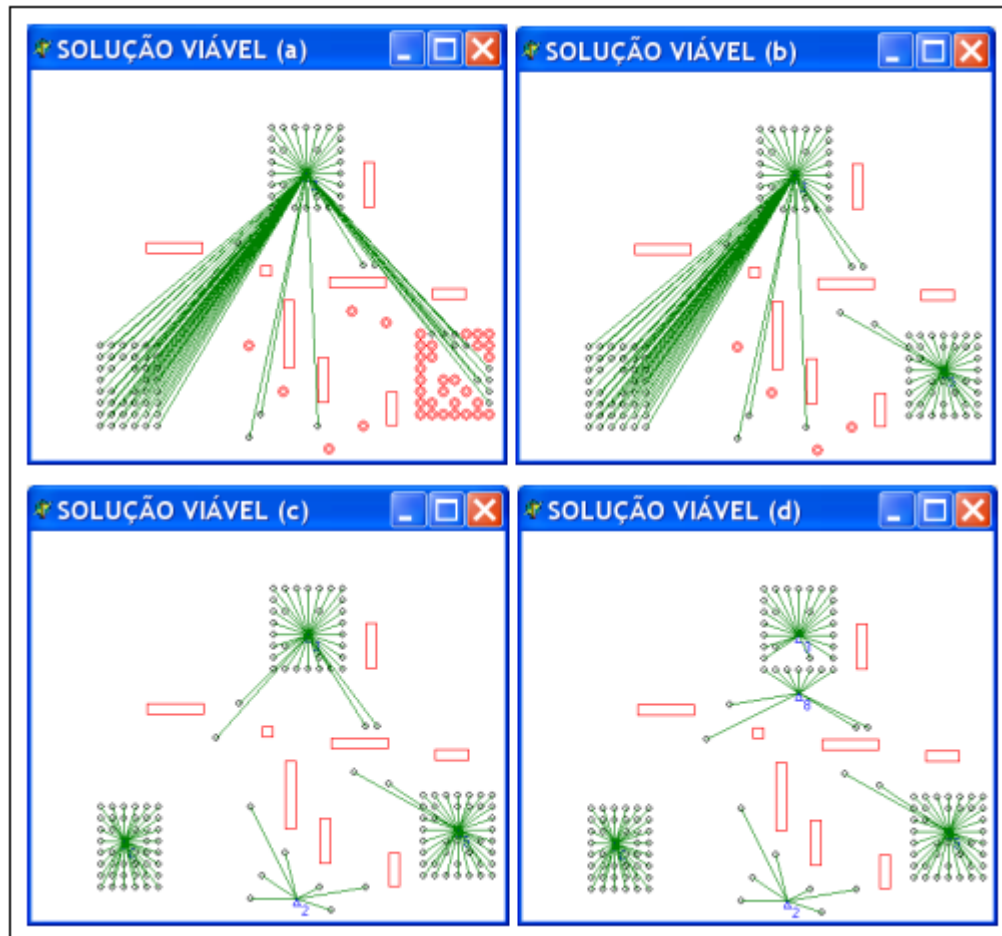


Figura 12. Exemplo gráfico da procura por uma boa solução.

HOFFMAN & GÓMEZ (2003), na localização de antenas de transmissão, formulam a função objetivo procurando apenas maximizar o atendimento aos pontos de demanda. Esta solução não aproxima as antenas dos clientes que, de acordo com a teoria dos raios, (SMIT, 1988), poderá ocasionar em uma redução na qualidade do sinal recebido, uma vez que a densidade de potência varia inversamente proporcional ao quadrado da distância, como demonstrado no Capítulo 2. Isto pode ser observado na Figura 13, que ilustra uma provável solução que não considera as distâncias.

Embora todos os pontos de demanda tenham sido atendidos com apenas três facilidades, esta solução não apresenta bons resultados por não aproximar as antenas dos clientes, não melhorando a qualidade do sinal recebido.

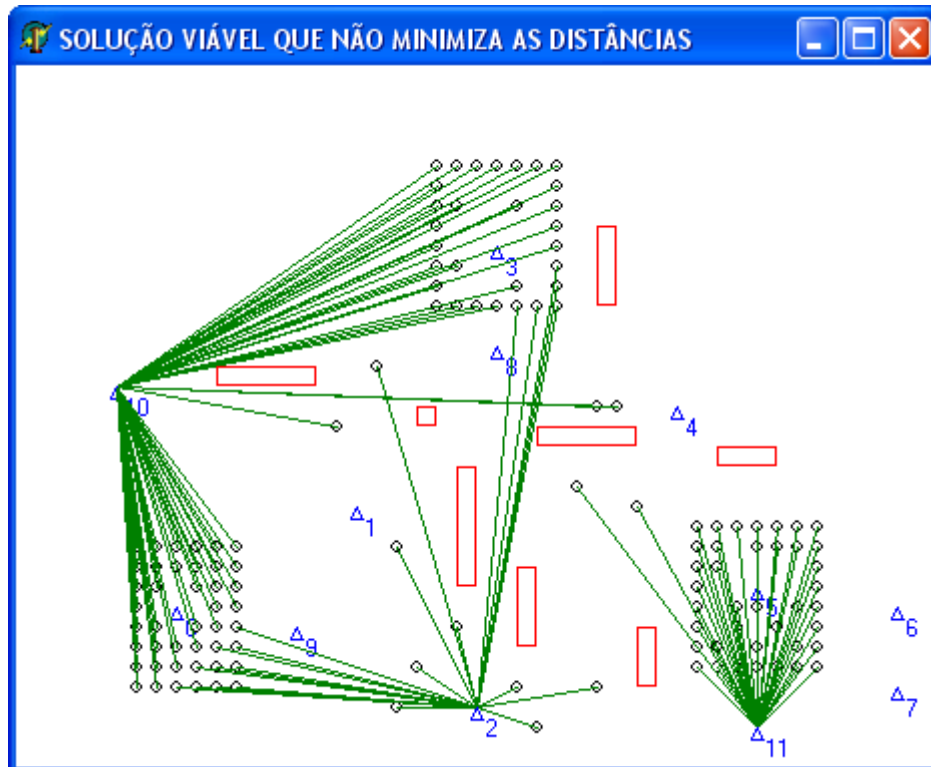


Figura 13. Exemplo gráfico de uma solução viável que não minimiza a distância entre as antenas e os pontos de demanda.

4.2.1 Definição dos obstáculos

Pode-se observar que o tratamento dos obstáculos não está inserido diretamente na formulação matemática do problema, estando a função objetivo um pouco relaxada.

Este trabalho considera apenas os obstáculos densos como sendo interferentes. Esta situação é muito comum em regiões com relevo acidentado onde pode-se observar a presença de muitas montanhas. Estas montanhas são aproximadas de sua forma real para um paralelepípedo de dimensões tais que a comportem totalmente em seu interior. A Figura 14 ilustra esta situação onde são obtidos os maiores valores de cada coordenada para que seja formado um paralelepípedo com dimensões correspondentes.

Devido a esta aproximação, em alguns casos, as heurísticas podem prever o não atendimento a uma dada demanda. Na prática, os resultados podem ser ainda melhores, pois poderá haver uma linha de visada direta entre o ponto de demanda, teoricamente não atendido, e uma antena. Neste caso, o sinal poderá passar

próximo a montanha, sem tocá-la, propiciando o seu recebimento pelo ponto de demanda.

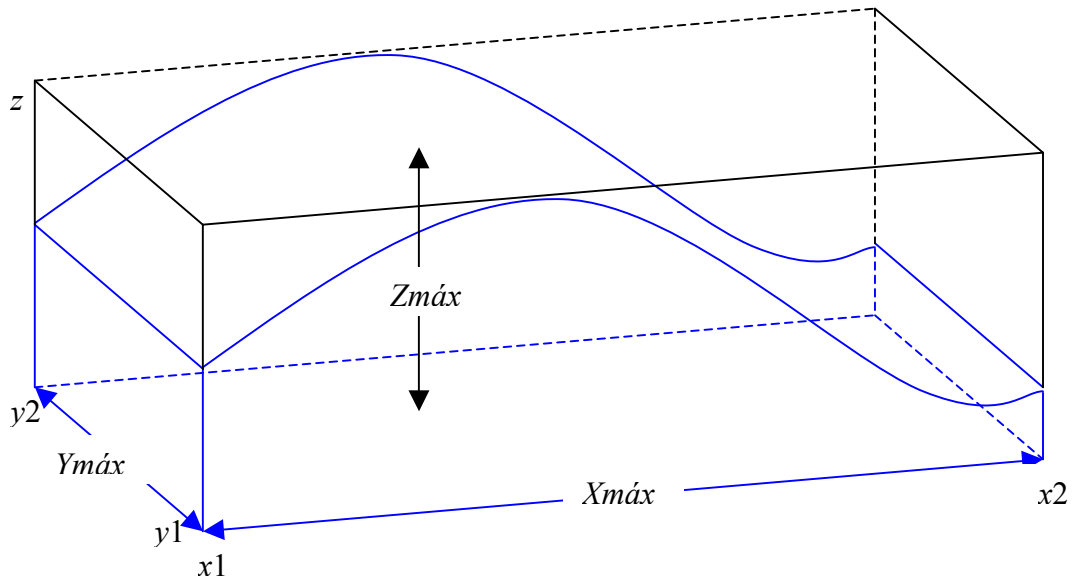


Figura 14. Obstáculo em sua forma real aproximado para o formato de um paralelepípedo.

Consideram-se como dados de entrada do sistema, as coordenadas dos obstáculos representadas por x_1 , x_2 , y_1 , y_2 e z . Desta forma, os obstáculos são definidos como:

$O = \{1, \dots, q\}$ conjunto contendo q obstáculos, com coordenadas x_1 , x_2 , y_1 , y_2 e $z_{máx}$.

4.2.1.1 Cálculo da interferência causada pelos obstáculos

Um obstáculo é considerado totalmente interferente, se a linha reta que une a facilidade e o ponto de demanda testado, o tocar em qualquer uma de suas partes. A interferência é detectada conforme segue:

- **Passo um** - Análise em um plano cartesiano, onde não são consideradas as alturas dos obstáculos, facilidades ou pontos de demanda. Caso a linha reta que une um ponto de demanda testado passar por qualquer obstáculo, deve-se verificar as alturas envolvidas (passo dois). A Figura 15 mostra esta situação. Pode-se observar que a linha reta formada entre a antena a e ponto de demanda b_3 , sofre interferência, se analisada apenas pelo plano cartesiano.

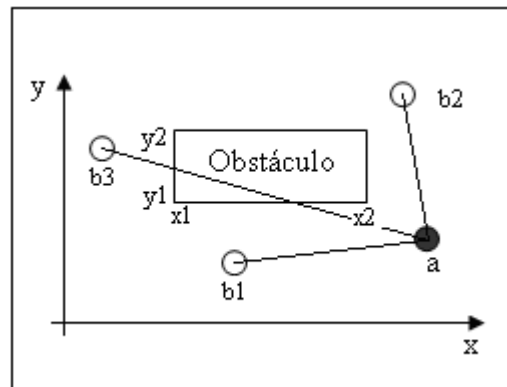


Figura 15. Análise da interferência pelas coordenadas x e y .

- **Passo dois** - Caso haja interferência no plano cartesiano é preciso analisar os ângulos envolvidos entre o obstáculo interferente e a antena, e entre o ponto de demanda e a antena. Se o ângulo calculado pelo obstáculo for inferior, não haverá interferência. A Figura 16 ilustra os cálculos necessários, sendo usadas as seguintes definições:

dcs_{ij} = Distância no plano cartesiano entre o local candidato a_j e o ponto de demanda b_i .

dcs_{ob} = Distância no plano cartesiano entre o obstáculo e o ponto de demanda b_i .

d_{ob} = Distância de visada entre o obstáculo (ob) e o ponto de demanda b_i .

φ = Ângulo entre o local candidato a_j e o ponto de demanda b_i .

φ' = Ângulo entre o ponto interferente do obstáculo ob e o ponto de demanda b_i .

Z_a = Altura da antena a_j .

Z_b = Altura do ponto de demanda b_i .

Z_{ob} = Altura do obstáculo.

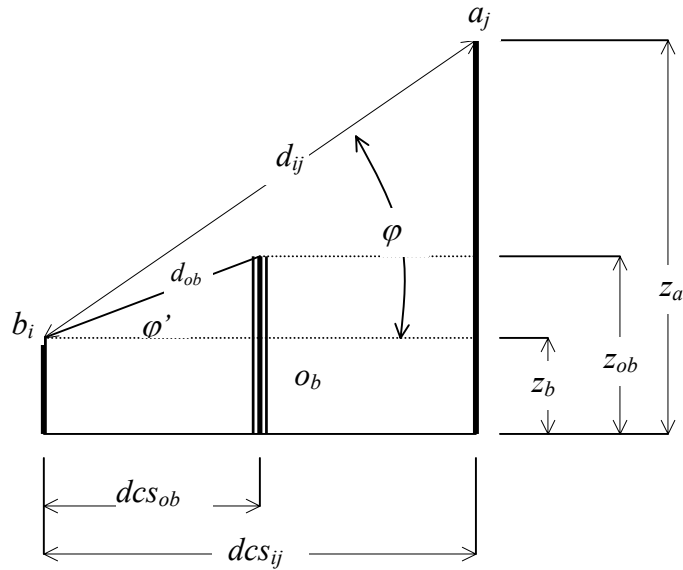


Figura 16: Visada direta entre uma antena e um ponto de demanda, passando por cima de um obstáculo.

Em seqüência, são apresentados os cálculos matemáticos necessários para a detecção da interferência através dos ângulos φ e φ' , subdivididos em quatro passos, descritos a seguir.

- Calcular a distância real d_{ij} entre um ponto de demanda b_i com coordenadas $(x_b, y_b$ e $z_b)$ e um local candidato a_j com coordenadas $(x_a, y_a$ e $z_a)$, conforme Eq.(5).

$$d_{ij} = \sqrt{(z_a - z_b)^2 + dcs_{ij}^2} \quad (5)$$

- Calcular o ângulo φ entre o ponto de demanda b_i e o local candidato a_j , Eq.(6).

$$\varphi = \arcsin\left[\frac{z_a - z_b}{d_{ij}}\right] \quad (6)$$

- Calcular a distância de visada d_{ob} entre o ponto interferente do obstáculo e o ponto de demanda b_i , Eq.(7).

$$d_{ob} = \sqrt{(z_{ob} - z_b)^2 + dcs_{ob}^2} \quad (7)$$

- Calcular o ângulo φ' de todos os pontos interferentes do obstáculo no plano cartesiano, entre a antena e o ponto de demanda analisado, Eq.(8).

$$\alpha' = \arcsin \left[\frac{z_{ob} - z_b}{d_{ob}} \right] \quad (8)$$

O Pseudocódigo mostrado na Figura 17 resume todos os cálculos para a detecção da interferência no plano 3D. Os valores das alturas do local candidato, ponto de demanda e obstáculo atualmente analisados são passados para o procedimento *Ângulo*, que verifica inicialmente se o obstáculo é mais alto que o local candidato e o ponto de demanda (linha 1), ou se é menor que ambos (linha 4). Caso contrário, os ângulos φ e φ' são calculados. Este processo deverá ser feito para todos os obstáculos do problema analisado.

```

Procedimento Ângulo ( $z_a, z_b, z_{ob}$ );
1: Se  $z_{ob} > z_b$  e  $z_{ob} > z_a$  então
2:    $Distância \leftarrow \infty$ ; //ausência de comunicação
3: senão
4:   se  $z_{ob} < z_b$  e  $z_{ob} < z_a$  então
5:      $Distância \leftarrow d_{ij}$ ;
6:   senão
7:     se  $\varphi > \varphi'$  então
8:        $Distância \leftarrow d_{ij}$ ;
9:     senão
10:       $Distância \leftarrow \infty$ ; //ausência de comunicação
11:   fim_se
12: fim_se
13: fim_se
14: Retornar  $Distância$ ;
Fim Procedimento

```

Figura 17. Pseudocódigo do procedimento que detecta a interferência considerando as alturas envolvidas.

A Figura 18 exibe a alocação das facilidades em um problema montado sobre uma maquete. Estrategicamente, a heurística *GRASP-ADD* alocou duas facilidades e, apesar da presença dos obstáculos, todos os cem pontos de demanda foram completamente atendidos. Pode-se observar que em muitos casos, ao atender um ponto de demanda, o sinal de telecomunicações passa por cima dos obstáculos, cabendo ao algoritmo fazer a análise dos ângulos. A Figura 19, ilustra a solução heurística obtida.



Figura 18. Visão superior de uma maquete montada para o problema de alocação-localização de facilidades.

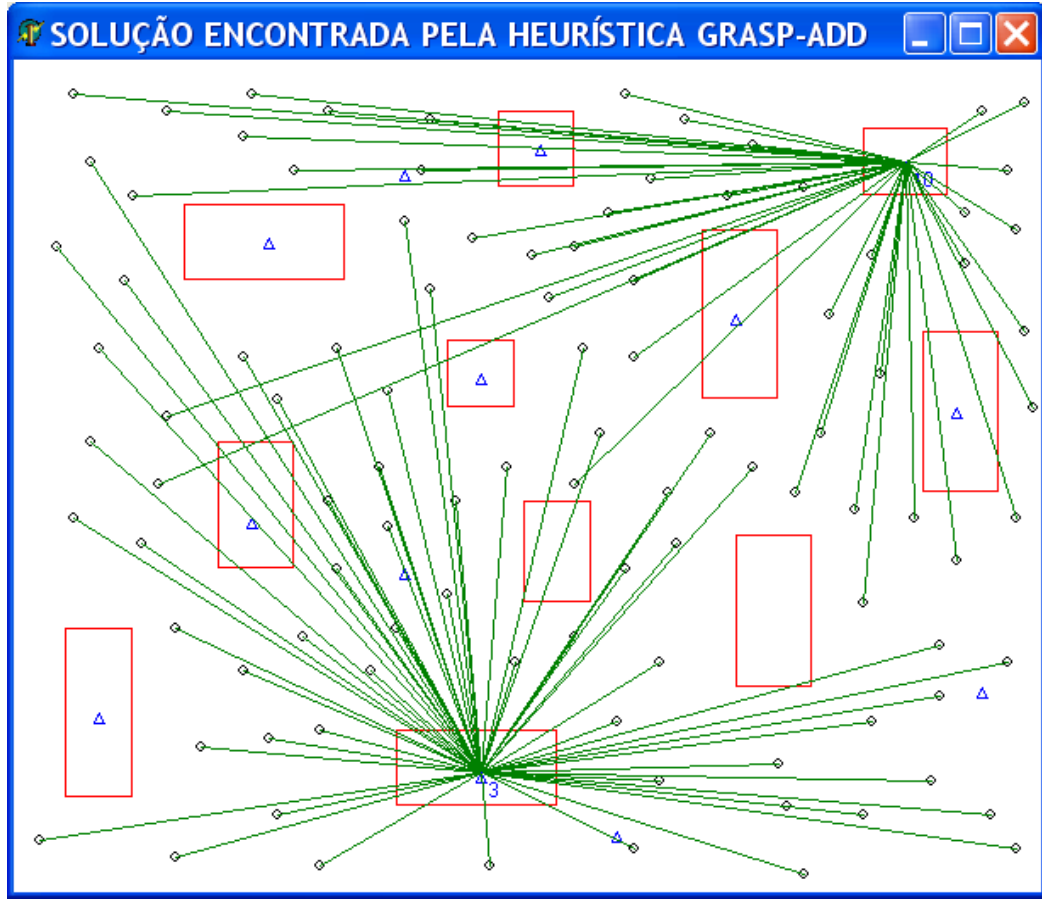


Figura 19. Instância da maquete resolvida pela heurística *GRASP-ADD*.

4.3 CÁLCULO DA MATRIZ DE CUSTOS

Sendo calculada apenas no início do problema, a matriz de custos tem por finalidade informar a distância, d_{ij} , entre um ponto de demanda i e um local candidato j . Para isto, são considerados todos os obstáculos e também o máximo alcance de raio D da facilidade.

Havendo um obstáculo que interfira no sinal entre os pontos i e j , $d_{ij} \leftarrow -\infty$, indicando a ausência de comunicação entre local candidato e o ponto de demanda em análise. O mesmo é válido quando não existe um obstáculo interferindo o sinal, mas a visada direta possui uma distância superior ao máximo alcance das antenas.

A matriz de custos é utilizada pelas heurísticas propostas, pois através dela será possível determinar se os pontos de demanda escolhidos apresentam bons resultados. O Pseudocódigo mostrado na Figura 20 calcula os custos entre os locais candidatos e os pontos de demanda.

Sejam:

$Ob = \{1, \dots, q\}$, um conjunto de obstáculos;

D = Máxima distância que uma antena poderá alcançar em um raio de 360° .

