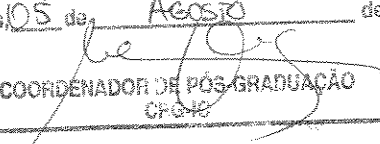


Este exemplar corresponde à redação final da  
Tese/Dissertação devidamente corrigida e defendida  
por: SILVANA LIVRAMENTO  
e aprovada pela Banca Examinadora,  
Campinas 05 de Agosto de 2004  
  
COORDENADOR DE PÓS-GRADUAÇÃO  
CPQ-18

**Algoritmos Genéticos para o Problema de Localização  
de Recursos em Rede Telefônica**

*Silvana Livramento*

**Dissertação de Mestrado**

# **Algoritmos Genéticos para o Problema de Localização de Recursos em Rede Telefônica**

**Silvana Livramento<sup>1</sup>**

Junho de 2004

## **Banca Examinadora:**

- Prof. Dr. Flávio Keidi Miyazawa  
Instituto de Computação, Unicamp (Orientador)
- Prof. Dr. Cid Carvalho de Souza  
Instituto de Computação, Unicamp
- Prof. Dr. Vinícius Amaral Armentano  
Faculdade de Engenharia Elétrica e da Computação, UNICAMP
- Prof. Dr. Ricardo Dahab  
Instituto de Computação, Unicamp (Suplente)

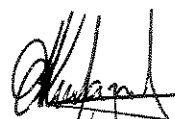
---

<sup>1</sup>Auxílio financeiro do CNPq e CPqD

# **Algoritmos Genéticos para o Problema de Localização de Recursos em Rede Telefônica**

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Silvana Livramento e aprovada pela Banca Examinadora.

Campinas, 23 de junho de 2004.



Prof. Dr. Flávio Keidi Miyazawa  
Instituto de Computação, Unicamp (Orientador)

Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

## TERMO DE APROVAÇÃO

Tese defendida e aprovada em 21 de maio de 2004, pela Banca examinadora composta pelos Professores Doutores:



Prof. Dr. Vinicius Amaral Armentano  
FEEC - UNICAMP



Prof. Dr. Cid Carvalho de Souza  
IC - UNICAMP



Prof. Dr. Flávio Keidi Miyazawa  
IC - UNICAMP

UNIDADE	FE
Nº CHAMADA	
	TI/UNICAMP
	L767a
	EX
OMBO BC/	59875
ROC.	6. 117-04
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREÇO	22.000
DATA	
Nº CPD	

2.6 Id 321975

## FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP

Livramento, Silvana

L767a      Algoritmos genéticos para o problema de localização de recursos  
em rede telefônica / Silvana Livramento -- Campinas, [S.P. :s.n.], 2004.

Orientador : Flávio Keidi Miyazawa

Dissertação (mestrado) - Universidade Estadual de Campinas,  
Instituto de Computação.

1. Otimização combinatória. 2. Algoritmos genéticos. 3.  
Telecomunicações. I. Miyazawa, Flávio Keidi. II. Universidade  
Estadual de Campinas. Instituto de Computação. III. Título.

200424086

© Silvana Livramento, 2004.  
Todos os direitos reservados.

# Resumo

Desenvolvemos Algoritmos Genéticos (AGs) para resolver problemas no projeto de redes de telecomunicações. Um problema consiste em particionar uma grande área de projeto urbana em pequenas seções de serviços, as quais são controladas por um único equipamento de comunicação. O AG desenvolvido para este problema incorpora informações geométricas e topológicas da área de projeto operando diretamente com uma malha de pontos de demanda geograficamente dispersos. Dada uma seção de serviço, a outra fase deste projeto, consiste em agrupar os pontos de demanda em grupos pequenos e posicionar outros equipamentos em postes da rede elétrica, para fazer a comunicação entre estes grupos e o equipamento de comunicação da seção de serviço. Outro AG foi desenvolvido para este problema, e também incorpora informações geométricas e topológicas, pois trabalha diretamente sobre o grafo de vizinhança dos postes existentes numa seção de serviço e a ligação destes com os pontos de demanda. Os resultados computacionais mostraram que os dois AGs são técnicas promissoras para projetar uma rede de telecomunicações, obtendo resultados favoráveis em tempo computacional razoável. Todos os testes foram realizados com instâncias reais tomadas de grandes áreas da cidade de São Paulo.

# Abstract

We propose Genetic Algorithms (GAs) to solve problems in telecommunication network design. The first problem is to partition a large urban project area into smaller service sections, which can be controlled by a single standard communication switch. The GA for this problem, incorporates geometric and topological information from the project area by operating directly with a grid of geographically dispersed demand points. Given a service section, the second problem, consists to group the demand points in smaller areas and to position another equipments in poles of the electric net, to make the communication between these groups and the service section switch. Another GA is developed to this problem, and also incorporates geometric and topological information, since it works directly through the neighborhood graph of existents poles in a service section and the connection between these poles and the demand points. Computation results show both AGs to be a promising technique for telecommunication network design. In the tests, we used real instances taken from large areas in the city of São Paulo.



*A Deus e à minha família.*

# Agradecimentos

Em primeiro lugar quero agradecer a Deus, por estar sempre comigo, por ser a força que me move, e por proporcionar tudo o que foi preciso durante o tempo necessário para este projeto. Pela minha mãe Santíssima que sempre intercede e roga por mim.

Quero agradecer a minha família, que mesmo longe, sempre apoiou as minhas escolhas, confiando no meu sucesso. Ao Renato (vulgo Chenca) e ao Borin, por estarem comigo no início de minha vinda à Campinas, sendo companhia nos momentos de lazer.

Às amigas da primeira república Karla, Renata, Flávia e Taciana. E agora, Ju e Carmen que dividem comigo a casa, a vida e partilham os tempos de folga. Não posso deixar de ressaltar a companhia da Ju e da Taci no horário da novela.

Aos amigos de laboratório, pelas conversas, brincadeiras e grandes ajudas na hora do trabalho: Renato, Guido, Alberto, Daniel, Nilton, Antônio Alberto, Evandro, Eduardo, Ju. Aos muitos amigos e amigas que fiz neste instituto, tornando a estadia aqui muito agradável e alegre: Amanda, Leizza, Fernando, Fábio, Bartho, Zeh, Gregório, Sheila, Victor, e muitos outros que ainda estão aqui e que já seguiram o seu rumo .

Aos amigos de fé e do GPP, pelas nossas partilhas, pelas muitas orações dedicadas a mim, e pelos ideais que buscamos juntos, Justo, Mário, José, Aline, Bete, Júlia, e principalmente à Livia quem primeiro me acolheu em Campinas e me levou a conhecer muitas destas pessoas.

Ao meu namorado Eduardo, por sempre estar por perto me dando força e ânimo para terminar este trabalho, ele muito fez por mim, lendo e corrigindo inúmeras vezes esta dissertação, o artigo e até mesmo a apresentação. E é claro por me proporcionar momentos muito felizes.

Tenho muito a agradecer também às pessoas envolvidas neste projeto, pelas reuniões que fizemos, onde tudo foi discutido, debatido e amadurecido. Agradeço ao Flávio, Alberto, Sheila, Daniel, Nilton, Artur, Arnaldo e ao pessoal do CPqD que acompanhou o projeto: Mário, Rogério, Narbal, Zelmann, Daniel, Brenelli.

Tenho que fazer um agradecimento especial ao Nilton, pela enorme ajuda técnica dada a mim durante todo este tempo de mestrado, tanto sendo um *help online* em todas as áreas em especial sobre *Linux* e configurações.

Por fim, quero agradecer ao CPqD pela ajuda financeira durante o mestrado.

*“O valor das coisas não está no tempo em que elas duram, mas na intensidade com que acontecem. Por isso existem momentos inesquecíveis, coisas inexplicáveis e pessoas incomparáveis.”*

(Fernando Pessoa)

*“Comece pelo necessário, faça o que for possível, ao final você terá feito o impossível.”*

(São Francisco de Assis)

# Sumário

<b>Resumo</b>	<b>vii</b>
<b>Abstract</b>	<b>viii</b>
<b>Agradecimentos</b>	<b>x</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Objetivos do Trabalho . . . . .	3
1.2 Trabalhos Relacionados . . . . .	3
1.3 Organização do Texto . . . . .	4
<b>2 Algoritmos Genéticos</b>	<b>5</b>
2.1 O que são AGs? . . . . .	6
2.2 Como AGs Funcionam? . . . . .	7
2.2.1 Inicialização e Representação . . . . .	7
2.2.2 Função de Aptidão . . . . .	8
2.2.3 Seleção . . . . .	8
2.2.4 Reprodução . . . . .	9
2.3 Por que AGs Funcionam? . . . . .	11
2.3.1 Teorema do Esquema . . . . .	11
2.3.2 <i>Building Blocks</i> . . . . .	12
2.3.3 Diversificação e Intensificação . . . . .	12
<b>3 Descrição do Problema</b>	<b>13</b>
3.1 O Problema . . . . .	13
3.2 Definição de Seções de Serviços . . . . .	14
3.2.1 Formulação Matemática . . . . .	14
3.3 Posicionamento de Caixas Terminais . . . . .	15
3.3.1 Formulação Matemática . . . . .	16

<b>4</b>	<b>AG para Definição de Seções de Serviço</b>	<b>19</b>
4.1	Introdução . . . . .	19
4.2	Modelagem . . . . .	19
4.2.1	Representação . . . . .	19
4.2.2	População Inicial . . . . .	21
4.2.3	Cruzamento . . . . .	22
4.2.4	Mutação . . . . .	26
4.2.5	Função de Avaliação e Penalidade . . . . .	27
4.2.6	Seleção . . . . .	29
4.2.7	Outras Características . . . . .	29
4.3	Resultados . . . . .	30
4.3.1	Ambiente Computacional . . . . .	30
4.3.2	Parâmetros . . . . .	30
4.3.3	Solução . . . . .	35
4.3.4	Comparação com Busca Tabu e GRASP . . . . .	53
<b>5</b>	<b>AG para Posicionamento de Caixas Terminais</b>	<b>56</b>
5.1	Introdução . . . . .	56
5.2	Modelagem . . . . .	56
5.2.1	Representação . . . . .	57
5.2.2	População Inicial . . . . .	57
5.2.3	Cruzamento . . . . .	59
5.2.4	Mutação . . . . .	60
5.2.5	Função de Avaliação . . . . .	61
5.2.6	Otimização . . . . .	63
5.2.7	Outras Características . . . . .	65
5.3	Resultados . . . . .	65
5.3.1	Ambiente Computacional . . . . .	65
5.3.2	Parâmetros . . . . .	65
5.3.3	Instâncias . . . . .	66
5.3.4	Teste de Parâmetros . . . . .	66
5.3.5	Conclusão dos resultados . . . . .	76
<b>6</b>	<b>Conclusão</b>	<b>87</b>
6.1	Trabalhos Futuros . . . . .	88
	<b>Bibliografia</b>	<b>89</b>

# Lista de Tabelas

4.1	Probabilidades para seleção da célula que entra no corte. . . . .	24
4.2	Instâncias. . . . .	30
4.3	Dados do projeto. . . . .	30
4.4	Dados do algoritmo genético. . . . .	31
4.5	Plano de teste para $K_U = 240$ . . . . .	32
4.6	Resultado dos testes para $K_U = 240$ . . . . .	32
4.7	Plano de teste para $K_U = 300$ . . . . .	34
4.8	Resultado dos testes para $K_U = 300$ . . . . .	34
4.9	Testes para o Bairro da Liberdade. . . . .	45
4.10	Resultado dos testes para o Bairro da Liberdade. . . . .	45
4.11	Resultados da BT . . . . .	53
4.12	Resultados do GRASP . . . . .	55
4.13	Comparação entre AG, BT e GRASP . . . . .	55
5.1	Dados do projeto de posicionamento de caixas. . . . .	65
5.2	Dados do AG. . . . .	66
5.3	Dados numéricos sobre as instâncias. . . . .	66
5.4	Plano de testes para AGPC. . . . .	67
5.5	Resultados para a instância 1. . . . .	68
5.6	Resultados para a instância 2. . . . .	71
5.7	Resultados para a instância 3. . . . .	72
5.8	Resultados para a instância 4. . . . .	75

# Lista de Figuras

1.1	Rede telefônica cabeada . . . . .	2
2.1	Operações básicas de um AG . . . . .	8
2.2	<i>Crossover</i> de um único ponto . . . . .	10
2.3	Uma única mutação . . . . .	11
4.1	Malha de Vizinhança. . . . .	20
4.2	Um cromossomo. . . . .	20
4.3	Os quatro tipos de corte na malha. . . . .	23
4.4	Cruzamento. . . . .	25
4.5	Filho resultante depois da operação de reparo. . . . .	26
4.6	Seleção por roleta. . . . .	29
4.7	Distribuição de demanda. . . . .	33
4.8	Distribuição de demanda com $K_U = 300$ . . . . .	36
4.9	Solução para o Sul da Vila Mariana: demanda por seção. . . . .	37
4.10	Solução para o Sul da Vila Mariana: custo por seção. . . . .	38
4.11	Solução para o Sul da Vila Mariana: custo total por iteração. . . . .	39
4.12	Solução para o Sul da Vila Mariana: custo de cabeamento por iteração. . . . .	39
4.13	Solução para o Sul da Vila Mariana. . . . .	40
4.14	Solução para o Leste da Vila Mariana: demanda por seção. . . . .	41
4.15	Solução para o Leste da Vila Mariana: custo por seção. . . . .	42
4.16	Solução para o Leste da Vila Mariana: custo total por iteração. . . . .	43
4.17	Solução para o Leste da Vila Mariana: custo de cabeamento por iteração. . . . .	43
4.18	Solução para o Leste da Vila Mariana. . . . .	44
4.19	Solução para o Norte da Vila Mariana: demanda por seção . . . . .	45
4.20	Solução para o Norte da Vila Mariana: custo total por iteração . . . . .	46
4.21	Solução para o Norte da Vila Mariana: custo de cabeamento por iteração . . . . .	46
4.22	Solução para o Norte da Vila Mariana. . . . .	47
4.23	Solução para o Norte da Vila Mariana: demanda por seção . . . . .	48
4.24	Solução para o Norte da Vila Mariana: custo por seção . . . . .	48

4.25	Solução para o Norte da Vila Mariana: custo total por iteração . . . . .	49
4.26	Solução para o Norte da Vila Mariana: custo de cabeamento por iteração . . . . .	49
4.27	Solução para o Norte da Vila Mariana. . . . .	50
4.28	Solução para o do Bairro da Liberdade: demanda por seção. . . . .	51
4.29	Solução para o do Bairro da Liberdade: custo por seção. . . . .	52
4.30	Solução para o do Bairro da Liberdade: custo total por iteração. . . . .	52
4.31	Solução para o do Bairro da Liberdade: demanda por seção. . . . .	53
4.32	Solução para o Bairro da Liberdade. . . . .	54
5.1	Representação de um cromossomo. É um vetor de tamanho igual ao número de pontos de demanda. Cada posição do vetor contém o identificador do poste que atende o cliente e o identificador da caixa deste poste onde realmente ele está conectado. Cada posição só pode conter identificadores de postes adjacentes ao cliente. Estas adjacências são dadas pelo grafo de clientes e postes. . . . .	58
5.2	Uma caixa terminal, onde $P$ é o identificador do poste no qual ela está instalada, $C$ é a capacidade, $N$ indica se a caixa foi instalada pelo algoritmo e $D$ é a demanda . . . . .	59
5.3	Decodificação do cromossomo em caixas terminais. . . . .	60
5.4	Algoritmo de inicialização de um indivíduo $\pi$ . . . . .	61
5.5	Cruzamento . . . . .	62
5.6	Algoritmo de reparo. . . . .	63
5.7	Uma mutação . . . . .	63
5.8	Instância 1: Distribuição de demanda nas caixas com capacidade 10. . . . .	69
5.9	Instância 1: Distribuição de demanda nas caixas com capacidade 20. . . . .	70
5.10	Instância 1: Gráfico de evolução do AGPC. Custo x Geração. . . . .	71
5.11	Instância 1: Custo de caixas e fiação. . . . .	72
5.12	Instância 1: Distribuição de demanda nas caixas com capacidade 10. . . . .	73
5.13	Instância 1: Distribuição de demanda nas caixas com capacidade 20. . . . .	73
5.14	Instância 1: Solução. . . . .	74
5.15	Instância 2: Distribuição de demanda nas caixas com capacidade 10. . . . .	75
5.16	Instância 2: Distribuição de demanda nas caixas com capacidade 20. . . . .	76
5.17	Instância 2: Gráfico de evolução do AGPC. Custo x Geração. . . . .	77
5.18	Instância 2: Distribuição de demanda nas caixas com capacidade 10. . . . .	78
5.19	Instância 2: Distribuição de demanda nas caixas com capacidade 20. . . . .	78
5.20	Instância 2: Solução. . . . .	79
5.21	Instância 3: Distribuição de demanda nas caixas com capacidade 10. . . . .	80
5.22	Instância 3: Distribuição de demanda nas caixas com capacidade 20. . . . .	80
5.23	Instância 3: Gráfico de evolução do AGPC. Custo x Geração. . . . .	81
5.24	Instância 3: Distribuição de demanda nas caixas com capacidade 10. . . . .	81



5.25	Instância 3: Distribuição de demanda nas caixas com capacidade 20. . . . .	82
5.26	Instância 3: Solução. . . . .	83
5.27	Instância 4: Distribuição de demanda nas caixas com capacidade 10. . . . .	84
5.28	Instância 4: Distribuição de demanda nas caixas com capacidade 20. . . . .	84
5.29	Instância 4: Gráfico de evolução do AGPC. Custo x Geração. . . . .	85
5.30	Instância 4: Distribuição de demanda nas caixas com capacidade 10. . . . .	85
5.31	Instância 4: Distribuição de demanda nas caixas com capacidade 20. . . . .	86
5.32	Instância 4: Solução. . . . .	86

# Capítulo 1

## Introdução

O sistema urbano de telecomunicações é organizado como um grande e sofisticado sistema de hierarquias possibilitando a comunicação de centenas de milhares de clientes (ou assinantes).

Para obter maior facilidade no planejamento, projeto e na administração de uma rede telefônica, cidades são divididas em grandes áreas, denominadas *áreas de projeto* e cada uma dessas áreas é atendida por uma *estação telefônica*. A estação telefônica provê acesso e comunicação para todos os clientes pertencentes a uma mesma área de projeto.

Uma área de projeto, por sua vez, é dividida em áreas menores, chamadas *seções de serviço*. Cada seção de serviço é atendida por um único equipamento, chamado *armário de distribuição*. A rede de cabos que conecta os armários de distribuição existentes numa área de projeto à sua estação telefônica é chamada de *rede alimentadora ou primária*. Geralmente a rede alimentadora é subterrânea e composta por fibras óticas, mas podendo ser também por satélite ou rádio.

A *rede de distribuição ou secundária* provê a conexão entre um armário de distribuição e os clientes de uma mesma seção de serviço, esta conexão é feita, geralmente, por pares de fios de cobre empacotados em um mesmo cabo.

Os clientes não são conectados diretamente aos armários de distribuição, eles precisam ser agrupados em pequenas *caixas terminais* que são instaladas nos postes pertencentes à rede elétrica. A figura 1.1 ilustra o posicionamento dos equipamentos de um sistema de rede telefônica.

O projeto de uma rede de telecomunicação é muito complexo e custoso, visto que os equipamentos utilizados são muito caros e a mão-de-obra para realizá-lo tem que ser muito especializada. Por outro lado, as empresas de telecomunicações devem ser capazes de prover atendimento imediato às necessidades que surgem, com preços competitivos. No mercado de Telecomunicações as regras são claras, as empresas precisam ter cada vez mais clientes, comprando ainda mais serviços, a custos cada vez mais atraentes. Para as operadoras, os ganhos reais estão diretamente relacionados com a qualidade dos serviços prestados, sustentados pela otimização dos recursos e dos processos das redes existentes. Por isso um bom projeto que

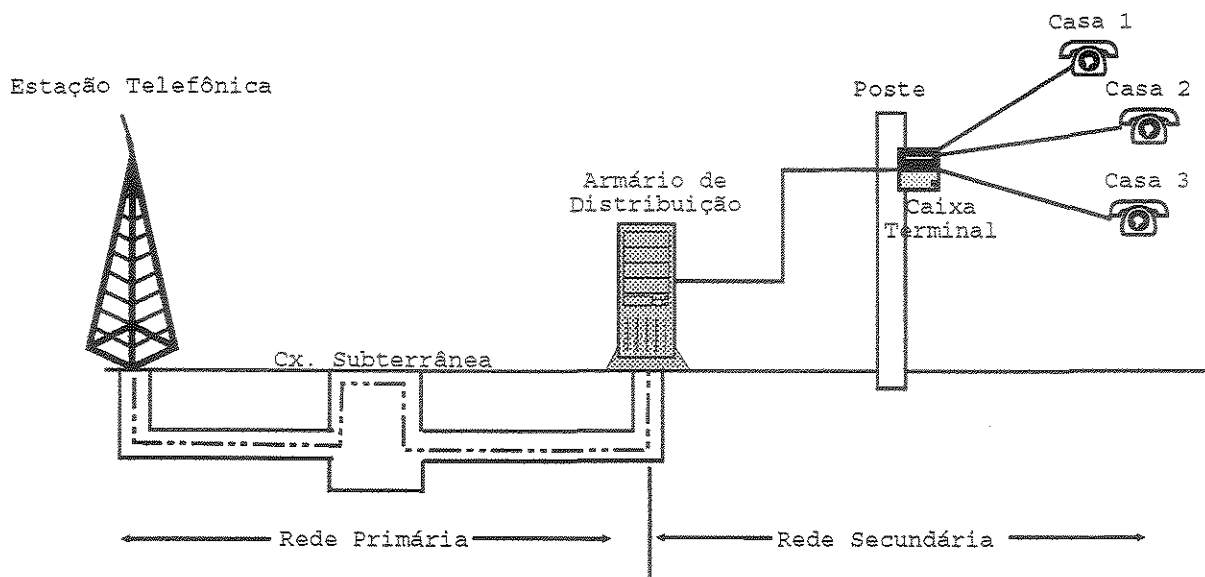


Figura 1.1: Rede telefônica cabeada

minimiza o custo total dos equipamentos, mão-de-obra é muito importante.

O objetivo deste trabalho é desenvolver algoritmos genéticos para auxiliar no planejamento e projeto de redes de distribuição de uma única área de projeto, que envolve os seguintes itens:

- Definição de seções de serviço.
- Posicionamento dos armários de distribuição dentro das seções.
- Posicionamento das caixas terminais dentro de cada seção.
- Ligação dos clientes às caixas terminais.

A motivação para o estudo desta parte do problema reside no fato de que a rede de distribuição possui um maior percentual do custo total do sistema de hierarquia da rede. Pois como foi dito, a rede de alimentação é hoje composta por fibra ótica, que é muito mais barata do que fios de cobre, pois atendem muito mais linhas. Hoje em dia o projeto de redes de distribuição é feito por engenheiros elétricos que podem levar aproximadamente seis meses para definir as seções e posicionar os armários em uma área de projeto não muito grande.

É claro que não pretendemos realizar o projeto final de uma rede de distribuição, mas sim um pré-projeto onde um engenheiro especializado poderá se guiar para acertar as restrições geográficas, limites de áreas, restrições de posicionamento de armários em locais que o proprietário não aceita, etc.

Decidimos usar Algoritmos Genéticos (AGs) para solucionar este problema, visto que ele é um problema combinatório muito grande e pertencente a classe NP-difícil [9], e portanto é improvável que haja métodos exatos capazes de resolvê-lo de forma eficiente.

Algoritmos Genéticos são métodos heurísticos baseados na seleção natural do mais apto, e seguem a Teoria da Evolução de Charles Darwin. O poder do AG vem do fato de ser uma técnica robusta, podendo lidar com sucesso para uma grande variedade de áreas de problemas, incluindo aquelas que são difíceis para outros métodos resolverem [2]. Não é garantido que AGs achem a solução ótima global para um problema, mas eles são geralmente bons em achar soluções “aceitavelmente boas” num tempo “razoavelmente bom”.

## 1.1 Objetivos do Trabalho

Os objetivos do trabalho de mestrado são:

- Desenvolver dois algoritmos genéticos para resolver o projeto de rede de distribuição telefônica que inclui:
  - Um algoritmo genético para resolver o problema de “Definição de Seções de Serviço e Posicionamento de Armários de Distribuição”, também chamado Fase 1. Dada uma área de projeto, este problema consiste em dividi-la em seções de serviço, posicionando seus respectivos armários.
  - Um algoritmo genético para resolver o problema de “Posicionamento de Caixas Terminais Considerando a Rede Existente”, também chamado de Fase 2. A partir de uma seção de serviço, este problema consiste em definir o posicionamento de caixas terminais em postes, bem como determinar as ligações entre clientes e caixas.
- Fazer com que os algoritmos possibilitem uma economia de recursos e tempo quando utilizados por engenheiros de projeto responsáveis pela elaboração e planejamento da rede de distribuição.