$B = \{1, \dots, n\}$, um conjunto de pontos de demanda;

$A = \{1, \dots, m\}$, um conjunto de pontos potenciais (candidatos) para instalação de facilidades;

$C = \{1, \dots, m; 1 \dots n\}$, uma matriz de custos entre os locais candidatos e os pontos de demanda.

```

Procedimento matriz_custos( $D$ );
  Para ( $j \leftarrow 1, 2, \dots, m$ ) faça:
    Para ( $i \leftarrow 1, 2, \dots, n$ ) faça:
       $interf \leftarrow false$ ;
      Para ( $Ob \leftarrow 1, 2, \dots, q$ ) faça:
        Se  $Ob$  causar interferência então
           $d_{ij} \leftarrow \infty$ ;
           $Ob \leftarrow q$ ;
           $interf \leftarrow true$ ;
        fim_se
      fim_para
      Se  $interf \leftarrow false$  então Calcule  $d_{ij}$ ;
      Se  $d_{ij} > D$  então  $d_{ij} \leftarrow \infty$ ;
       $c[j, i] \leftarrow d_{ij}$ ;
    fim_para
  fim_para
Fim Procedimento

```

Figura 20: Pseudocódigo para o cálculo da matriz de custos.

5 METAHEURÍSTICAS UTILIZADAS

Há vários fatores que influenciam a escolha de uma técnica para resolução de um determinado problema, como a natureza dos dados, o tamanho do conjunto de entrada, etc. Muitas vezes opta-se pelo uso de heurísticas ou metaheurísticas para auxiliar na obtenção de uma solução de boa qualidade, abrindo-se mão da garantia de se obter uma solução ótima pela diminuição do tempo de processamento.

Este Capítulo apresenta a descrição de duas metaheurísticas. A primeira denominada *GRASP* (*Greedy Randomized Adaptive Search Procedure*) proposta por FEO & RESENDE (1995), que mescla as características dos algoritmos puramente gulosos e dos procedimentos aleatórios na fase de construção, e aplica o processo de busca local na solução viável, inicialmente construída. A segunda metaheurística, corresponde aos Algoritmos Genéticos propostos por HOLLAND (1975), que criam uma população inicial e através de um processo de seleção, cruzamento e mutação de indivíduos, atualizam a população, obedecendo a um determinado critério de parada.

5.1 METAHEURÍSTICA *GRASP*

Problemas de otimização combinatória que envolvem um número muito grande de variáveis, aparecem com frequência em setores públicos e privados. Tais problemas teoricamente são possíveis de serem resolvidos pelo método exato, que enumera as soluções e avalia cada uma em relação ao objetivo desejado, mas de uma perspectiva prática é ineficiente seguir tal estratégia, porque o número de combinações cresce exponencialmente com o tamanho do problema.

Segundo RESENDE (1995), alocação-localização de redes de telecomunicações são exemplos de problemas intratáveis, seja por sua natureza ou por serem suficientemente grandes para impedir o uso de algoritmos exatos.

A metaheurística *GRASP* é um processo de busca adaptativa gulosa e randomizada, composto por duas fases:

- Uma fase de construção, na qual uma provável solução é construída, elemento a elemento.
- Uma fase de busca local, na qual um ótimo local na vizinhança da solução construída é pesquisado.

O melhor resultado encontrado ao longo de todas iterações *GRASP* realizadas é retornado como resultado. Seu pseudocódigo pode ser descrito conforme Figura 21.

```

Procedimento GRASP (Max_Iterações)
1:   $s^* \leftarrow \emptyset$ ;
2:   $f(s^*) \leftarrow \infty$ ;
3:  Para ( $h \leftarrow 1, 2, \dots, \text{Max\_Iterações}$ ) faça:
4:     $s \leftarrow \text{Construção\_GRASP}$ ;
5:     $s \leftarrow \text{BuscaLocal}(s)$ ;
6:    Se ( $f(s) < f(s^*)$ ) então
7:       $s^* \leftarrow s$ ;
8:    fim_se;
9:  fim_para;
10: Retorne  $s^*$ ;
Fim GRASP

```

Figura 21. Pseudocódigo da metaheurística *GRASP*.

Na fase de construção da *GRASP*, uma provável solução é construída iterativamente, um elemento da solução por vez, até que a solução esteja completa. Os elementos candidatos que não compõem a provável solução são ordenados em uma lista, chamada de lista de candidatos (*LC*). Esta lista é ordenada por uma função gulosa que mede o benefício que o mais recente elemento escolhido concede à parte da solução já construída.

Um subconjunto denominado lista de candidatos restrita (*LCR*) é formado pelos melhores elementos que compõem a lista de candidatos. O tamanho da lista de candidatos restritos é controlado por um parâmetro $\alpha \in [0,1]$, onde para $\alpha = 0$ tem-se um comportamento puramente guloso do algoritmo e para $\alpha = 1$, um comportamento aleatório. A componente probabilística do método é devida à escolha aleatória de um elemento da lista de candidatos restritos. Este procedimento permite que diferentes soluções de boa qualidade sejam geradas. A Figura 22 descreve um algoritmo para fase de construção da *GRASP*.

```

Procedimento Construção GRASP
   $s \leftarrow \emptyset$ ;
  Inicializar o conjunto de candidatos  $C$ ;
  Enquanto  $C \neq \emptyset$  faça
     $t_i \leftarrow \min \{g(t) \mid t \in C\}$ ;
     $t_s \leftarrow \max \{g(t) \mid t \in C\}$ ;
     $LCR = \{t \in C \mid g(t) < t_i + \alpha (t_s - t_i)\}$ ;
    Selecione aleatoriamente, um elemento  $t \in LCR$ ;
     $s \leftarrow s \cup \{t\}$ ;
    Atualizar o conjunto de candidatos  $C$ ;
  fim enquanto
  Retornar  $s$ ;
fim procedimento

```

Figura 22. Pseudocódigo da fase de construção da *GRASP*.

Assim como em muitas técnicas determinísticas, as soluções geradas pela fase de construção da *GRASP* provavelmente não são localmente ótimas com respeito à definição de vizinhança adotada. Daí a importância da fase de busca local, a qual objetiva melhorar a solução construída. Um algoritmo de busca local pode ser descrito conforme mostrado na Figura 23.

```

Procedimento Busca Local GRASP
   $s \leftarrow$  solução inicial;
  Enquanto critério de parada não satisfeito faça
    Encontrar uma solução  $s' \in N(s)$ ;
    Se  $f(s') < f(s)$  então  $s \leftarrow s'$ ;
  Fim Enquanto
  Retornar  $s$ ;
Fim Procedimento

```

Figura 23: Algoritmo básico de busca local *GRASP*

A eficiência da busca local depende da qualidade da solução construída. A fase de construção tem então um papel importante, uma vez que as soluções construídas constituem bons pontos de partida para a busca local, permitindo assim a obtenção de melhores resultados.

O parâmetro α , que determina o tamanho da lista de candidatos restrita, (*LCR*) deverá estar bem ajustado na implementação da fase de construção *GRASP*. Sabe-se que valores de α que levam a uma *LCR* de tamanho muito limitado (ou seja, valor de α próximo da escolha gulosa) implicam soluções finais de qualidade muito próxima àquela obtida de forma puramente gulosa, obtidas com um baixo esforço computacional. Em contrapartida, provocam uma baixa diversidade de soluções construídas. Já uma escolha de α próxima da seleção puramente aleatória leva a uma grande diversidade de soluções construídas, mas por outro lado, muitas das soluções construídas são de qualidade inferior, tornando mais lento o processo de busca local.

O procedimento *GRASP* procura, portanto, conjugar bons aspectos dos algoritmos puramente gulosos com aqueles dos procedimentos aleatórios de construção de solução.

5.2 ALGORITMOS GENÉTICOS

Na natureza, existe um processo de seleção dos seres vivos. Numa determinada população, quando há escassez de recursos sejam eles, comida, água, espaço ou qualquer outro recurso essencial, os indivíduos mais aptos têm mais chances de sobreviver e por consequência manter para as próximas gerações algumas de suas características. Indivíduos bons também podem surgir da exploração de alguma característica ainda não desenvolvida na população. Porém, se a natureza tentasse descobrir tais características selecionando apenas os melhores e assim possibilitar o cruzamento dentro de um mesmo grupo, certamente não atingiria o sucesso, tendo em vista que depois de muitas gerações, os diversos indivíduos compartilhariam do mesmo código genético. Desta maneira, para que não ocorra tal fato, a natureza insere de forma aleatória materiais genéticos diferentes através de um processo conhecido como mutação. Caso o indivíduo que recebeu o

material genético diferenciado tenha um grau de aptidão satisfatório, suas chances são grandes no futuro processo de seleção.

Os Algoritmos Genéticos (AGs) são um tipo de algoritmo de busca que se utilizam do paradigma genético/evolucionário (HOLLAND, 1975). Algoritmos Genéticos foram criados com o intuito de imitar alguns dos processos observados na evolução natural das espécies.

Segundo BITTENCOURT (1998), os AGs são o ramo mais conhecido da computação evolutiva e tiveram origem no trabalho de Holland, nos anos da década de 1960.

No começo dos anos 70, John Holland, quando pesquisava as características da evolução natural, acreditou que se estas características fossem adequadamente incorporadas a algoritmos computacionais, poderia produzir uma técnica para solucionar problemas difíceis, da mesma forma que a natureza fazia para resolver os seus problemas, ou seja, usando a evolução. Acreditando nisto, ele deu início a uma pesquisa sobre algoritmos que manipulavam “strings” de 0's (zeros) e 1's (uns), a qual ele deu o nome de cromossomos. Os algoritmos de Holland realizavam a evolução simulada de populações destes cromossomos. Desta forma, imitando a natureza, seus algoritmos resolviam muito bem o problema de encontrar bons cromossomos através da manipulação do material contido nos cromossomos.

Segundo GOLDBARG & LUNA (2000), os Algoritmos Genéticos possuem as seguintes características gerais:

- Operam em um conjunto de pontos, denominados população, e não a partir de pontos isolados;
- Operam em um espaço de busca de soluções codificadas e não diretamente no espaço de busca;
- Necessitam, como informação de avaliação, somente o valor de uma função objetivo, denominada função de adaptabilidade ou *fitness*;
- Usam transições probabilísticas e não regras determinísticas.

Os Algoritmos Genéticos têm sido aplicados com sucesso em problemas com funções objetivo muito complexas. Na literatura pode-se encontrar abordagens completas de Algoritmos Genéticos em (REEVES, 1993 ; DOWSLAND, 1996).

5.2.1 A função de avaliação

Um ponto interessante nas técnicas desenvolvidas por Holland, no início dos anos 70, é que assim como na natureza, os cromossomos não tinham conhecimento nenhum sobre o tipo de problema que estavam resolvendo. A única informação que eles dispunham era uma avaliação de cada cromossomo produzido. O objetivo desta avaliação era verificar quais os cromossomos que estavam mais adaptados e, com base nisto, aumentar as suas chances de serem selecionados para a reprodução.

O elemento de ligação entre os Algoritmos Genéticos e o problema a ser resolvido, é a função de avaliação. A função de avaliação, chamada de função *fitness*, toma como entrada um cromossomo e retorna um número ou lista de números que representam a medida de performance do cromossomo, com relação ao problema a ser resolvido. Esta função desempenha nos Algoritmos Genéticos o mesmo papel desempenhado pelo meio ambiente na teoria da evolução natural das espécies.

5.2.2 O processo de seleção

A finalidade do processo de seleção em um algoritmo é escolher os elementos da população que devem se reproduzir, de tal forma que dê maior chance de reprodução aos membros da população mais adaptados ao meio ambiente, isto é, àqueles que apresentam melhor *fitness*. A mais conhecida e utilizada forma de se fazer a seleção é a roleta, ou algoritmo Monte Carlo (DAVIS, 1996). Neste método, cada indivíduo da população é representado na roleta proporcionalmente ao seu índice de aptidão, como é mostrado na Figura 24. Assim, aos indivíduos com alta aptidão é dada uma porção maior da roleta, enquanto aos de aptidão mais baixa é dada uma porção relativamente menor da roleta. Finalmente, a roleta é girada um determinado número de vezes, dependendo do tamanho da população, e são escolhidos como indivíduos que participarão da próxima geração aqueles sorteados na roleta.

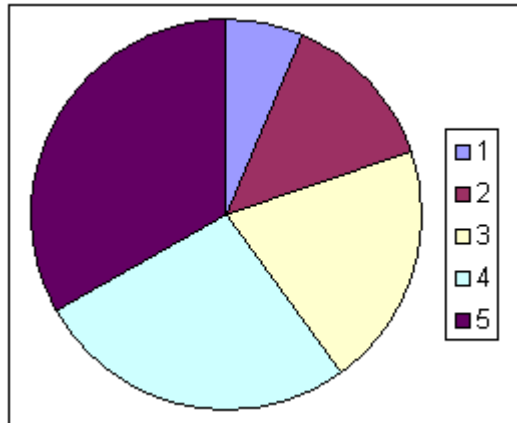


Figura 24: Exemplo de uma roleta de Seleção do Algoritmo Genético.

Existem diversas maneiras para melhorar a convergência dos Algoritmos Genéticos. O Elitismo é o método mais utilizado, sendo uma adição aos métodos de seleção, que os força a reter um certo número de “melhores” indivíduos em cada geração (YEPES, 2000). Tais indivíduos podem ser perdidos se eles não forem selecionados para reprodução ou se eles forem destruídos por cruzamento ou mutação.

5.2.3 O processo de cruzamento e mutação

O objetivo final dos operadores de cruzamento e mutação é fazer com que os cromossomos criados durante o processo de reprodução sejam diferentes dos cromossomos dos pais. O operador de cruzamento é responsável por combinar os cromossomos dos pais na criação dos cromossomos filhos, e o operador de mutação é responsável pela introdução de pequenas mudanças aleatórias nos cromossomos dos filhos.

A Figura 25 ilustra os operadores de seleção, cruzamento e mutação, intimamente relacionados no modelo básico de um algoritmo genético, que fazem a evolução de uma população acontecer.

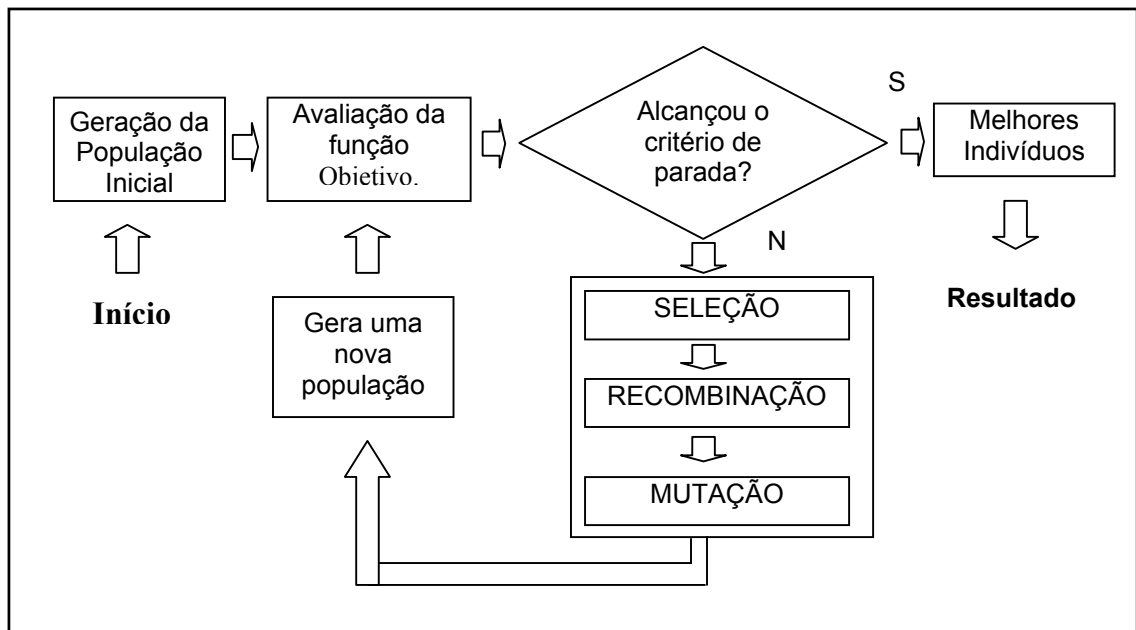


Figura 25: Representação gráfica de um Algoritmo Genético.

5.2.4 Parâmetros que influenciam no comportamento dos Algoritmos Genéticos

Alguns parâmetros influenciam no comportamento dos Algoritmos Genéticos, devendo ser estabelecidos conforme as necessidades do problema e dos recursos disponíveis. São eles:

- **Tamanho da População** - Determina o número de cromossomos na população, afetando o desempenho global e a eficiência dos Algoritmos Genéticos. Com uma população pequena o desempenho poderá cair, pois deste modo a população fornece uma pequena cobertura do espaço de busca do problema. Por outro lado, com uma grande população geralmente é fornecida uma cobertura representativa do domínio do problema, prevenindo também convergências prematuras para soluções locais ao invés de globais, entretanto, para se trabalhar com grandes populações são necessários maiores recursos computacionais, ou então, um maior tempo computacional para a obtenção dos resultados.

- Taxa de Cruzamento - Determina a probabilidade em que um cruzamento ocorrerá. Quanto maior for esta taxa, mais rapidamente novas estruturas serão introduzidas na população. Mas se ela for muito alta, a maior parte da população será substituída, e pode ocorrer perda de estruturas de alta aptidão. Com um valor baixo, o algoritmo pode tornar-se muito lento.
- Taxa de Mutação - Determina a probabilidade em que uma mutação ocorrerá. Uma baixa taxa de mutação previne que uma dada posição fique estagnada em um valor, causando uma convergência prematura, além de possibilitar que se chegue em qualquer ponto do espaço de busca. Com uma taxa muito alta a busca se torna essencialmente aleatória.

Os parâmetros genéticos citados anteriormente afetam diretamente o desempenho dos Algoritmos Genéticos. Os efeitos decorrentes da escolha inadequada destes parâmetros vão desde o aumento no tempo de convergência, convergência prematura, estagnação da busca, maior necessidade de recursos computacionais até a não convergência para uma solução viável.

5.2.5 Exemplos de aplicabilidade dos AGs em problemas de otimização

Os Algoritmos Genéticos podem ser utilizados em qualquer problema de otimização desde que devidamente modelados e observados os parâmetros que possam influenciar no seu comportamento, uma vez que muitos foram aplicados em tais problemas com sucesso (REEVES, 1993).

HOSAGE & GOODCHILD (1986) codificou as soluções do problema das p -medianas, como uma cadeia de n dígitos binários (genes). Esta codificação não garante a seleção exatamente de p instalações em cada solução. Os autores usam uma função de penalização para evitar que não sejam usadas mais ou menos facilidades do que o permitido. Devido a esta penalização, há uma perda na qualidade da solução final. Esta estratégia, porém, é interessante quando busca-se também reduzir o emprego das facilidades, que inclusive é o caso deste trabalho.

DIBBLE & DENSHAM (1993) descrevem um Algoritmo Genético multicritério para o problema de localização de facilidades e resolvem um problema com $n = 150$

e $p = 9$ com uma população de tamanho igual a 1000. Segundo os autores, o algoritmo fornece prováveis soluções quase tão boas quanto as soluções geradas por algoritmos de troca, embora exijam um maior esforço computacional.

MORENO-PEREZ *et al.*, (1994) desenvolveram um algoritmo paralelo genético para o problema das p -medianas, onde múltiplos grupos de população existem, ocorrendo uma troca entre os indivíduos dos grupos. Esta característica tem efeitos semelhantes para mutação, desde que previna-se a homogenização prematura nos grupos das populações. Os autores não informam os resultados computacionais.

BOZKAYA *et al.*, (2003) descrevem um Algoritmo Genético bastante complexo para o problema das p -medianas. Segundo os autores, este algoritmo pode produzir soluções que são melhores que as soluções de um algoritmo de troca. Porém, a convergência é muito lenta. Em contrapartida, o algoritmo desenvolvido neste trabalho é muito mais simples e produz rapidamente boas soluções.

6 METAHEURÍSTICA *GRASP* APLICADA AO PROBLEMA DE ALOCAÇÃO-LOCALIZAÇÃO DE ANTENAS

Este Capítulo apresenta uma abordagem da metaheurística *GRASP*, adaptada ao problema de alocação-localização de antenas. São apresentadas duas versões denominadas: *GRASP-ADD* e *GRASP-ADD-HÍBRIDA*.

A *GRASP-ADD* inicia o processo com todas as facilidades fechadas e vai adicionando (abrindo) facilidades que resultem em um mínimo valor da função objetivo. A *GRASP-ADD-HÍBRIDA* utiliza uma técnica híbrida, que mescla a solução atual com a melhor solução encontrada, aplicando na seqüência um procedimento *DROP*, que consiste em retirar as facilidades que não são comuns em ambas as soluções.

6.1 INTRODUÇÃO

O método *ADD* foi utilizado para a implementação da fase construtiva da *GRASP*. Esta primeira fase procura encontrar um número adequado de facilidades (antenas) a serem utilizadas para o atendimento a uma determinada demanda. Após a fase de construção é executada a segunda fase da *GRASP* (fase de busca local), que consiste em trocar as facilidades escolhidas na fase de construção com outras ainda não utilizadas.

Como este problema não é tratado especificamente como um problema das p -medianas, uma vez que o número de medianas varia, a fase de busca local poderá também inserir mais ou menos facilidades à solução entregue pela fase construtiva da *GRASP*. A cada troca efetua-se o cálculo da função objetivo, de forma a armazenar sempre a melhor solução.

O processo de acrescentar as facilidades (*ADD*) e a conseqüente busca local é repetido a cada iteração da *GRASP*. As iterações produzem uma relação custo / benefício, pois quanto maior o seu número, maior a probabilidade de se encontrar boas soluções, mas por outro lado, maior será o esforço computacional, principalmente em problemas de grande porte.

6.2 A FASE DE CONSTRUÇÃO DA *GRASP-ADD*

Considerando A_0 como sendo o conjunto de todos os locais candidatos à instalação das facilidades, a fase de construção da *GRASP-ADD* é inicializada com $A_1 = \emptyset$ (conjunto que irá retornar a solução construída, inicialmente com todas as facilidades fechadas) e função objetivo $f = \infty$ (um valor positivo muito grande). A heurística é então repetida da seguinte maneira: procure pela facilidade $j \in A_0$ cuja adição em A_1 produza um menor valor da função objetivo f . Se ao inserir a facilidade j houver uma redução em f , adicione j em A_1 . Caso contrário finalize o procedimento.

Evidentemente este método guloso não deve ser usado diretamente na *GRASP*, sendo completamente determinístico, gerando soluções idênticas em todas as iterações. Neste caso, a fase de construção *GRASP-ADD* utiliza o parâmetro α , que determina o nível de aleatoriedade e gulosidade durante o processo de abertura de facilidades. Na Figura 26 ilustra-se o pseudocódigo da fase de construção da *GRASP*.

Inicializando a fase de construção com $A_1 = \emptyset$ e $f = \infty$, para cada $j \in A_0$ calcula-se a função objetivo ao adicionar j a A_1 . Ordena-se o conjunto $A_0 = \{1, \dots, m\}$ de facilidades fechadas em ordem decrescente da função objetivo, sendo m o número de locais candidatos para a instalação de facilidades. O conjunto A_0 é denominado *lista de candidatos*. A cada iteração da fase construtiva, uma facilidade j é selecionada aleatoriamente de um subconjunto $A_2 = \{1, \dots, x\} \subseteq A_0$, (Figura 27), e adicionada no conjunto A_1 : $A_0 = A_0 - \{j\}$, $A_1 = A_1 \cup \{j\}$. O conjunto A_2 é denominado *lista de candidatos restrita (LCR)* e seu tamanho é definido como $x = \max(1, \alpha \times m)$, onde $\alpha \in [0, 1]$ é o parâmetro de aleatorização. Note que se $\alpha = 0$, a facilidade a ser escolhida é $j = 1$ (escolha gulosa). Se $\alpha = 1$, tem-se $p = m$ e a escolha da facilidade será de forma totalmente aleatória. A fase construtiva finaliza quando não for mais possível minimizar a função objetivo f ao inserir facilidades no conjunto A_1 .

```

Procedimento CONSTRUÇÃO-GRASP ();
1:  $f^* \leftarrow \infty$ ;
2:  $\text{melhor} \leftarrow \text{true}$ ;
3:  $A_1 \leftarrow \emptyset$ ;
4:  $A_2 \leftarrow \text{máx}(1, \alpha \times |A_0|)$ ;
5: enquanto ( $\text{melhor} = \text{true}$ ) faça:
6:    $j \leftarrow \text{rand}(0, A_2)$ ;
7:    $A_1 \leftarrow A_1 \cup \{j\}$ ;
8:   se ( $f(A_1) < f^*$ ) então
9:      $f^* \leftarrow f(A_1)$ ;
10:     $A_0 \leftarrow A_0 - \{j\}$ ;
11:     $A_2 \leftarrow \text{máx}(1, \alpha \times A_0)$ ;
12:   senão
13:      $\text{melhor} \leftarrow \text{false}$ ;
14:      $A_1 \leftarrow A_1 - \{j\}$ ;
15:   fim_se
16: fim_enquanto
17: retorne  $A_1$ ;
Fim Procedimento

```

Figura 26. Pseudocódigo da fase de construção *GRASP*.

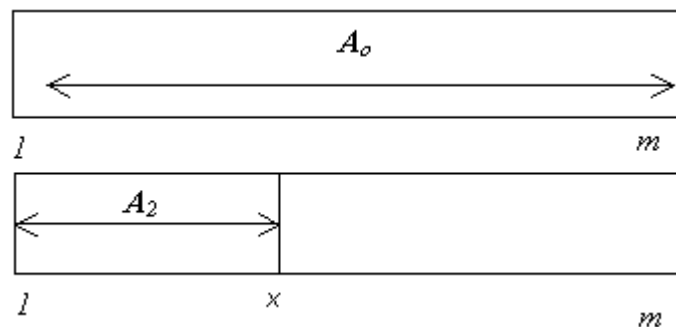


Figura 27. Lista de candidatos e lista de Candidatos restrita da *GRASP*.

Do ponto de vista apenas da fase construtiva, o acréscimo das facilidades pertencentes à *LCR* dá-se de forma gradual visando buscar um ponto de equilíbrio que consiga atender a uma maior quantidade de pontos de demanda. Isto pode ser observado na Figura 28, que ilustra os acréscimos gradativos correspondentes à fase de construção. Inicialmente é usada apenas uma antena (a), que deixa de atender a diversos pontos de demanda (destacados na cor vermelha).

O acréscimo de mais antenas (b) e (c), melhora o atendimento dos pontos de demanda. O processo de busca por solução pára, quando o acréscimo da próxima

facilidade não mais melhorar o valor final da função objetivo. A fase de busca local terá início a partir desta solução viável, gerada na fase de construção. A Figura 29 exibe uma solução já melhorada pela fase de busca local.

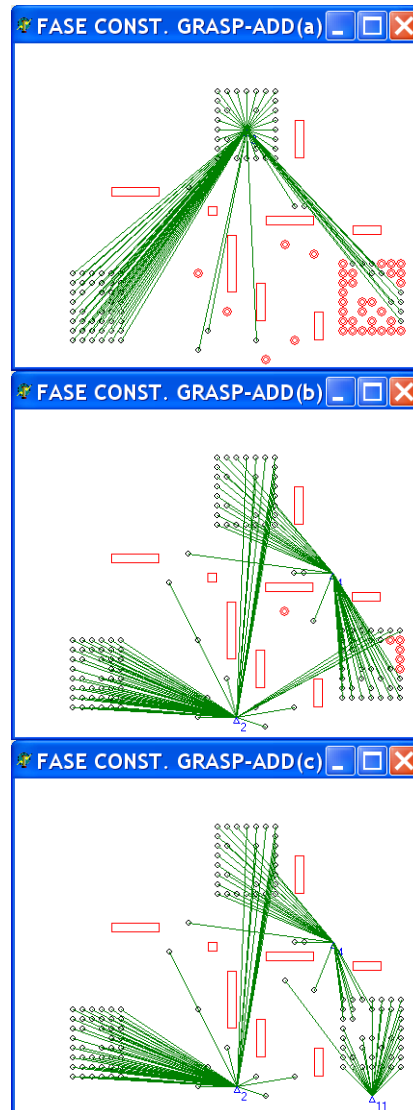


Figura 28. Evolução gráfica da fase de construção da *GRASP-ADD*.

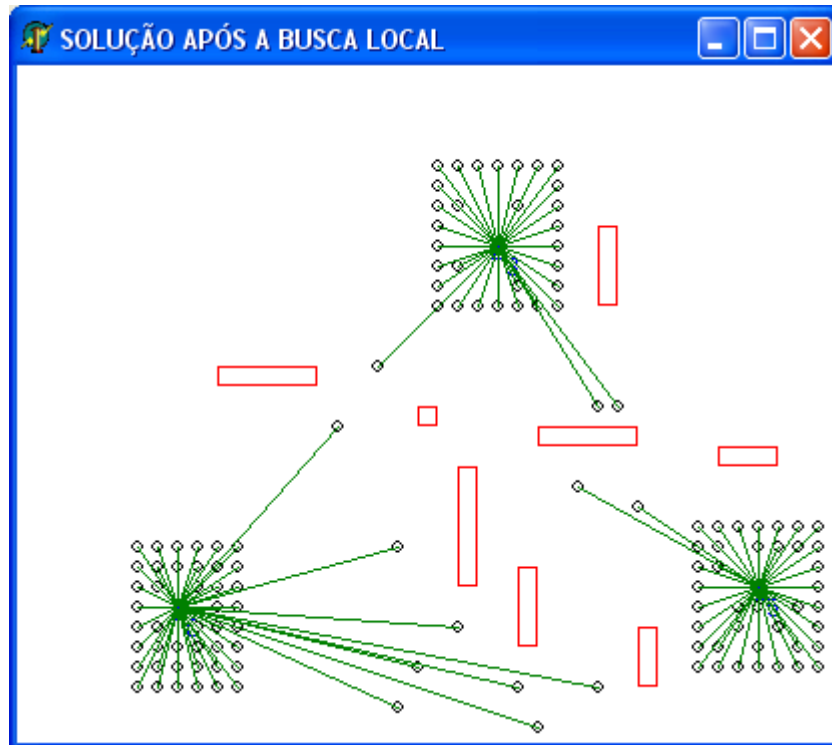


Figura 29: Solução gráfica já melhorada pela fase de busca local para uma instância hipotética de problema.

6.3 FASE DE BUSCA LOCAL

Métodos de busca local em problemas de otimização constituem uma família de técnicas baseadas na noção de vizinhança, ou seja, são métodos que exploram o espaço de soluções passando, iterativamente, de uma solução viável para outra que seja sua vizinha.

Geralmente, uma busca local refina uma solução viável encontrada por uma heurística. A heurística explora o espaço de busca procurando regiões promissoras, a busca local explora essas regiões procurando ótimos locais.

Assim como em muitas técnicas determinísticas, as prováveis soluções geradas pela fase de construção *GRASP* não são localmente ótimas com respeito à definição de vizinhança adotada. Daí a importância da fase de busca local, a qual objetiva melhorar a solução viável, inicialmente construída.

Pelo fato do número de facilidades a serem usadas não ser fixo, este trabalho, na fase de busca local utiliza três tipos de vizinhanças, descritas a seguir:

- Vizinhança 1: uma solução vizinha é determinada adicionando na solução corrente duas novas facilidades e removendo uma.
- Vizinhança 2: uma solução vizinha é determinada adicionando na solução corrente uma nova facilidade e removendo duas.
- Vizinhança 3: uma solução vizinha é determinada adicionando na solução corrente uma nova facilidade e removendo outra.

A cada iteração da *GRASP* são escolhidas aleatoriamente as vizinhanças 1 ou 2, uma vez que não é conhecido o número correto de facilidades a serem usadas. Ao final de todas as iterações é aplicada uma busca local à melhor solução encontrada com o objetivo de refinar a qualidade da solução. Esta busca local usa a vizinhança 3, ou seja é fixado o número de facilidades.

A Figura 30 exibe o pseudocódigo da *GRASP-ADD*. A cada iteração inicia-se a fase de construção (linha 4) e na seqüência é selecionada aleatoriamente a vizinhança1 ou a vizinhança2 para a execução da fase de busca local (linhas 5 e 6). Ao final das iterações, aplica-se uma busca local fixa representada pela vizinhança 3 (linha 10), sendo retornada a melhor solução viável encontrada (linha 11).

```

Procedimento GRASP-ADD ();
1:  $i \leftarrow 1$ ;
2:  $f^* \leftarrow \infty$ ;
3: enquanto ( $i \leq \text{iterações\_grasp}$ ) faça:
4:    $x \leftarrow \text{CONSTRUÇÃO-GRASP}()$ ;
5:    $k \leftarrow \text{Escolha aleatoriamente uma vizinhança entre 1 e 2}$ ;
6:    $x' \leftarrow \text{BuscaLocal\_Vizinhança}_k(x)$ ;
7:   se  $f(x') < f^*$  então  $x^* \leftarrow x'$ ;
8:    $i \leftarrow i + 1$ ;
9: fim\_enquanto
10:  $x'' \leftarrow \text{BuscaLocal\_Vizinhança}_3(x^*)$ ;
11: retorne  $x''$ ;
Fim Procedimento

```

Figura 30: Pseudocódigo da *GRASP-ADD*.

A Figura 31 apresenta o pseudocódigo da busca local *Vizinhança1* que consiste em testar a troca das facilidades contidas em A_1 com todas as facilidades não escolhidas (FNS) em A_0 (linhas 6 à 8). Observe que a troca é efetuada a uma proporção de duas facilidades retiradas de FNS , para uma facilidade retirada de A_1 . A cada troca efetua-se o cálculo da função objetivo, de forma a armazenar sempre a solução viável com o menor valor (linha 9). Este processo é repetido enquanto existirem facilidades a serem adicionadas, desde que o valor da função objetivo melhore (linha 3).

```

Procedimento Busca_Local_Vizinhança_1(s)
  // s = A1 (facilidades utilizadas)
  1: melhorou ← true;
  2: r ← 0;
  3: Enquanto (melhorou = true and |FNS| > 1) faça
  4:   melhorou ← false;
  5:   remova  $A_1[r]$ ;
  6:   Para (x = 0,1,2,..., |FNS| - 2) faça
  7:     Para (y = x+1, x+2,..., |FNS| - 1) faça
  8:       Adicione  $FNS[x]$  e  $FNS[y]$  em  $A_1$ ;
  9:       se  $f(A_1) < f(s)$  então
 10:         melhorou ← true;
 11:         retorne com as facilidades retiradas de  $FNS$ ;
 12:         k ← x;
 13:         z ← y;
 14:         s ←  $A_1$ ;
 15:       fim_se
 16:     fim_para
 17:   fim_para
 18:   Se melhorou = true então
 19:     Adicione definitivamente  $FNS[k]$  e  $FNS[z]$  em  $A_1$ ;
 20:     r = r + 1;
 21:   fim_se
 22: fim_enquanto
 23: retorne s;
Fim procedimento;

```

Figura 31. Pseudocódigo da busca local que adiciona duas facilidades e remove uma (*Vizinhança1*).

A Figura 32 apresenta o pseudocódigo da busca local *Vizinhança2*, que de forma análoga remove duas facilidades de A_1 e adiciona uma. A cada troca é efetuado o cálculo da função objetivo (linha 9) de forma a guardar sempre o melhor valor. O processo se repete, enquanto houverem facilidades a serem retiradas em A_1 , desde que tenha ocorrido uma melhoria na função objetivo.

```

Procedimento Busca_Local_Vizinhança_2(s)
  // s = A1 (facilidades utilizadas)
  1: melhorou ← true;
  2: r ← 0;
  3: Enquanto (melhorou = true and |A1| > 1) faça
  4:   melhorou ← false;
  5:   Para (x = 0, 1, 2, ..., |A1| - 2) faça
  6:     Para (y = x+1, x+2, ..., |A1| - 1) faça
  7:       Remova A1[x] e A1[y];
  8:       Adicione FNS[r] em A1;
  9:       se f(A1) < f(s) então
 10:         melhorou ← true;
 11:         retorne com as facilidades retiradas de A1;
 12:         retorne com a facilidade retirada de FNS;
 13:         k ← x;
 14:         z ← y;
 15:         s ← A1;
 16:       fim_se
 17:     fim_para
 18:   fim_para
 19:   Se melhorou = true então
 20:     Remova definitivamente A1[k] e A1[z] de A1;
 21:     Adicione definitivamente FNS[r] em A1;
 22:     r = r + 1;
 23:   fim_se
 24: fim_enquanto
 25: retorne s;
Fim Procedimento;

```

Figura 32. Pseudocódigo da busca local que remove duas facilidades e adiciona uma (*Vizinhança2*).

A Figura 33 apresenta o pseudocódigo da busca local *Vizinhança3*, que é executada ao final de todas as interações *GRASP*, quando é retornada a melhor solução viável encontrada. Observe que neste caso, não são adicionadas e nem retiradas facilidades de *A*₁. Elas são apenas trocadas com as facilidades não sorteadas (*LFNS*) em *A*₀. Enquanto o valor final da função objetivo melhorar, atualiza-se *A*₁ (linhas 16 à 21) efetuando-se neste ponto, a troca da próxima facilidade de *A*₁ com seus vizinhos (linhas 4 à 6). O processo repete-se, testando todos os vizinhos de todas as facilidades contidas em *A*₁, parando somente quando após todas as trocas, não for encontrada uma solução melhor (linha 2). Por último, retorna-se o melhor resultado encontrado (linha 24).

```

Procedimento BuscaLocal_Vizinhança_3( $A_1$ ,  $FNS$ )
//  $s = A_1$  (facilidades utilizadas)
1:  $cont \leftarrow true$ ;
2: Enquanto ( $cont = true$ ) faça
3:    $cont \leftarrow false$ ;
4:   Para ( $x = 1, 2, 3, \dots, |FNS|$ ) faça
5:      $melh \leftarrow false$ ;
6:     Para ( $i = 1, 2, 3, \dots, |A_1|$ ) faça
7:        $k \leftarrow A_1[i]$ ;
8:        $A_1[i] \leftarrow FNS[x]$ ;
9:       Se  $f(A_1) < f(s)$  então
10:         $y \leftarrow i$ ;
11:         $melh \leftarrow true$ ;
12:         $f(s) \leftarrow f(A_1)$ ;
13:      fim_se
14:       $A_1[i] \leftarrow k$ ;
15:    fim_para
16:    Se  $melh = true$  então
17:       $A_1[y] \leftarrow FNS[x]$ ;
18:      atualize  $FNS$ ;
19:       $cont \leftarrow true$ ;
20:       $s \leftarrow A_1$ ;
21:    fim_se
22:  fim_para
23: fim_enquanto
24: retorne  $s$ ;
Fim_Procedimento

```

Figura 33. Pseudocódigo da busca local *Vizinhança3*.

A Figura 34 compara uma solução viável encontrada na fase de construção da *GRASP-ADD* com outra encontrada, após todas as iterações *GRASP* e aplicação da fase de busca local fixa (*Vizinhança3*).

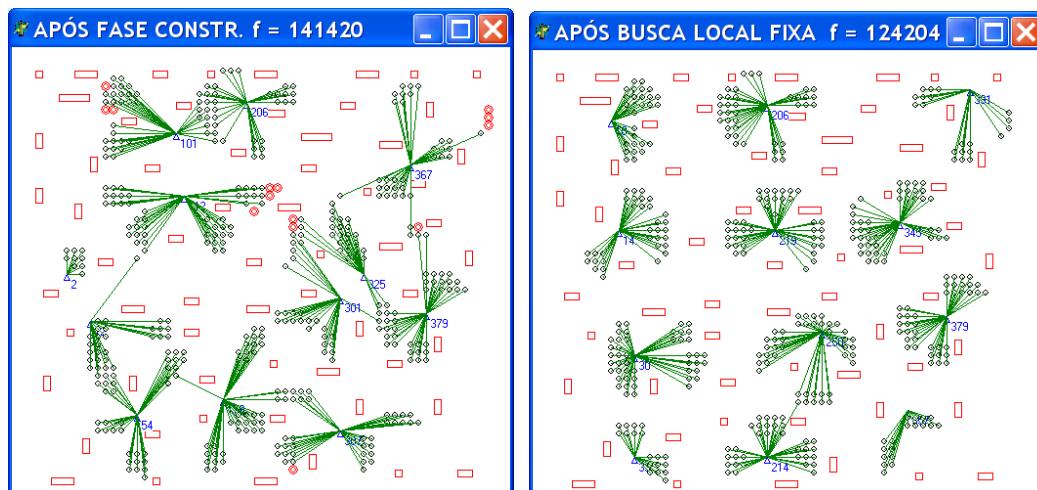


Figura 34. Comparação gráfica entre a fase de construção *ADD* e a fase de busca local.

6.4 A GRASP HÍBRIDA

Técnicas híbridas são apresentadas por diversos autores para a solução de problemas NP-completos, com a finalidade de melhorar ainda mais as soluções encontradas. Como exemplo, pode-se citar o trabalho de DELMAIRE *et al.* (1998), que combina estratégias da busca tabu com técnicas que mesclam elementos da *GRASP* e da busca tabu para resolver o problema de localização de facilidades capacitadas. A heurística *GRASP* proposta pelos autores, baseia-se em dois estágios distintos: o primeiro seleciona as facilidades, podendo ser considerado como uma fase construtiva, onde muitos conjuntos de facilidades abertos são selecionados, e uma designação inicial é obtida no segundo estágio, que corresponde à fase de melhoria. A busca tabu utiliza estratégias de diversificação e intensificação. Os autores construíram dois conjuntos de diferentes problemas. O primeiro contendo 33 problemas com 20 facilidades, no máximo, e 50 clientes. O segundo que trata de problemas um pouco maiores, com tamanho máximo de 30 facilidades e 90 clientes. Os resultados mostram que a busca tabu é muito promissora, mesmo se aplicada apenas com memória de curto prazo. Os melhores resultados ocorreram para os procedimentos híbridos.