## 1.2 Trabalhos Relacionados

Existem muitos trabalhos onde são desenvolvidos algoritmos para as várias partes do projeto de um sistema de rede telefônica, como a rede de alimentação, cabeamento, definição de seções, caixas terminais, entre outros.

Cabral em [5] estuda a rede de alimentação como uma analogia à rede de Steiner e o problema de transporte com uma única fonte, resolvendo o problema em três fases: inicialização, heurística e otimização. Costa em [7], também estuda a rede de alimentação, mais voltado para a localização de centros de comutação, formulando o problema matematicamente como um problema de localização de facilidades capacitado. Já Almeida et al em [1] apresenta um Algoritmo Evolutivo Multi-objetivo para resolver o problema de localização de estações telefônicas.

Existe um grande número de trabalhos relacionados com a rede de distribuição, alguns estão citados a seguir. Em [14], Mateus et. al. estuda a expansão de uma rede de distribuição considerando a definição de seções de serviço bem como a modularidade dos cabos e suas emendas. O problema é formulado como um modelo matemático. Em [13], Mateus et. al. trata do problema de projeto de rede de acesso generalizado que consiste em achar uma topologia, atribuições de capacidade e roteamento. Formula o problema como programação inteira usando uma heurística de relaxação Lagrangeana e o método do subgradiente. Já em [15], Mateus et. al. estuda o problema de localização exata de equipamentos na rede também usando relaxação Lagrangeana. Em [8], ele trata o problema de cabeamento entre o armário de distribuição e as caixas terminais usando também relaxação Lagrangeana. Por fim, Khuri et. al., em [11], estuda o problema de posicionamento de terminais, o qual pode ser entendido como o problema de posicionamento de caixas terminais, usando duas heurísticas uma gulosa e um algoritmo genético.

Em [6], uma seção inteira é dedicada a algoritmos evolutivos para problemas em redes de telecomunicação. Esta seção inclui estudos em segurança, rede para uma rápida transferência de dados, planejamento da rede e rede ótica, entre outros.

O nosso trabalho possui um diferencial no sentido de usar uma técnica diferente na modelagem de seção de serviço. Muitos dos trabalhos nesta área consideram a existência de seções de serviço e/ou alguns possíveis locais para o posicionamento de seus armários. O nosso trabalho está exatamente nesta parte, definir as seções levando em consideração as restrições geográficas, físicas e topológicas, e depois desta fase definir a posição das caixas terminais.

O trabalho na definição de seções de serviço já resultou em um artigo publicado [12].

## 1.3 Organização do Texto

O restante desta dissertação está organizado da seguinte forma. O capítulo 2 define conceitos básicos sobre algoritmos genéticos. O capítulo 3 define os problemas que iremos tratar, as restrições e formulações. O capítulo 4 apresenta a modelagem em algoritmos genéticos e os resultados obtidos para o problema de “Definição de Seções de Serviço”. O capítulo 5 apresenta a modelagem em algoritmos genéticos e os resultados obtidos para o problema de “Posicionamento de Caixas Terminais”. Finalmente, o capítulo 6 apresenta as conclusões do trabalho de mestrado, e propõe algumas extensões para trabalhos futuros.

## Capítulo 2

# Algoritmos Genéticos

Algoritmos Genéticos (AGs) são métodos adaptativos que podem ser usados para resolver problemas de busca e otimização. Eles são baseados nos processos genéticos dos organismos biológicos. Depois de muitas gerações, populações naturais evoluem de acordo com os princípios da seleção natural e da “sobrevivência do mais apto”, discutidos, primeiramente, por Charles Darwin em “*The Origin of the Species*”. Imitando este processo, AGs são capazes de “evoluir” obtendo boas soluções para problemas do mundo real, se forem codificados apropriadamente.

Os princípios básicos dos AGs foram esboçados primeiro e rigorosamente por Holland [10], porém existem vários textos que os descrevem muito bem ([2, 3, 16, 17]).

Na natureza, indivíduos numa mesma população competem entre si por recursos como comida, água e abrigo. Membros da mesma espécie também competem entre si para atrair um parceiro. Aqueles indivíduos que são mais bem sucedidos em sobreviver e atrair parceiros terão um número relativamente grande de descendentes. Indivíduos menos aptos produzirão poucos ou até mesmo nenhum descendente. Isto significa que genes dos indivíduos altamente adaptados, ou mais aptos, estarão presentes num número crescente de indivíduos, em cada geração sucessiva. A combinação de características boas de diferentes ancestrais pode, às vezes, produzir descendentes “super-aptos”, cuja aptidão é muito maior do que a dos pais. Desta maneira, espécies evoluem para se tornarem cada vez mais adaptadas ao seu ambiente.

AGs usam uma analogia direta com o processo evolutivo natural. Eles trabalham com uma população de “indivíduos”, também chamados de *cromossomos*, cada um representando uma possível solução para um dado problema. Uma “contagem de aptidão” (*fitness score*) é atribuída a cada indivíduo de acordo com quão boa é a sua solução para o problema que codifica. Aos indivíduos mais aptos são dadas oportunidades de se “reproduzirem” por cruzamento com outros indivíduos na população. Assim, são produzidos novos indivíduos, que compartilham algumas características tomadas de cada “pai”. Os membros menos aptos da população são menos prováveis de serem selecionados para reprodução, e então tendem a desaparecer.

Uma nova população de possíveis soluções é então produzida selecionando-se os melho-

res indivíduos da “geração” corrente, e agrupando-os para produzir novos indivíduos. Esta nova geração contém uma alta proporção de características dos membros aptos da geração anterior. Desta maneira, depois de muitas gerações, características boas são propagadas através da população, sendo misturadas e trocadas com outras características. Favorecendo a junção de indivíduos mais aptos, as áreas mais promissoras do espaço de busca são exploradas. Se o AG foi bem projetado, a população irá *convergir* para uma solução adequada para o problema.

O poder dos AGs vem do fato de serem uma técnica robusta, podendo lidar de forma bem sucedida com uma grande variedade de problemas, incluindo aqueles que são difíceis para outros métodos resolverem. AGs não garantem achar a solução ótima global para um problema, mas eles são geralmente bons para achar uma solução “aceitavelmente boa” para problemas “num tempo razoavelmente rápido”. Nas próximas seções detalharemos o que são AGs, como e porque eles funcionam.

## 2.1 O que são AGs?

Daremos uma breve resposta que está em [21]:

*“Um AG pode ser visto como um tipo não usual de estratégia de busca. Num AG, há um conjunto de soluções candidatas para um problema; tipicamente este conjunto é inicialmente preenchido ao acaso com possíveis soluções, não necessariamente todas distintas. Cada candidato é tipicamente (ainda que não em todos os AGs) um vetor ordenado de tamanho fixo de valores (chamados ‘alelos’) para atributos (‘genes’). Cada gene é considerado atômico; o conjunto de alelos para este gene é o conjunto de valores que o gene pode possuir. Então, construindo um AG para um problema específico a primeira tarefa é decidir como representar possíveis soluções.”*

Algoritmos genéticos pertencem à classe dos algoritmos probabilísticos, e combinam elementos de busca direta e estocástica. Outra propriedade importante de tais algoritmos é que eles mantêm uma população de soluções potenciais – todos os outros métodos processam um único ponto do espaço de busca por vez.

Como descrito em [16], um algoritmo genético para um problema particular deve possuir os seguintes componentes:

1. uma representação genética para soluções potenciais do problema;
2. uma maneira de criar uma população inicial de soluções potenciais;
3. uma função capaz de comparar soluções em termos de sua “aptidão”;
4. operadores genéticos que alteram a composição dos pais, produzindo filhos;

5. valores para os vários parâmetros que o algoritmo genético usa (tamanho da população, probabilidades de aplicar os operadores genéticos, entre outros).

## 2.2 Como AGs Funcionam?

Algoritmos genéticos resolvem um problema gerando, mudando e avaliando soluções candidatas para aquele problema. Uma solução candidata para o problema é chamada de *cromossomo*. Um cromossomo é usualmente uma cadeia de *bits* ou qualquer outra codificação da solução.

Inicialmente, uma população aleatória de cromossomos é gerada. Mudanças nos cromossomos são feitas por mutação e/ou operadores de *crossover*. Os cromossomos são avaliados por uma função de aptidão dependente do domínio, que primeiro decodifica o cromossomo e depois avalia sua qualidade como solução para o problema particular.

Segue um esboço do fluxo de controle de um algoritmo genético básico, que também está ilustrado na figura 2.1:

1. Inicialize e codifique uma população de cromossomos aleatoriamente. É chamada de 'população corrente'.
2. Avalie a aptidão de cada cromossomo na população corrente.
3. Produza uma geração intermediária, selecionando estocasticamente cromossomos da população corrente de acordo com sua aptidão. Eles serão os pais da próxima geração.
4. Aplique os operadores de mutação e *crossover* nos cromossomos selecionados na população intermediária, produzindo assim uma nova geração de cromossomos. Esta é agora a população corrente.
5. Repita 2-4 até encontrar uma solução adequada, ou atingir um critério de parada.

Os detalhes de como um AG funciona são explicados nas seções seguintes.

### 2.2.1 Inicialização e Representação

Há várias maneiras de inicializar e codificar a população inicial: binário ou não-binário, cadeias de tamanho fixo ou variável, e assim por diante. Holland [10] primeiro sugeriu que o ideal seria usar um alfabeto binário e uma cadeia de tamanho fixo. Entretanto, freqüentemente isso não é desejável nem natural. No estágio inicial, o sistema apenas gera aleatoriamente cromossomos válidos e avalia cada um.



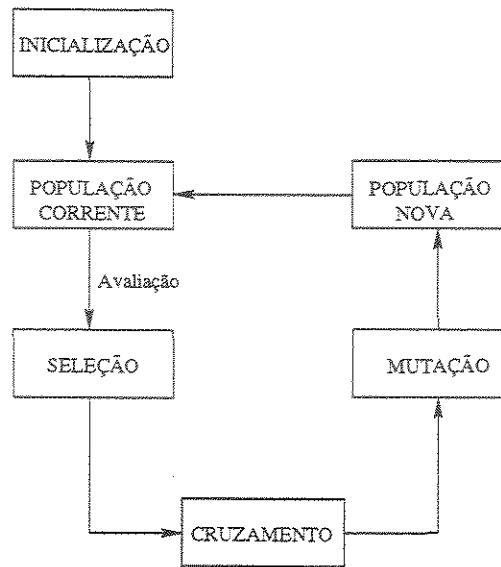


Figura 2.1: Operações básicas de um AG

### 2.2.2 Função de Aptidão

Em termos genéticos, o conjunto de parâmetros representado por um cromossomo particular é referenciado como um *genótipo*. O genótipo contém as informações necessárias para construir um organismo – que refere-se ao *fenótipo*. Os mesmos termos são usados nos AGs. A aptidão de um indivíduo depende do desempenho do fenótipo. Isto pode ser inferido do genótipo – isto é, pode ser computado do cromossomo, usando a função de aptidão (ou avaliação).

Para cada problema, devemos construir uma função de avaliação. Dado um cromossomo particular, a função de avaliação retorna um único número relativo à sua “aptidão” (*fitness*), que é proporcional à “qualidade” da solução que o cromossomo representa.

### 2.2.3 Seleção

O objetivo da seleção é retornar pais selecionados probabilisticamente. A chance de cada pai ser selecionado está de alguma maneira relacionada com sua aptidão.

Michalewicz em [16], destaca que existem dois casos importantes no processo de evolução da busca genética: diversidade da população e pressão seletiva (*selective pressure*). Estes fatores estão fortemente relacionados: um aumento na pressão seletiva diminui a diversidade da população, e vice-versa. Em outras palavras, uma pressão seletiva forte leva a convergência prematura da busca do AG; uma pressão seletiva fraca pode fazer a busca ineficiente. Portanto, é importante obter um balanço entre estes dois fatores; isto é o objetivo dos mecanismos de amostragem.

Em [16], existe uma listagem de variações para a seleção no AG, entre elas temos:

### Seleção Baseada na Aptidão

O método padrão e original de seleção é a seleção por *Roleta* ou seleção baseada na aptidão. Neste tipo de seleção cada cromossomo tem uma chance de seleção diretamente proporcional à sua aptidão. O seu efeito depende fortemente da variação dos valores da aptidão na população corrente. Por exemplo, se a aptidão varia de 5 a 10, então o melhor cromossomo tem duas vezes mais chances de ser selecionado do que o pior. Porém, se acrescentarmos 1000 a cada aptidão para que a variação da aptidão fique entre 1005 e 1010, então a chance relativa de cada indivíduo ser selecionado é igual para todos. Em muitas aplicações isto não é desejado, e leva a estagnação na busca. Uma maneira para resolver este problema é escalar a aptidão antes da seleção.

### Seleção Baseada em Ordenação (*Rank*)

Neste método as probabilidades de seleção são baseadas numa ordenação ou posição relativa dos cromossomos na população, ao invés da aptidão absoluta. Existem muitas possibilidades de realizar este método, dependendo de como os valores a serem ordenados são convertidos em aptidão.

### Seleção Baseada em Torneio

A seleção por torneio original escolhe  $k$  cromossomos aleatoriamente e retorna o mais apto entre eles. Outros métodos existem, como por exemplo selecionar dois ou mais pelo método da roleta e retornar o mais apto.

### Seleção Uniforme

Este método seleciona aleatoriamente um indivíduo da população e retorna-o. Todos os indivíduos possuem a mesma chance de serem selecionados.

## 2.2.4 Reprodução

Como descrito em Beasley et al. [2], durante a fase de reprodução do AG, indivíduos são selecionados da população e recombinados, produzindo filhos os quais comporão a próxima geração. Os pais são selecionados da população corrente usando um esquema que favorece os indivíduos mais aptos. Indivíduos bons provavelmente serão selecionados várias vezes numa geração. Os piores podem nem ser selecionados.

Tendo selecionado dois pais, seus cromossomos são *recombinados*, tipicamente usando os mecanismos de *crossover* e *mutação*. A forma mais básica destes operadores é descrita a seguir.

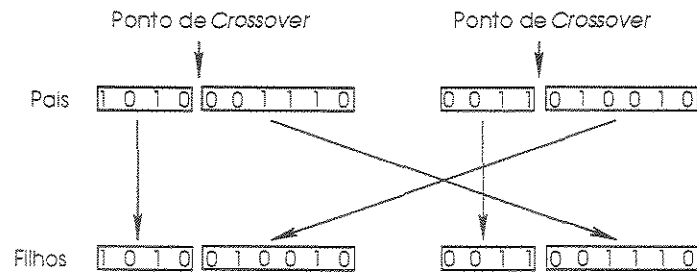


Figura 2.2: *Crossover* de um único ponto

### *Crossover*

O operador de cruzamento (ou *crossover*) é o operador mais importante em AGs. Ele é um processo que lida com recombinação de cadeias de bits trocando segmentos entre os pares de cromossomos. Existem muitos tipos de cruzamento, ilustramos abaixo quatro deles.

- *Cruzamento de um-ponto*: o procedimento do cruzamento de um-ponto é gerar aleatoriamente um número (menor ou igual ao tamanho do cromossomo) para ser o ponto de cruzamento. Os bits que estão antes do número permanecem inalterados e troca os bits que estão depois do ponto de cruzamento entre os dois pais. A figura 2.2 ilustra este procedimento.
- *Cruzamento de dois-pontos*: O procedimento do cruzamento de dois-pontos é parecido com o de um ponto, exceto que precisamos selecionar duas posições e apenas os bits que estão entre as duas posições são trocados.
- *Cruzamento de  $n$ -pontos*: O procedimento do cruzamento de  $n$ -pontos é semelhante aos de um e dois pontos, exceto que precisamos selecionar  $n$  posições e apenas os bits que estão entre as posições ímpares e pares são trocados. Os bits entre as posições pares e ímpares não mudam.
- *Cruzamento Uniforme*: Cada gene do primeiro pai tem probabilidade 0.5 de ser trocado com o gene correspondente do segundo pai.

### *Mutação*

A mutação tenta fazer com que todos os possíveis cromossomos sejam alcançáveis. Ela é muito útil pois a operação de cruzamento não pode produzir novos cromossomos se eles não aparecerem na geração inicial.

A mutação é aplicada a cada filho individualmente, depois da operação de cruzamento, alterando-se aleatoriamente cada gene com uma probabilidade baixa. A figura 2.3 mostra o quinto gene do cromossomo alterado.

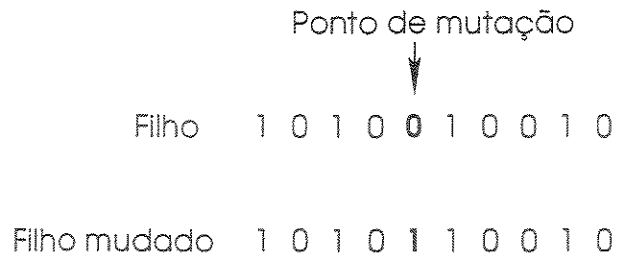


Figura 2.3: Uma única mutação

## 2.3 Por que AGs Funcionam?

Muitas pesquisas em AGs estão concentradas em achar regras empíricas para provar porque se desempenham bem. Não existe uma “teoria geral” aceita que explica exatamente porque AGs têm as propriedades que eles possuem. Entretanto, muitas hipóteses têm sido introduzidas que podem explicar parcialmente o sucesso dos AGs. Elas podem nos ajudar a implementar boas aplicações com AGs.

### 2.3.1 Teorema do Esquema

O *teorema do esquema* de Holland [10] foi a primeira explicação rigorosa de como AGs funcionam. Um esquema é um padrão de valores para os genes que pode ser representado (numa codificação binária) por uma cadeia de caracteres no alfabeto  $\{0,1,\#\}$ . Um cromossomo contém um esquema particular se ele é compatível com um esquemata, com o símbolo “#” casando com qualquer caracter do alfabeto. Então, por exemplo, o cromossomo “1010” contém, entre outros, os esquematas “10##”, “#0#0”, “##1#”, “101#”. A *ordem* de um esquema é o número de símbolos diferentes de # que ele contém (2,2,1,3 respectivamente no exemplo). O *tamanho* de um esquema é a distância, mais um, entre os símbolos extremos (esquerdo e direito) não-# (2, 3, 1, 3 respectivamente no exemplo).

O *teorema do esquema* explica o poder de um AG em termos de como um esquemata é processado. São dadas oportunidades de reprodução a indivíduos da população, e assim produzem uma descendência. O número de tais oportunidades que um indivíduo recebe é proporcional à sua aptidão – assim os melhores indivíduos contribuem com mais genes para a próxima geração. É assumido que a alta aptidão de um indivíduo é devido ao fato dele possuir bons esquematas. Passando um ou mais desses esquematas bons para a próxima geração, a probabilidade de achar mais soluções boas aumenta.

Holland mostrou que a maneira ótima de explorar o espaço de busca é dar oportunidades de reprodução aos indivíduos, proporcionalmente à sua aptidão relativa ao resto da população. Desta maneira, bons esquematas recebem um número crescente exponencialmente de oportuni-

dades em gerações sucessivas. Isto é chamado o teorema do esquema.

### 2.3.2 *Building Blocks*

O teorema do esquema nos torna capazes de formular tentativas e idéias iniciais para entender porque um AG funciona bem e como podemos torná-lo mais eficiente. O ponto mais claro sugerido pelo teorema do esquema é que esquemas altamente aptos de baixa ordem e tamanho pequeno parecem ser particularmente importantes para um AG. De fato, eles são tão importantes que merecem ter seu próprio nome, e são chamados de *building blocks*. Eles são esquemas de tamanho pequeno consistindo de *bits* que trabalham bem juntos, e tendem a lidar com um desempenho melhorado quando incorporados em um indivíduo. Uma codificação do problema bem sucedida é aquela que força a formação de *building blocks* assegurando que:

1. Genes relacionados estão em posições próximas no cromossomo, enquanto
2. existe uma pequena interação entre genes.

*Interação* entre genes significa que a contribuição de um gene para a aptidão depende do valor de outros genes no cromossomo. Se estas regras são observadas, então um AG será tão eficiente como predito pelo teorema do esquema.

Infelizmente, as condições (1) e (2) não são fáceis de alcançar. Em muitos casos, a natureza exata do relacionamento entre os genes pode não ser conhecida pelo programador, mesmo que haja apenas relações simples, ainda pode ser impossível arrumar o código para refleti-las.

### 2.3.3 *Diversificação e Intensificação*

Qualquer algoritmo de otimização eficiente deve usar duas técnicas para achar um máximo (mínimo) global: *diversificação* para investigar áreas novas e desconhecidas no espaço de busca, e *intensificação* para usar o conhecimento encontrado em pontos previamente visitados para achar pontos melhores. Estas duas técnicas são contraditórias, e um bom algoritmo de busca deve achar um equilíbrio entre as duas.

Holland mostrou que um AG combina diversificação e intensificação *ao mesmo tempo* de uma maneira ótima. Entretanto, mesmo que isso seja *teoricamente* verdadeiro para um AG, há problemas inevitáveis na prática. Isto acontece porque Holland fez certas suposições simplificadas, incluindo:

1. que o tamanho da população é infinito,
2. que a função de avaliação reflete precisamente a utilidade de uma solução, e
3. que os genes num cromossomo não interagem significativamente.

# Capítulo 3

## Descrição do Problema

### 3.1 O Problema

Usualmente, e particularmente no Brasil, uma rede de telecomunicações urbana é organizada como um grande e sofisticado sistema de hierarquias. Nos níveis mais altos encontramos um conjunto de *estações telefônicas*, onde cada estação pode englobar uma cidade inteira. Nos níveis mais baixos existem *armários de distribuição*, dos quais cada cliente recebe sua linha individual.

Para melhor entender o problema em questão, a rede de telecomunicações é particionada em dois segmentos lógicos. O primeiro é formado por grandes entroncamentos, tipicamente conectados por fibra óptica, rádio ou satélite. Este primeiro segmento termina numa estação telefônica, que pode controlar milhares de linhas telefônicas, porém este número pode variar consideravelmente. O segundo segmento compreende a área controlada por uma destas estações, a qual chamamos de *área de projeto*. Cada área de projeto, por sua vez, é subdividida num número de áreas menores, chamadas *seções de serviço*, cada seção de serviço tipicamente controla em torno de quinhentas linhas telefônicas. Dentro de cada seção de serviço é localizado um *armário de distribuição* independente, ligado por fibra óptica à estação que controla a área de projeto onde a seção de serviço está localizada. Tipicamente, pares de fios de cobre (empacotados em cabos) saem do armário de distribuição para atenderem os clientes, estes também chamados *pontos de demanda*. Um ponto de demanda pode conter mais do que uma linha. Além disto, como a área de projeto também leva em consideração uma estimativa de futuras requisições para pares de comunicação, um ponto de demanda pode ser um número fracionário.

Os pares de fios de cobre que saem do armário de distribuição não são conectados diretamente aos clientes. Eles são levados até pequenas *caixas terminais*, fixadas nos postes pertencentes à rede elétrica local, onde cada caixa terminal acomoda de 10 a 20 pares. Também é necessário definir o número, tipo e posição das caixas terminais a serem instaladas, e escolher quais pontos de demanda serão conectados a cada caixa terminal.

Finalmente, existe o problema do cabeamento. Cada cabo empacota de dezenas a centenas de pares de cobre. Os cabos são fixados nos postes, formando uma rede aérea. Os cabos saem dos armários de distribuição e grupos de pares de cobre são extraídos e conectados às caixas terminais, num número suficiente para atender a demanda local, associada a cada caixa, localizada na rota do cabo.

Neste trabalho de mestrado abordamos dois problemas que chamamos de “Definição de Seções de Serviço” e “Posicionamento de Caixas Terminais”, respectivamente. Nas subseções seguintes explicaremos com mais detalhes estes dois problemas.

## 3.2 Definição de Seções de Serviços

Dada uma área de projeto que possui um conjunto de pontos de demanda, representando uma estimativa da sua necessidade presente e futura por linhas de telecomunicação, este problema consiste em particionar a área em seções de serviço e localizar os armários de distribuição dentro de cada seção, respeitando um número de restrições de projeto e minimizando uma função de custo.

Este problema possui restrições que estão listadas a seguir:

1. Todo ponto de demanda deve pertencer a uma única seção de serviço.
2. Pontos com demanda maior que um valor  $K_U$ , são considerados seções de serviço locais. Para cada ponto deste tipo um ou mais armários de distribuição são instalados somente para atendê-lo.
3. Um armário de distribuição de capacidade  $C$  pode atender uma demanda em torno de  $C \times k$  e  $C \times K$ , onde  $0 < k < K \leq 1$ . As constantes  $k$  e  $K$  são fatores de carga mínima e máxima, respectivamente.
4. Seções de serviço próximas devem atender, se possível, aproximadamente a mesma demanda.