Recentemente RESENDE & WERNECK (2005) apresentaram uma heurística de multi-reinício para o problema de localização de facilidades não capacitadas, baseando-se em um método originalmente desenvolvido para resolver o problema das p -medianas. Foi utilizada a técnica *Path-relinking* que é um procedimento de intensificação, comumente utilizado com a busca tabu, mas freqüentemente empregado em outros métodos como a *GRASP*. Esta técnica consiste em inicialmente selecionar duas soluções, S_a e S_b . O algoritmo começa com a solução viável S_a e gradualmente a transforma em S_b . As operações que transformam S_a em S_b , são as mesmas utilizadas na busca local: inserções, remoções, e trocas. Soluções de elite, contendo uma coletânea de boas soluções, são atualizadas baseando-se na noção da diferença simétrica entre duas soluções S_a e S_b , definida como $|S_a / S_b| + |S_b / S_a|^2$.

Neste trabalho, no algoritmo *GRASP* é inserida uma técnica que consiste na combinação de soluções como em um Algoritmo Genético. A solução viável X retornada pela Busca Local é unida à melhor solução armazenada (Y), dando origem

à solução Z . Na seqüência aplica-se um procedimento *DROP* em Z , retornando-a em sua forma reduzida.

6.4.1 A *GRASP-ADD-HÍBRIDA*

A *GRASP-ADD-HÍBRIDA* consiste em selecionar uma solução inicial X , entregue pela fase de busca local *GRASP-ADD*, a cada iteração, contendo algumas facilidades abertas e em seguida aplicar o procedimento de cruzamento, que une a solução viável X , obtida pela fase de busca local, com a melhor solução já encontrada (Y), gerando-se outra solução (Z). Após o cruzamento, aplica-se um procedimento *DROP* em Z , que fecha as facilidades que não são comuns à X e Y . O procedimento *DROP* repete-se enquanto o valor final da função objetivo melhorar. Ao final deste processo, retorna-se Z na forma reduzida. A solução Y que será usada no procedimento Crossover da próxima iteração é a melhor das soluções viáveis já encontradas até o momento.

A Figura 35 (a e b) compara uma solução viável X com a melhor solução encontrada Y em uma dada iteração *GRASP*. Observa-se que o valor da função objetivo em (a) é ligeiramente superior ao valor da função objetivo em (b), embora ambas as soluções utilizem o mesmo número de facilidades e atendam a todos os pontos de demanda. O valor de f é menor em (b) porque as facilidades estão melhor alocadas, ficando mais próximas dos pontos de demanda. A união de X e Y é exibida em (c) e, logicamente não apresenta bons resultados devido ao elevado número de facilidades usadas. Mostra-se em (d) os melhores resultados obtidos, correspondendo à forma reduzida de Z .

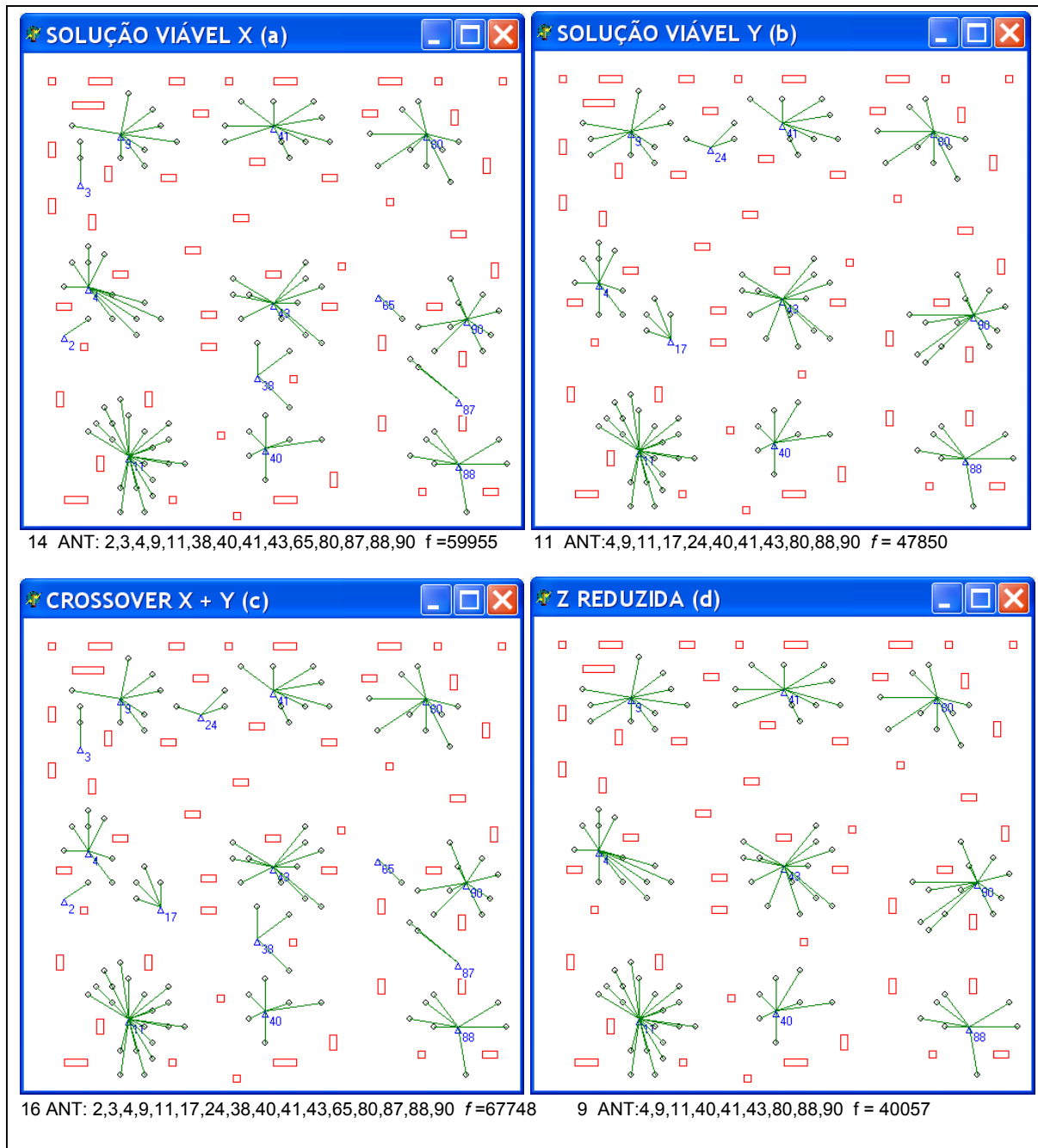


Figura 35: Comparativo entre as soluções X, Y e Z.

O pseudocódigo mostrado na Figura 36 sintetiza o procedimento *União-Drop*. Este procedimento efetua o cruzamento de X e Y originando a solução Z, (linha 1). Na sequência, gera o conjunto F , que contém as facilidades que não são comuns à Y e X (linha 3) e efetua um procedimento *DROP*, retirando as facilidades de Z pertencentes ao conjunto F (linhas 6 à 8). Este procedimento é repetido enquanto o valor da função objetivo melhorar (linha 5).

Sejam:

$X = \{1, \dots, m1\}$, solução entregue pela fase de busca local *GRASP*.

$Y = \{1, \dots, m2\}$, melhor solução encontrada.

$Z = X \cup Y$, solução X unida com a solução Y .

Procedimento União-Drop (Y, X)

```

1: Gere a solução  $Z$  unindo as facil. abertas de  $Y$  e  $X$  ( $Z = Y \cup X$ );
2: Calcule o valor da função objetivo da nova solução  $Z$ :  $f(Z)$ ;
3: Defina o conjunto de facilidades:  $F = Y \cup X - Y \cap X$ ;
4:  $Melhora = true$ ;
5: enquanto  $Melhora = true$  faça
6:   Determine a facilidade  $j \in F$  que produza o menor incremento da
   função objetivo quando  $j$  é removida da solução  $Z$ .
7:   se existe a facilidade  $j$  então
8:      $Z = Z - \{j\}$ ;
9:   senão
10:     $Melhora = false$ ;
11:   fim_se;
12: fim_enquanto;
13: retorne  $Z$ ;
Fim Procedimento;

```

Figura 36. Pseudocódigo do procedimento União-DROP da *GRASP-ADD- HÍBRIDA*.

A Figura 37 exhibe o pseudocódigo que sintetiza toda a heurística *GRASP-ADD-HÍBRIDA*. São fornecidos como dados de entrada o número máximo de iterações a serem executadas (N_{iter}). A cada iteração t , o procedimento *Construção-grasp* constrói uma solução X e em seguida esta solução é submetida a um procedimento *busca-local* obtendo uma solução melhorada X^* (linhas 03 e 04). A partir da segunda iteração a solução X^* é unida com a melhor solução Y encontrada até o momento, obtendo uma solução Z . Após de fazer a união é aplicada um procedimento de redução (*Drop*) na nova solução Z , que consiste em excluir ou fechar facilidades, de forma gulosa, enquanto o valor da função objetivo seja melhorado (linha 06). A cada iteração a melhor solução encontrada sempre será armazenada (linhas 05 e 07, respectivamente).

```

Procedimento GRASP_ADD_HIBRIDA ( $N\_iter$ )
1:  $f(Y) \leftarrow \infty$ ;
2: Para  $t \leftarrow 1$  até  $N\_iter$  faça
3:    $X = \text{construção\_grasp}()$ ;
4:    $k \leftarrow \text{Escolha aleatoriamente uma vizinhança entre 1 e 2}$ ;
5:    $X^* \leftarrow \text{BuscaLocal\_Vizinhança\_k}(X)$ ;
6:   se  $t > 1$  então
7:      $Z \leftarrow \text{União-Drop}(Y, X^*)$ ;
8:     se  $f(Z) < f(Y)$  então  $Y \leftarrow Z$ ;
9:   fim_se
10: fim_Para;
11: Retorne  $Y$ ;
Fim Procedimento

```

Figura 37. Pseudocódigo da heurística *GRASP-ADD- HÍBRIDA*.

6.5 CONSIDERAÇÕES SOBRE A GRASP

A *GRASP*, se corretamente adaptada ao problema, apresenta bons resultados em baixo tempo computacional. O fator α que mede o nível de aleatoriedade e gulosidade no sorteio das facilidades, deve ser devidamente calibrado, pois influi na qualidade das soluções obtidas. Para um α muito pequeno corre-se o risco de se sortear muitas vezes a mesma solução. Por outro lado, adotando-se um alto valor de α , soluções muito aleatórias podem ser geradas na fase de construção, contendo facilidades que pouco melhorem o valor da função objetivo.

A fase de busca local possui um papel importante na melhoria da qualidade das soluções, uma vez que faz a troca com os vizinhos, mas ela não necessariamente usa o mesmo número de facilidades entregues pela fase de construção *GRASP*, uma vez que tal número é desconhecido.

Foi aplicado após todas as iterações uma busca local fixa pelo fato de explorar um universo de vizinhos maiores que as buscas locais anteriores. Este último processo tem a finalidade de refinar um pouco mais a qualidade das soluções, geralmente aproximando um pouco mais as antenas dos pontos de demanda.

O método híbrido, que combina a solução *ADD*, com a melhor solução, apresenta na maioria dos casos, resultados melhores do que o não híbrido, embora

exija mais esforço computacional. Estas considerações podem ser observadas no Capítulo 7 que exhibe os testes computacionais realizados em diversas instâncias de problemas.

7 TESTES COMPUTACIONAIS COM O USO DA GRASP

Neste Capítulo é comparada a qualidade das soluções obtidas pelos algoritmos *GRASP-ADD* e *GRASP-ADD-HÍBRIDO*. Foram gerados 24 problemas de médio porte, contendo até 400 pontos de demanda e locais candidatos à instalação de facilidades, e 18 problemas de grande porte, onde foram gerados entre 500 e 1000 pontos de demanda e locais candidatos. Dos 42 problemas, 21 foram gerados manualmente (conjunto 1 de problemas) e 21 de forma aleatória (conjunto 2 de problemas).

Os problemas manuais são hipotéticos e foram gerados inserindo-se estrategicamente obstáculos, pontos de demanda e locais candidatos com o objetivo de tornar difícil a busca por uma boa solução. Se um obstáculo estiver entre um ponto de demanda e um local candidato, sempre causará interferência nos problemas manuais.

As instâncias de problemas aleatórios geraram obstáculos, pontos de demanda e locais candidatos, com alturas variáveis, o que torna possível em muitos casos, o atendimento a um dado ponto de demanda por uma facilidade, pelo fato do sinal transmitido pela antena passar por cima dos obstáculos.

Para os conjuntos 1 e 2 de problemas, considerou-se uma cidade com 33 Km de comprimento por 30 Km de largura. As antenas utilizadas são onidirecionais, com um alcance máximo de 10 km. O custo da instalação de uma facilidade, adotado para todas as instâncias de problemas testados, é de 7000.

O nome dos problemas testados inseridos nas Tabelas, informam a quantidade de pontos de demanda e locais candidatos, respectivamente. Por exemplo, o problema de nome 100_100, contém 100 pontos de demanda e 100 locais candidatos para a instalação de facilidades.

Os algoritmos foram implementados utilizando-se a linguagem Object Pascal do Delphi 7.0, com o uso de um computador Sempron 3.4 Ghz, 1G Mb de RAM.

7.1 ANÁLISE DO PARÂMETRO α

O desempenho da *GRASP* foi testado usando os valores de $\alpha = 0,0; 0,1; 0,2; 0,3; 0,5; 0,75$ e $1,0$. Os testes foram realizados usando a *GRASP-ADD* e *GRASP-ADD-HÍBRIDA*, com 100 iterações, nos conjuntos 1 e 2 de problemas. Os resultados estão mostrados nas Tabelas 2 e 3, respectivamente. Os valores destacados em negrito representam as médias aritméticas das melhores soluções obtidas por ambas as implementações. O objetivo é encontrar o melhor valor de α e usá-lo igualmente em todos os testes computacionais da *GRASP*.

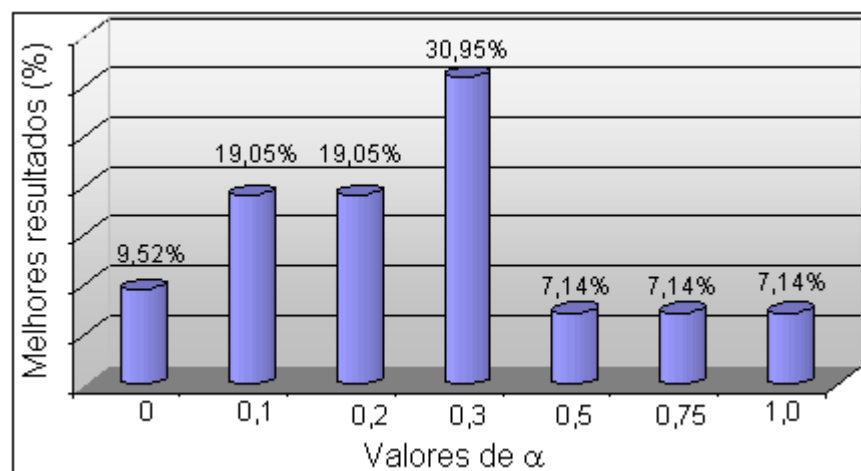
Tabela 2. Valores médios da função objetivo em relação ao parâmetro α (conjunto 1).

Conjunto 1 (n_m)		VALORES DE α						
		0	0,1	0,2	0,3	0,5	0,75	1
1		73086	31630	35288	29794	33876	30818	31626
2	100_100	24422	44598	22688	24440	23042	22430	24422
3		39020	39658	39508	39015	42006	41628	42056
4		170435	144800	99864	95342	100669	100668	96950
5	200_200	67126	65790	67176	69186	72253	65232	73242
6		146814	52900	60070	56264	53639	53442	63329
7		87526	89366	98494	91604	93314	101010	90036
8	300_300	103614	105788	102253	107800	115715	109777	107638
9		289914	227404	158176	159690	159746	159967	156741
10		125447	118735	148966	134174	124001	122910	136523
11	400_400	104632	100738	100544	103972	104694	106590	104283
12		125154	127756	134871	123582	123010	144064	132140
13		138238	142498	131627	137916	144048	137145	135931
14	500_500	139198	144146	134164	129839	140936	144812	145018
15		417774	150378	215179	152830	159616	169612	170056
16		177860	204821	177861	165954	183474	171644	213058
17	800_800	130178	149381	132956	132936	125084	132995	138486
18		157595	146454	158049	146272	186531	140617	169567
19		248584	237782	377780	242830	233999	238074	227776
20	1000_1000	169020	191986	186155	186159	186115	171870	203288
21		147680	331334	169197	164032	151418	242469	177284

Tabela 3. Valores médios da função objetivo em relação ao parâmetro α (conjunto 2).

Conjunto 2 (n_m)		VALORES DE α						
		0	0,1	0,2	0,3	0,5	0,75	1
1	100_100	58634	56929	57018	56159	59857	57370	57018
2		57935	58632	58380	60936	59572	57794	58774
3		74574	51979	49398	49981	51280	49690	51280
4	200_200	77885	69573	70948	68929	70948	70920	75324
5		76006	75482	71845	70148	71845	73858	75316
6		78085	79069	75732	77885	79369	77903	76502
7	300_300	97094	94302	98512	96385	96616	105246	96525
8		108669	104448	105574	111076	106900	109860	111826
9		124240	124357	124178	123183	126982	126158	123040
10	400_400	132808	133199	136380	129062	137082	168278	135562
11		129147	133794	128206	124870	165989	131648	148781
12		125753	129247	128610	133847	134327	140746	128296
13	500_500	147209	140102	148727	236149	171086	155416	148487
14		196018	193577	203972	226420	197914	196710	199820
15		172736	185084	172252	175649	175214	199256	181206
16	800_800	289675	285860	280024	314921	289605	295910	286760
17		231376	235336	236666	238649	256394	249170	249768
18		234516	233081	247174	248301	235868	250630	235896
19	1000_1000	321670	348462	349371	317402	320732	326542	310420
20		279084	277740	287830	277014	284254	277864	283496
21		285415	279008	282768	272002	278625	296748	290053

A Figura 38 mostra o percentual de melhores soluções encontradas para os valores testados de α . Fazendo-se $\alpha = 0,3$, os melhores resultados foram encontrados em 30,95 % dos problemas. Este valor foi usado para todas as implementações da *GRASP*.

Figura 38. Comparativo para diversos valores de α da *GRASP*.

7.2 COMPARAÇÕES ENTRE AS IMPLEMENTAÇÕES GRASP

Fazendo-se $\alpha = 0,3$, as instâncias de problemas dos conjuntos 1 e 2 foram testadas com 200 iterações e os resultados computacionais comparados.

7.2.1 Testes computacionais no conjunto 1 de problemas

O conjunto 1 de problemas foi testado comparando-se os valores da função objetivo (f), tempo computacional em horas, minutos e segundos (t), total de pontos de demanda atendidos (PA) e total de facilidades usadas (TF), mostrados nas Tabelas 4 e 5. Os valores destacados em negrito representam a melhor solução encontrada.

Tabela 4. Testes computacionais realizados com o uso da heurística GRASP nas instâncias de problemas de médio porte do conjunto 1.

Conjunto 1 (Médio Porte)	ADD				ADD HÍB.			
	PA	TF	t	f	PA	TF	t	f
1	100	7	0:00:01	30818	99	6	0:00:01	28771
2 100_100	100	13	0:00:01	27525	100	8	0:00:01	20144
3	100	9	0:00:01	40057	95	7	0:00:01	38249
4	200	8	0:00:03	94571	187	7	0:00:03	96113
5 200_200	200	16	0:00:04	69186	195	13	0:00:05	63287
6	200	7	0:00:02	62183	198	5	0:00:02	50346
7	300	10	0:00:12	91472	285	8	0:00:13	91736
8 300_300	299	12	0:00:15	109721	296	11	0:00:15	105786
9	300	34	0:00:38	159878	276	28	0:00:44	157062
10	400	6	0:00:23	133298	372	4	0:00:24	123700
11 400_400	400	12	0:00:28	109811	382	8	0:00:31	101217
12	400	12	0:00:37	124204	396	11	0:00:39	120355

Tabela 5. Testes computacionais realizados com o uso da heurística *GRASP* nas instâncias de problemas de grande porte do conjunto 1.

Conjunto 1 (Grande Porte)		ADD				ADD HÍB.			
		PA	TF	t	f	PA	TF	t	f
13	500_500	500	11	0:00:46	153094	500	8	0:00:51	122737
14		500	12	0:01:02	157314	484	7	0:01:16	121083
15		500	10	0:01:07	151473	493	9	0:01:10	147341
16	800_800	800	12	0:03:19	189052	800	8	0:04:20	142856
17		800	21	0:07:25	149224	800	14	0:09:30	111149
18		800	10	0:03:51	163425	800	7	0:04:57	129120
19	1000_1000	1000	19	0:12:16	259895	1000	16	0:15:18	225764
20		1000	13	0:07:19	208821	1000	9	0:08:25	163554
21		1000	13	0:11:28	172445	1000	10	0:11:20	147384

Com base nos resultados apresentados nas Tabelas 4 e 5, referentes às instâncias de problemas do conjunto 1, observa-se que o tempo computacional aumenta à medida que os pontos de demanda aumentam e principalmente quando é adicionado um maior número de locais candidatos à instalação de facilidades, como pode ser observado no gráfico exibido na Figura 39. O tempo computacional gasto pela heurística *GRASP-ADD-HÍBRIDA* é ligeiramente superior à *GRASP-ADD*, o que vale a pena, tendo em vista os resultados obtidos.

Com relação ao valor final da função objetivo, a *GRASP-ADD-HÍBRIDA* apresentou melhores resultados em 90,47 % dos casos, embora tenha atendido à uma quantidade de pontos de demanda menor ou igual à *GRASP-ADD*. Em todos estes casos foi utilizada uma menor quantidade de antenas.

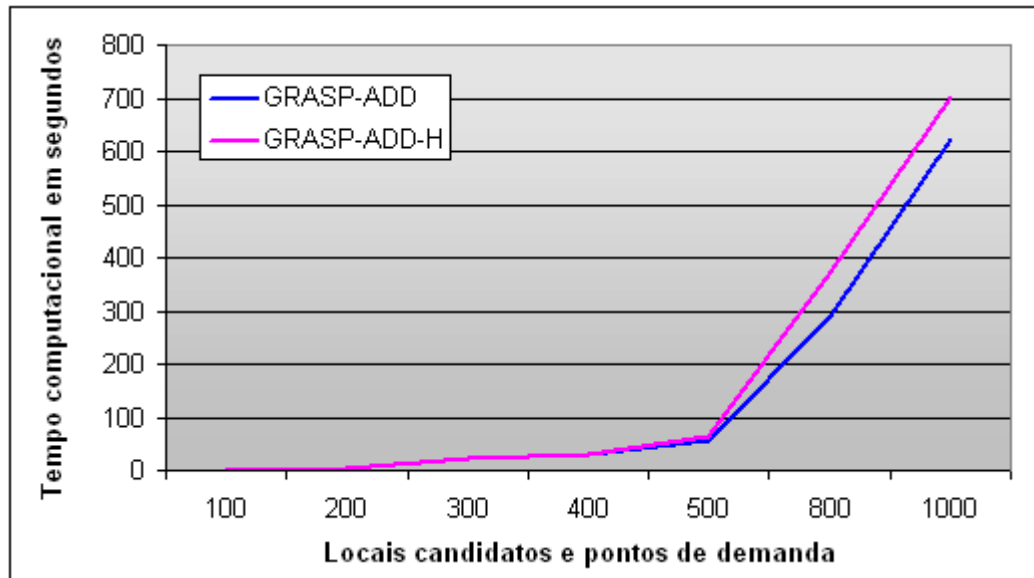


Figura 39. Tempo médio computacional gasto pelas implementações *GRASP* para instâncias de problemas do conjunto1 entre 100_100 e 1000_1000.

7.2.2 Testes computacionais no conjunto 2 de problemas

O conjunto 2, que contém as instâncias de problemas geradas aleatoriamente, foi testado comparando-se os valores da função objetivo (f), tempo computacional em horas, minutos e segundos (t), total de pontos de demanda atendidos (PA) e total de facilidades usadas (TF), mostrados nas Tabelas 6 e 7.

Tabela 6. Testes computacionais realizados com o uso da heurística *GRASP* nas instâncias de problemas de médio porte do conjunto 2.

Conjunto 2 (Médio Porte)		ADD				ADD HÍB.			
		PA	TF	t	f	PA	TF	t	f
1	100_100	91	5	0:00:01	56159	91	5	0:00:01	56159
2		92	6	0:00:01	56641	85	5	0:00:01	57775
3		88	5	0:00:01	49981	88	5	0:00:01	49981
4	200_200	197	9	0:00:05	71568	191	7	0:00:05	68072
5		199	11	0:00:04	81899	185	6	0:00:05	70113
6		197	10	0:00:05	79968	187	7	0:00:05	75802
7	300_300	298	12	0:00:15	98474	289	9	0:00:15	93425
8		299	13	0:00:20	108587	281	8	0:00:21	106155
9		294	18	0:00:19	126306	282	14	0:00:20	117319
10	400_400	399	23	0:01:09	140503	391	17	0:01:09	125112
11		397	22	0:00:59	139100	381	15	0:01:05	120085
12		396	22	0:01:02	137837	370	15	0:01:02	129857

Tabela 7. Testes computacionais realizados com o uso da heurística *GRASP* nas instâncias de problemas de grande porte do conjunto 2.

Conjunto 2 (Grande Porte)		ADD				ADD HÍB.			
		PA	TF	t	f	PA	TF	t	f
13	500_500	498	25	0:01:39	153494	488	17	0:01:51	133725
14		484	19	0:01:23	202991	441	12	0:01:35	188857
15		497	22	0:01:50	183873	474	16	0:02:16	168750
16	800_800	787	23	0:07:44	287508	733	16	0:09:06	277111
17		798	21	0:05:59	254485	782	15	0:06:40	216605
18		799	17	0:05:03	261197	754	9	0:05:48	229142
19	1000_1000	993	19	0:10:14	317402	952	14	0:11:28	309418
20		997	20	0:11:53	305125	977	13	0:13:43	262064
21		993	17	0:13:32	280192	980	13	0:13:49	266591

Após todos os testes computacionais realizados nas instâncias de problemas dos conjuntos 1 e 2, pode-se observar que a heurística *GRASP-ADD-HÍBRIDA* apresentou melhores resultados em 95,24% dos casos. A Figura 40 exibe os tempos computacionais envolvidos.

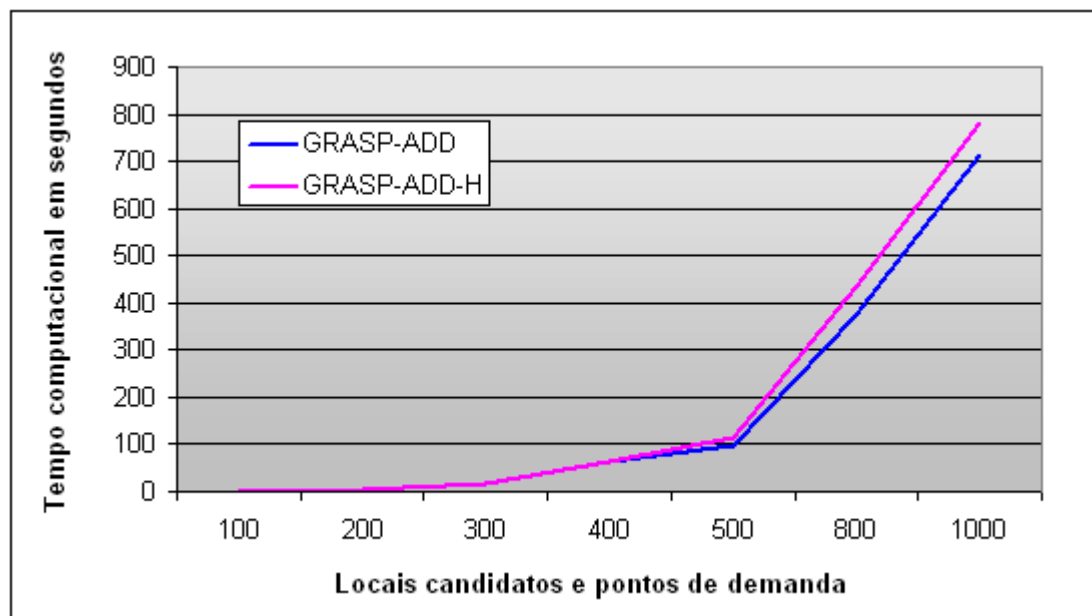


Figura 40. Tempo médio computacional gasto pelas implementações *GRASP* para instâncias de problemas do conjunto2 entre 100_100 e 1000_1000.

8 ALGORITMO GENÉTICO ADAPTADO AO PROBLEMA DE ALOCAÇÃO-LOCALIZAÇÃO DE ANTENAS

Este Capítulo apresenta o desenvolvimento de um Algoritmo Genético adaptado ao problema de alocação-localização de antenas. São implementadas duas versões: (1) com o uso de uma população inicial gerada de forma aleatória; (2) com o uso de um algoritmo construtivo na geração da população inicial, onde é procurado inserir facilidades que possam apresentar boas soluções.

Após a criação da população inicial, duas soluções pais são selecionadas e cruzadas dando origem a uma solução filha, que herda as características de seus pais. Aplica-se na seqüência um procedimento *DROP* na solução filha, retirando as facilidades com menor peso no valor final da função objetivo. Logo em seguida, a população é atualizada. O processo é repetido até atingir o critério de parada.

O Algoritmo Genético, implementado desta forma, tem propriedades desejáveis: é simples, gera soluções excelentes e é rápido. Ele é similar ao algoritmo proposto por DREZNER *et al.*, (2003), diferenciando-se pelo fato de trabalhar com uma população contendo cromossomos de tamanhos diferentes, uma vez que não é fixado o número de facilidades a serem instaladas e, por possibilitar a geração de uma população inicial por um processo construtivo, o que melhora em alguns casos, a qualidade das soluções obtidas.

8.1 CONSTRUÇÃO DA POPULAÇÃO INICIAL

No Algoritmo Genético proposto neste trabalho, os genes de um cromossomo correspondem aos índices das facilidades selecionadas entre os locais candidatos. Por exemplo, (8, 4, 3, 25) é um cromossomo representando uma possível solução para o problema, utilizando quatro facilidades (antenas) e atendendo a uma certa quantidade de pontos de demanda. O universo de soluções é muito grande e podem haver melhores soluções com um número maior ou menor de facilidades. Uma vez que o número de facilidades é desconhecido, a população é gerada de forma a conter cromossomos de diferentes tamanhos (contendo mais ou menos antenas).

Com relação ao tamanho da população (quantidade de prováveis soluções), este deverá ser cuidadosamente estipulado, uma vez que populações grandes reduzem a velocidade do Algoritmo Genético, enquanto populações pequenas podem possuir uma reduzida diversidade genética. Segundo DREZNER et al. (2003), o tamanho da população (P) deverá estar diretamente relacionado com a quantidade de pontos de demanda (n) e com o número de facilidades a serem alocadas (p), conforme mostrado nas Figuras 41 e 42.

Fonte: DREZNER et al., 2003, p. 26

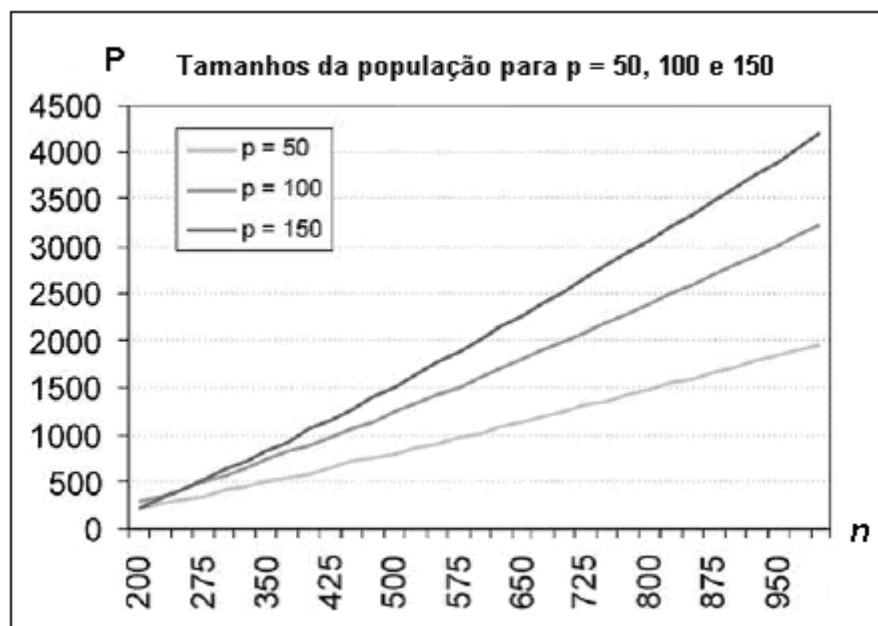


Figura 41. Tamanho da população de um A.G. em função de n para valores de p .

Fonte: DREZNER *et al.*, 2003, p. 26

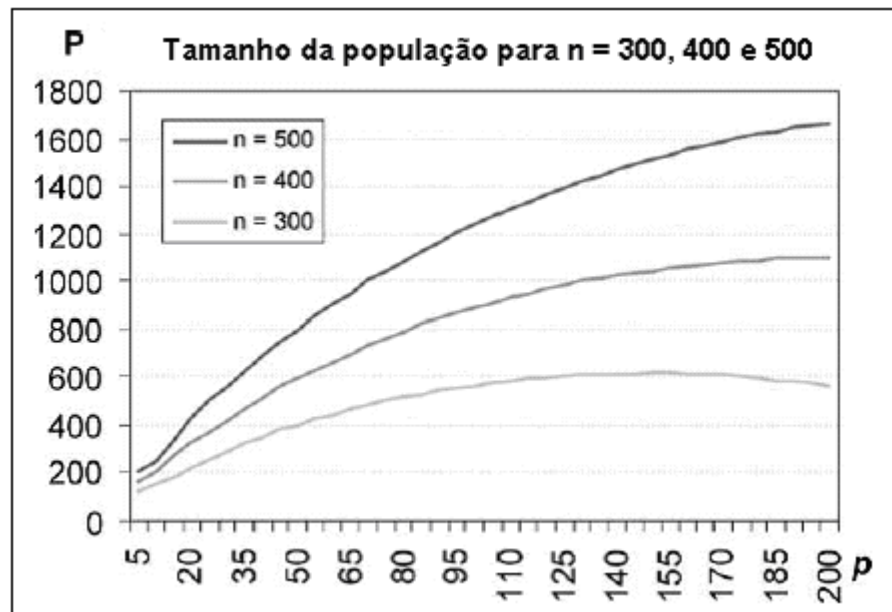


Figura 42. Tamanho da população de um A.G. em função de p para valores de n .

Esta relação não se aplica exatamente neste trabalho, pois o número de facilidades não é conhecido, servindo apenas como um norteio para testes computacionais.

Em diversos testes realizados, resultados muito satisfatórios foram obtidos relacionando-se o tamanho da população (P) com os pontos de demanda (n) como mostrado na Eq.(9).

$$P = \frac{n}{2} \quad (9)$$

8.1.1 Geração da população inicial com a aplicação de um algoritmo construtivo

A geração da população inicial de forma construtiva é baseada em uma heurística para o problema das p -medianas proposta por RESENDE & WERNECK (2005). Os autores apresentam um algoritmo híbrido que combina elementos de várias metaheurísticas tradicionais, tendo conseguido bons resultados em um baixo tempo computacional. Dentre os algoritmos construtivos apresentados destaca-se o

método *sample* (*sample greedy*), que é semelhante ao algoritmo guloso, que escolhe a facilidade que mais atende a pontos de demanda, sucessivamente. Em vez de selecionar o melhor entre todos os locais candidatos, gastando um elevado tempo computacional, considera-se somente $q < m$ locais candidatos à instalação de facilidades (escolhidas uniformemente ao acaso) em cada iteração. Segundo os autores, o valor de q deverá ser suficientemente pequeno para que haja uma redução do tempo computacional, estando relacionado com a quantidade de locais candidatos à instalação de facilidades (m) e com a quantidade de medianas (f) como mostrado na Eq.(10).

$$q = \log_2\left(\frac{m}{f}\right) \quad (10)$$

Este processo foi adaptado ao problema de alocação-localização de antenas para a criação da população inicial do Algoritmo Genético. Os cromossomos são *construídos* pelo processo de adição da melhor facilidade (gene) de um conjunto $Q = \{1,2,3,\dots,q\}$, que é escolhido aleatoriamente a cada iteração. O processo de adição das facilidades em um cromossomo permanece enquanto o valor final da função objetivo melhorar.

Como neste trabalho não é conhecido o número de antenas, diversos valores estimados para facilidades foram testados (f_{estim}), nas instâncias de problemas manuais de pequeno e grande porte. As Tabelas 8 e 9 comparam estes valores. Pode-se observar pela Tabela 10, que o valor de $f_{estim} = 4$ apresentou melhores resultados na maior parte das instâncias de problemas testados. Os valores de q relacionam-se diretamente com o número estimado de facilidades (f_{estim}) e o número de locais candidatos (m). Esta relação pode ser visualizada na Tabela 11. Pode-se ainda observar nesta Tabela, que os valores de q não são muito distantes uns dos outros mesmo para valores elevados de f_{estim} , pelo fato de se tratar de uma função logarítmica.

Tabela 8. Valores da função objetivo em relação ao número estimado de facilidades nas instâncias de problemas de médio porte do conjunto 1.

Conjunto 1 (Médio Porte)		Número estimado de facilidades								
		4	6	8	10	12	14	16	18	20
1	100_100	28871	28805	28805	28837	28837	28837	28837	28961	28961
2		19928	20870	20870	22306	22306	22306	22306	21820	21820
3		38930	39456	39456	39018	39018	39018	39018	39001	39001
4	200_200	94571	94703	94703	96143	96143	96143	96143	94616	94616
5		59425	61299	61299	59439	59439	59439	59439	60383	60383
6		50519	50485	50485	50532	50532	50532	50532	50600	50600
7	300_300	86767	86767	85986	85986	85986	86698	86698	86698	86698
8		104217	104217	104659	104659	104659	105030	105030	105030	105030
9		156978	156978	156962	156962	156962	156488	156488	156488	156488
10	400_400	118419	116545	116545	118291	118291	118291	118291	122933	122933
11		98108	99520	99520	97170	97170	97170	97170	99255	99255
12		119670	119545	119545	119510	119510	119510	119510	120250	120250

Tabela 9. Valores da função objetivo em relação ao número estimado de facilidades nas instâncias de problemas de grande porte do conjunto 1.

Conjunto 1 (Grande Porte)		Número estimado de facilidades								
		4	6	8	10	12	14	16	18	20
13	500_500	120866	121138	121138	121138	121138	121138	121138	121138	121138
14		120715	120865	120865	120865	122171	122171	122171	122171	122171
15		151855	148579	148579	148579	144354	144354	144354	144354	144354
16	800_800	142933	142856	142856	142856	142856	142856	142856	142861	142861
17		111149	111154	111154	111149	111149	111149	111149	111154	111154
18		129253	129159	129159	129120	129120	129120	129120	129165	129165
19	1000_1000	225811	220241	220241	220241	223605	223605	223605	223605	223605
20		152277	152317	152317	152317	152277	152277	152277	152277	152277
21		139414	139406	139406	139406	139741	139741	139741	139741	139741

Tabela 10. Número de vezes (nv) que f_{estim} apresentou melhores resultados

f_{estim}	4	6	8	10	12	14	16	18	20
nv	9	7	7	8	8	8	8	3	3

Tabela 11. Valores de q em função do número de locais candidatos (m) e do número estimado de facilidades (f_{estim}).

m	Número estimado de facilidades								
	4	6	8	10	12	14	16	18	20
100	5	4	4	3	3	3	3	2	2
200	6	5	5	4	4	4	4	3	3
300	6	6	5	5	5	4	4	4	4
400	7	6	6	5	5	5	5	4	4
500	7	6	6	6	5	5	5	5	5
800	8	7	7	6	6	6	6	5	5
1000	8	7	7	7	6	6	6	6	6

O pseudocódigo mostrado na Figura 43 ilustra a geração da população inicial com a aplicação do algoritmo construtivo. O conjunto de pontos potenciais para a instalação das facilidades (A) é inicializado para cada cromossomo da população (linha 4). As facilidades (antenas) contidas em Q são testadas no conjunto das antenas (linhas 9 à 15). A melhor facilidade é então adicionada ao conjunto de antenas (linhas 16 à 19). O processo de adição das facilidades permanece enquanto a função objetivo melhorar (linhas 7 e 20). Este processo repete-se para cada cromossomo da população (linha 1). Sejam:

$Pop = \{1, \dots, p\}$, um registro representando uma população contendo p cromossomos.

f_{estim} = número estimado de facilidades usadas pela melhor solução.