### 3.2.1 Formulação Matemática

A área de projeto é dada por um conjunto  $A$  de triplas  $(x, y, d) \in \mathbb{R}^3$ , onde  $x$  e  $y$  são as coordenadas do ponto de demanda, e  $d$  representa a demanda, em termos de linhas de telecomunicação, para ser atendida nesta posição.

Uma solução para este problema é uma partição  $\pi$  de  $A$ , que satisfaça restrições de capacidade máxima e mínima do armário, localizado em cada seção. Cada membro de  $\pi$  representa uma seção de serviço. Inicialmente, não sabemos quantas seções devem existir.

Desejamos uma partição  $\pi$  de  $A$  tal que o custo total do projeto seja minimizado. Este custo é calculado como a soma dos custos de cada seção de serviço existente em  $\pi$ .

O custo de uma seção de serviço é medido pelo custo do armário de distribuição, mais uma estimativa do custo para instalar os cabos que vão do armário de distribuição até os pontos por ele atendidos. Definimos o custo de uma seção de serviço  $s$  como:

$$w_s = c_a + f \sum_{i \in s} [d_i \times \text{dist}(p_a, p_i)], \quad (3.1)$$

onde  $c_a$  é o custo fixo de um armário de distribuição,  $p_i$  e  $d_i$  são a posição e a demanda do ponto  $i$ , respectivamente,  $p_a$  é a posição do armário de distribuição em  $s$ ,  $\text{dist}$  é a distância euclideana, e  $f = \sigma\beta$  é uma constante, onde  $\sigma$  representa o custo médio por unidade de comprimento do cabo, e  $\beta > 1$  é um fator de correção que é aplicado na distância pois, na realidade, o cabeamento não segue o caminho mínimo. As constantes  $\sigma$  e  $\beta$  foram obtidas fazendo a média sobre redes existentes em áreas de projeto similares.

Assim o custo de uma solução  $\pi$  é dado por:

$$w(\pi) = \sum_{s \in \pi} w_s. \quad (3.2)$$

O objetivo é encontrar uma solução  $\pi$  que minimiza  $w(\pi)$ , e que satisfaça as restrições de capacidade de um armário. Assim, a demanda total de uma seção de serviço está sujeita a valores máximos e mínimos. Cada seção de serviço  $s \in \pi$  tem que satisfazer  $k \times C \leq D_s \leq K \times C$ , onde  $C$  é a capacidade de um armário,  $K$  é o fator de carga máxima da capacidade,  $k$  é o fator de carga mínima e  $D_s$  é a demanda total em  $s$  (i.e.  $D_s = \sum_{i \in s} d_i$ ).

### 3.3 Posicionamento de Caixas Terminais

Também chamamos este problema de “Posicionamento de Caixas Terminais Considerando a Rede Existente”, pois consideramos as demandas que já estão instaladas na rede atual. São dados:

- um conjunto  $I$  de triplas  $(x, y, d)$ , chamados pontos de demanda (ou *clientes*), onde  $x$  e  $y$  são as coordenadas do ponto e  $d$  representa a demanda em termos de linhas telefônicas;
- um conjunto de postes  $J$ , onde cada poste possui um identificador;
- um conjunto  $E = I \times J$  de ligações entre pontos de demanda e postes e
- um conjunto  $T$  de caixas terminais existentes, instaladas em postes. Cada caixa terminal é formada por um identificador do poste no qual ela está instalada, um tipo indicando a sua capacidade e um subconjunto de  $I$ , indicando os pontos que ela atende.



Uma resolução deste problema consiste em instalar novas caixas em postes para atender os pontos de demanda do conjunto  $I$  que ainda não estão instalados, bem como efetuar o alívio de caixas existentes, ou seja, se existirem caixas que estão com a capacidade saturada (atendem mais clientes do que o permitido), é necessário transferir clientes desta caixa para outras.

A instalação de novas caixas e transferências de clientes deve ser guiada pela minimização do custo de tais operações. O objetivo é realizar o planejamento e projeto de uma nova rede, posicionando caixas terminais e estabelecendo conexões a clientes de modo a minimizar o custo total do projeto.

Este problema possui algumas restrições que estão listadas a seguir:

1. Cada ponto de demanda deve estar conectado à alguma caixa terminal.
2. Toda a demanda de um ponto deve ser atendida por uma mesma caixa.
3. Existe uma distância máxima entre a caixa e os pontos que ela atende.
4. Pontos com demanda maior que um valor  $d_p$ , são considerados pontos de demanda predial (prédios), e estas demandas serão atendidas por um tipo de caixa diferente, chamada predial.
5. Uma caixa de capacidade  $s$ , pode atender de  $k \times s$  até  $K \times s$ , onde  $0 < k < K \leq 1$ . As constantes  $k$  e  $K$  são fatores de carga mínima e máxima, respectivamente.
6. Num poste podem ser alocadas no máximo duas caixas terminais.
7. Postes que possuem equipamentos indutores instalados, como transformadores, não podem receber caixas terminais.
8. Caixas terminais que atendem regiões próximas devem ter, de preferência, demandas balanceadas.

### 3.3.1 Formulação Matemática

Considere que são dados os seguintes itens:

1.  $I$ , um conjunto de pontos de demanda que devem ser atendidos por caixas terminais.
2.  $J$ , um conjunto de postes que podem acomodar no máximo 2 caixas terminais.
3.  $L$ , um conjunto de tipos de caixas terminais que podem ser instaladas em um poste.
4.  $E \subseteq I \times J$ , um conjunto de arestas (possíveis ligações) entre clientes e postes.
5.  $F = J \times L$ , um conjunto de possíveis caixas terminais em postes.

6.  $c_{ijl}$  o custo de instalar o ponto de demanda  $i \in I$  na caixa terminal  $l \in L$  do poste  $j \in J$ .
7.  $f_{jl}$  o custo de instalar a caixa terminal  $l \in L$  no poste  $j \in J$ .
8.  $d_i$  a demanda do cliente  $i \in I$ .
9.  $s_l$  a capacidade da caixa terminal  $l \in L$ .

O objetivo do problema é minimizar a função de custo do projeto. O custo do projeto engloba: o custo de instalação de novas caixas, custo de remoção de caixas já instaladas, custo de ligação de um cliente a uma caixa e o custo de transferência de clientes.

Uma formulação para este problema é:

$$\text{Minimizar} \quad \sum_{((i,j),l) \in E \times L} c_{ijl} x_{ijl} + \sum_{(j,l) \in F} f_{jl} y_{jl} \quad (3.3)$$

$$\text{sujeito a} \quad \sum_{(j,l) \in F \wedge (i,j) \in E} x_{ijl} = 1 \quad \forall i \in I \quad (3.4)$$

$$\sum_{l \in L} y_{jl} \leq 2 \quad \forall j \in J \quad (3.5)$$

$$k s_l y_{jl} \leq \sum_{i \in I \wedge (i,j) \in E} d_i x_{ijl} \leq K s_l y_{jl} \quad \forall (j,l) \in F \quad (3.6)$$

$$x_{ijl} \leq y_{jl} \quad \forall ((i,j),l) \in E \times L \quad (3.7)$$

$$x_{ijl} \in \{0, 1\} \quad \forall ((i,j),l) \in E \times L$$

$$y_{jl} \in \{0, 1\} \quad \forall (j,l) \in F$$

Onde:

$$x_{ijl} = \begin{cases} 1 & \text{se o cliente } i \text{ está instalado} \\ & \text{na caixa } l \text{ do poste } j. \\ 0 & \text{c.c.} \end{cases} \quad y_{jl} = \begin{cases} 1 & \text{se a caixa terminal } l \text{ está} \\ & \text{instalada no poste } j. \\ 0 & \text{c.c.} \end{cases}$$

A restrição (3.4) garante que todo ponto de demanda será atendido por uma caixa. A restrição (3.5) limita a quantidade de caixas por poste, garantido que teremos no máximo duas caixas por poste. A restrição (3.6) impede que uma caixa atenda mais pontos de demanda do que é permitido pela sua capacidade. Por fim, a restrição (3.7) implica que se um ponto está sendo atendido por uma caixa, esta deve estar instalada.

O custo  $c_{ijl}$  deve ser calculado de forma a refletir os custos de instalação de novos assinantes e/ou transferência de assinantes já instalados (antigos), e o custo  $f_{jl}$  deve refletir o custo de instalação de novas caixas.

Assim temos:

$$c_{ijl} = \begin{cases} 0 & \text{se } i \text{ é antigo e estava na caixa } l \text{ do poste } j \\ inst_{ijl} + rem_{itm} & \text{se } i \text{ é antigo e estava na caixa } m \text{ do poste } t \\ inst_{ijl} + rem_{itm} + crem_{mt} & \text{se } i \text{ é antigo e estava na caixa } m \text{ do poste } t \\ & \text{e a caixa } m \text{ ficou vazia} \\ inst_{ijl} & \text{se } i \text{ é um ponto novo} \end{cases}$$

onde,  $inst_{ijl} = d_i \times dist(i, j) \times c_d$ , sendo  $d_i$  a demanda do ponto  $i$ ,  $dist(i, j)$  a distância entre o ponto e o poste aonde está a caixa e  $c_d$  o custo por unidade de medida do fio que liga o ponto à caixa;  $rem_{ijl} = c_r \times d_i$ , com  $c_r$  igual ao custo de remoção de um assinante e  $crem_{lt}$  é o custo de retirada da caixa  $l$  do poste  $t$ .

$$f_{jl} = \begin{cases} 0 & \text{se } l \text{ já estava instalada no poste } j \\ cinst_{jl} & \text{se } l \text{ é uma caixa nova} \end{cases}$$

onde  $cinst_{jl}$  é o custo de instalação, que depende da capacidade da caixa.

# Capítulo 4

## AG para Definição de Seções de Serviço

### 4.1 Introdução

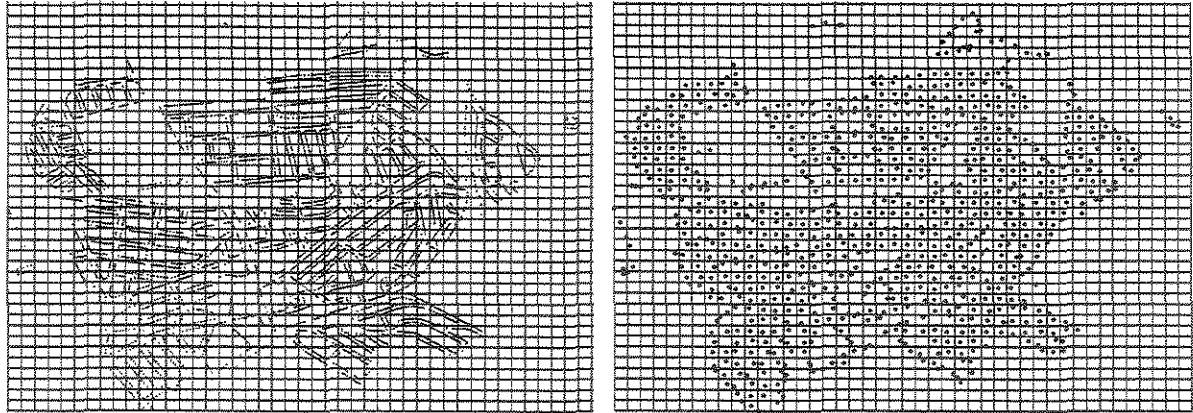
Neste capítulo apresentamos as idéias implementadas para modelar o problema de Definição de Seção de Serviço descrito no capítulo 3.2, chamamos este algoritmo de AGSS. Apresentamos também, os resultados obtidos pelo AGSS, quando aplicado em instâncias reais.

### 4.2 Modelagem

#### 4.2.1 Representação

Quando usamos AGs para resolver um problema de otimização, o primeiro passo, e o mais importante, é definir a forma de representação de uma solução deste problema como um cromossomo, o qual possa ser submetido facilmente aos operadores de mutação e/ou cruzamento. Para o problema de definição de seção de serviços, a representação usual de vetor não é adequada, pois não sabemos inicialmente quantas seções de serviço existem e nem a posição de seus armários de distribuição. Também temos que assegurar uma geometria razoável para as seções, o que é uma medida subjetiva, para ser julgada por engenheiros de projeto. Assim, propomos uma representação diferente.

Circunscrevemos a área de projeto por um retângulo e o dividimos em células iguais de tamanho  $\ell \times \ell$ . Atribuimos um único ponto de demanda para cada célula, com demanda igual a soma das demandas de todos os pontos que estão no seu interior. Sua posição dentro da célula é obtida pela média ponderada das posições dos pontos que estão no interior da célula. A figura 4.1 ilustra um exemplo desta operação. A malha resultante é chamada de *Malha de Vizinhança*. Ela armazena informações geométricas, topológicas e de vizinhança sobre todos os pontos de demanda. Agora o problema se reduz em particionar a malha de vizinhança em seções de serviços.



(a) Área de Projeto com Pontos de Demanda Reais

(b) Área de Projeto com Pontos de Demanda Concentrada

Figura 4.1: Malha de Vizinhança.

Um cromossomo é formado por uma malha de vizinhança  $M$  e um vetor  $S$  de seções (veja a figura 4.2). Cada célula de  $M$  aponta para uma posição de  $S$ , indicando a seção que ela pertence. Uma seção é descrita por uma lista de suas células, um par de inteiros representando

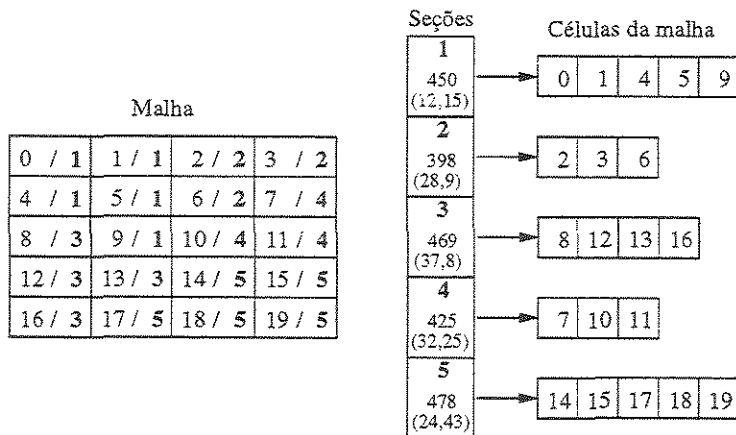


Figura 4.2: Um cromossomo.

a posição do seu armário de distribuição e um número indicando a sua demanda total. A malha  $M$  e o vetor  $S$  contêm informações redundantes mas, por questões de eficiência, mantemos os dois. Com a malha sabemos facilmente em qual seção um ponto está, e com o vetor de seções sabemos quais os pontos que estão numa mesma seção.

### 4.2.2 População Inicial

Para obter o  $i$ -ésimo indivíduo na população de tamanho  $n$ , começamos rotacionando a área de projeto por um ângulo igual a  $\frac{\pi i}{2n}$ , no sentido horário. Depois a área de projeto é particionada recursivamente por retas horizontais e verticais, até que cada subárea retangular resultante possua uma demanda total de no máximo  $K \times C$ . A cada iteração particionamos o maior lado de cada retângulo cuja demanda está acima de  $K \times C$ . O algoritmo que particiona uma área de projeto, já rotacionada, é descrito a seguir:

```

DIVIDE( $A, l, h$ )
1  if  $W[A] \leq K \times C$ 
2    then return  $\{A\}$ 
3  if  $H_x[A] > H_y[A]$ 
4    then  $orientacao \leftarrow Y$ 
5    else  $orientacao \leftarrow X$ 
6   $me \leftarrow \text{MEDIANAPONDERADA}(A, l, h, orientacao)$ 
7   $i \leftarrow \text{PARTICIONA}(A, l, h, me, orientacao)$ 
8  return  $\{\text{DIVIDE}(A, l, i), \text{DIVIDE}(A, i + 1, h)\}$ 

```

O algoritmo recebe como entrada uma área de projeto  $A$ , formada por um vetor  $P$  de pontos de demanda.  $l$  e  $h$  são índices inferior e superior do vetor, respectivamente.

Temos ainda que,  $W[A]$  representa a demanda total de  $A$  e  $H_x[A]$  e  $H_y[A]$  representam a dimensão da região de pontos paralelamente ao eixo das coordenadas e abscissas, respectivamente.

$$\begin{aligned}
 W[A] &= \sum_{P_i \in A} d_i, \\
 H_x[A] &= |\max_{P_i \in A}(x_i) - \min_{P_i \in A}(x_i)| \text{ e} \\
 H_y[A] &= |\max_{P_i \in A}(y_i) - \min_{P_i \in A}(y_i)|,
 \end{aligned}$$

A função MEDIANAPONDERADA acha o elemento  $me$  em  $A$  satisfazendo:

$$\sum_{(P_i \in A) \wedge (x_i < me)} d_i \leq \frac{W[A]}{2} \quad \text{e} \quad \sum_{(P_i \in A) \wedge (x_i > me)} d_i \leq \frac{W[A]}{2},$$

se o parâmetro  $orientacao = X$ , ou:

$$\sum_{(P_i \in A) \wedge (y_i < me)} d_i \leq \frac{W[A]}{2} \quad \text{e} \quad \sum_{(P_i \in A) \wedge (y_i > me)} d_i \leq \frac{W[A]}{2},$$

se  $orientacao = Y$ .

A função PARTICIONA, divide o vetor  $A$  em duas partes:

- $A[l..i]$  com toda tripla  $P_i \in A$  e  $x_i \leq me$ .

- $A[i + 1..h]$  com as triplas  $P_i \in A$  e  $x_i > me$ .

se  $orientacao = X$ , ou:

- $A[l..i]$  com toda tripla  $P_i \in A$  e  $y_i \leq me$ .
- $A[i + 1..h]$  com as triplas  $P_i \in A$  e  $y_i > me$ .

se  $orientacao = Y$ . O valor de retorno é o índice  $i$ .

O uso combinado das funções MEDIANAPONDERADA e PARTICIONA, transforma o vetor  $A$  em dois subvetores  $A_1 = A[l..i]$  e  $A_2 = A[i+1..h]$  de tal forma que  $W[A_1] \approx W[A_2] \approx \frac{W[A]}{2}$ .

Podemos ver com isso, que a idéia do algoritmo é sempre dividir a região em duas outras sub-regiões de demanda aproximadamente igual à metade da demanda original. E a cada uma destas sub-regiões o algoritmo é aplicado recursivamente.

### 4.2.3 Cruzamento

Tendo a representação, temos que definir um operador de cruzamento apropriado para ela. Como o cruzamento consiste na troca de informações entre os pais, precisamos criar uma maneira de dividir um cromossomo em duas partes, para serem trocadas. Chamamos de *corte* uma operação que divide a malha de vizinhança em duas partes. Criamos quatro tipos de corte, os quais chamamos de horizontal, vertical, sudoeste e noroeste, conforme a figura 4.3.

Dada a malha  $M$ , um corte é criado da seguinte forma:

1. Selecionamos uma célula da malha para iniciar o corte. A seleção desta célula depende do tipo do corte.
  - (a) No corte horizontal a seleção da célula inicial é feita sobre as células da primeira coluna. Cada célula tem uma probabilidade de ser selecionada, e esta probabilidade segue a distribuição binomial, onde as células que estão no meio da primeira coluna têm mais chance de serem selecionadas.
  - (b) No corte vertical a seleção da célula inicial é feita sobre as células da primeira linha. Cada célula tem uma probabilidade de ser selecionada, e esta probabilidade segue a distribuição binomial, onde as células que estão no meio da primeira linha têm mais chance de serem selecionadas.
  - (c) No corte sudoeste a seleção da célula inicial é feita sobre as células da primeira linha e da primeira coluna. Cada célula tem uma probabilidade de ser selecionada, e esta probabilidade segue a distribuição binomial, onde as células que estão no canto superior esquerdo têm mais chance de serem selecionadas.

- (d) No corte noroeste a seleção da célula inicial é feita sobre as células da primeira coluna e da última linha. Cada célula tem uma probabilidade de ser selecionada, e esta probabilidade segue a distribuição binomial, onde as células que estão no canto inferior esquerdo têm mais chance de serem selecionadas.
2. O corte é aumentado selecionando aleatoriamente uma dentre três células vizinhas da última célula presente no corte. As posições das células vizinhas depende do tipo de corte que está sendo feito.
  3. Repetimos este ciclo até que uma fronteira da malha é alcançada.

Quando selecionamos a próxima célula do corte, atribuímos probabilidades para cada célula vizinha. Estas probabilidades dependem da posição da célula candidata e se ela está na mesma seção de serviço do que a célula presente na aresta do corte. Assim, de acordo com a orientação do corte que está sendo feito, dividimos a malha em duas partes (lado 1 e lado 2). Na figura 4.3 mostramos como é o comportamento do corte, quais são as posições vizinhas de acordo com sua orientação, e quais posições ficam de um mesmo lado da malha, sendo que as células do corte sempre ficam do lado 1.

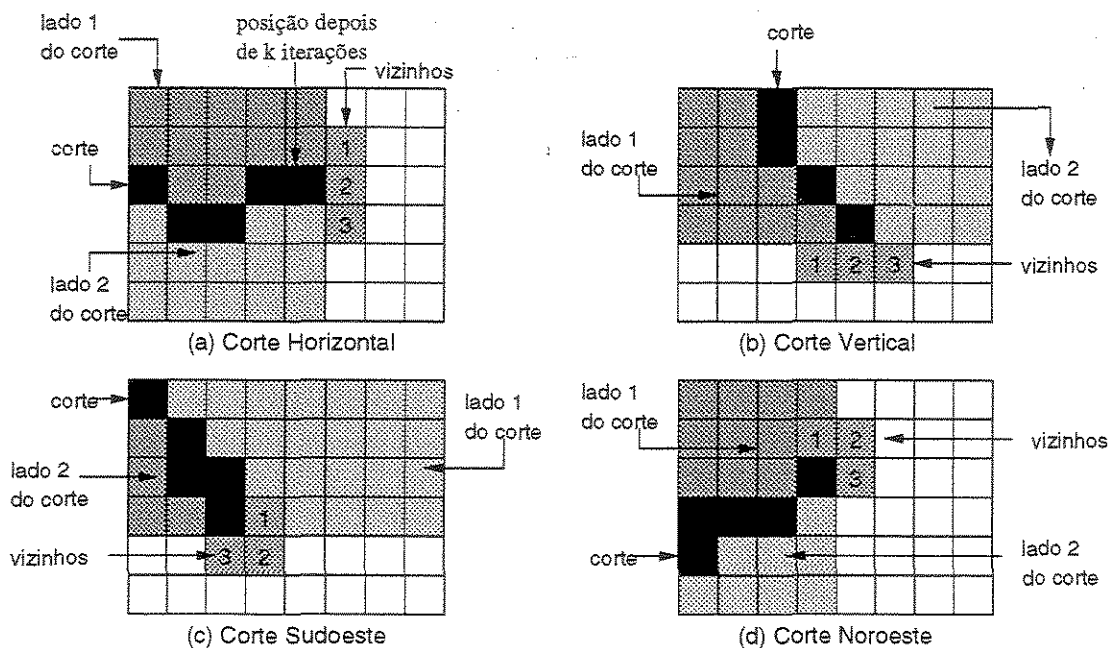


Figura 4.3: Os quatro tipos de corte na malha.

Para calcular a probabilidade de cada célula ser selecionada para entrar no corte, foram utilizadas duas propriedades como segue: (1) sua posição e (2) sua seção com relação à célula escolhida anteriormente pelo corte.



Para cada propriedade temos três casos, e a probabilidade de uma célula é calculada pela multiplicação dos valores referentes às suas propriedades, como especificado na tabela 4.1. Nesta tabela,  $p_1$ ,  $p_2$  e  $p_3$  representam as probabilidades referentes às células vizinhas da última célula do corte, 1, 2 ou 3, respectivamente. Os valores  $p_i$  e  $p_d$  indicam o valor da probabilidade para os casos da seção ser igual ou diferente da célula escolhida anteriormente, e se a célula não possui demanda o valor de sua probabilidade é dado por  $p_v$ . As probabilidades  $p_1$ ,  $p_2$ ,  $p_3$ ,  $p_i$ ,  $p_d$ ,  $p_v$  são parâmetros e seus valores podem ser alterados.

Seção\Posição	1	2	3
<b>igual</b>	$p_1 \times p_i$	$p_2 \times p_i$	$p_3 \times p_i$
<b>diferente</b>	$p_1 \times p_d$	$p_2 \times p_d$	$p_3 \times p_d$
<b>vazia</b>	$p_1 \times p_v$	$p_2 \times p_v$	$p_3 \times p_v$

Tabela 4.1: Probabilidades para seleção da célula que entra no corte.