$Q = \{1, \dots, q\}$, um conjunto de facilidades escolhidas aleatoriamente. $q = \log_2\left(\frac{m}{f_{estim}}\right)$

```

Procedimento População_construtiva() ;
1: para ( $i \leftarrow 1, 2, 3, \dots, p$ ) faça
2:    $ant \leftarrow 1$ ;
3:    $f^* \leftarrow \infty$ ;
4:   Inicialize  $A$ ;
5:    $melhor \leftarrow 0$ ;
6:    $cont \leftarrow \text{true}$ ;
7:   enquanto ( $cont = \text{true}$ ) faça //insere facilid. enqto. Melhorar
8:     Inicialize  $Q$ ;
9:     para ( $j \leftarrow 1, 2, 3, \dots, q$ ) faça //Testa a adição das facilidades
10:       $Pop[i].c[ant] \leftarrow Q[j]$ ;
11:      se  $f(Pop[i].c[ant]) < f^*$  então
12:         $f^* \leftarrow f(Pop[i].c[ant])$ ;
13:         $melhor \leftarrow j$ ;
14:      fim se
15:    fim para
16:    se  $melhor > 0$  então //Insere a melhor facilidade testada
17:       $Pop[i].c[ant] \leftarrow Q[melhor]$ ;
18:      atualize  $Q$ ;
19:       $ant \leftarrow ant + 1$ ;
20:    senão
21:       $cont \leftarrow \text{false}$ ;
22:      retire a última antena inserida;
23:    fim se
24:  fim enquanto
25:fim para
Fim Procedimento

```

Figura 43. Pseudocódigo de um algoritmo construtivo que gera uma população inicial para o A.G.

8.1.2 Geração da população inicial de forma aleatória

A população inicial do Algoritmo Genético também pode ser gerada aleatoriamente, sem obedecer a nenhum processo construtivo. Neste caso, como não é conhecido o número de antenas que um indivíduo deverá conter, os cromossomos (prováveis soluções) são construídos com tamanhos diferentes. Eles são gerados de forma a conter uma quantidade de antenas que varia entre 10% e 50% do total de locais candidatos à instalação das facilidades. A Tabela 12 exibe o valor final da função objetivo obtido pelos testes computacionais realizados nas instâncias de problemas do conjunto 1.

Tabela 12. Resultados obtidos para diferentes quantidades de genes dos cromossomos da população inicial gerada de forma aleatória.

(m_n)	Porcentagem de genes por cromossomo da população inicial.					
	10% e 50%	20% e 60%	30% e 70%	40% e 80%	50% e 90%	60% e 100%
100_100	29058	28904	29201	33425	33323	34354
200_200	99637	99727	99795	99899	99954	99992
300_300	92048	92570	92477	92582	92654	92723
400_400	139136	139298	139323	139444	139487	139481
500_500	148824	148831	148752	148954	148929	148993
800_800	160254	160053	160154	160189	160547	160984
1000_1000	234227	234458	234498	234754	234997	235024

O Pseudocódigo exibido na Figura 44 resume a criação da população inicial de forma aleatória. Pode-se observar nas linhas 1, 2 e 5, a geração dos cromossomos com tamanhos variáveis (entre 10% e 50% de m locais candidatos). Sejam:

$A = \{1, \dots, m\}$, um conjunto de pontos potenciais (candidatos) para instalação de facilidades.

$Pop = \{1, \dots, p\}$, um registro, representando uma população inicial, contendo p cromossomos.

$c = \{1, \dots, k\}$, um vetor representando um cromossomo p da população, contendo k genes | $1 \leq k \leq m$.

```

Procedimento População_Aleatória ( $P, m$ )
1:  $v_i \leftarrow 0.1 * m$ ;
2:  $v_s \leftarrow 0.5 * m$ ;
3: para ( $i \leftarrow 1, 2, 3, \dots, P$ ) faça
4:   Inicialize  $A$ ;
5:    $p \leftarrow v_i + \text{rand}(v_s - v_i)$ ;
6:   para ( $j \leftarrow 1, 2, 3, \dots, k$ ) faça
7:      $sort \leftarrow \text{rand}(m)$ ;
8:      $Pop[i].c[j] \leftarrow A[sort]$ ;
9:     Retire a  $sort$  de  $A$ ;
10:     $m \leftarrow m - 1$ ;
11:   fim para
12: fim para
fim

```

Figura 44. Pseudocódigo que gera uma população inicial de forma aleatória.

8.2 SELEÇÃO E CRUZAMENTO

Segundo DREZNER (2003), a seleção de pais escolhidos com mecanismos de seleções aleatórios ou não, não representaram impactos significantes no desempenho de um algoritmo para o problema das p -medianas. Através de testes computacionais, o autor conclui que tanto a seleção aleatória quanto a não aleatória de pais, produzem resultados excelentes e em alguns casos os resultados são melhores quando a seleção é feita aleatoriamente, sendo que, a seleção aleatória é inclusive mais fácil de ser implementada.

Neste trabalho optou-se por fazer a escolha aleatória dos pais. Eles serão responsáveis pela geração de um único cromossomo filho, que herdará no processo de cruzamento, todos os genes de seus pais. A Figura 45 ilustra o processo de seleção (1) e cruzamento dos pais (2).

Na sequência, aplica-se o procedimento *DROP* ao cromossomo gerado pelo processo de cruzamento, sendo testadas a retirada dos genes individualmente (3). O procedimento *DROP* permanece enquanto a retirada das facilidades melhorar o valor da função objetivo (4). Por último, o cromossomo filho é retornado (5).

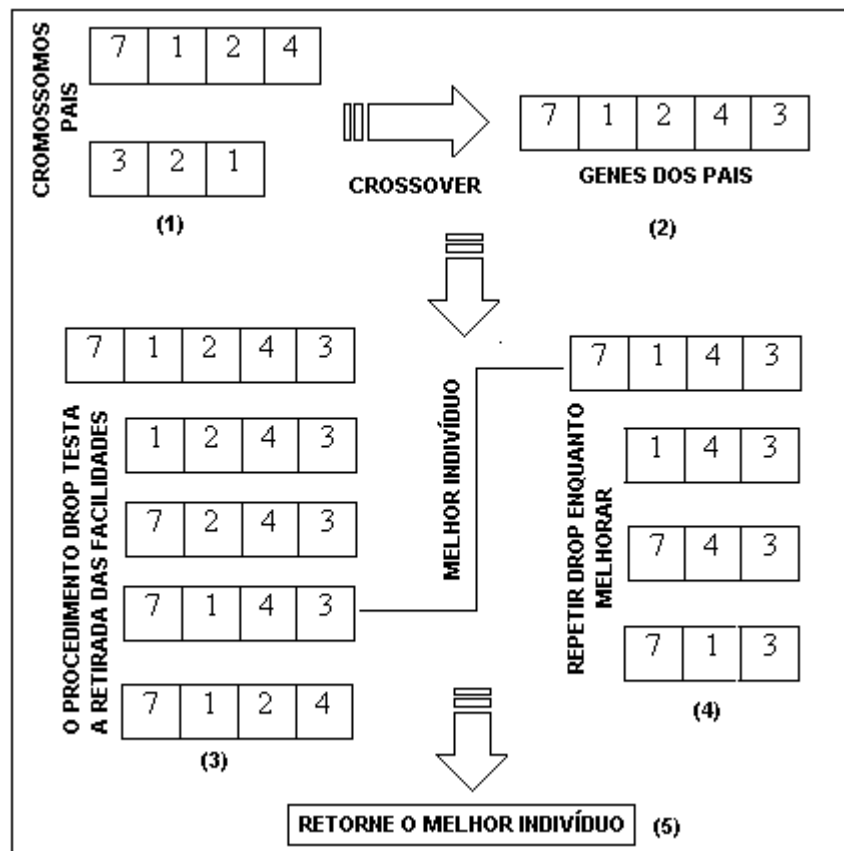


Figura 45. Exemplificação da seleção e cruzamento dos cromossomos de um A.G.

O Pseudocódigo exibido na Figura 46 ilustra o procedimento *DROP*. Sejam:

$s1, s2 = \{1, \dots, p\}$, os vetores representando os cromossomos escolhidos no processo de seleção para serem os genitores.

$S = s1 \cup s2$, um vetor resultante do cruzamento de $s1$ e $s2$.

```

Procedimento DROP(S);
   $f^* \leftarrow f(S)$ ;
  cont  $\leftarrow$  true;
  enquanto (cont = true) faça
    para ( $i \leftarrow 1, 2, 3, \dots, |S|$ ) faça
      Retire indivíduo i de S;
      Calcule  $f(S)$ ;
      Retorne o indivíduo i em S;
    fim para
    cont  $\leftarrow$  false;
    se  $\min(f(S)) < f^*$  então
       $f^* \leftarrow \min(f(S))$ ;
      retire o indivíduo de S que provocou um  $\min f(S)$ ;
      cont  $\leftarrow$  true;
    fim se
  fim enquanto
Fim procedimento

```

Figura 46. Pseudocódigo do procedimento *DROP* aplicado ao cromossomo filho do Algoritmo Genético.

8.3 ATUALIZAÇÃO DA POPULAÇÃO

A população deverá ser constantemente atualizada, de forma que a cada atualização, ela melhore a qualidade das prováveis soluções representadas por seus cromossomos. Este mecanismo básico de evolução das soluções está implementado obedecendo aos seguintes critérios:

- O cromossomo filho retornado pelo processo de seleção e cruzamento poderá ser inserido na população desde que seja único (não podem existir soluções iguais em uma mesma população).
- O cromossomo filho deverá possuir um grau de aptidão maior que o cromossomo de menor grau de aptidão da população. Neste caso, o pior cromossomo é substituído (retira-se o conjunto de facilidades que possuem o menor valor final da função objetivo da população, inserindo no lugar o cromossomo filho testado).

8.4 CRITÉRIO DE PARADA

O critério de parada informa quando deve terminar a busca pela evolução da população. Ele foi implementado de duas formas distintas:

- Após um número de vezes (*nvsm*) que o cromossomo filho, ao término da aplicação do procedimento *DROP*, não puder ser inserido na população, por possuir um baixo valor de aptidão ou por não ser o único. O valor de *nvsm* poderá ser estipulado de acordo com cada problema. Em testes computacionais com pontos de demanda e locais candidatos para instalação de facilidades que variam entre 100 e 1000, adotou-se *nvsm* = 50 obtendo-se bons resultados.
- Calculando-se o valor da função objetivo o mesmo número de vezes que outras heurísticas, para comparações computacionais. Esta implementação foi utilizada nos testes apresentados no Capítulo 9.

O Pseudocódigo exibido na Figura 47, sintetiza as fases da geração da população inicial (1 a 5), seleção dos pais (9 e 10), cruzamento e retirada dos piores indivíduos (11 e 12) e atualização da população (13 a 18).

```

Procedimento Algoritmo_Genético(P,m);
1: se criar pop aleatória então
2:   População_Aleatória (P,m);
3: senão
4:   População_Construtiva ();
5: fim se;
6:  $f^* \leftarrow f(S^*)$ ; //indivíduo com pior fitness
7:  $y \leftarrow 0$ ;
8: enquanto ( $y < \text{Critério de parada}$ ) faça
9:    $s1 \leftarrow P[\text{random}(p)].c$ ;
10:   $s2 \leftarrow P[\text{random}(p)].c$ ;
11:   $S \leftarrow s1 \cup s2$ ;
12:  DROP(S);
13:  se  $f(S) < f^*$  e  $S \notin P$  então
14:    retire  $S^*$  da população;
15:    insira indivíduo  $S$  na população;
16:     $f^* \leftarrow f(s)$ ;
17:  senão
18:     $y \leftarrow y + 1$ ;
19:  fim se;
20: fim enquanto;
21: retorne o melhor indivíduo;
fim

```

Figura 47. Pseudocódigo do A.G. adaptado ao problema.

9 TESTES E ANÁLISE DOS RESULTADOS COM O USO DO ALGORITMO GENÉTICO

Neste Capítulo é comparada a qualidade das soluções obtidas pelo Algoritmo Genético com a população inicial criada de forma aleatória e de forma construtiva. O critério de parada usado pelo Algoritmo Genético com a população criada aleatoriamente consistiu em avaliar a função objetivo o mesmo número de vezes que a heurística *GRASP-ADD*, para que ambos possam ser comparados. Já o Algoritmo Genético com população criada de forma construtiva avaliou a função objetivo o mesmo número de vezes que a *GRASP-ADD-HÍBRIDA*.

9.1 TESTES COMPUTACIONAIS

O número médio de vezes que cada função objetivo foi avaliada pode ser visualizado na Tabela 13. Os testes computacionais realizados com o uso do Algoritmo Genético analisaram o valor final da função objetivo (f), os pontos de demanda atendidos (PA), o total de facilidades utilizadas (TF) e o tempo computacional gasto, incluindo a geração da população inicial, em horas, minutos e segundos (t). Estes resultados estão mostrados nas Tabelas 14 e 15 (conjunto 1 de problemas) e nas Tabelas 16 e 17 (conjunto 2 de problemas) .

Tabela 13. Número médio de vezes que a função objetivo foi avaliada nas instâncias de problemas dos conjuntos 1 e 2.

	A.G. POP. ALEATÓRIA		A.G. POP. CONSTRUTIVA	
	Conjunto 1	Conjunto 2	Conjunto 1	Conjunto 2
100_100	43513	35308	50488	37252
200_200	78491	117652	94491	131329
300_300	229819	206836	274976	229101
400_400	214605	421409	235363	455657
500_500	281008	431613	325308	524632
800_800	572301	712684	754232	843572
1000_1000	787021	896784	918093	1004017

Tabela 14. Testes computacionais realizados com o uso do Algoritmo Genético nas instâncias de problemas de médio porte do conjunto 1.

Conjunto 1 (Médio Porte)		POP. ALEATÓRIA				POP. CONSTRUTIVA			
		PA	TF	t	f	PA	TF	t	f
1	100_100	99	6	0:00:01	28904	99	6	0:00:01	28771
2		99	8	0:00:01	20926	100	8	0:00:01	20039
3		94	8	0:00:01	42368	98	8	0:00:01	38948
4	200_200	195	8	0:00:08	99637	187	7	0:00:03	96113
5		199	14	0:00:11	63231	197	13	0:00:05	61365
6		195	6	0:00:06	60387	197	5	0:00:02	50472
7	300_300	300	10	0:00:43	92048	298	9	0:00:12	86655
8		290	11	0:00:43	111067	282	9	0:00:15	104020
9		272	30	0:01:40	169686	268	27	0:00:48	160232
10	400_400	388	6	0:01:18	139136	395	5	0:00:22	116458
11		387	10	0:01:30	108716	391	9	0:00:29	98349
12		376	12	0:02:26	147699	388	10	0:00:40	120166

Tabela 15. Testes computacionais realizados com o uso do Algoritmo Genético nas instâncias de problemas de grande porte do conjunto 1.

Conjunto 1 (Grande Porte)		POP. ALEATÓRIA				POP. CONSTRUTIVA			
		PA	TF	t	f	PA	TF	t	f
13	500_500	486	9	0:02:34	148752	500	8	0:00:46	121520
14		469	9	0:05:06	154554	492	8	0:01:03	124052
15		471	10	0:04:08	180893	490	9	0:01:10	150176
16	800_800	785	8	0:21:00	160053	800	8	0:03:23	143099
17		800	15	0:37:47	118473	800	14	0:08:23	111582
18		787	8	0:22:50	155816	800	7	0:03:54	129120
19	1000_1000	993	16	0:49:54	234227	1000	16	0:13:44	226457
20		983	8	0:44:36	170840	1000	8	0:07:29	152545
21		984	14	0:43:19	199368	1000	9	0:12:08	139856

Analisando os resultados obtidos nas instâncias de problemas do conjunto 1, exibidos nas Tabelas 14 e 15, observa-se que Algoritmo Genético implementado com a população inicial criada de forma construtiva apresentou melhores resultados em 100% dos problemas. Ele também usou um menor número de facilidades do que o Algoritmo Genético com população criada aleatoriamente.

O tempo computacional requerido com a implementação da população inicial de forma aleatória é superior ao da população criada de forma construtiva, como pode ser observado na Figura 48. Uma análise minuciosa das heurísticas mostra que apenas o processo de geração da população inicial de forma construtiva é mais lento. O algoritmo que usa a população construtiva torna-se mais rápido pelo fato do indivíduo gerado durante cruzamento conter um elevado *fitness* (os pais possuem um alto valor de aptidão), requerendo muito pouco do procedimento *DROP*, que procura retirar as facilidades ruins. Normalmente, o mesmo não ocorre para a população criada de forma aleatória, que contém um elevado número de genes que não são bons.

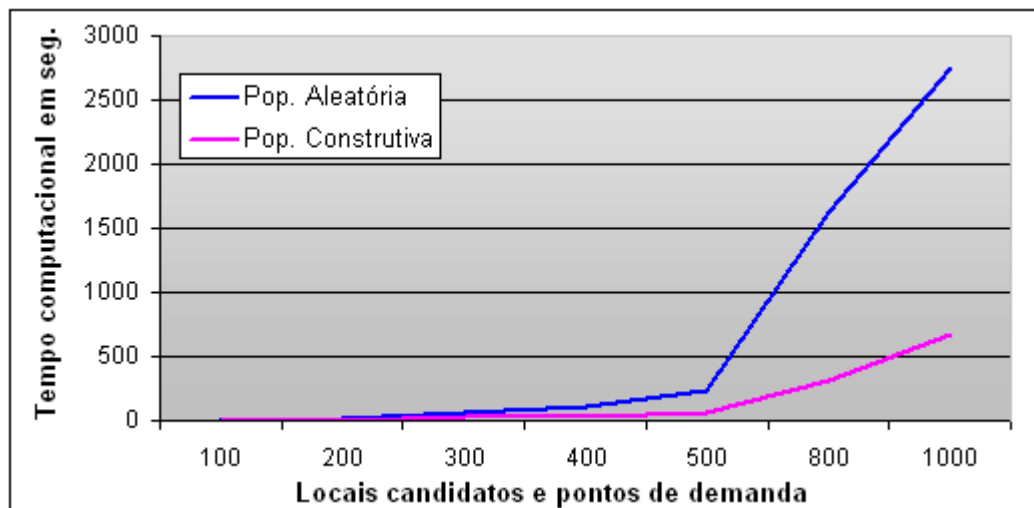


Figura 48. Tempo computacional médio do A.G. para as instâncias de problemas do conjunto 1.

Tabela 16. Testes computacionais realizados com o uso do Algoritmo Genético nas instâncias de problemas de médio porte do conjunto 2.

Conjunto 2 (Médio Porte)		POP. ALEATÓRIA				POP. CONSTRUTIVA			
		PA	TF	<i>t</i>	<i>f</i>	PA	TF	<i>t</i>	<i>f</i>
1	100_100	82	5	0:00:01	61572	91	5	0:00:01	56159
2		87	6	0:00:01	61048	94	6	0:00:01	56118
3		87	5	0:00:01	50330	94	6	0:00:01	49398
4	200_200	195	8	0:00:10	71207	192	7	0:00:05	67695
5		183	6	0:00:11	75571	192	7	0:00:04	68877
6		183	7	0:00:11	77059	193	8	0:00:05	74792
7	300_300	279	8	0:00:37	99189	296	10	0:00:15	92930
8		284	10	0:00:59	107455	291	10	0:00:20	103587
9		271	15	0:00:54	131426	277	13	0:00:20	118206
10	400_400	377	17	0:03:23	135853	391	17	0:01:20	124532
11		382	17	0:03:08	138607	378	15	0:01:05	123855
12		380	16	0:03:07	128920	385	16	0:01:09	123437

Tabela 17. Testes computacionais realizados com o uso do Algoritmo Genético nas instâncias de problemas de grande porte do conjunto 2.

Conjunto 2 (Grande Porte)		POP. ALEATÓRIA				POP. CONSTRUTIVA			
		PA	TF	<i>t</i>	<i>f</i>	PA	TF	<i>t</i>	<i>f</i>
13	500_500	477	18	0:06:07	155612	488	17	0:01:51	135031
14		459	16	0:04:42	209613	454	14	0:01:32	192809
15		469	16	0:06:45	181670	461	14	0:02:04	166655
16	800_800	725	17	0:29:44	296772	765	19	0:08:34	273613
17		773	17	0:22:34	250481	774	15	0:06:34	223413
18		758	11	0:21:22	250834	772	10	0:05:15	224796
19	1000_1000	939	14	0:46:10	320155	953	14	0:10:54	309867
20		975	14	1:02:33	275963	983	14	0:12:22	262057
21		975	15	0:50:24	281726	976	14	0:14:16	268535

Conforme os resultados obtidos nas instâncias de problemas do conjunto 2, mostrados nas Tabelas 16 e 17, observa-se que o Algoritmo Genético que cria a população inicial de forma construtiva obteve um desempenho superior, apresentando melhores 100% dos casos.

Analogamente aos problemas do conjunto 1, o tempo computacional requerido com a implementação da população inicial de forma construtiva é bastante inferior à população criada aleatoriamente. O tempo computacional também aumenta quando são analisadas uma maior quantidade de locais candidatos e pontos de demanda.