O operador de cruzamento funciona da seguinte forma:

1. Selecionamos dois indivíduos, pelo método da roleta [16], para serem os pais do cruzamento.
2. Realizamos  $k$  cortes nos pais, escolhidos aleatoriamente, e selecionamos os dois cortes que geraram os filhos com menor custo estimado.
3. Trocamos as partes de cada pai resultantes do corte, criando os dois filhos.

A figura 4.4 ilustra uma operação de cruzamento, onde regiões com diferentes números representam seções distintas (estes números são os identificadores da seção, cada seção possui um identificador único, mesmo seções de diferentes cromossomos) e células vazias têm demanda zero. As figuras 4.4(a) e 4.4(b) representam os cromossomos pais. A figura 4.4(c) ilustra um corte noroeste feito no pai; o mesmo corte é feito na mãe. A figura 4.4(d) ilustra um dos filhos criados pelo operador de cruzamento quando trocamos as partes dos pais.

Depois do cruzamento, algumas seções podem ficar desconexas (como as seções 13 e 10 na figura 4.4(d)). Cada parte conexa é transformada em uma seção diferente, com demanda igual a soma das demandas de suas células. Depois disso os filhos resultantes passam por um operador de reparo onde as seções com demanda muito baixa são incorporadas nas seções vizinhas. Um

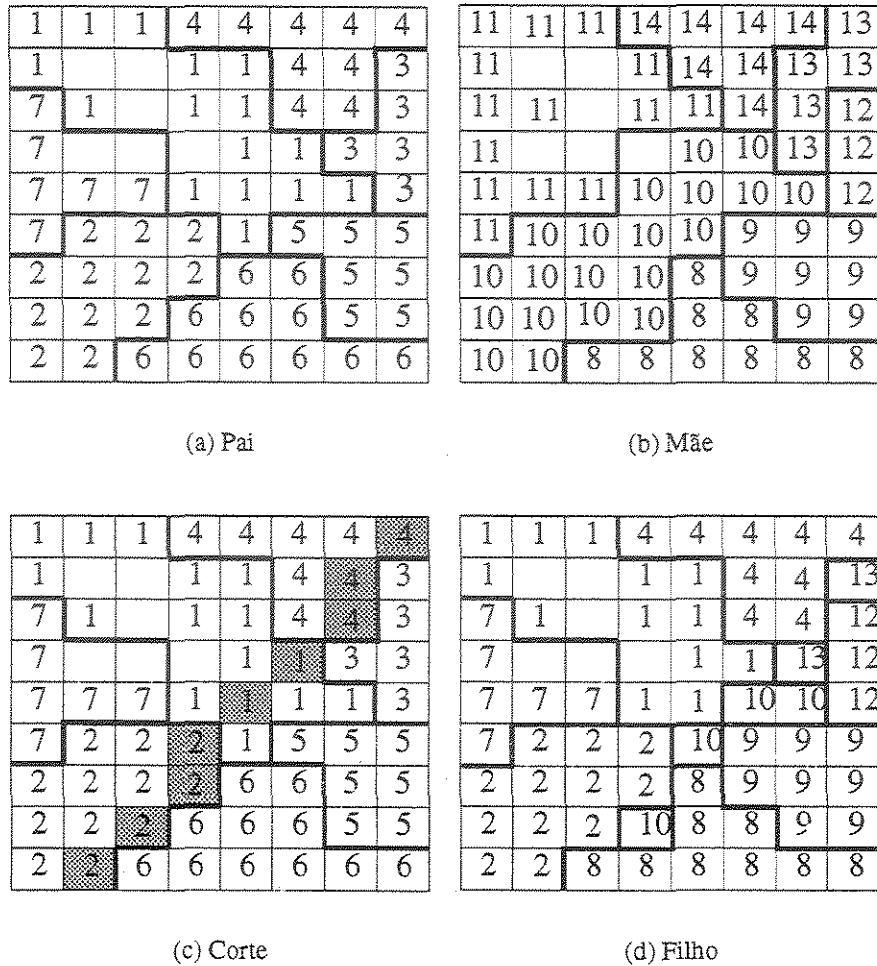


Figura 4.4: Cruzamento.

pseudo-código para esta operação é descrita a seguir:

CROSSREPARO( $\pi$ )

- 1  $S \leftarrow \{s : s \text{ é uma seção de } \pi\}$
- 2 **for** cada  $s \in S$
- 3   **do if**  $D_s < k \times C$
- 4     **then repeat**
- 5       Distribua as células que estão na borda de  $s$  para as seções
- 6       vizinhas, sem exceder a capacidade máxima permitida.
- 7     **until** ( $s \neq \emptyset$ ) ou (não existir tal seção vizinha)

Porém, uma seção permanecerá com demanda baixa, se ela não possuir seções vizinhas suficientes e capazes de incorporar todos os seus pontos de demanda.

1	1	1	4	4	4	4	4
1			1	1	4	4	4
7	1		1	1	4	4	12
7				1	1	1	12
7	7	7	1	1	1	9	12
7	2	2	2	2	9	9	9
2	2	2	2	8	9	9	9
2	2	2	2	8	8	9	9
2	2	8	8	8	8	8	8

Figura 4.5: Filho resultante depois da operação de reparo.

A figura 4.5 ilustra o resultado do algoritmo de reparo aplicado no filho ilustrado na figura 4.4(d).

#### 4.2.4 Mutação

Desenvolvemos dois operadores de mutação, chamados *MutacaoSecao* e *MutacaoPonto*, ambos possuem um carácter de busca local. Começamos selecionando aleatoriamente um certo número de cromossomos para sofrerem mutação. Cada cromossomo selecionado irá sofrer um dos dois tipos de mutação. Para escolher qual operador de mutação usar, verificamos o número de seções do cromossomo selecionado. Se este número é maior do que o mínimo possível, calculado por  $\lceil (\sum_{s \in \pi} D_s) / (K \times C) \rceil$ , usamos o operador *MutacaoSecao*, caso contrário usamos *MutacaoPonto*.

##### *MutacaoSecao*

Este operador tenta diminuir o número de seções do cromossomo  $\pi$  selecionado e também tenta balancear a demanda entre elas. Ele necessita de três parâmetros  $d_{min}$ ,  $d_{med}$  e  $d_{max}$ , com  $d_{min} \leq d_{med} \leq d_{max}$ , todos proporcionais a  $C$ .

Assim, selecionamos aleatoriamente 40% das seções de  $\pi$ , cada seção  $s$  tendo probabilidade de ser selecionada proporcional a  $(D_s - K \times C)^2$ . Para cada seção  $s$  selecionada, se  $D_s < d_{min}$ , fazemos  $s$  perder todos os seus pontos para as seções vizinhas, e se  $D_s > d_{max}$  fazemos  $s$  perder pontos até sua demanda ficar abaixo de  $K \times C$ . Quando  $d_{min} < D_s < d_{med}$ ,  $s$  receberá pontos das seções vizinhas. Estas operações estão descritas abaixo.

PERDEPONTOS( $s, D$ )

- 1 **while**  $D_s > D$
- 2 **do for** cada  $p$  na fronteira de  $s$
- 3 **do** Coloque  $p$  na seção vizinha que possui menos demanda

RECEBEPONTOS( $s, D$ )

- 1 **while**  $D_s < D$
- 2 **do for** cada  $p$  na fronteira de  $s$
- 3 **do** Coloque em  $s$  os pontos vizinhos de  $p$  que estão em seções diferentes de  $s$

onde, se  $D_s \leq d_{min}$  então  $D = D_s$ , se  $D_s \geq d_{max}$  então  $D = d_{max} - C \times K$ , se  $D_s \leq d_{med}$  então  $D = C \times K - D_s$ .

Este operador de mutação ajusta o número de seções de serviço na solução.

### *MutacaoPonto*

Este operador visa diminuir o custo de cabeamento da solução, para isso ele troca um ponto de uma seção  $s_1$  para uma seção vizinha  $s_2$  se o custo de instalá-lo na seção  $s_2$  é menor do que o seu custo de instalação em  $s_1$ .

Um pseudo-código para esta operação está descrito a seguir, onde  $PosArmario(s)$  dá a posição do armário em  $s$  e  $m$  corresponde a 40% do número de seções do cromossomo  $\pi$ .

MUTACAO PONTO( $\pi, m$ )

- 1  $S \leftarrow \{s : s \text{ é uma seção de } \pi\}$
- 2 **for**  $i \leftarrow 0$  até  $m - 1$
- 3 **do** Selecione  $s \in S$  com probabilidade  $w_s / \sum_{s \in S} w_s$
- 4 **for** cada  $p$  na fronteira de  $s$
- 5 **do if**  $\exists s_v \in S : dist(PosArmario(s_v), p) < dist(PosArmario(s), p)$
- 6 **then** Remova  $p$  de  $s$
- 7 **do** Coloque  $p$  em  $s_v$

### 4.2.5 Função de Avaliação e Penalidade

A função de avaliação precisa refletir quão boa é uma solução para o problema. Como o objetivo deste problema é minimizar o custo da solução, a aptidão deve ser inversamente proporcional a ele. Porém, precisamos analisar como tratar os indivíduos que não respeitam as restrições de capacidades do armário.

Muitas metodologias foram desenvolvidas para tratar restrições em AGs, porém muitas delas são dependentes do problema. Dentre estas metodologias, destacamos aquelas que: (1) tentam preservar a factibilidade das soluções usando operadores genéticos especiais, (2) tentam

preservar a factibilidade usando um operador de reparo ou (3) avaliam soluções usando uma função de avaliação combinada, ou seja, dando uma penalidade aos indivíduos infactíveis.

Em [17] existe uma comparação entre as duas últimas metodologias. O processo de reparar indivíduos infactíveis está relacionado com uma combinação de aprendizado e evolução. Aprendizado (em geral como uma busca local e em particular como uma busca local pela solução factível mais próxima) e evolução interagem entre si. O valor de aptidão (ou avaliação) da melhora local é transferido para o indivíduo. O ponto fraco destes métodos está na sua dependência do problema, pois diferentes procedimentos de reparação devem ser projetados para cada problema particular.

Já para aplicar penalidades aos indivíduos inválidos, [17] recomenda que a aproximação mais comum para manipular soluções infactíveis é prover uma penalidade para sua infactibilidade somando ou diminuindo este valor na função de avaliação. E este valor pode representar uma penalidade para um indivíduo infactível ou um custo de reparar tal indivíduo. A intuição sugere que a penalidade deve ser mantida tão pequena quanto possível, bem acima do ponto onde as soluções infactíveis se tornam factíveis. Mas, geralmente é muito difícil de implementar esta regra efetivamente.

As restrições existentes para este problema são de capacidade mínima e máxima de um armário. Uma solução que possui seções com demanda um pouco maior (menor) do que a capacidade máxima (mínima) não necessariamente deve ser eliminada, assim optamos por aplicar uma penalidade aos indivíduos que violam estas restrições.

Quando aplicamos os operadores de cruzamento e mutação, algumas seções podem ainda violar as restrições de capacidade. Assim, a aptidão de um cromossomo é calculada somando uma penalidade para violações de capacidade das suas seções. Esta penalidade é calculada por  $\tau(\pi) = \sum_{s \in \pi} \tau_s$ ,  $\pi$  é uma solução, e  $\tau_s$  é a penalidade associada à seção  $s$ , definida como:

$$\tau_s = \begin{cases} \gamma \times (k \times C - D_s) & \text{se } D_s < k \times C \\ \gamma \times (D_s - K \times C) & \text{se } D_s > K \times C \\ 0 & \text{se } k \times C \leq D_s \leq K \times C, \end{cases}$$

onde  $\gamma > 0$  é um parâmetro.

Assim, a aptidão de uma solução  $\pi$ , é dada por

$$\varphi(\pi) = [N - (w(\pi) + \tau(\pi))] / F,$$

onde  $N$  é um número suficientemente grande para tornar  $\varphi$  positiva para todos os indivíduos da população,  $w$  é o custo da solução (definido em 3.2),  $\tau$  é a penalidade associada a ele, e  $F$  é um fator de normalização uniforme dado pela soma de todas as parcelas  $[N - (w(\pi) + \tau(\pi))]$  sobre a população inteira.

### 4.2.6 Seleção

Para selecionar os pais que irão sofrer o operador de cruzamento, usamos o método de seleção da roleta (2.2.3), que é baseado na aptidão de cada indivíduo.

Para visualizar este método, considere uma “roleta” dividida em  $n$  regiões (tamanho da população), onde a área de cada região é proporcional à aptidão do indivíduo. Veja a figura 4.6. Após uma rodada, a posição do cursor indica o indivíduo selecionado. Evidentemente, os indivíduos cujas regiões possuem maior área terão maior probabilidade de serem selecionados e, conseqüentemente, a seleção de indivíduos pode conter várias cópias de um mesmo indivíduo enquanto outros podem desaparecer.

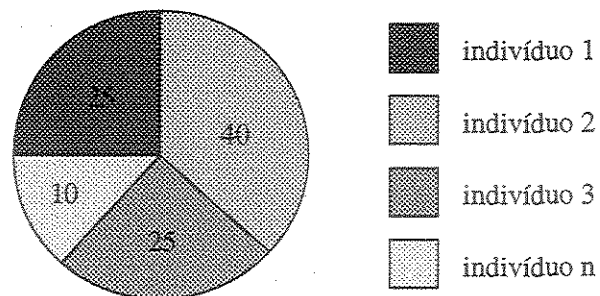


Figura 4.6: Seleção por roleta.

### 4.2.7 Outras Características

- A mutação só é aplicada nos indivíduos que sofreram cruzamento.
- A seleção dos indivíduos que sofrem mutação é feita de forma aleatória.
- Usamos elitismo, ou seja, os melhores indivíduos sempre sobrevivem nas gerações seguintes.
- Ao invés de termos probabilidades para aplicar as operações de cruzamento e mutação, temos taxas de cruzamento e mutação, isto é, uma porcentagem da população sofre cruzamento e destes uma porcentagem sofre mutação.
- O número de melhores indivíduos que sempre sobrevivem nas próximas gerações é dado por uma taxa de elite.

## 4.3 Resultados

### 4.3.1 Ambiente Computacional

O AG foi implementado em C++ e os testes foram realizados num processador Intel Pentium IV com um clock de 2 GHz e 1GB de memória RAM. Todas as instâncias de testes são dados reais, que representam áreas urbanas da cidade de São Paulo. A tabela 4.2 apresenta algumas características numéricas das instâncias utilizadas, como o número de pontos e a demanda total destes.

Inst.	Nome	Num. pontos	Demanda
1	Sul Vila Mariana	6.276	11.616,40
2	Norte Vila Mariana	8.482	11.350,35
3	Leste Vila Mariana	19.007	34.432,75
4	Bairro Liberdade	42.453	90.679,80

Tabela 4.2: Instâncias.

### 4.3.2 Parâmetros

Realizamos uma variedade de testes para ajustar os parâmetros recebidos pelo algoritmo genético. Os testes apresentados nesta seção foram realizados para a região Sul da Vila Mariana, instância 1. Existem dois conjuntos de parâmetros, os referentes ao projeto de seção de serviço e os referentes ao algoritmo genético, listados na tabela 4.3 e na tabela 4.4, respectivamente.

Dados do Projeto		
Símbolo	Descrição	Valor
$\sigma$	custo médio dos cabos por unidade de distância	R\$ 0,0323
$\beta$	fator de correção da distância euclideana para a real	1,9
$c_a$	custo do armário	R\$ 230.000,00
$C$	capacidade do armário	600
$K$	fator de carga máxima no armário	80%
$k$	fator de carga mínima no armário	30%
$K_U$	demanda localizada mínima para caracterizar seção de serviço local	300

Tabela 4.3: Dados do projeto.

Para medir a qualidade de uma solução estamos levando em consideração o custo total do projeto e a distribuição de demanda entre as seções.

Dados do AG	
Símbolo	Descrição
$lado$	tamanho do lado da malha de vizinhança
$ngen$	números de gerações do AG
$npop$	tamanho da população
$t_c$	taxa de cruzamento
$t_e$	taxa de elite
$t_s$	taxa de sobreviventes ( $t_s = 1 - t_c - t_e$ )
$t_m$	taxa de mutação
$\gamma$	coeficiente da função de penalidade
$p_1$	probabilidade para o vizinho 1 durante o corte
$p_2$	probabilidade para o vizinho 2 durante o corte
$p_3$	probabilidade para o vizinho 3 durante o corte
$p_i$	probabilidade se a unidade vizinha for da mesma seção
$p_d$	probabilidade se a unidade vizinha for de seção diferente
$p_v$	probabilidade se a unidade vizinha for vazia

Tabela 4.4: Dados do algoritmo genético.

O único parâmetro da tabela 4.3 na página anterior que foi alterado durante os testes, foi o  $K_U$  (seção de serviço local). Este valor influencia fortemente no custo do projeto visto que para cada ponto com demanda maior ou igual a  $K_U$  um ou mais armários de distribuição deverão ser instalados para atendê-lo, e o custo do armário é muito elevado. Além disso, também influencia na distribuição de demanda entre as seções, pois removendo estes pontos temos menos demanda para alocar em armários, podendo resultar numa demanda média baixa em relação à carga máxima do armário.

### Plano de Testes

Nesta seção apresentamos os testes realizados e seus resultados. Como o algoritmo genético é uma heurística com escolhas probabilísticas, cada teste foi realizado cinco vezes, e o resultado apresentado é a média das cinco evoluções.

#### *Demanda Localizada Mínima de 240*

A tabela 4.5 na próxima página apresenta os parâmetros dos testes realizados levando em consideração os dados da tabela 4.3 na página anterior com  $K_U = 240$ . Para estes testes temos três pontos que se tornaram seção de serviço local, com demandas iguais a 1.200, 303,6 e 264. Estes pontos não serão considerados durante a evolução do AG, pois para cada um deles será alocado um (ou mais) armário de distribuição. A demanda total do projeto que o AG deve tratar é 9.848,79.



Teste	lado	ngen	npop	$t_e$ %	$t_c$ %	$t_s$ %	$t_m$ %	$p_1$ %	$p_2$ %	$p_3$ %	$p_i$ %	$p_d$ %	$p_v$ %	$\gamma$
1	20	500	50	10	80	10	20	25	25	50	40	10	50	800
2	20	1000	50	40	50	10	20	25	25	50	40	10	50	800
3	20	500	50	10	80	10	1	25	25	50	40	10	50	800
4	20	1000	50	10	80	10	10	25	25	50	40	10	50	800
5	20	1000	50	10	80	10	40	35	35	30	20	10	70	650

Tabela 4.5: Plano de teste para  $K_U = 240$ .

Neste conjunto de testes, variamos as taxas de cruzamento e mutação (a variação da taxa de cruzamento influencia nas taxas de elite e sobrevivência), para sabermos quão sensíveis são estes parâmetros para o algoritmo (testes: 1, 2, 3 e 4). Uma outra abordagem nos testes foi variar as probabilidades utilizadas durante a operação de corte, juntamente com uma diminuição do coeficiente utilizado na função de penalidade (teste 5).

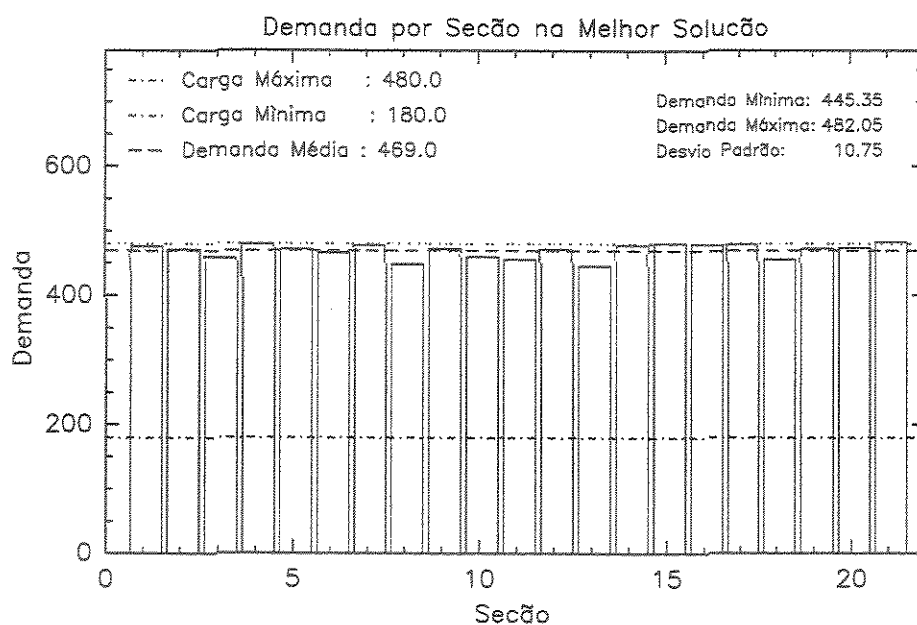
Na tabela 4.6 apresentamos os resultados para os testes com os parâmetros da tabela 4.5. As características que ressaltamos das soluções são o número total de seções do projeto, o custo total, o custo aproximado de cabeamento, a porcentagem do custo de cabo em relação ao custo total, a demanda máxima, mínima e média das seções do projeto e o desvio padrão desta distribuição. O custo total do projeto apresentado na tabela 4.6 não contém o custo relativo às seções de serviço locais listadas acima, que resultam em um custo equivalente a cinco armários de distribuição. Os custos total e de cabos estão listados em milhares de reais (MR\$).

teste	tempo (min)	número de seções	custo total	custo cabos	% cabo	demanda			
						desvio	mínima	máxima	média
1	19	21	4.914,21	84,21	1,71	8,87	444,00	483,30	468,99
2	24	21	4.911,44	81,44	1,66	11,60	439,00	484,75	468,99
3	19	21	4.913,10	83,10	1,69	12,80	441,00	489,55	468,99
4	35	21	4.911,25	81,25	1,65	11,80	436,00	484,68	468,99
5	39	20	4.684,91	84,91	1,81	17,10	470,00	537,95	492,40

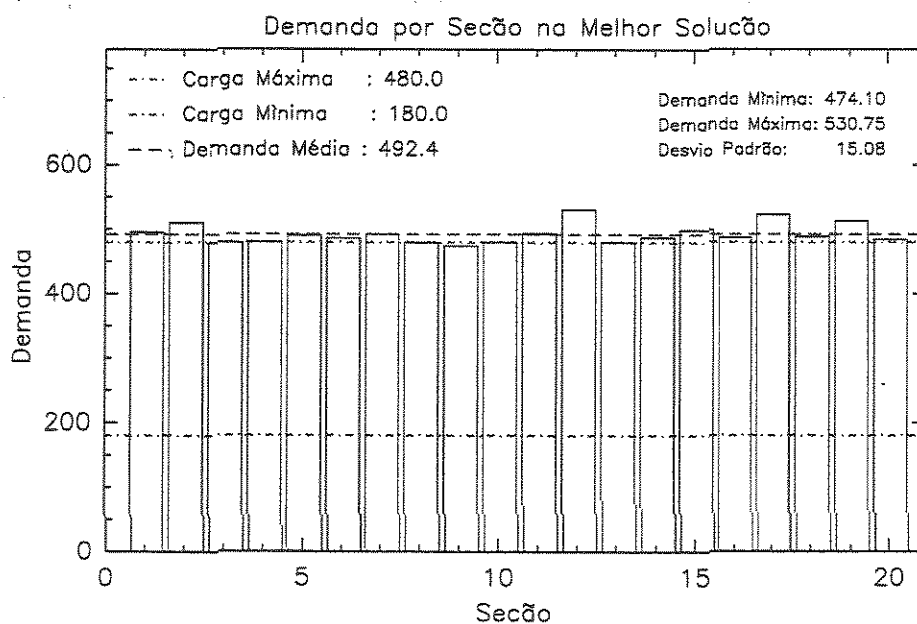
Tabela 4.6: Resultado dos testes para  $K_U = 240$ .

Da tabela 4.6, concluímos que a diferença de custo entre os testes de 1 a 4 é muito pequena, variando no máximo de R\$3.000,00, referente a variação no custo de cabo. Apenas quando alteramos o coeficiente da função de penalidade (teste 5), damos uma maior liberdade para aumentar a carga de um armário, assim o AG conseguiu diminuir uma seção, porém a demanda máxima quase chegou ao limite físico do armário (537,95), e a demanda média ultrapassou a carga máxima do mesmo (492,4). As figuras 4.7(a) e 4.7(b) ilustram a distribuição de demanda nas soluções obtidas pelos testes 2 e 5 respectivamente.

#### *Demanda Localizada Mínima de 300*



(a) Teste 2



(b) Teste 5

Figura 4.7: Distribuição de demanda.