10 COMPARAÇÃO E ANÁLISE DOS RESULTADOS ENTRE A *GRASP* E O ALGORITMO GENÉTICO

Este capítulo compara as melhores versões das heurísticas *GRASP* e os Algoritmos Genéticos sendo mostrados gráficos comparativos para os conjuntos de problemas 1 e 2. Também são plotadas as facilidades utilizadas em um dado problema, com os respectivos pontos de demanda por elas atendidos.

10.1 A *GRASP* VERSUS ALGORITMO GENÉTICO

Como pôde ser observado nos testes computacionais realizados nos capítulos 7 e 9, os melhores resultados foram respectivamente obtidos pela heurística *GRASP-ADD-HÍBRIDA* e pelos Algoritmos Genéticos com população criada de forma construtiva. As Tabelas 18 e 19 exibem os valores da função objetivo e o tempo computacional requerido em ambas as implementações para as instâncias de problemas do conjunto 1.

Tabela 18. Valores da função objetivo e tempo computacional requerido em instâncias de problemas de médio porte do conjunto 1.

Conjunto 1 (Md. Pte.)	GRASP – ADD - HÍBRIDA		A.G. POP. CONSTRUTIVA	
	<i>f</i>	<i>t</i>	<i>f</i>	<i>t</i>
1	28771	0:00:01	28771	0:00:01
2 100_100	20144	0:00:01	20039	0:00:01
3	38249	0:00:01	38948	0:00:01
4	96113	0:00:03	96113	0:00:03
5 200_200	63287	0:00:05	61365	0:00:05
6	50346	0:00:02	50472	0:00:02
7	91736	0:00:13	86655	0:00:12
8 300_300	105786	0:00:15	104020	0:00:15
9	157062	0:00:44	160232	0:00:48
10	123700	0:00:24	116458	0:00:22
11 400_400	101217	0:00:31	98349	0:00:29
12	120355	0:00:39	120166	0:00:40

Tabela 19. Valores da função objetivo e tempo computacional requerido em instâncias de problemas de grande porte do conjunto 1.

Conjunto 1 (GDE. Pte.)	GRASP – ADD - HÍBRIDA		A.G. POP. CONSTRUTIVA	
	<i>f</i>	<i>t</i>	<i>f</i>	<i>t</i>
13	122737	0:00:51	121520	0:00:46
14 500_500	121083	0:01:16	124052	0:01:03
15	147341	0:01:10	150176	0:01:10
16	142856	0:04:20	143099	0:03:23
17 800_800	111149	0:09:30	111582	0:08:23
18	129120	0:04:57	129120	0:03:54
19	225764	0:15:18	226457	0:13:44
20 1000_1000	163554	0:08:25	152545	0:07:29
21	147384	0:11:20	139856	0:12:08

Conforme os testes computacionais exibidos nas Tabelas 18 e 19 pode-se observar que Algoritmo Genético com população criada de forma construtiva apresentou melhores resultados em 11 problemas, contra 8, encontrados pela GRASP-ADD-HÍBRIDA. A Figura 49 compara graficamente estes resultados.

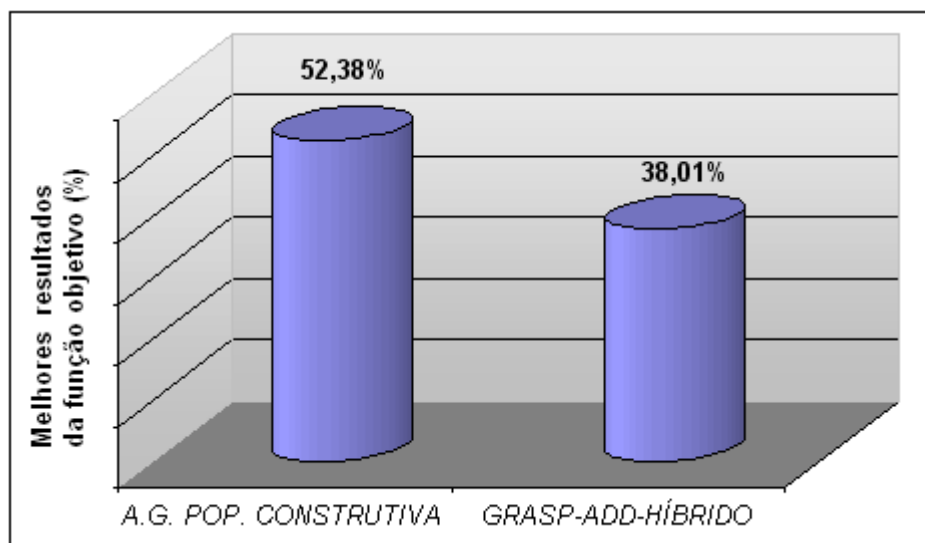


Figura 49. A.G. com população construtiva x *GRASP-ADD-HÍBRIDA* nas instâncias de problemas do conjunto 1.

Observa-se também, que ambas as implementações requereram tempos computacionais similares nas instâncias de problemas do conjunto 1 (Figura 50).

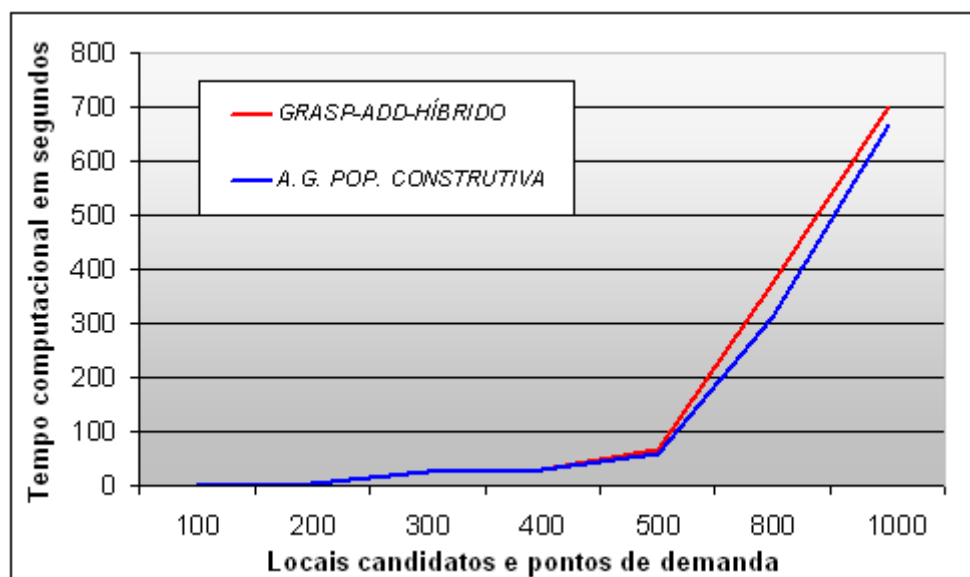


Figura 50. Tempo computacional médio das para as instâncias de problemas do conjunto 1.

As instâncias de problemas do conjunto 2 são comparadas pelas Tabelas 20 e 21, onde são exibidos os valores da função objetivo e os tempos computacionais envolvidos.

Tabela 20. Valores da função objetivo e tempo computacional requerido em instâncias de problemas de médio porte do conjunto 2.

Conjunto 2 (Md. Pte.)		GRASP – ADD - HÍBRIDA		A.G. POP. CONSTRUTIVA	
		<i>f</i>	<i>t</i>	<i>f</i>	<i>t</i>
1	100_100	56159	0:00:01	56159	0:00:01
2		57775	0:00:01	56118	0:00:01
3		49981	0:00:01	49398	0:00:01
4	200_200	68072	0:00:05	67695	0:00:05
5		70113	0:00:05	68877	0:00:04
6		75802	0:00:05	74792	0:00:05
7	300_300	93425	0:00:15	92930	0:00:15
8		106155	0:00:21	103587	0:00:20
9		117319	0:00:20	118206	0:00:20
10	400_400	125112	0:01:09	124532	0:01:20
11		120085	0:01:05	123855	0:01:05
12		129857	0:01:02	123437	0:01:09

Tabela 21. Valores da função objetivo e tempo computacional requerido em instâncias de problemas de grande porte do conjunto 2.

Conjunto 2 (GDE. Pte.)		GRASP – ADD - HÍBRIDA		A.G. POP. CONSTRUTIVA	
		<i>f</i>	<i>t</i>	<i>f</i>	<i>t</i>
13	500_500	133725	0:01:51	135031	0:01:51
14		188857	0:01:35	192809	0:01:32
15		168750	0:02:16	166655	0:02:04
16	800_800	277111	0:09:06	273613	0:08:34
17		216605	0:06:40	223413	0:06:34
18		229142	0:05:48	224796	0:05:15
19	1000_1000	309418	0:11:28	309867	0:10:54
20		262064	0:13:43	262057	0:12:22
21		266591	0:13:49	268535	0:14:16

Analogamente às instâncias de problemas do conjunto 1, o Algoritmo Genético com população inicial criada de forma construtiva apresentou melhores resultados em 61,90% dos casos, seguido pela *GRASP-ADD-HÍBRIDA*, que apresentou melhores resultados em 28,57 % dos casos. Ambas as implementações apresentaram resultados iguais em 9,53% dos problemas.

Se compararmos os resultados obtidos nas instâncias de problemas dos conjuntos 1 e 2, o Algoritmo Genético com população inicial criada de forma construtiva apresentou melhores resultados em 24 instâncias (57,14%). Já a *GRASP-ADD-HÍBRIDA* apresentou melhores resultados em 14 instâncias (33,33%).

10.1.2 Comparações gráficas

As Figuras 51 a 57 exibem os pontos de demanda (atendidos e não atendidos), facilidades e obstáculos usados nas melhores implementações de algumas instâncias de problemas do conjunto 1. A Figura 51 compara as soluções encontradas para a instância 3 (100_100). A *GRASP-ADD-HÍBRIDA* apresentou uma menor função objetivo pelo fato de usar uma facilidade a menos que o Algoritmo Genético, embora tenha deixado de atender a 5 pontos de demanda, contra 3 pontos de demanda sem atendimento pelo Algoritmo Genético. Mesmo assim o resultado foi melhor porque a adição de mais uma facilidade atende a apenas mais 3 pontos de demanda. Esta situação se inverte na Figura 52, onde o Algoritmo Genético encontra melhores resultados para a instância 5 (200_200).

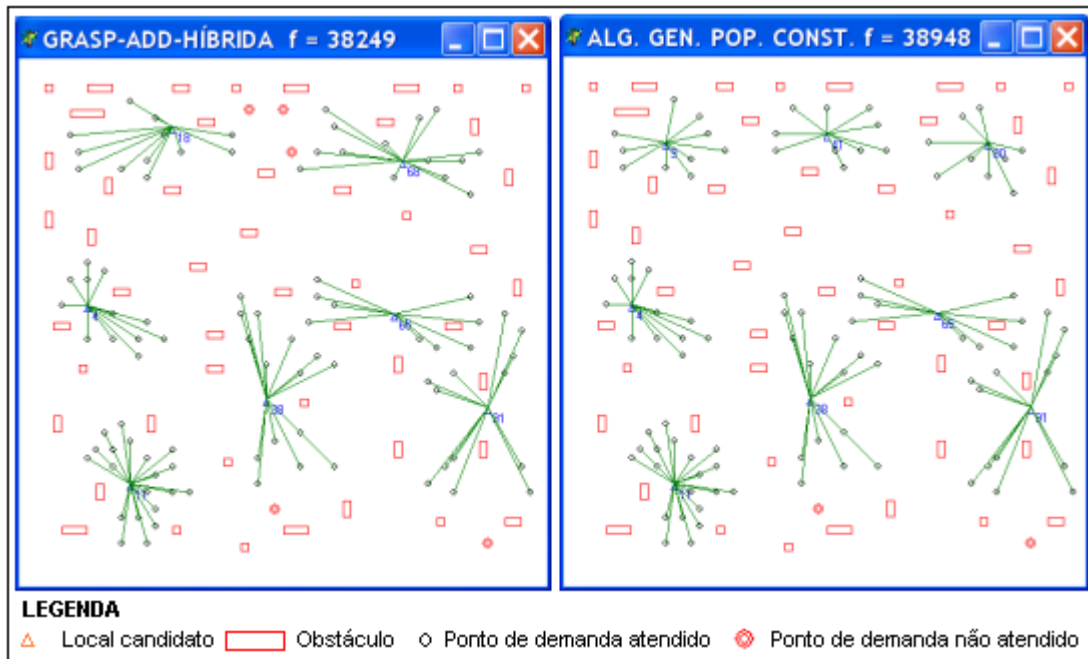


Figura 51. Gráfico comparativo para a instância 3 do conjunto 1 de problemas.

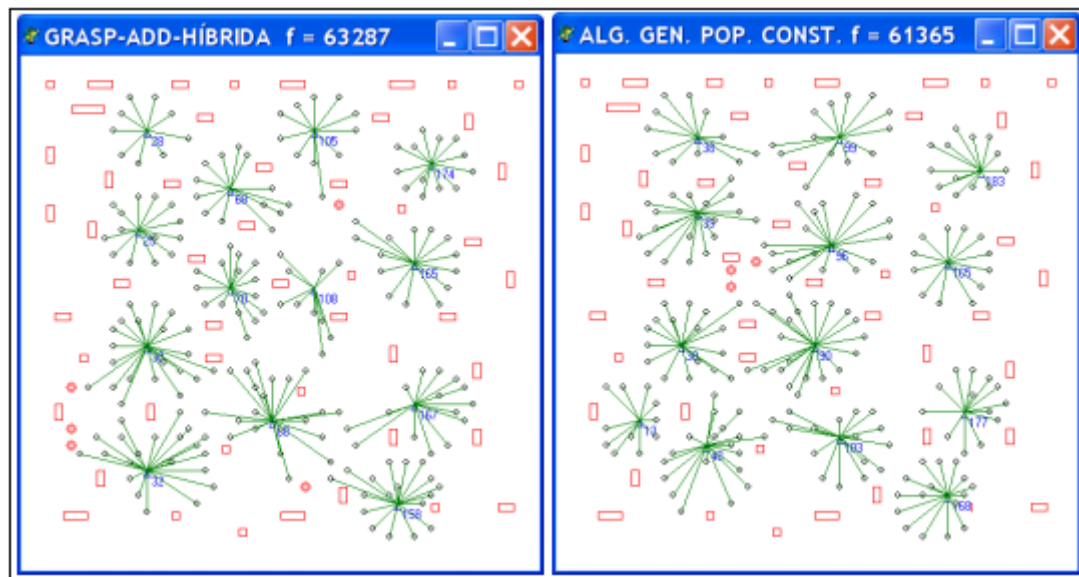


Figura 52. Gráfico comparativo para a instância 5 do conjunto 1 de problemas.

A Figura 53 compara as soluções para a instância 7 (300_300) do conjunto 1. Neste caso, pode ser observado que a adição de mais uma facilidade, usada pelo Algoritmo Genético permitiu o atendimento a mais 13 pontos de demanda, quando comparado com a *GRASP-ADD-HÍBRIDA*. Analogamente, o mesmo ocorreu para a instância 10 (400_400) do conjunto 1 (Figura 54).

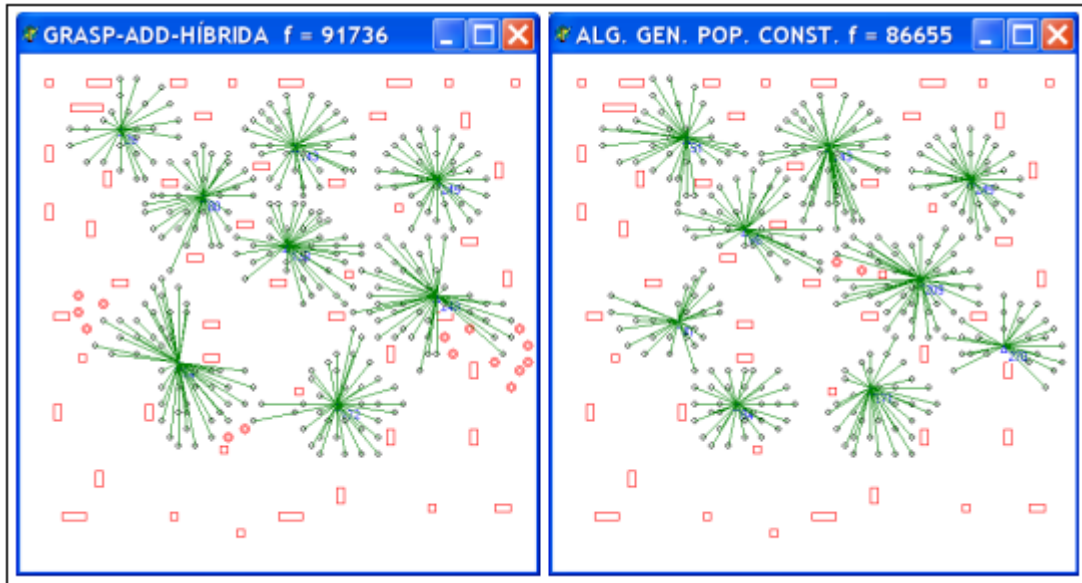


Figura 53. Gráfico comparativo para a instância 7 do conjunto 1 de problemas.

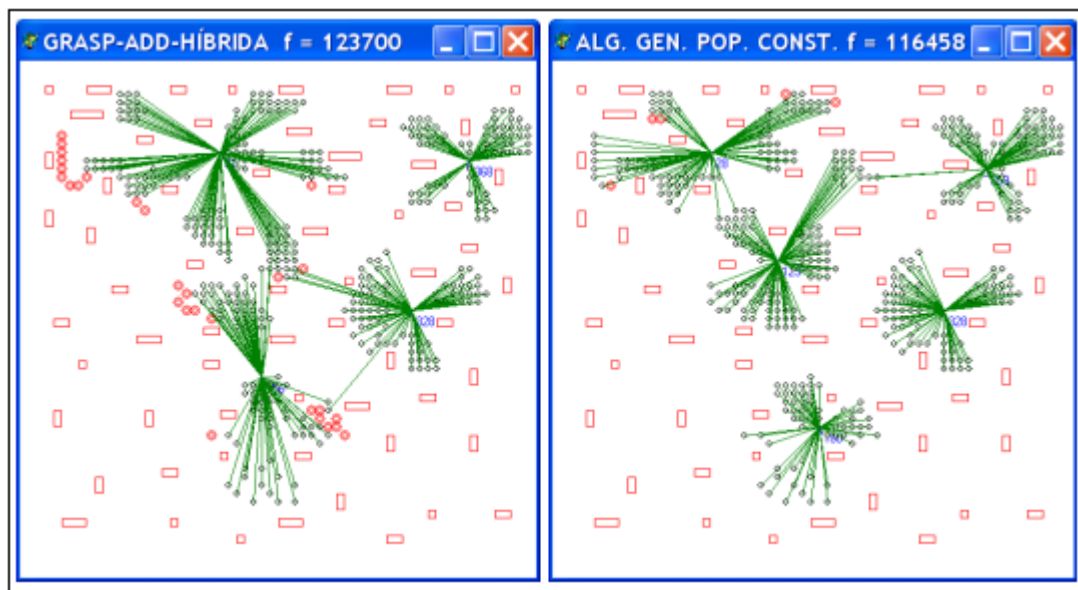


Figura 54. Gráfico comparativo para a instância 10 do conjunto 1 de problemas.

A prova de que as implementações também procuram aproximar as facilidades dos pontos de demanda pode ser obtida pela Figura 55, que compara as soluções para a instância 13 (500_500) do conjunto 1. Ambas as metaheurísticas usaram 8 facilidades, porém os melhores resultados foram obtidos pelo Algoritmo Genético, que encurtou a distância de atendimento para vários pontos de demanda. Da mesma forma, a Figura 56 compara as soluções para a instância 16 (800_800) e apresenta uma ligeira melhoria em prol da *GRASP-ADD-HÍBRIDA*, alocando de uma melhor maneira, as 8 facilidades.

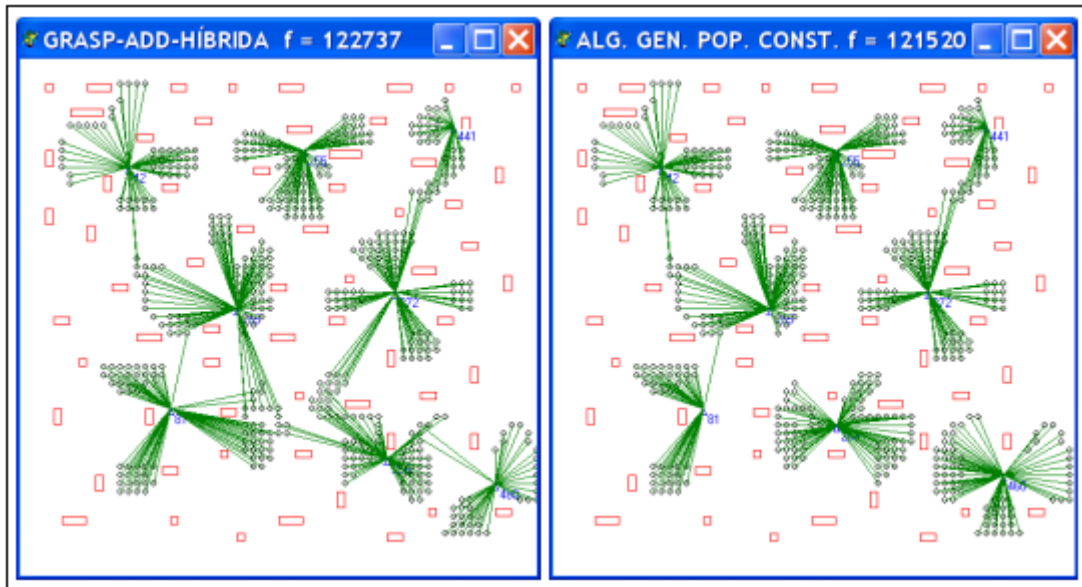


Figura 55. Gráfico comparativo para a instância 13 do conjunto 1 de problemas.

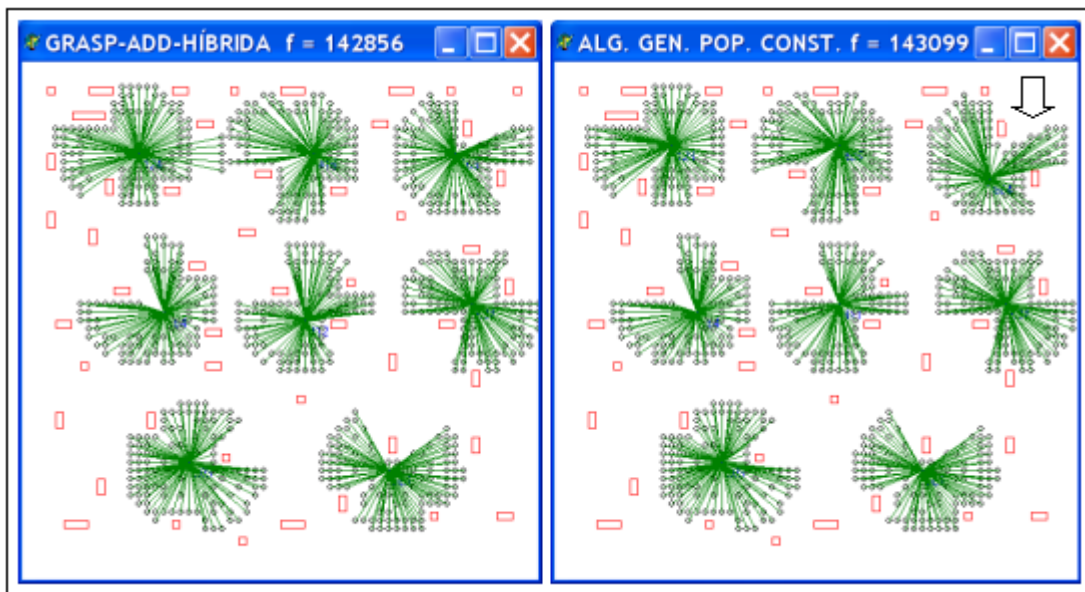


Figura 56. Gráfico comparativo para a instância 16 do conjunto 1 de problemas.

As heurísticas também procuram reduzir o número de facilidades utilizadas. Isto pode ser observado pela Figura 57, que compara as soluções para a instância 20 (1000_1000) do conjunto 1. As heurísticas atenderam a todos os pontos de demanda, porém o Algoritmo Genético foi melhor por utilizar uma facilidade a menos.

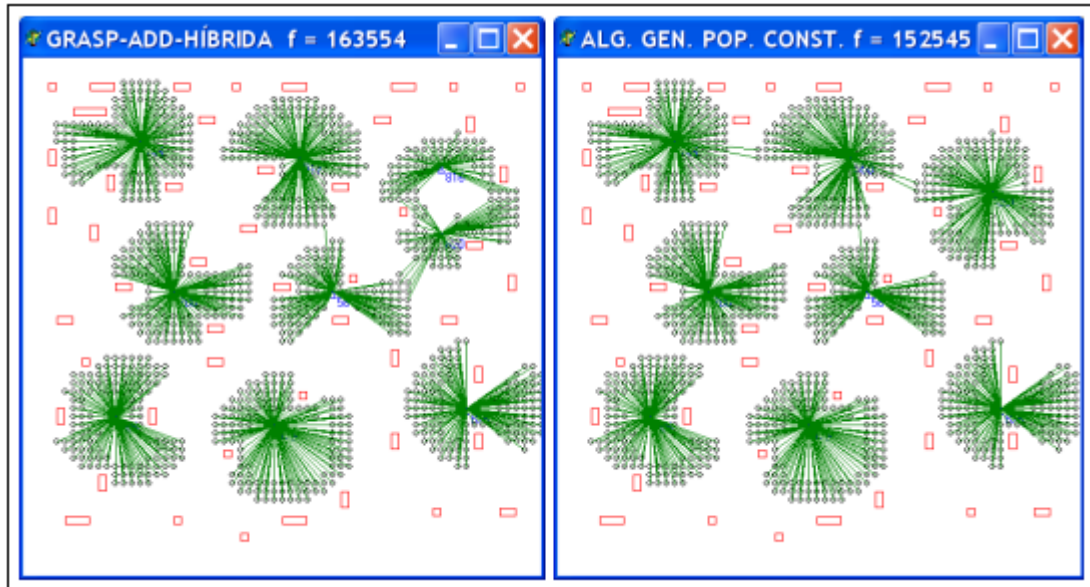


Figura 57. Gráfico comparativo para a instância 20 do conjunto 1 de problemas.

A Figura 58 exibe a solução encontrada pelo Algoritmo Genético para a instância 1 (100_100) do conjunto 2 de problemas. O conjunto 2 de problemas gerou aleatoriamente pontos de demanda, facilidades candidatas e obstáculos com alturas variáveis. Pode-se observar que a facilidade de número 35, que possui uma altura de 344 m atende ao ponto de demanda de número 50 que possui uma altura de 13 m, enviando o sinal por cima do obstáculo 57, com altura de 45 m e do obstáculo 22, com altura de 17 m. Esta Figura também mostra o raio de alcance, que é o mesmo para todas as facilidades. Da mesma forma, o atendimento aos pontos de demanda com passagem do sinal por cima dos obstáculos pode ser observado em outros obstáculos, mostrados na Figura 59, da instância de problema 20 (1000_1000) do conjunto 2.

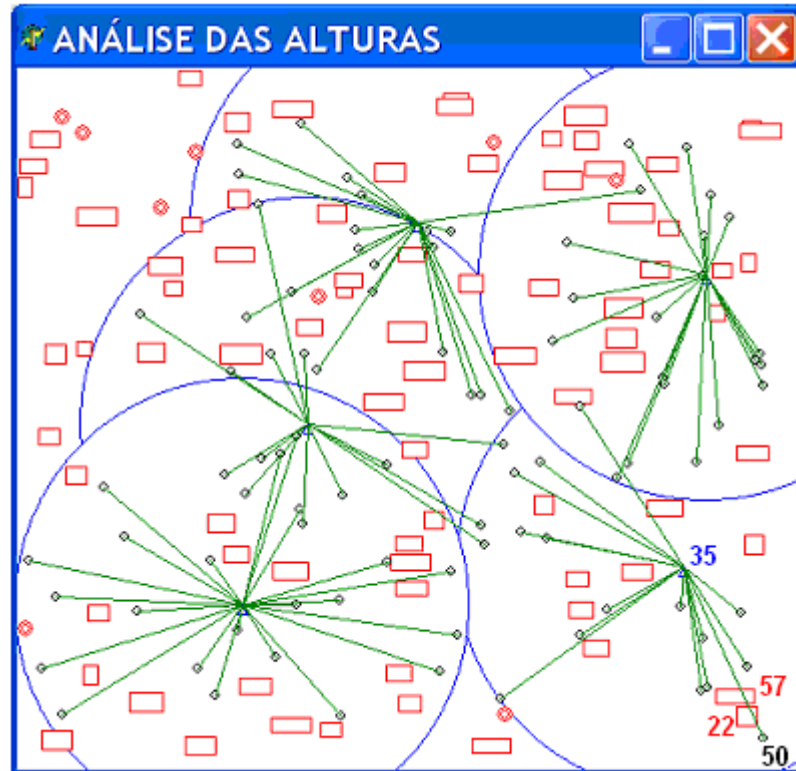


Figura 58. Passagem do sinal por cima dos obstáculos na instância de problema 1 do conjunto 2.

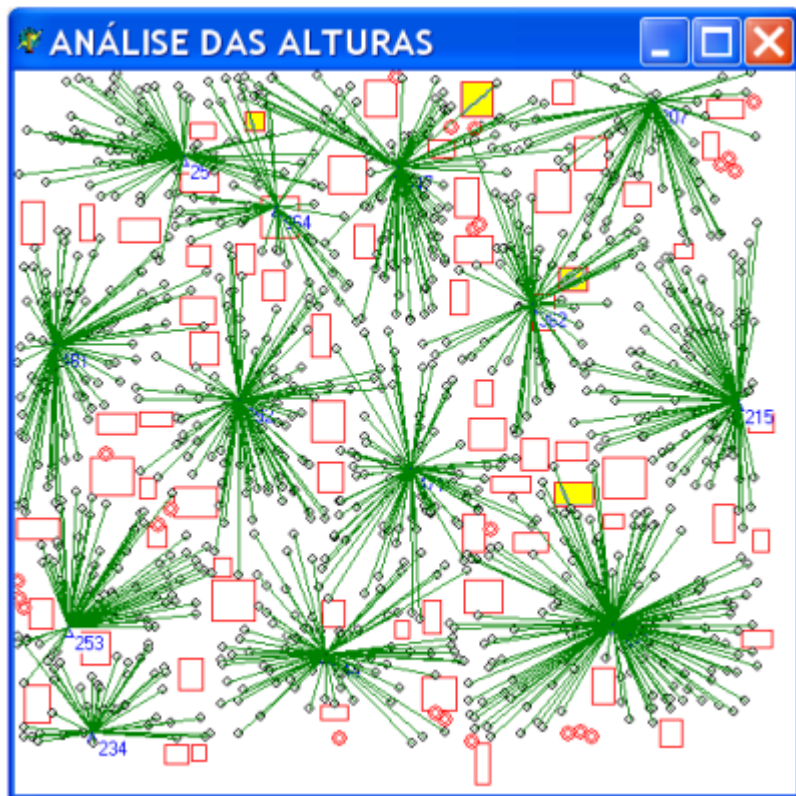


Figura 59. Passagem do sinal por cima dos obstáculos na instância de problema 20 do conjunto 2.

11 CONCLUSÃO

O objetivo deste trabalho foi desenvolver e implementar duas heurísticas diferentes para a solução do problema de localização de antenas de transmissão. Procurou-se aproximar mais o modelo matemático das situações reais, onde em muitos casos é desconhecido o número de facilidades a serem instaladas, e, restrições como alturas de obstáculos, pontos de demanda e facilidades, além do alcance dos equipamentos de telecomunicações devem ser consideradas.

Diferentes técnicas de implementação das heurísticas, como o Algoritmo Genético com população inicial criada de forma construtiva e a técnica híbrida aplicada à *GRASP* foram utilizadas, e os resultados comparados em diversas instâncias de problemas de grande porte com até 1000 pontos de demanda e locais candidatos. Os valores obtidos puderam ser comparados em 42 problemas, onde foi possível a análise da função objetivo, tempo computacional, pontos de demanda atendidos e facilidades utilizadas, além da visualização gráfica do atendimento aos pontos de demanda, mostrados no Capítulo 10.

As heurísticas *GRASP-ADD* e *GRASP-ADD-HÍBRIDA* usaram uma fase construtiva que consiste em abrir facilidades à medida que a solução melhora e, na sequência, aplicaram a fase de busca local que diferencia-se do problema das p -medianas, por não fixar o número de facilidades. Ao final de todas as iterações *GRASP*, foi aplicada uma busca local fixa, explorando uma quantidade maior de vizinhos, melhorando em muitos casos os resultados obtidos. A *GRASP-ADD-HÍBRIDA* apresentou excelentes resultados, se comparada à *GRASP-ADD*, pois mesclou facilidades da melhor solução encontrada até um dado momento, com a melhor solução retornada pela fase de busca local, mantendo as facilidades comuns

em ambos os resultados e retirando as demais através do procedimento *DROP*.

O Algoritmo Genético que usa uma população inicial criada de forma construtiva, baseou-se na técnica *sample greedy*, mostrada no Capítulo 8 e foi a melhor de todas as heurísticas implementadas, mostrando a importância de habitar a população inicial com bons indivíduos. O mesmo não ocorreu para o Algoritmo Genético com população inicial criada de forma aleatória, que inseriu um elevado número de facilidades com pequeno ou pouco valor nos indivíduos da população inicial.

Alocação-localização de facilidades é um problema amplamente discutido na literatura, sendo tratado com o emprego de inúmeras heurísticas pelo fato de ser um problema NP-difícil, mas até o presente momento, inexistente uma abordagem que trate da alocação de facilidades com um número variável de medianas, que minimize o seu uso ao mesmo tempo em que procura aproximá-las dos pontos de demanda. Este modelo, se corretamente adaptado, poderá ser aplicado com sucesso em outros problemas de alocação-localização de facilidades que possuam estes mesmos objetivos, podendo ainda ser melhorado.

Como objeto de futuros estudos pode-se destacar o tratamento dos obstáculos em sua forma real, a análise dos pontos de demanda e obstáculos através de um sistema de posicionamento geográfico e a ampliação do modelo para frequências mais baixas do espectro de frequências, onde os fenômenos de refração, difração e espalhamento do sinal de telecomunicações incidente em obstáculos são mais intensos.

Sob o ponto de vista de sua aplicabilidade, este trabalho poderá contribuir em muito para o progresso dos sistemas de telecomunicações. Como benefícios pode-se citar principalmente a satisfação da população com a melhoria da qualidade dos serviços e a redução dos custos de implementação, por parte das empresas prestadoras de serviços.

12 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.

BITTENCOURT, G. *Inteligência artificial: ferramentas e teorias*. Florianópolis: Editora da UFSC, v. 1. 362 p. 1998.

BOZKAYA, B. ; ZHANG J. ; ERKUT E. *An Efficient Genetic Algorithm for the p-Median Problem*. Journal Annals of Operations Research. Berlin, Springer Netherlands, v. 122, p. 21-42, 2003.

CHRISTOFIDES N., BEASLEY J.E. *Extensions to a Lagrangean location problem*. European Journal of Operational Research, Vol. 12, pp. 19-28, 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.; REVELLE, 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.

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.

DASKIN M.S. *Network and Discrete Location, Models, Algorithms, and Applications*, Wiley & Sons, Inc., New York, 1995.

DAVIS, L. *Handbook of Genetic Algorithms*. New York: Van Nostrand Reinhold, 1996.

DELMAIRE H.J. ; DÍAZ J.A. ; FERNÁNDEZ E. ; ORTEGA M. *Reactive grasp and tabu search based heuristics for the single source capacitated plant location problem*. Information Systems and Operations Research, Vol. 37, No. 3, pp. 194-225, 1998.

DENSHAM, P.J. ; RUSHTON G. *A More Efficient Heuristic for Solving Large p -Median Problems*. *Papers in Regional Science* 71, 307–329, 1992.

DIBBLE, C. ; DENSHAM P.J. *Generating Interesting Alternatives in GIS and SDSS Using Genetic Algorithms*. *GIS/LIS* 1993.

DOWNS, B.; CAMM, J. *An exact algorithm for the maximal covering location problem*. *Naval Research Logistics*, v. 43, n. 3, p. 435–461, 1996.

DOWSLAND, K.A. *Genetic Algorithms – A Tool for OR?*. *Journal of Operational Research Society*. 47, 550–561, 1996.

DREZNER, Z ; OSMAN ALP ; ERHAN ERKUT. *An Efficient Genetic Algorithm for the p -Median Problem*. *Annals of Operations Research*. 122, 21–42, 2003.

DWYEAR, 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.

FELDMAN E., LEHRER F.A., RAY T.L. *Warehouse location under continuous economies of scale*, *Management Science*, Vol. 12, No. 9, pp. 670-684, 1966.

FERRARI, M.A. *Telecomunicações Evolução e Revolução*. Érica, 1991. 297 p.

FEO, T.A; RESENDE, M.G.C., *Greedy randomized adaptive search procedures*. *Journal of Global Optimization*, 6:109-133, 1995.

GALVÃO, R.D.; RAGGI L.A. *A Method for Solving to Optimality Uncapacitated Location Problems*. *Annals of Operations Research* 18, 225–244, 1989.

GAREY, M.; JOHNSON, D. *Computers and intractability: a guide to the theory of NP-completeness*. San Francisco: W.H.Freeman and Co, 1979.

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.

GOLDBARG, M. C.; LUNA, H. P. L. *Otimização combinatória e programação linear: modelos e algoritmos*. Rio de Janeiro : Campus, 2000.

GOLDEN, B. ; SKISCIM C. *Using Simulated Annealing to Solve Routing and Location Problems*. Naval Research Logistic Quarterly 33, 261–279, 1986.

HAKIMI, I.S. *Optimum location of switching centers and the absolute centers and medians of a graph*, Operations Research, Vol. 12, pp. 450-459, 1964.

HAKIMI, I.S. *Optimum location of switching centers in a communications network and some related graph theoretic problems*, Operations Research, Vol. 13, pp. 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.

HILLSMAN, E. L. *The p-Median Structure as a Unified Linear Model for Location-Allocation Analysis*. Environmental and Planning A, 16: 305-318, 1984.

HOFFMAN, L. T.; GÓMEZ, A.T. *Uma abordagem do problema de localização de torres de rádio transmissão auxiliado por um sistema de informação geográfica*. SBPO, 2003.

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

HOSAGE, C.M. ; GOODCHILD M.F. *Discrete Space Location–Allocation Solutions from Genetic Algorithms*. Annals of Operations Research. 6, 35–46, 1986.

HOUGLAND, E. S.; STEPHENS, N. T. *Air pollutant monitor siting by analytical techniques*. Journal of Air Pollution Control Association, v. 26, p. 52–53, 1976.

KUEHN A.A., HAMBURGER M.J. *A heuristic program for locating warehouses*, Management Science, Vol. 9, pp. 643-666, 1963.

LORENA, L.A.N. ; SENNE, E. L. F. ; PAIVA, J. A. C. E PEREIRA M. A.: *Integração de modelos de localização a sistemas de informações geográficas*. Gestao e Produção 8(2):180-195, 2001.

MANIEZZO, V. ; MINGOZZI A. ; BALDACCI R. *A Bionomic Approach to the Capacitated p-Median Problem*. Journal of Heuristics 4, 263–280, 1998.

MORENO-PEREZ, J.A. ; MORENO-VEGA J.M. ; MLADENOVIC N. *Tabu Search and Simulated Annealing in p-Median Problem*. Proceedings of the Canadian Operational Research Society Conference. Montreal, 1994.

OSMAN, I.H. ; CHRISTOFIDES N. *Capacitated Clustering Problems by Hybrid Simulated Annealing and Tabu Search*. International Transactions in Operational Research 1, 317–336, 1994.

PIRKUL H. *Efficient algorithms for the capacitated concentrator location problem*, Computer & Operations Research, Vol. 14, pp. 197-208, 1987.

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

REEVES, C.R. *Genetic Algorithms*. Modern Heuristic Techniques for Combinatorial Problems. 4, pp. 151–196, 1993.

RESENDE, T. A. F. . M. G. C. *Greedy randomized adaptive search procedures*. Journal of Global Optimization, v. 6, p. 109, 1995.

RESENDE, M.G.C; WERNECK, R.F. *A Hybrid Multistart Heuristic for the Uncapacitated Facility Location Problem*. Journal of Heuristics, 10: 59–88, 2005.

ROLLAND, E., D.A. SCHILLING; CURRENT J.R. *An Efficient Tabu Search Procedure for the p-Median Problem*. European Journal of Operational Research 96, 329–342, 1997.

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.

SMIT, JAROSLAV. *Ondas e Antenas*. 2 ed. Érica, 1988. 301 p.

SMIT, JAROSLAV. *Rádio Propagação*, 4 ed. Érica, 1986 . 138 p.

TEITZ, M.B. ; BART. P. *Heuristic Methods for Estimating the Generalized Vertex Median of a Weighted Graph*. *Operations Research* 16, 955–961, 1968.

TOREGAS, C.; REVELLE, C. *Optimal location under time or distance constraints*. *Papers of the Regional Science Association*, v. 28, p. 131-143, 1972.

YEPES, I. *Uma incursão aos algoritmos genéticos*. 2000. Disponível em <<http://www.geocities.com/igoryepes/>>. Acesso em 03 mar. 2007.

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.