Realizamos alguns testes com demanda localizada mínima igual a 300, concluímos que esta é a melhor configuração para a região testada, visto que a demanda média das seções (481,56) ficou quase igual a carga máxima desejada, o desvio padrão da distribuição foi pequeno e possui uma seção local a menos, resultando num custo total somado com os custos das seções locais menor do que o custo com  $K_U = 240$ .

A tabela 4.7 apresenta os testes realizados levando em consideração os dados da tabela 4.3 na página 30 com  $K_U = 300$ . Para estes casos de teste existem dois pontos considerados seções de serviço local, com demandas 1.200 e 303,6. A demanda total agora é 10.112,8.

Teste	lado	ngen	npop	$t_e$ %	$t_c$ %	$t_s$ %	$t_m$ %	$p_1$ %	$p_2$ %	$p_3$ %	$p_i$ %	$p_d$ %	$p_v$ %	$\gamma$
1	50	10000	50	10	80	10	20	25	25	50	40	10	50	800
2	20	1000	50	10	80	10	0	25	25	50	40	10	50	800
3	20	2000	50	10	80	10	10	25	25	50	40	10	50	800
4	20	1000	50	10	80	10	10	35	35	30	20	10	70	800
5	20	500	50	10	80	10	40	25	25	50	40	10	50	650
6	20	500	50	20	80	0	20	25	25	50	40	10	50	500
7	20	1000	50	40	50	10	20	25	25	50	40	10	50	800

Tabela 4.7: Plano de teste para  $K_U = 300$ .

teste	tempo (min)	número de seções	custo total	custo cabos	% cabo	demanda			
						desvio	mínima	máxima	média
1	90	21	4.911,30	81,30	1,66	4,88	471,80	492,00	481,56
2	34	22	5.141,38	81,39	1,58	27,26	336,95	476,27	459,68
3	60	21	4.912,13	82,13	1,67	3,17	473,70	490,15	481,56
4	41	21	4.914,17	84,28	1,72	3,91	472,43	490,90	481,56
5	37	21	4.912,52	82,52	1,68	7,45	468,00	499,95	481,56
6	37	20	4.682,33	92,33	1,99	27,90	472,00	578,40	505,60
7	29	21	4.912,02	82,02	1,67	4,99	471,20	499,80	481,56

Tabela 4.8: Resultado dos testes para  $K_U = 300$ .

A tabela 4.8 apresenta os resultados para os testes da tabela 4.7. Podemos concluir que:

- Sem taxa de mutação o algoritmo não conseguiu diminuir uma seção, resultando numa média de demanda por seção baixa e um custo mais elevado (teste 2).
- Variando a taxa de mutação entre 10% e 40% o custo permanece quase constante (testes 3 e 5).
- Variando a taxa de elite entre 10% e 40% o custo permanece quase constante (testes 3 e 7).

- A variação das probabilidades de corte influencia no custo, e resultou numa diferença de aproximadamente R\$ 2.000,00 a mais do que a melhor. Cabe salientar que esta diferença é do custo do cabeamento que está sendo aproximado, pois utilizamos a distância euclidiana para estimar o comprimento do cabo (teste 4).
- Diminuindo o coeficiente da penalidade,  $\gamma$ , o algoritmo tem maior liberdade para sobrecarregar um armário, e neste caso conseguiu diminuir uma seção, aumentando a demanda média por seção (teste 6). Nas figuras 4.8(a) e 4.8(b) na próxima página ilustramos a distribuição de demanda para o teste 3 (com distribuição mais uniforme e perto de  $K \times C = 480$ ) e o teste 6 (com demanda média muito alta, mas com uma seção a menos), respectivamente. Portanto este parâmetro parece ser bastante útil. Quando utilizado em regiões onde já se sabe que a estimativa de crescimento é baixa, pode-se colocar um valor baixo, caso contrário o parâmetro deve ter um valor mais alto.

### 4.3.3 Solução

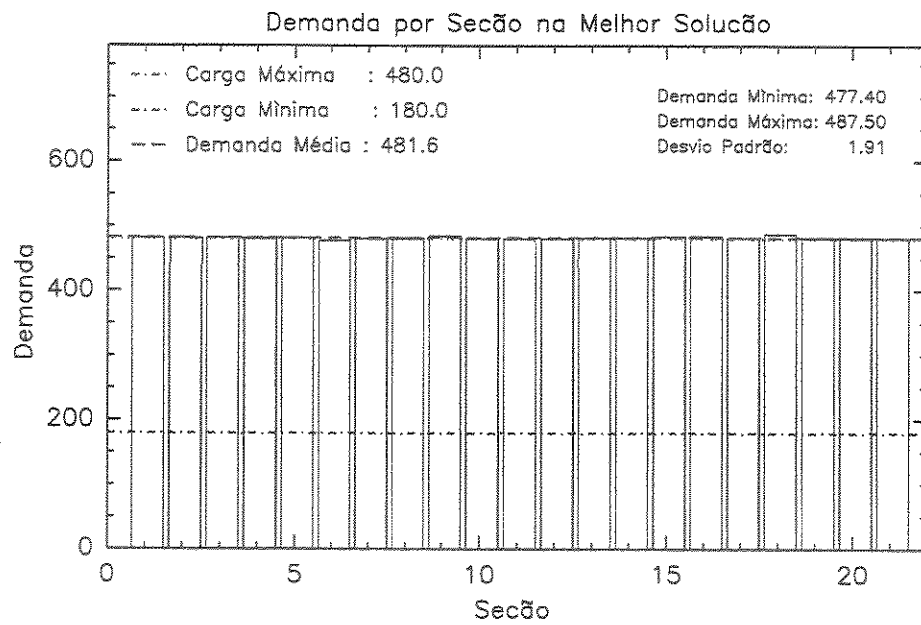
Nesta seção apresentamos os gráficos das soluções usando os parâmetros do teste 7 da tabela 4.7 na página anterior, aplicados às regiões Norte, Leste e Sul da Vila Mariana e ao Bairro da Liberdade.

#### Região Sul da Vila Mariana

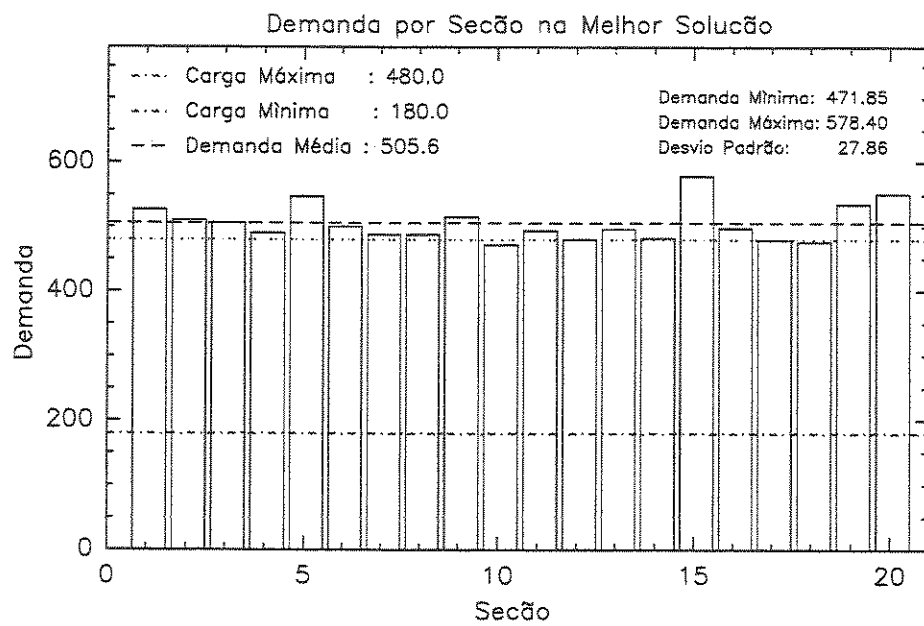
As figuras desta seção retratam os gráficos da solução resultante da aplicação do algoritmo genético para a região Sul da Vila Mariana.

A figura 4.9 na página 37 apresenta a distribuição de demanda entre as seções da solução. A partir desta figura observamos que a solução possui o número mínimo de armários, que é dado pela demanda total dividida pela carga máxima do armário ( $\frac{10.1128}{480} = 21,07$ ), e estes armários estão todos equilibrados, bem perto da carga máxima permitida. A média da distribuição de demanda entre as seções é igual a 481,6, e o desvio padrão desta distribuição é igual a 4,15, ou seja, todas as seções possuem exatamente a demanda desejada que foi passada como parâmetro para o algoritmo ( $C \times K = 480$ ). Não temos nenhuma seção abaixo da carga mínima (180), e nenhuma muito perto da capacidade do armário (600), sendo que a seção que possui a menor demanda atende 472,95 pontos (muito próximo da carga máxima do armário). Resultando em seções uniformes que atendem a demanda projetada, e possuem espaço físico para possíveis alterações e aumento da rede.

A figura 4.10 na página 38 apresenta o custo de cabeamento em cada seção. Observamos que algumas seções possuem um custo muito maior do que outras, devido à extensão da seção. Quanto maior for sua área de atendimento, mais alto será o custo de atender os pontos que a ela pertencem. O número de seções é inversamente proporcional ao custo de cabo, ou seja, quanto



(a) Teste 3



(b) Teste 6

Figura 4.8: Distribuição de demanda com  $K_U = 300$ .

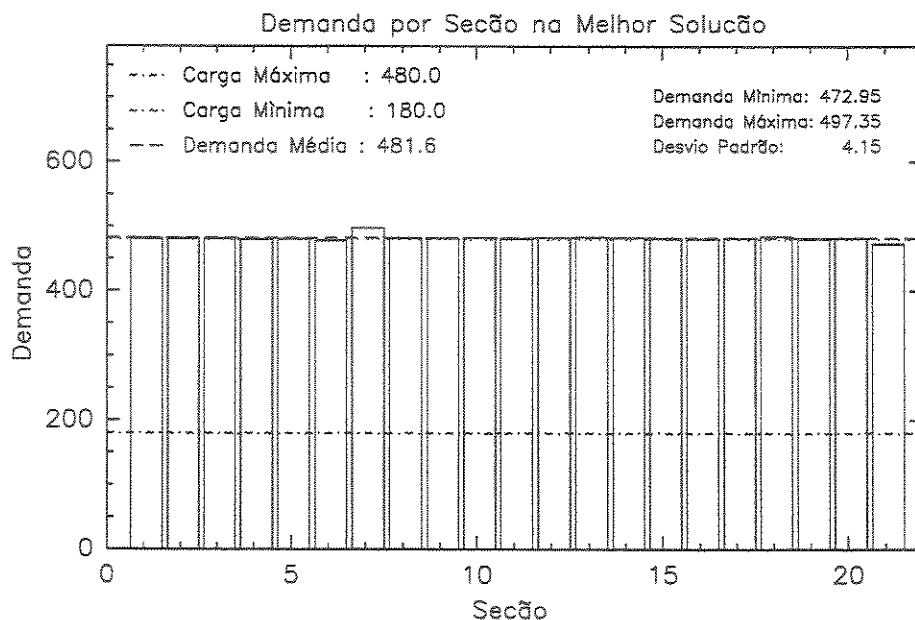


Figura 4.9: Solução para o Sul da Vila Mariana: demanda por seção.

maior o número de seções menor será o custo de cabeamento, podendo até reduzir o custo de cabeamento a zero, se o número de seções for igual ao número de pontos de demanda.

Podemos observar, também, que o custo de cabo é muito inferior ao custo dos armários, portanto o algoritmo tenta diminuir ao máximo o número de seções, para então tentar distribuir os pontos de demanda entre as seções de forma que o custo do cabo diminua. Mais uma vez salientamos que os custos ilustrados na figura 4.10, são custos estimados, calculados pela distância do ponto ao armário que o atende, ponderado pela sua demanda. Ao aplicar esta solução na realidade, estes custos irão certamente modificar, visto que a posição do armário nem sempre poderá ser àquela proposta pelo algoritmo.

A figura 4.11 na página 39 ilustra a evolução do algoritmo genético, ou seja, indica o custo total da melhor solução por iteração. A linha tracejada representa o custo da melhor solução e a linha contínua ilustra o custo mais a penalidade. Os pontos mostram quando o algoritmo conseguiu uma solução melhor. A queda brusca do custo que observamos nas primeiras gerações é devido à diminuição de um armário (cujo custo é muito alto). Observamos também, que após a diminuição do armário, o algoritmo não possui uma melhora significativa, porém, é nesta fase que os pontos de demanda são distribuídos entre as seções, para que o custo do cabo diminua. Nesta figura, apresentamos também o custo total (4.910.908,46), o custo de armários (4.830.000,00) e o custo de cabeamento (80.908,46) da melhor solução encontrada.

Na figura 4.12 na página 39 a linha contínua ilustra custo do cabo por geração e a linha tracejada ilustra a quantidade de seções (armários) por geração. Esta figura representa o que acontece com o algoritmo nas iterações seguintes à diminuição de um armário, o que não é

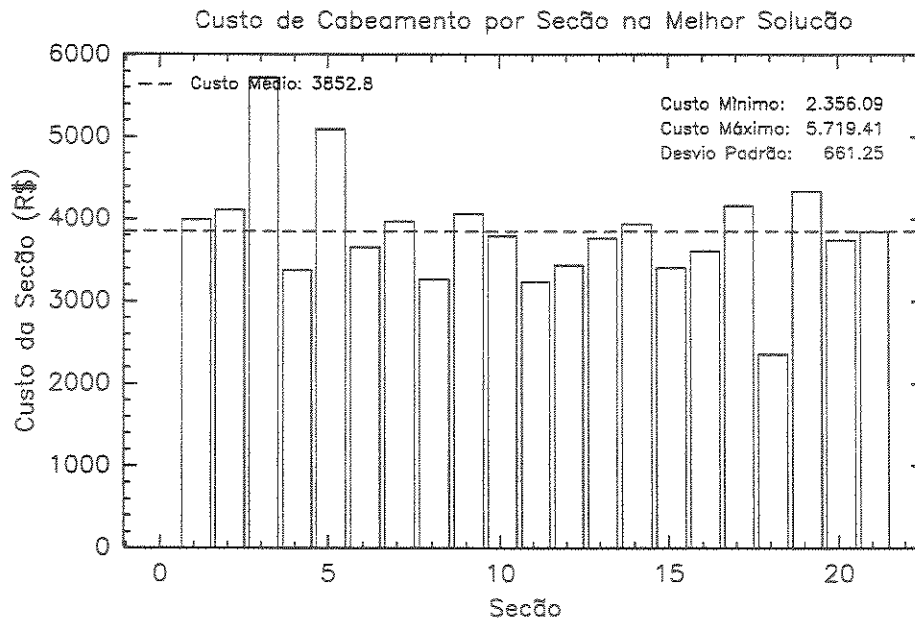


Figura 4.10: Solução para o Sul da Vila Mariana: custo por seção.

possível de observar pela figura 4.11. Observamos que ao diminuir um armário o custo de cabo aumenta consideravelmente, isto porque algumas seções tiveram que atender pontos mais distantes de seus armários, e durante a evolução do AG, o custo de cabo foi reduzido, sendo menor do que o custo de cabo quando a solução possuía uma seção a mais (primeiras iterações).

A figura 4.13 na página 40 mostra o particionamento dos pontos de demanda em seções de serviço.

### Região Leste da Vila Mariana

As figuras desta seção ilustram os gráficos resultantes da aplicação do algoritmo genético para a região Leste da Vila Mariana.

A figura 4.14 apresenta a distribuição de demanda entre as seções resultantes da solução. A demanda total desta região é 11.349,6, portanto o número mínimo de armários é igual a 24. É o que obtemos nesta solução. A média da distribuição de demanda entre as seções é igual a 472,9, e o desvio padrão desta distribuição é igual a 4,15, ou seja, as seções estão muito próximas da carga máxima desejada com uma variação muito pequena entre as demandas de cada uma. Como consideramos a distribuição de demanda entre as seções como medida de qualidade de uma solução, podemos concluir que a solução apresentada possui uma alta qualidade.

A figura 4.15 apresenta o custo de cabeamento em cada seção. Observamos que algumas seções possuem um custo muito maior do que outras, devido à extensão da seção. Quanto maior for sua área de atendimento, mais alto será o custo de atender os pontos que a ela pertencem.

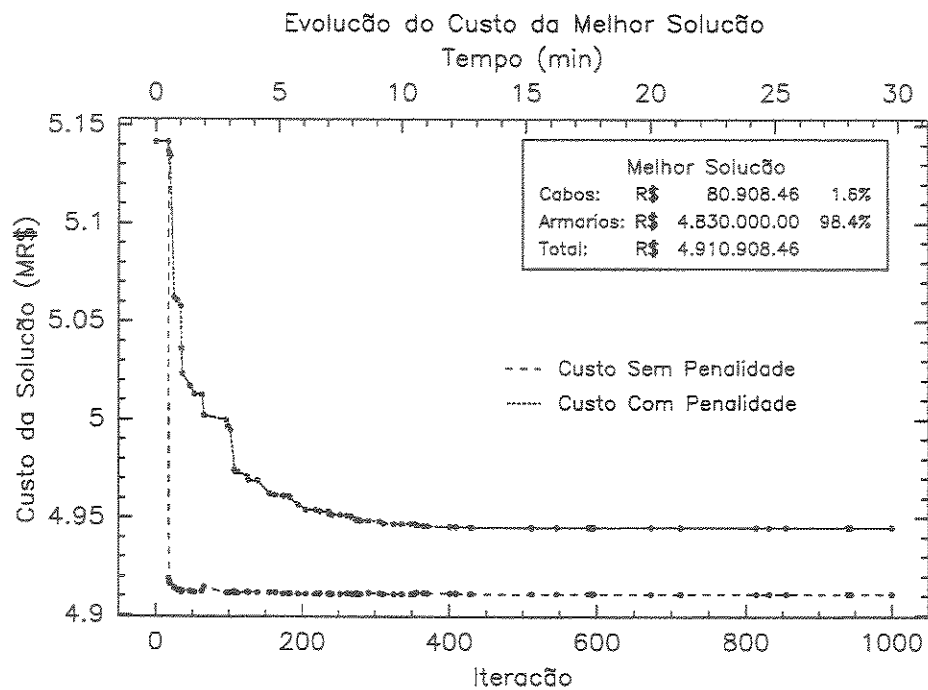


Figura 4.11: Solução para o Sul da Vila Mariana: custo total por iteração.

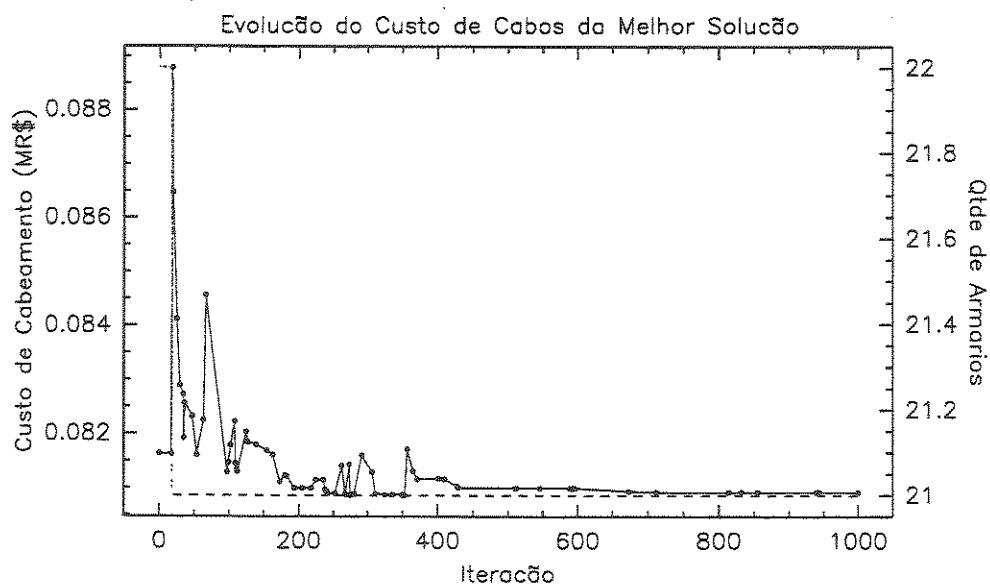


Figura 4.12: Solução para o Sul da Vila Mariana: custo de cabeamento por iteração.





Figura 4.13: Solução para o Sul da Vila Mariana.

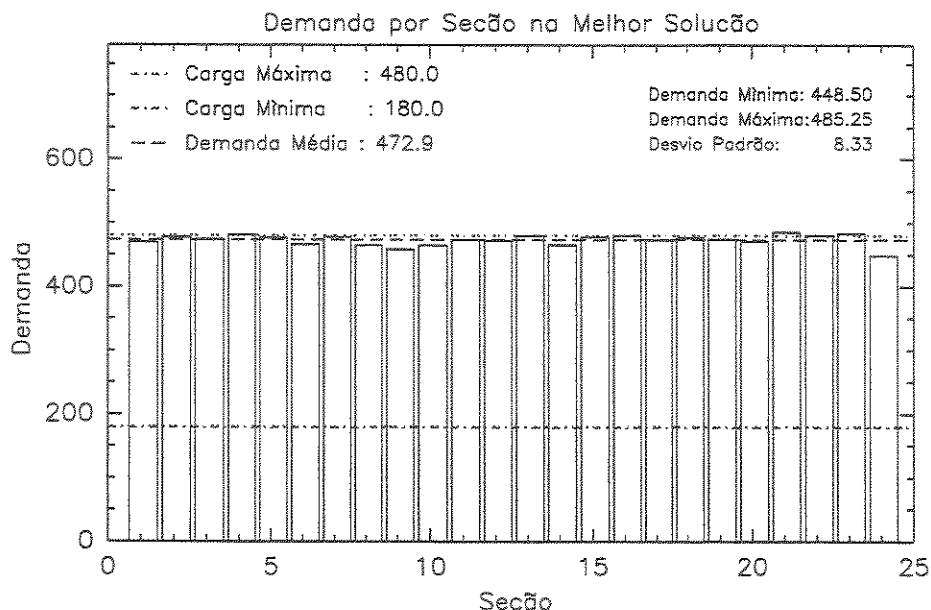


Figura 4.14: Solução para o Leste da Vila Mariana: demanda por seção.

A figura 4.16 na página 43 ilustra a evolução do algoritmo genético durante as gerações, ou seja, indica o custo total da melhor solução por iteração. A linha tracejada representa o custo da melhor solução e a linha contínua ilustra este custo mais a penalidade. Observamos que a penalidade aplicada às melhores soluções é pequena, isto indica que existem poucas seções que estão acima da carga máxima permitida, e a quantidade ultrapassada é pequena. A queda brusca no custo que observamos nas primeiras gerações é devido à diminuição de um armário. Nesta figura, apresentamos também o custo total (5.607.074,85), o custo de armários (5.520.000,00) e o custo de cabeamento (87.074,85) da melhor solução encontrada.

Na figura 4.17 na página 43 a linha contínua ilustra o custo do cabo por geração e a linha tracejada ilustra a quantidade de seções por geração. A primeira iteração representa a solução gerada pelo algoritmo de solução inicial, este resultou numa solução com 25 armários e um custo de cabeamento elevado. No final da execução do algoritmo conseguimos uma solução com 24 armários e um custo de cabeamento inferior ao inicial.

A figura 4.18 na página 44 mostra o particionamento dos pontos de demanda em seções de serviço.

### Região Norte da Vila Mariana

A figura 4.19 na página 45 ilustra a distribuição de demanda nas seções da melhor solução encontrada pelo AG, para os parâmetros do caso 7 da tabela 4.7 na página 34. Nesta figura observamos que a demanda média das seções é igual a 476,1 e possui 72 seções. A distribuição de demanda está bastante uniforme com um desvio padrão de 9,03.

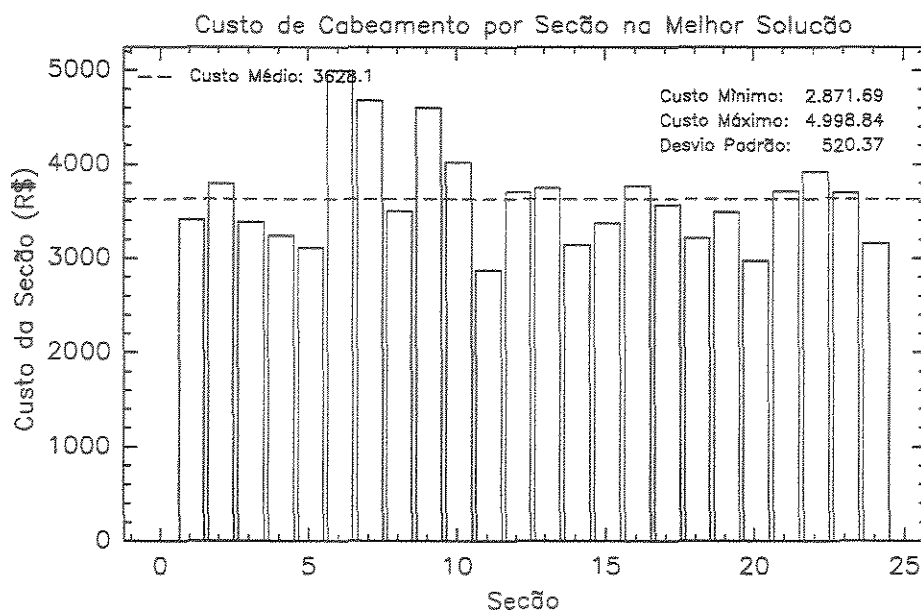


Figura 4.15: Solução para o Leste da Vila Mariana: custo por seção.

As figuras 4.20 e 4.21 na página 46 ilustram a evolução do algoritmo genético quanto ao custo total e custo de cabeamento da melhor solução da população, respectivamente.

A figura 4.22 na página 47 mostra o particionamento dos pontos de demanda em seções de serviço.

Aumentando a taxa de mutação para 80%, o algoritmo diminuiu o número de seções para 71 e a média de demanda por seção subiu para 482,8, resultando numa solução com um custo bem inferior ao custo ilustrado anteriormente. A distribuição de demanda para este caso pode ser observada na figura 4.23 na página 48, onde constatamos que a maior demanda de uma seção é 529,30 e a menor é 463,50 e o desvio padrão da distribuição de demanda entre as seções é de 10,46.

A figura 4.24 na página 48 ilustra o custo de cabeamento de cada seção da solução. A figura 4.25 ilustra a evolução do algoritmo genético quanto ao custo total da melhor solução por iteração. Nas primeiras iterações o número de seções da melhor solução oscila entre 71 e 72, devido à penalidade aplicada às seções que estão acima da carga máxima.

A figura 4.26 apresenta a evolução do custo de cabo da melhor solução por iteração, nesta figura observamos que ao tentar distribuir os pontos de demanda nas seções, o custo de cabo aumenta e diminui, conforme o número de armários oscila, até que o algoritmo consegue uma configuração dos pontos tal que o custo do cabo diminua e o número de armários fique mínimo.

A figura 4.27 na página 50 apresenta a divisão da área de projeto em seções de serviço para o teste com taxa de mutação igual a 80%.

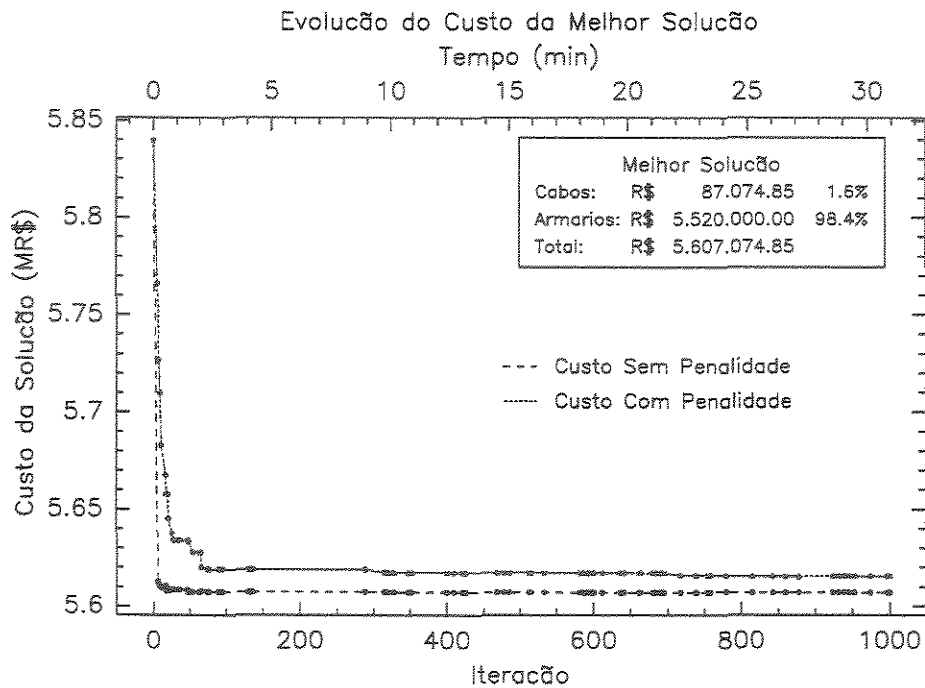


Figura 4.16: Solução para o Leste da Vila Mariana: custo total por iteração.

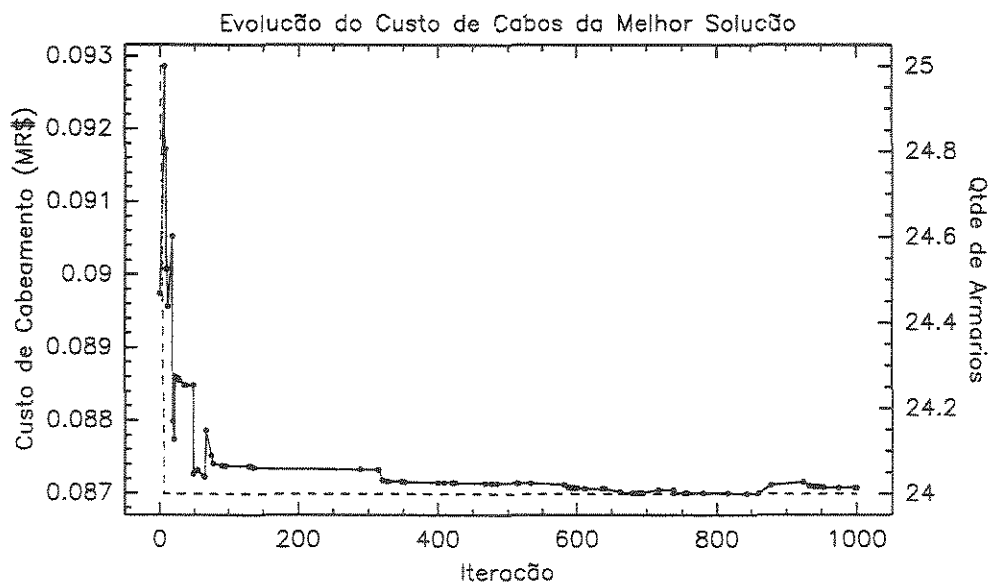


Figura 4.17: Solução para o Leste da Vila Mariana: custo de cabeamento por iteração.

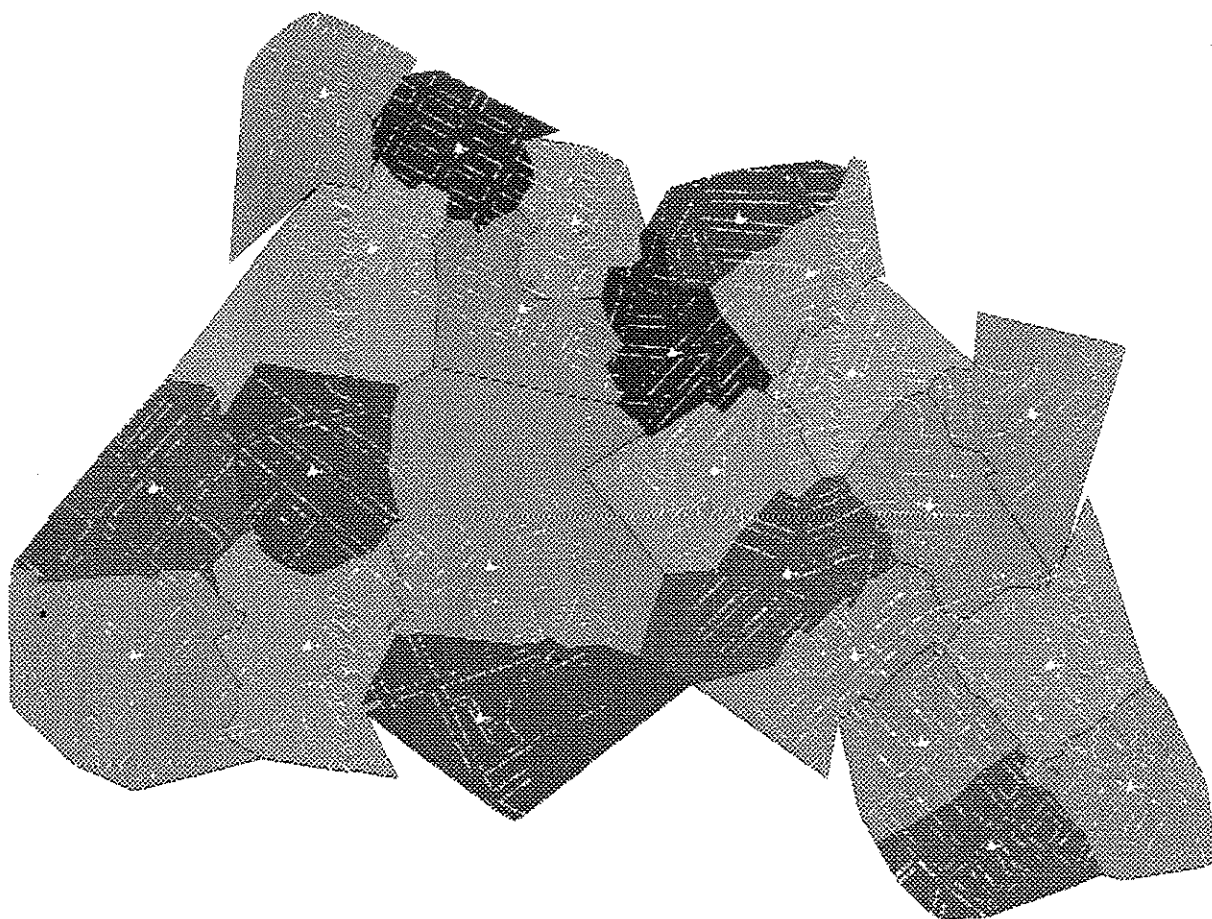


Figura 4.18: Solução para o Leste da Vila Mariana.

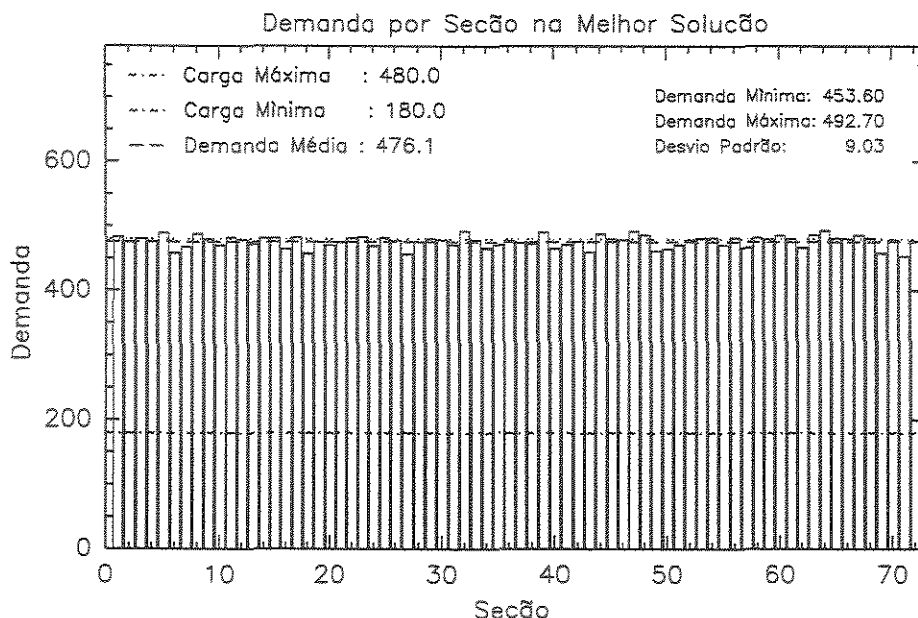


Figura 4.19: Solução para o Norte da Vila Mariana: demanda por seção

### Bairro da Liberdade

Assim como para a Vila Mariana, vários testes foram realizados para o Bairro da Liberdade. Esta instância é muito maior que a anterior, visto que possui muitos pontos e com demandas altas bem concentradas.

A tabela 4.9 indica os parâmetros de dois testes realizados para o Bairro da Liberdade que iremos detalhar nesta seção. O teste 1 resultou numa solução melhor geometricamente, porém o teste 2 resultou numa solução com três armários a menos e portanto num custo bem inferior, como podemos comparar pela tabela 4.10.

Teste	lado	ngen	npop	$t_e$ %	$t_c$ %	$t_s$ %	$t_m$ %	$p_1$ %	$p_2$ %	$p_3$ %	$p_i$ %	$p_d$ %	$p_v$ %	$\gamma$
1	20	1000	50	10	80	10	20	25	25	50	40	10	50	1000
2	20	1000	50	10	80	10	40	25	25	50	40	10	50	650

Tabela 4.9: Testes para o Bairro da Liberdade.

teste	tempo (min)	número de seções	custo total	custo cabos	% cabo	demanda			
						desvio	mínima	máxima	média
1	550	190	44.363,63	663,63	1,5	19,34	423,00	525,30	473,80
2	600	187	43.661,22	651,22	1,5	21,46	411,10	560,05	481,40

Tabela 4.10: Resultado dos testes para o Bairro da Liberdade.

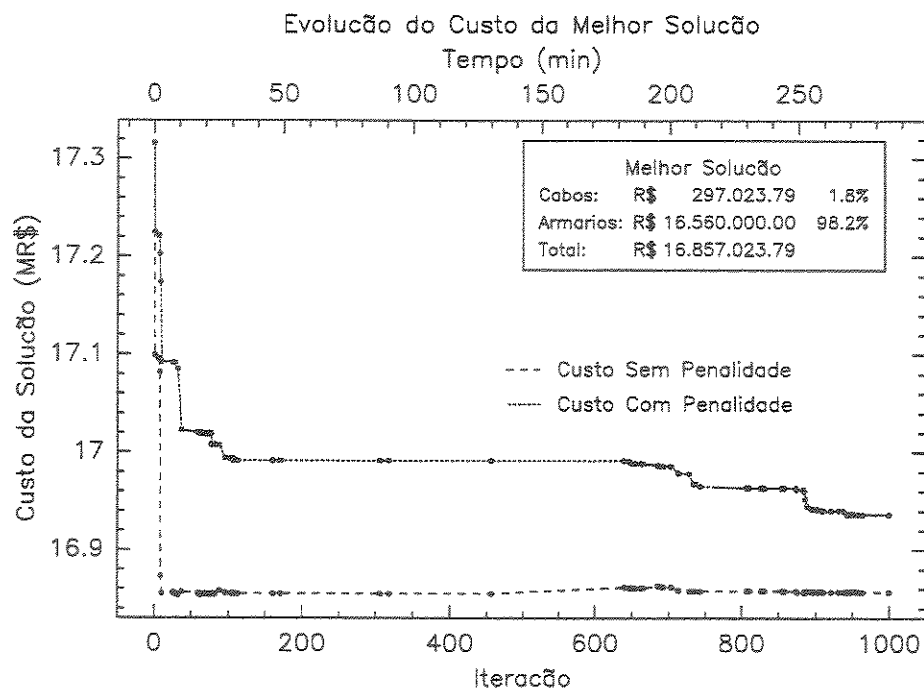


Figura 4.20: Solução para o Norte da Vila Mariana: custo total por iteração

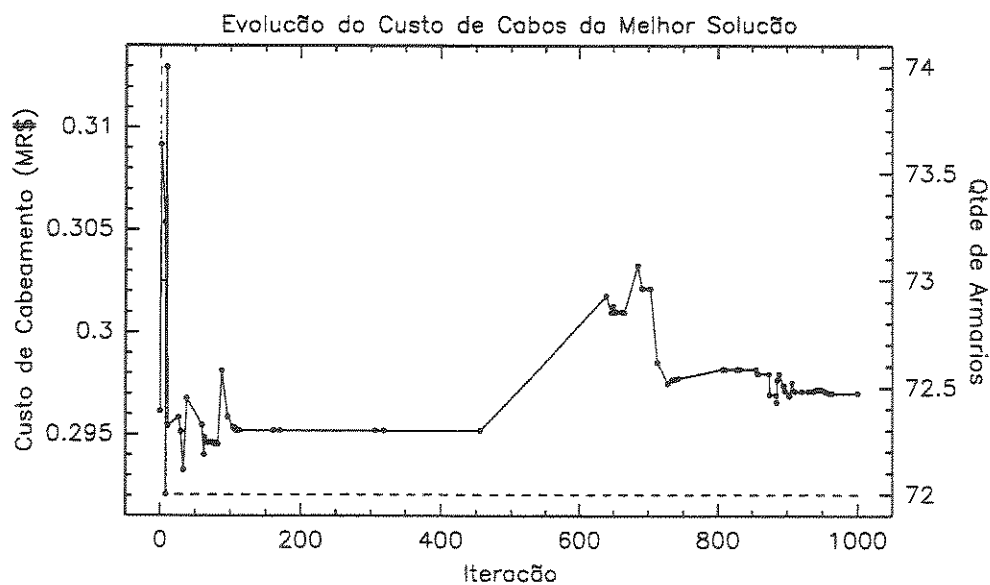


Figura 4.21: Solução para o Norte da Vila Mariana: custo de cabeamento por iteração



Figura 4.22: Solução para o Norte da Vila Mariana.



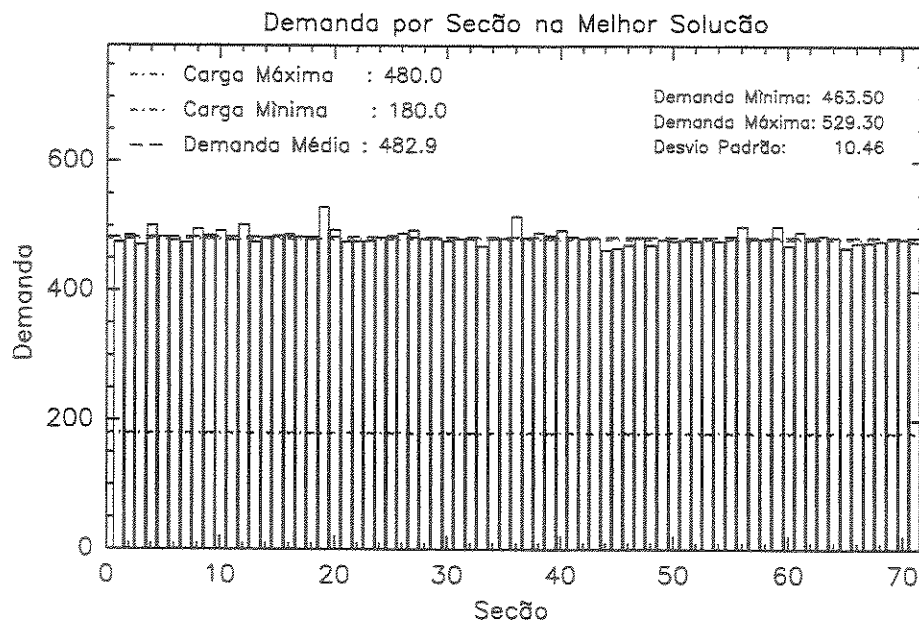


Figura 4.23: Solução para o Norte da Vila Mariana: demanda por seção

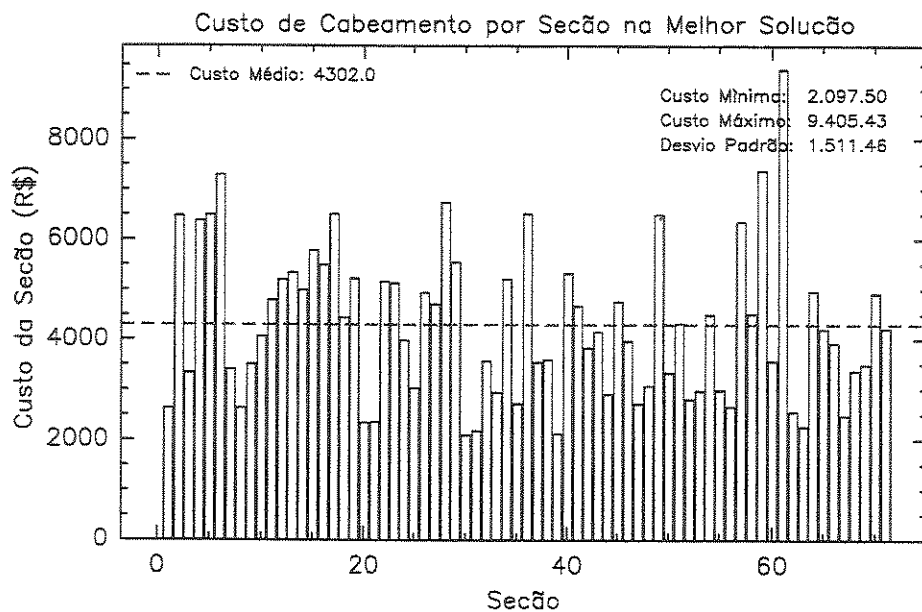


Figura 4.24: Solução para o Norte da Vila Mariana: custo por seção

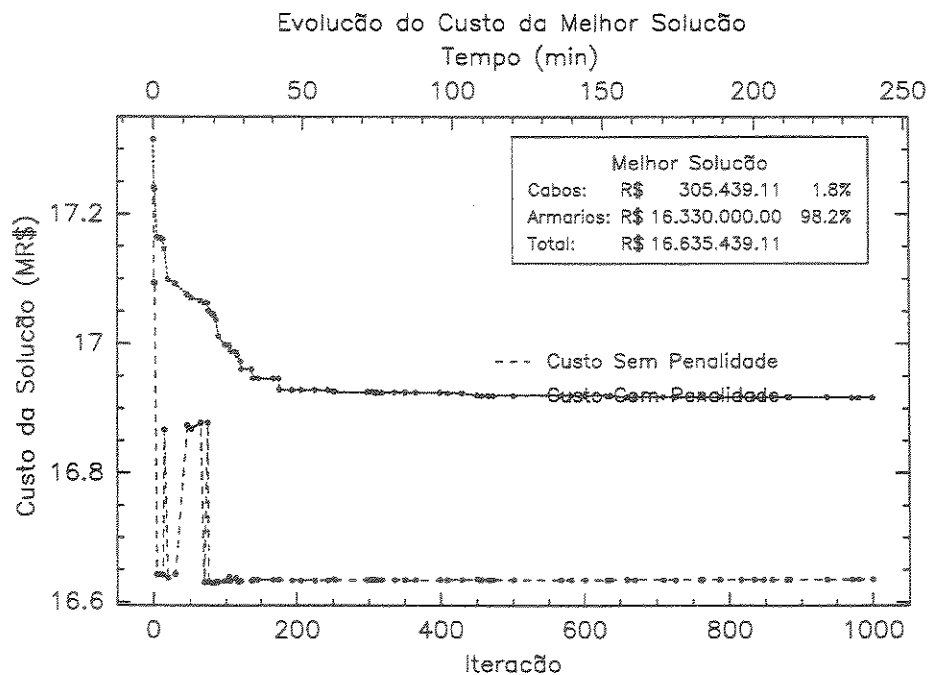


Figura 4.25: Solução para o Norte da Vila Mariana: custo total por iteração

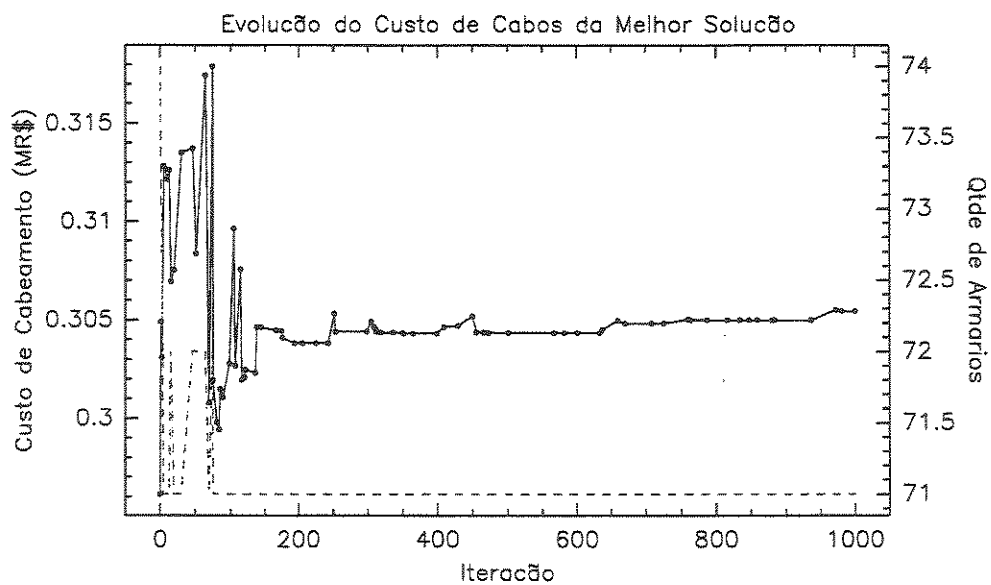


Figura 4.26: Solução para o Norte da Vila Mariana: custo de cabeamento por iteração



Figura 4.27: Solução para o Norte da Vila Mariana.

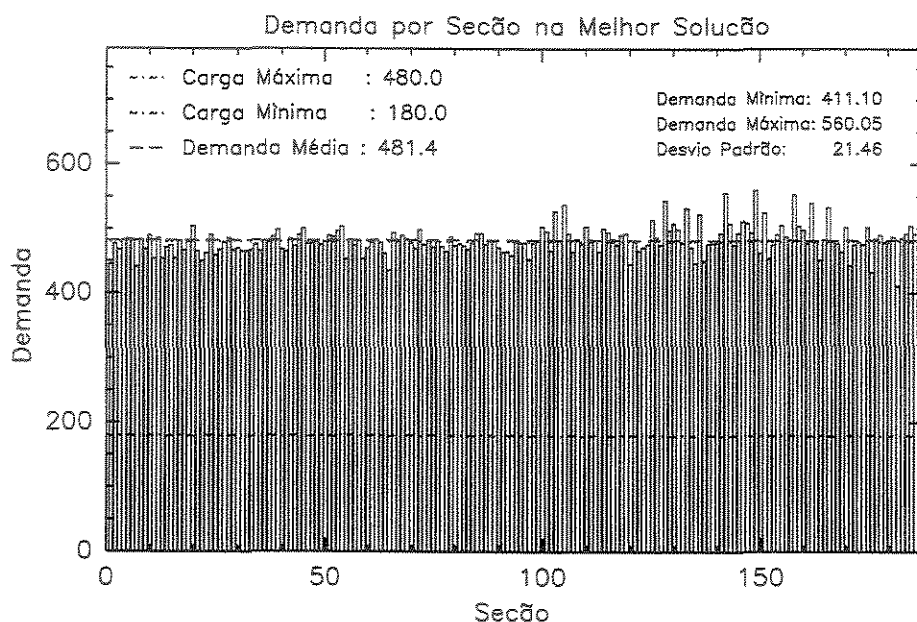


Figura 4.28: Solução para o do Bairro da Liberdade: demanda por seção.

As figuras listadas a seguir apresentam a solução do algoritmo para o conjunto de parâmetros do teste 2 da tabela 4.9. A figura 4.28 ilustra a distribuição de demanda entre as seções da solução, nela observamos que a demanda média é de 481,4 muito próxima da carga máxima desejada. A seção com demanda máxima atende 560,05 pontos de demanda e a com demanda mínima atende 411,10 pontos. O desvio padrão desta distribuição é 21,46. Esta instância é muito maior, densa e complexa do que as anteriores, pois possui muito mais seções e pontos de demanda concentrados. Dado estes fatores, concluímos que o resultado encontrado pelo algoritmo foi adequado.

A figura 4.29 na próxima página mostra o custo de cabeamento das seções, onde o menor custo é igual a R\$150,00 o que indica que existe uma seção com poucos pontos de demanda alta. Por outro lado, existem seções com muitos pontos dispersos, o que resulta num custo elevado de cabeamento.

Pela figura 4.30 na página seguinte acompanhamos a evolução do custo da melhor solução durante as gerações do algoritmo. Percebemos várias quedas no gráfico referentes às quedas nos custos de cabo e armários, principalmente. O algoritmo proporcionou uma melhora de R\$ 6.000.000,00 a partir da solução inicial.

Na figura 4.31 na página 53 representamos a evolução do custo de cabeamento por geração, sem considerar os custos de armários. Por este gráfico conseguimos acompanhar como o algoritmo evolui tentando reduzir o custo de cabeamento. Quando ocorre uma diminuição do número de armários, o que podemos acompanhar pela linha tracejada na figura 4.31, o custo de cabeamento aumenta, até que o algoritmo consegue uma distribuição dos pontos de demanda

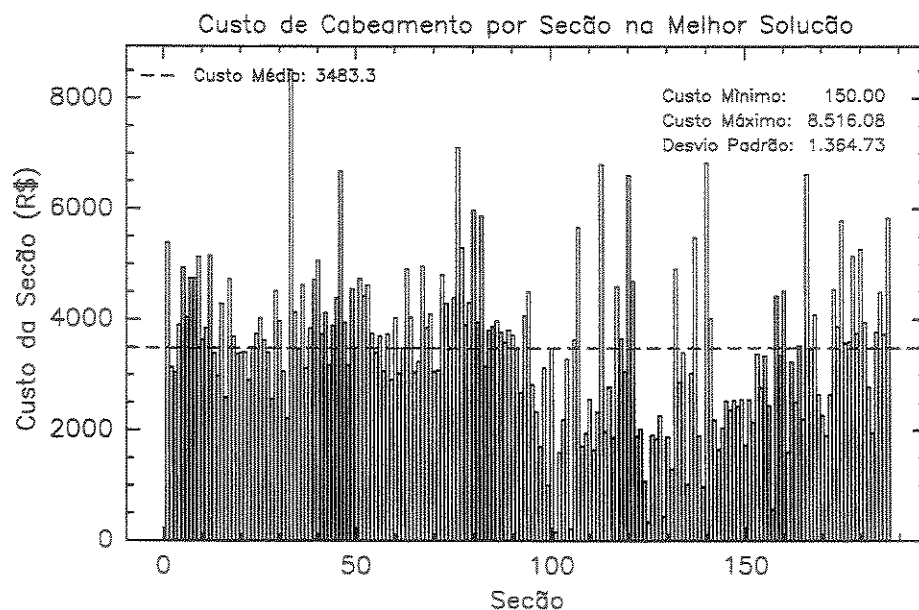


Figura 4.29: Solução para o do Bairro da Liberdade: custo por seção.

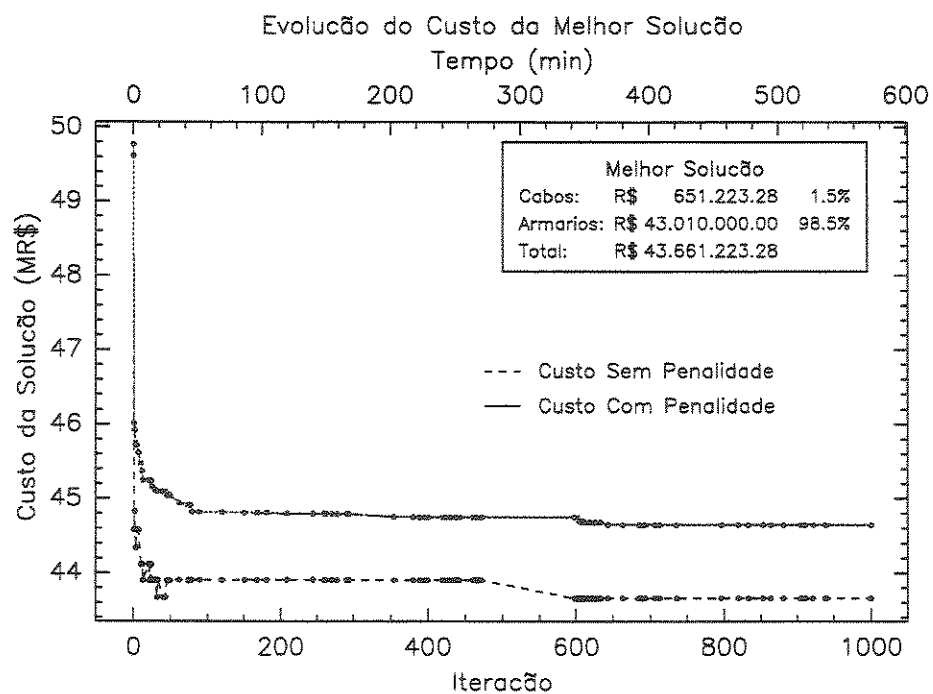


Figura 4.30: Solução para o do Bairro da Liberdade: custo total por iteração.

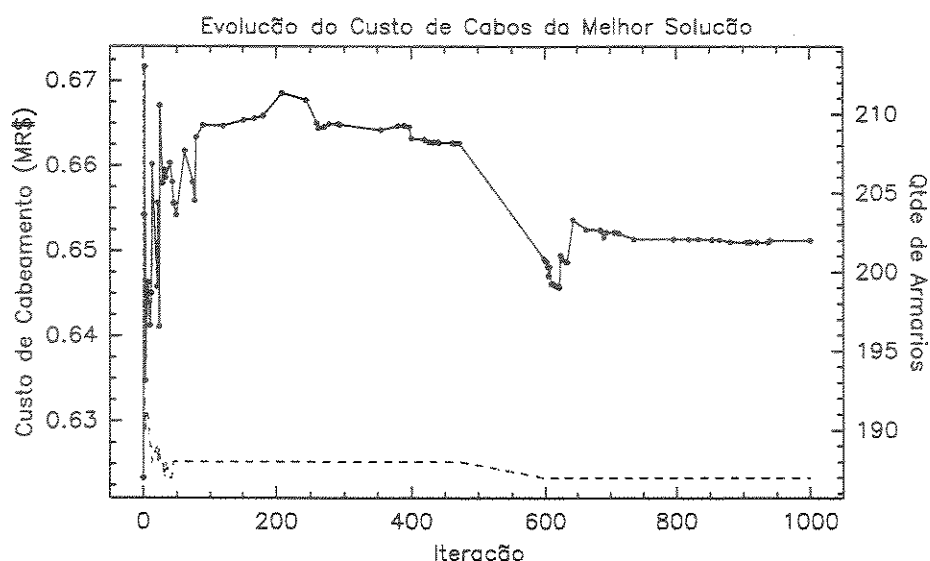


Figura 4.31: Solução para o do Bairro da Liberdade: demanda por seção.

entre as seções de forma que este custo diminua. Nesta figura, observamos também, como a melhor solução encontrada conseguiu reduzir consideravelmente o número de seções. A solução inicial possuía 213 seções, enquanto a melhor obteve 187, uma diferença de 1.380.000,00 reais.

A figura 4.32 na página seguinte mostra o particionamento dos pontos de demanda em seções de serviço.

#### 4.3.4 Comparação com Busca Tabu e GRASP

Uma busca tabu (BT) e um GRASP (*Greedy Randomized Adaptive Search Procedure*) também foram implementados para este problema e testados para as mesma instâncias. A implementação da Busca Tabu está descrita em [20], e a implementação do GRASP está descrita em [19]. A tabela 4.11 ilustra os resultados obtidos pela BT e a tabela 4.12 ilustra os resultados obtidos pelo GRASP.

teste	tempo (seg)	número de seções	custo total	custo cabos	%	demanda			
						desvio	mínima	máxima	média
1	20	21	4.907,80	77,80	1,6	63,70	324,00	529,20	481,56
2	30	24	5.606,27	86,27	1,5	68,63	263,00	527,10	472,89
3	100	72	16.822,34	262,34	1,6	79,18	212,55	592,20	476,10
4	400	200	46.533,83	533,83	1,1	102,67	11,10	583,80	450,10

Tabela 4.11: Resultados da BT



Figura 4.32: Solução para o Bairro da Liberdade.

teste	tempo (min)	número de seções	custo total	custo cabos	‰ cabo	demanda			
						desvio	mínima	máxima	média
1	7,5	21	4.909,49	79,49	1,7	66,82	284,35	597,20	481,56
2	9	24	5.613,81	93,81	1,7	49,78	349,10	586,40	472,89
3	15	71	16.631,60	301,60	1,8	80,62	264,20	593,00	482,90
4	20	187	43.706,61	696,61	1,6	93,51	210,40	598,80	481,40

Tabela 4.12: Resultados do GRASP

Quando o número de pontos de demanda é pequeno, BT obteve soluções com um custo de cabeamento um pouco menor do que o AG, mas o AG produz uma distribuição de demanda entre as seções melhor. Para grandes instâncias, o AG obteve as melhores soluções. Já a técnica GRASP também obteve o número mínimo de seções para todas as instâncias, contudo a distribuição de demanda entre as seções ficou bastante irregular. Esta é uma característica muito importante do AG, conseguir uma distribuição uniforme de demanda entre as seções, e ainda um custo total do projeto reduzido.

A tabela 4.13 resume os resultados obtidos pelo AG, pela BT e pelo GRASP, para facilitar a comparação. Nesta tabela GR significa GRASP.

teste	tempo			número seções			demanda					
	(min)	(seg)	(min)				média			desvio		
	AG	BT	GR	AG	BT	GR	AG	BT	GR	AG	BT	GR
1	29	20	7,5	21	21	21	481,56	481,56	481,56	4,99	63,70	66,82
2	30	30	9	24	24	24	472,89	472,89	472,89	7,09	68,63	49,78
3	200	100	15	71	72	71	480,62	476,10	480,62	17,87	79,18	80,62
4	600	400	20	187	200	187	481,40	450,10	481,40	21,46	102,67	93,51

Tabela 4.13: Comparação entre AG, BT e GRASP



## Capítulo 5

# AG para Posicionamento de Caixas Terminais

### 5.1 Introdução

Neste capítulo detalharemos o algoritmo genético desenvolvido para o problema de “Posicionamento de Caixas Terminais Considerando a Rede Existente”. Chamaremos este algoritmo de AGPC. Apresentamos também os resultados obtidos pelo AGPC quando aplicado em instâncias reais.

### 5.2 Modelagem

A entrada para este problema é formada pelos seguintes itens:

- Um conjunto  $I$  de pontos de demanda, onde cada ponto possui:
  - Coordenada do ponto:  $(x, y)$ .
  - Demanda:  $d$ . Representa o número de linhas telefônicas que este ponto necessita. Este número representa a demanda atual mais a demanda promitente. A demanda promitente é uma estimativa futura da necessidade deste ponto.
  - Identificador da caixa que o atende:  $id_c$ . Este valor é zero se é um ponto novo, ou seja ainda não é atendido por nenhuma caixa terminal.
  - Demanda que está sendo atendida atualmente pela caixa  $id_c$ :  $oc$ . Este valor é zero se é um ponto novo.
- Um conjunto  $J$  de postes, onde cada poste possui:

- Coordenada do poste:  $(x, y)$ .
  - Indicação se pode receber instalação de uma caixa terminal: *canInstall*.
  - Identificador: *id*.
- Um conjunto  $E = I \times J$  de ligações entre pontos de demanda e postes, onde cada ligação armazena a distância do ponto ao poste.
  - Um conjunto  $F$  de caixas terminais instaladas em postes. Cada caixa terminal possui:
    - Identificador: *id*.
    - Identificador do poste aonde ela está instalada:  $id_p$ .
    - Capacidade da caixa: *s*
    - Um subconjunto de  $I$  indicando os pontos de demanda atendidos por esta caixa e, para cada ponto, o valor que ela atende: *oc*

### 5.2.1 Representação

Um cromossomo deve ser capaz de indicar em qual poste um cliente está instalado e em que caixa deste poste o cliente se conecta. Na representação usamos um vetor de  $n$  posições ( $n$  é o número de pontos de demanda), onde em cada posição existem dois números, um indicando o identificador do poste e o outro um identificador para a caixa deste poste. Para auxiliar esta representação, temos um grafo de adjacências dos possíveis postes nos quais um cliente pode se conectar. Veja figura 5.1

Para auxiliar no cálculo da aptidão de um cromossomo e nas operações de cruzamento e mutação, mantemos uma decodificação do cromossomo. O cromossomo decodificado é composto por um vetor  $T$  de caixas terminais, instaladas em postes. Cada caixa terminal possui um identificador do poste no qual ela está instalada, um inteiro identificando a sua capacidade, um valor booleano indicando se a caixa é nova ou já instalada, um número indicando a demanda atendida por ela e um conjunto de pontos de demanda (ou seja, um subconjunto de  $I$ ) representando os pontos  $(x, y, d)$  que estão ligados a ela. A figura 5.2 ilustra uma caixa terminal e seus dados.

A figura 5.3 ilustra uma decodificação para o cromossomo da figura 5.1.

### 5.2.2 População Inicial

Como estamos tratando da rede existente, sendo necessário dimensioná-la para atender demandas futuras e novas, na população inicial devemos representar a rede atual e uma possível solução para as demandas ainda não instaladas. Na população inicial também devemos incorporar

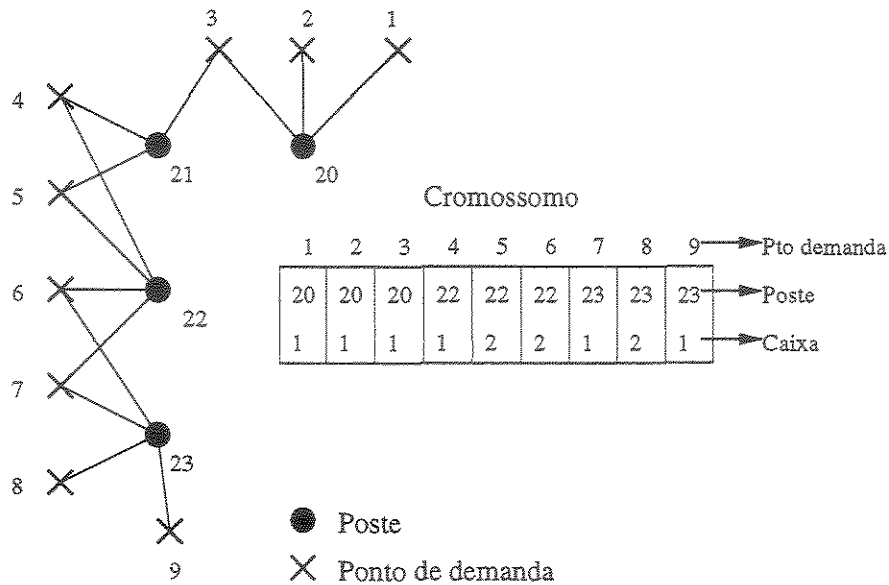


Figura 5.1: Representação de um cromossomo. É um vetor de tamanho igual ao número de pontos de demanda. Cada posição do vetor contém o identificador do poste que atende o cliente e o identificador da caixa deste poste onde realmente ele está conectado. Cada posição só pode conter identificadores de postes adjacentes ao cliente. Estas adjacências são dadas pelo grafo de clientes e postes.

a possibilidade de aliviar caixas existentes saturadas. Assim, decidimos que alguns indivíduos representarão soluções mantendo a configuração atual da rede e outros representarão soluções completamente diferente da rede atual.

Portanto, um indivíduo  $\pi$  é inicializado da seguinte forma:

1. Geramos aleatoriamente um número  $\beta$  entre 0 e 1.
2. Se  $\beta$  for maior que 0,2 então:
  - (a) todas as caixas existentes atualmente na rede são representadas, ou seja as posições em  $\pi$  dos pontos já instalados recebem o identificador do poste e da caixa que atendem o ponto representado pela posição.
  - (b) os pontos novos que ainda não foram atendidos por caixas são instalados em caixas selecionadas aleatoriamente dentro do conjunto de postes adjacentes ao ponto. Este passo é realizado da seguinte forma:
    - i. Selecionamos aleatoriamente um poste  $p_i$  adjacente ao ponto em questão  $\pi[j]$ .
    - ii. Verificamos se  $p_i$  possui alguma caixa já instalada com capacidade suficiente para instalar  $\pi[j]$ , tanto caixa antiga como nova.

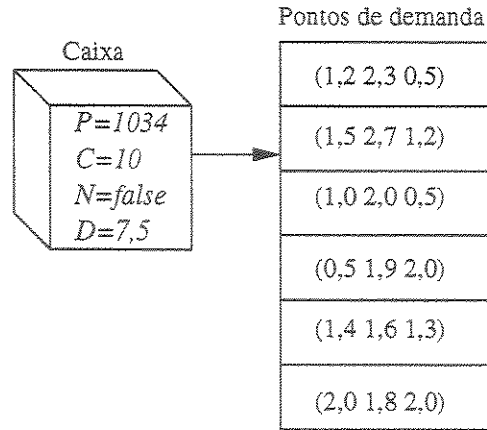


Figura 5.2: Uma caixa terminal, onde  $P$  é o identificador do poste no qual ela está instalada,  $C$  é a capacidade,  $N$  indica se a caixa foi instalada pelo algoritmo e  $D$  é a demanda

- iii. Repetimos os passos 2(b)i e 2(b)ii até encontrarmos algum poste que possua uma caixa com capacidade suficiente para instalar  $\pi[j]$  ou não exista tal poste.
  - iv. Se nenhum poste adjacente a  $\pi[j]$  foi encontrado, selecionamos aleatoriamente um poste adjacente a  $\pi[j]$  e instalamos nele uma nova caixa para atender  $\pi[j]$ .
3. Se  $\beta$  for menor que 0,2, então realizamos o passo 2b para todos os pontos de demanda, novos ou já instalados.

A figura 5.4 ilustra o algoritmo de inicialização de um indivíduo  $\pi$ , usando o conjunto de pontos de demanda  $I$ .

### 5.2.3 Cruzamento

Num AG tradicional, um operador de cruzamento simples como cruzamento de um-ponto, é freqüentemente usado. Este operador gera aleatoriamente um *ponto de cruzamento* e troca *segmentos* das cadeias de cada um dos pais, produzindo dois filhos.

Para assegurar que o filho descendente do cruzamento não é idêntico a nenhum dos pais, trabalhamos com um operador de um-ponto restrito, discutido em [4]. Sejam  $\pi_1 = \pi_1[1], \dots, \pi_1[n]$  e  $\pi_2 = \pi_2[1], \dots, \pi_2[n]$  os pais selecionados para o cruzamento. Este operador consiste simplesmente em gerar um ponto de cruzamento  $j$  onde

$$\min\{i : \pi_1[i] \neq \pi_2[i]\} \leq j < \max\{i : \pi_1[i] \neq \pi_2[i]\}.$$

Após gerado o ponto de cruzamento, trocamos as partes de cada pai, criando dois filhos. Esta operação está ilustrada na figura 5.5.

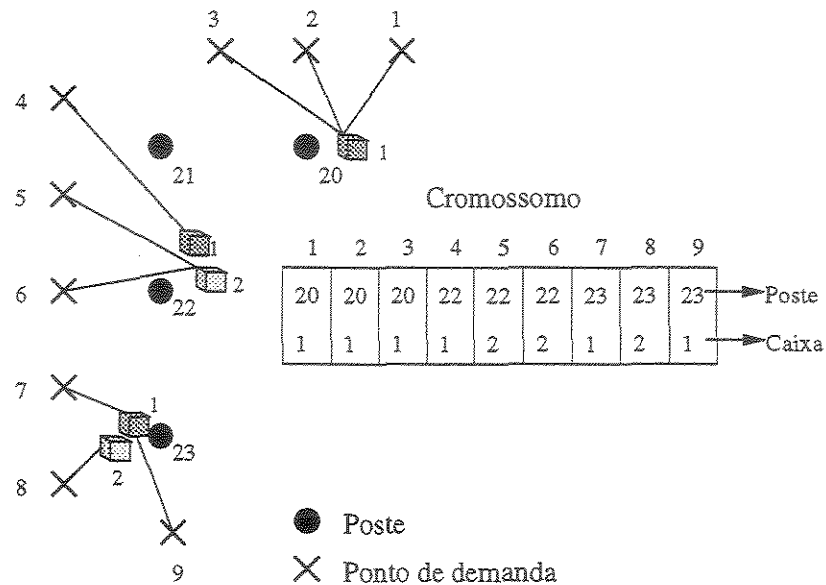


Figura 5.3: Decodificação do cromossomo em caixas terminais.

Porém este operador pode gerar soluções com caixas atendendo pouca demanda. Assim, após este operador, os filhos sofrem um operador de reparo. Este operador trabalha sobre o cromossomo decodificado e, para cada caixa terminal que possui demanda menor do que a mínima permitida, esta é desinstalada e pontos que ela atendia são atendidos por caixas vizinhas, sem ultrapassar a capacidade das últimas. Para caixas que estão acima da capacidade, ou seja saturadas, são feitas transferências de pontos entre elas e caixas vizinhas. Este algoritmo de reparo está descrito na figura 5.6.

A função  $DESINSTALAPONTOS(t, d)$  opera da seguinte forma. Para cada ponto de demanda  $p$  atendido por  $t$ , percorre os postes vizinhos a  $p$ , verificando se existe algum poste com caixa terminal com capacidade suficiente para atender a demanda de  $p$ . Se existir a caixa transfere o ponto para ela. Repete esse passo até que a quantidade de demanda subtraída da caixa  $t$  seja maior ou igual a  $d$ .

Com esse operador de reparo, garantimos que uma boa quantidade de caixas ficam com as capacidades no intervalo desejado. E como sempre instalamos os pontos em postes adjacentes a ele, sempre teremos soluções factíveis no sentido de que os pontos estão ligados a postes válidos.

### 5.2.4 Mutação

O operador de mutação utilizado é semelhante ao tradicional. Dado um cromossomo  $\pi = \pi[1], \dots, \pi[n]$ , selecionado para sofrer mutação, cada posição tem uma probabilidade de ser mudada. A mudança em uma posição é feita selecionando aleatoriamente um dos postes adjacentes

```

RANDOMINITIALIZE( $\pi, I$ )
1   $\beta \leftarrow \text{random}(0, 1)$ 
2  for  $i = 0$  até  $|\pi|$ 
3  do  $p[i].\text{poste} \leftarrow 0$ 
4      $p[i].\text{caixa} \leftarrow 0$ 
5     if  $\beta \geq 0.2$  e  $I_i$  já estava instalado
6     then  $\pi[i].\text{poste} \leftarrow$  identificador do poste no qual  $I_i$  está instalado
7          $\pi[i].\text{caixa} \leftarrow$  identificador da caixa na qual  $I_i$  está instalado
8     else
9         while  $\pi[i].\text{caixa} = 0$  (ou não existe mais opção de poste)
10        do  $\pi[i].\text{poste} \leftarrow$  identificador de um poste adjacente a ele ainda não selecionado
11            if existe uma caixa  $id_c$  instalada em  $\pi[i].\text{poste}$  com capacidade para atender  $I_i$ 
12            then  $\pi[i].\text{caixa} \leftarrow id_c$ 
13            else  $\pi[i].\text{caixa} \leftarrow 0$ 
14
15        if  $\pi[i].\text{caixa} = 0$ 
16        then  $\pi[i].\text{poste} \leftarrow$  identificador de um poste adjacente a  $\pi[i]$ 
17            Cria uma nova caixa  $id_c$  em  $\pi[i].\text{poste}$ 
18             $\pi[i].\text{caixa} \leftarrow id_c$ 

```

Figura 5.4: Algoritmo de inicialização de um indivíduo  $\pi$ .

ao ponto que a posição representa e ligando o cliente a uma caixa deste novo poste. Porém, só efetuamos a mudança se no poste selecionado existe alguma caixa com capacidade suficiente para atendê-lo.

A figura 5.7 ilustra esta mutação. Inicialmente o cromossomo estava como na figura 5.7(a). Selecionando a posição 6 do cromossomo para sofrer mutação, o cromossomo mudado é o ilustrado na figura 5.7(b)

### 5.2.5 Função de Avaliação

Para calcular a aptidão de um indivíduo temos que obter o custo da solução representada por ele. Para isso, usamos a decodificação do cromossomo num vetor  $T$  de caixas terminais. A aptidão de um cromossomo é igual ao custo da solução mais uma penalidade para a violação das restrições de capacidade das caixas. Vale lembrar que quanto menor o valor da função de avaliação, mais apto é o indivíduo.

O custo  $w(\pi)$  de uma solução  $\pi$  decodificada em um conjunto  $T$  de caixas terminais é:  $w(\pi) = \sum_{t \in T} (c_t + \sum_{p \in t} c_{tp})$  onde,  $c_t$  é o custo de instalação da caixa terminal, se esta não estava instalada (este valor depende da sua capacidade),  $p$  representa os pontos de demanda atendidos por  $t$  e  $c_{tp}$  reflete o custo de instalar um ponto novo ou de transferir um ponto já

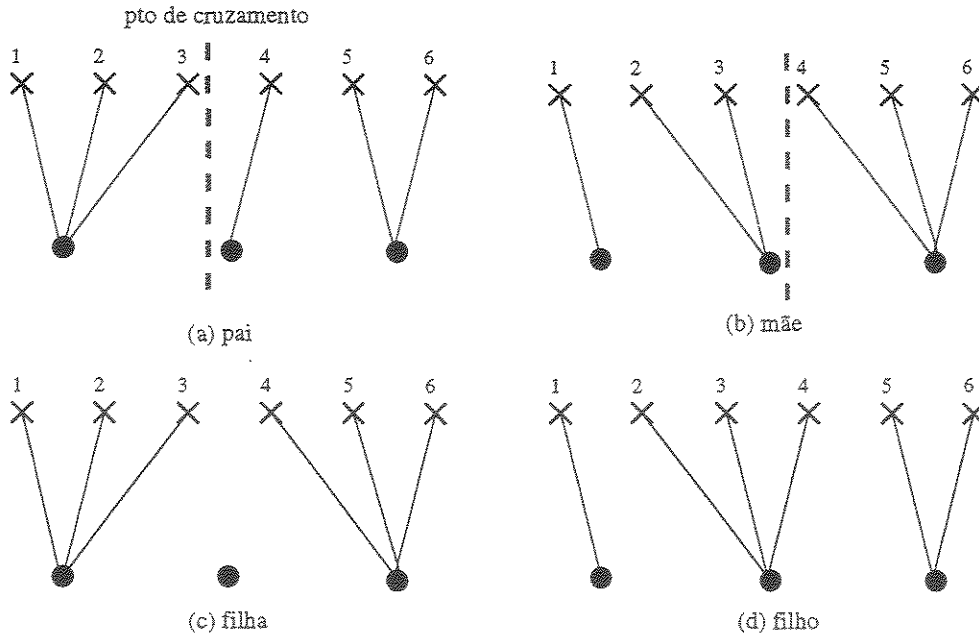


Figura 5.5: Cruzamento

instalado. Assim, temos:

$$c_{tp} = \begin{cases} 0 & \text{se } p \text{ já estava instalado na caixa } t \\ inst_{tp} + rem_{lp} & \text{se } p \text{ estava instalado na caixa } l \\ inst_{tp} + rem_{lp} + crem_l & \text{se } p \text{ estava instalado na caixa } l \text{ e a caixa } l \text{ ficou vazia} \\ inst_{tp} & \text{se } p \text{ é um ponto novo} \end{cases}$$

onde,  $inst_{tp} = d_{p_i} \times dist(t, p) \times c_d$ , sendo  $d_{p_i}$  a demanda do ponto  $i$ ,  $dist(t, p)$  a distância entre o ponto e o poste aonde está a caixa  $t$  e  $c_d$  o custo, por unidade de distância, do fio que liga o ponto à caixa;  $rem_{lp} = c_r \times d_{p_i}$ , com  $c_r$  igual ao custo de remoção de um assinante e  $crem_l$  é o custo de retirada da caixa  $l$ .

A penalidade,  $\gamma(\pi)$ , aplicada ao indivíduo  $\pi$ , decodificado num vetor  $T$  de caixas terminais, é igual a soma das penalidades em cada caixa existente na solução representada por ele, ou seja,  $\gamma(\pi) = \sum_{t \in T} \gamma_t$ . Dada uma caixa  $t$ , com capacidade  $s_t$  e demanda total que atende  $d_t$ , a penalidade associada a ela é dada por:

$$\gamma_t = \begin{cases} \alpha(d_t - K \times s_t) & \text{se } d_t > K \times s_t \\ \alpha(k \times s_t - d_t) & \text{se } d_t < k \times s_t \\ 0 & \text{se } k \times s_t \leq d_t \leq K \times s_t \end{cases}$$

onde  $\alpha$  é uma constante positiva,  $k$  e  $K$  são fatores de carga mínima e máxima da caixa, respectivamente.

```

CROSSREPAIR( $\pi$ )
1  //decodifica o cromossomo  $\pi$  num conjunto de caixas terminais T
2  DECODE( $\pi, T$ )
3  for  $t \in T$ 
4  do //onde  $d_t$  é a demanda atendida por  $t$  e  $s_t$  é a sua capacidade
5     if  $d_t < k \times s_t$ 
6     then DESINSTALAPONTOS( $t, d_t$ )
7     else if  $d_t > K \times s_t$ 
8     then DESINSTALAPONTOS( $t, d_t - K \times s_t$ )

```

Figura 5.6: Algoritmo de reparo.

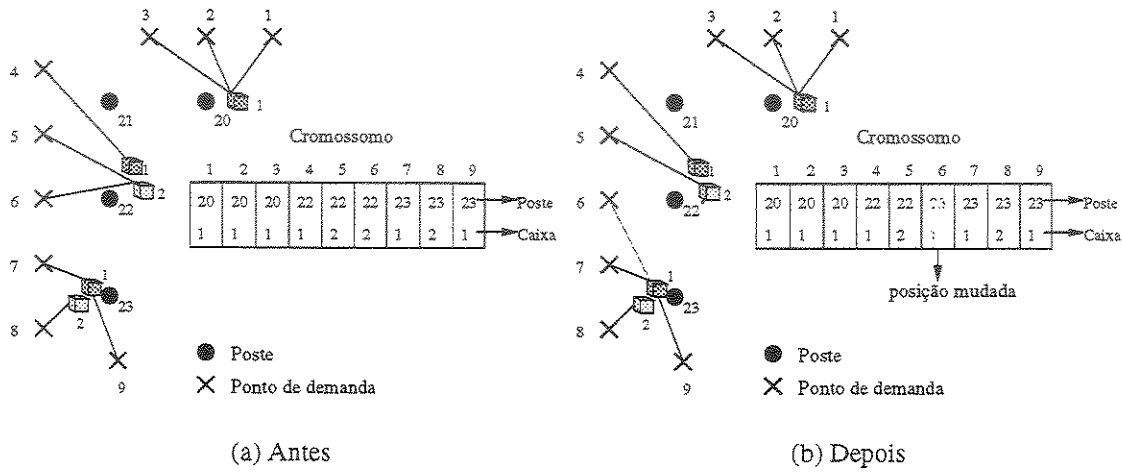


Figura 5.7: Uma mutação

Assim a função de avaliação de um indivíduo  $\pi$  é dada por:

$$f(\pi) = w(\pi) + \gamma(\pi)$$

### 5.2.6 Otimização

Usamos o algoritmo genético tradicional hibridizado com algumas buscas locais para direcionar a busca e evitar que ele convirja rapidamente para um ótimo local. Assim, se o algoritmo genético ficar um certo número de gerações sem evoluir, toda a população é submetida a um processo de otimização. Este número é calculado como 20% do número total de gerações. O processo de otimização consiste em:

- eliminação de caixas subutilizadas,



- alívio de caixas saturadas,
- balanceamento de caixas próximas e
- eliminação de atendimento muito distante.

### Eliminação de Caixas Subutilizadas e Alívio de Caixas Saturadas

Esta estratégia é a mesma usada como operador de reparo, após o cruzamento, e já descrito na seção 5.2.3. A idéia é tentar transferir os pontos da caixa subutilizada (saturada), para caixas que estão em postes adjacentes.

### Balanceamento de Caixas próximas

Esta estratégia tenta balancear caixas próximas que possuem demandas muito desequilibradas. Para cada caixa quase saturada, procuramos por caixas adjacentes que possuem capacidade para receber demanda dela e, assim, balancear a demanda atendida por ambas.

Ela procura por pares de caixas que possuem uma demanda média desejada (próxima da capacidade máxima de uma caixa) e atribui uma aptidão a cada par. A aptidão de um par de caixas  $t_1$  e  $t_2$ , com demandas  $d_1$  e  $d_2$ , e capacidades  $s_1$  e  $s_2$  é calculada da seguinte forma:

$$s_1 \left| \frac{d_1 + d_2}{s_1 + s_2} - \frac{d_1}{s_1} \right| + s_2 \left| \frac{d_1 + d_2}{s_1 + s_2} - \frac{d_2}{s_2} \right|$$

Dadas duas caixas em regiões próximas, uma boa solução é aquela em que as razões demanda/capacidade das caixas são aproximadamente as mesmas. A fórmula acima tenta refletir esta idéia. Quanto mais a razão demanda/capacidade estiver longe da razão média das duas caixas, então maior será a aptidão do par.

Trocamos pontos entre as caixas dos pares que possuem uma aptidão alta. Consideramos aptidão alta aquela que é pelo menos 70% do valor da aptidão do par mais apto.

### Eliminação de atendimento muito distante

Esta estratégia percorre todos os pontos de demanda verificando a sua distância em relação à caixa que o atende. Se este valor é maior do que a distância desejada (um parâmetro indicado pelo usuário), tentamos alocar o ponto no poste mais próximo. Esta tentativa é feita percorrendo os postes adjacentes em ordem crescente de distância e verificando se suas caixas possuem capacidade suficiente para atender a demanda do ponto em questão.

### 5.2.7 Outras Características

Na seleção para recombinação utilizamos o método da roleta, onde a probabilidade de seleção de cada indivíduo é inversamente proporcional a sua aptidão, e conseqüentemente, inversamente proporcional ao custo da solução. Outros métodos também foram testados. Usamos também elitismo, onde indicamos a quantidade de melhores indivíduos que sobreviverão na geração seguinte.

## 5.3 Resultados

### 5.3.1 Ambiente Computacional

O AGPC foi implementado em C++, executando tanto em ambiente Linux como em Windows. As instâncias apresentadas nesta seção foram testadas num processador Pentium IV com clock de 2GHz e uma memória RAM de 1GB.

### 5.3.2 Parâmetros

Realizamos vários testes para ajustar os parâmetros utilizados pelo algoritmo genético. A tabela 5.1 apresenta os parâmetros de projeto, definidos pelo usuário, e a tabela 5.2 apresenta os parâmetros utilizados pelo AGPC.

Dados do Projeto		
Símbolo	Descrição	Valor
$k$	fator de carga mínima da capacidade de uma caixa	30%
$K$	fator de carga máxima da capacidade de uma caixa	80%
$d_p$	demanda localizada mínima para caracterizar caixa predial	6
$c_{10}$	custo de instalação de uma caixa com capacidade 10	R\$ 60,77
$c_{20}$	custo de instalação de uma caixa com capacidade 20	R\$ 119,87
$crem$	custo de remoção de uma caixa terminal	R\$ 6,38
$rem$	custo de desinstalar uma demanda de uma caixa	R\$ 2,52
$c_d$	custo por metro do fio que liga uma caixa a uma demanda	R\$ 0,0194
$maxnbox$	número máximo de caixas por poste	2
$maxdist$	distância máxima desejada entre caixas e pontos de demanda	80
$maxdist1$	distância máxima permitida entre caixas e pontos de demanda	210

Tabela 5.1: Dados do projeto de posicionamento de caixas.

Dados do AG	
Símbolo	Descrição
$npop$	tamanho da população
$ngen$	número de gerações
$pcross$	probabilidade de cruzamento
$pmut$	probabilidade de mutação
$nelit$	número de melhores indivíduos que sobreviverão nas gerações seguintes
$\alpha$	coeficiente de penalidade

Tabela 5.2: Dados do AG.

### 5.3.3 Instâncias

Para os testes, usamos quatro instâncias que são seções de serviço do Bairro da Liberdade, na cidade de São Paulo. Estas seções foram calculadas pelo AGSS (algoritmo genético para definição de seções de serviço).

A tabela 5.3 mostra alguns dados numéricos sobre as quatro instâncias testadas. Os dados apresentados são o número de pontos que deverão ser alocados em caixas e a sua demanda total, a quantidade de pontos que possuem demanda maior ou igual a  $d_p$  e a sua demanda total, número de caixas terminais já instaladas na seção de serviço e a quantidade de demanda já instalada.

Dados	Inst. 1	Inst. 2	Inst. 3	Inst. 4
Número de pontos	309	317	227	360
Demanda total	487,4	557,7	416	498,2
Número de caixas prediais	5	4	0	2
Demanda das caixas prediais	54	39	0	18
Número de caixas instaladas	68	96	9	21
Número de caixas de 10	34	45	0	9
Número de caixas de 20	34	51	9	12
Demanda atendida pelas caixas	363	424	42	91

Tabela 5.3: Dados numéricos sobre as instâncias.

### 5.3.4 Teste de Parâmetros

Nesta seção apresentamos os casos de teste que realizamos para as quatro instâncias listadas acima, e também os resultados obtidos.

Estes resultados foram comparados com um limitante inferior obtido de uma formulação de programação linear inteira, descrita no relatório [18]. Esta formulação linear difere da formu-

lação matemática apresentada na seção 3.3. O limitante foi obtido durante a busca na árvore do *branch and bound* usando o resolvidor Xpress.

A tabela 5.4 apresenta os casos de testes realizados, indicando os valores para os parâmetros da tabela 5.2. Estes testes variam as probabilidades de cruzamento e mutação e o número de elite. O teste 1 usa uma probabilidade de cruzamento de 70%, uma probabilidade de mutação igual a 10% e um número de elite alto igual a 7. O teste 2 aumenta a probabilidade de cruzamento para 90%, o teste 3 diminui a probabilidade de mutação para 1%, o teste 4 diminui o número de elite para 3, o teste 5 aumenta a penalidade atribuída para caixas que violam as restrições de capacidade máxima e mínima. Por fim, o teste 6 não usa a configuração atual da rede, ou seja, ele assume que não existe caixas terminais instaladas.

teste	usa demanda instalada	$n_{gen}$	$n_{pop}$	$p_{cross}$	$p_{mut}$	$n_{elit}$	$\alpha$
1	sim	5000	50	0,7	0,1	7	1
2	sim	5000	50	0,9	0,1	7	1
3	sim	5000	50	0,9	0,01	7	1
4	sim	5000	50	0,9	0,01	3	1
5	sim	5000	50	0,9	0,01	7	10
6	não	5000	50	0,9	0,01	7	1

Tabela 5.4: Plano de testes para AGPC.

A seguir apresentamos os resultados obtidos, aplicando os casos de testes nas quatro instâncias. Como o algoritmo genético é uma heurística com escolhas probabilísticas, cada teste foi realizado cinco vezes, e o resultado apresentado é a média das cinco evoluções.

Os dados que apresentamos nas tabelas de resultados são: tempo de execução do algoritmo em minutos, o número total de caixas terminais, o número de caixas terminais que têm capacidade 10 e 20, o número de caixas terminais novas, o custo total do projeto, o limitante inferior para este custo, porcentagem da solução obtida pelo AGPC em relação ao limitante e dados numéricos sobre a distribuição de demanda entre as caixas de 10 e 20, como: demandas mínima e máxima. Os custos são dados em reais.

### Resultado dos testes para a instância 1

A tabela 5.5 lista os resultados obtidos pelo AGPC, realizando os testes da tabela 5.4, aplicado à instância 1 da tabela 5.3.

Dos resultados da tabela 5.5 vemos que a diferença de custo entre os testes que consideram a rede existente é bem pequena, e o teste que obteve o melhor custo foi o teste com probabilidade de cruzamento alta e de mutação baixa. Sendo que o custo obtido por este teste é um pouco mais do que 35% do custo ótimo. O número de caixas novas que o algoritmo alocou foi apenas

teste	tempo (min)	num. caixas	caixas novas	cx. 10	cx. 20	custo total	limit. inf.	% limit.	demanda			
									min10	max10	min20	max20
1	8	71	5	37	34	477,06	336,09	141,94	1,26	9,12	2,00	17,84
2	7	72	5	38	34	480,20	336,09	142,87	1,42	9,02	2,42	18,00
3	7	72	5	38	34	456,05	336,09	135,69	1,34	9,00	2,00	18,02
4	7	72	5	38	34	459,56	336,09	136,73	1,26	9,08	2,00	18,00
5	8	72	5	38	34	520,79	336,09	154,95	3,00	8,20	3,08	16,10
6	10	34	34	5	29	4.142,67	3.960,00	104,61	6,36	7,96	12,58	17,44

Tabela 5.5: Resultados para a instância 1.

5. Como inicialmente, esta instância tinha 68 caixas instaladas, concluímos que uma caixa foi removida, e cinco novas instaladas. A demanda entre as caixas não ficou muito balanceada devido ao fato das caixas existentes estarem desbalanceadas. Como o custo para remoção de caixas antigas e transferência de assinantes é muito grande, a preferência foi deixar a rede desbalanceada, como está. Aumentando o coeficiente de penalidade (teste 5), o custo da solução aumenta um pouco, mas não ocorre muita violação das restrições de capacidade.

Já o teste 6 mostra que o número de caixas necessário para atender a demanda desta área é muito menor do que já existe atualmente, e o custo obtido por este teste é um pouco mais do que 4% de um limitante do custo ótimo. E ele também conseguiu uma distribuição de demanda entre as caixas mais uniforme, não resultando em nenhuma caixa abaixo da capacidade mínima. As figuras 5.8 e 5.9 ilustram a distribuição de demanda entre as caixas de capacidade 10 e 20 respectivamente, de uma solução obtida com o teste 6.

A seguir apresentamos alguns gráficos da melhor solução, obtida do teste 3. A figura 5.10 ilustra o gráfico de evolução do AGPC durante as gerações. O eixo das abscissas representa as gerações, e o eixo das ordenadas o custo da melhor solução. Deste gráficos concluímos que o algoritmo obteve a melhor solução por volta da geração 1400.

O gráfico 5.11 ilustra o custo das caixas terminais, e o custo da fiação. O primeiro custo inclui o custo de instalação de novas caixas e remoção de caixas antigas. O segundo custo inclui o custo de ligação de pontos novos a caixas e o custo de transferência de pontos. Podemos ver neste gráfico que obtivemos uma melhora em ambos os custos. Isto se deve a um maior aproveitamento das caixas antigas e uma minimização da distância dos pontos às caixas.

As figuras 5.12 e 5.13 ilustram a distribuição de demanda entre as caixas, com capacidade 10 e 20, respectivamente. O eixo das abscissas representa as caixas e o eixo das ordenadas representa suas demandas. Os histogramas em cor cinza escuro representam caixas antigas e os em cor cinza claro representam caixas novas. Os histogramas pretos sob os histogramas cinza escuro, representam a demanda atual da rede, ou seja, a demanda que a caixa está atendendo atualmente.

A figura 5.14 ilustra a associação dos pontos às caixas terminais, e o posicionamento dessas

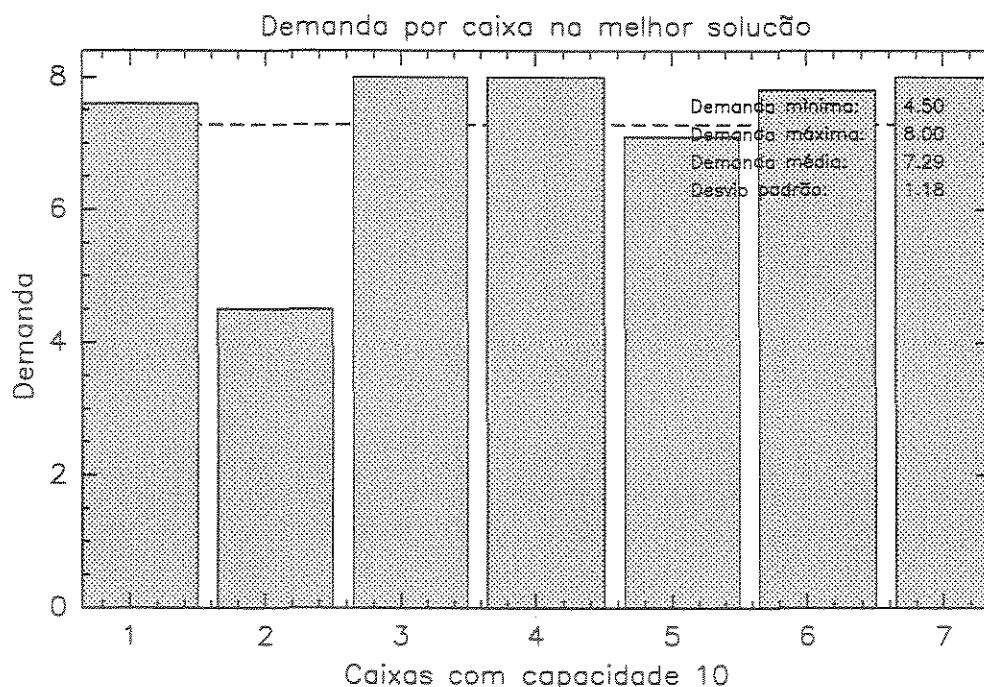


Figura 5.8: Instância 1: Distribuição de demanda nas caixas com capacidade 10.

caixas. Os círculos pretos nesta figura, representam os postes, os pontos vermelhos são postes que não podem receber instalação de caixas. Os textos acima de postes pretos, indicam as caixas terminais nele instaladas. Estes textos seguem o seguinte padrão: se for caixa antiga temos  $O/C/D$ , onde  $O$  indica que a caixa é antiga,  $C$  é a capacidade da caixa e  $D$  é a demanda que ela atende; se for uma caixa antiga o texto é dado simplesmente por  $C/D$ . Os pontos de demanda estão coloridos, e diferentes cores indicam diferentes caixas. Existe uma aresta ligando o ponto de demanda ao poste no qual a caixa que o atende está instalada.

### Resultado dos testes para a instância 2

Esta instância é um tanto particular, pois ela já possui muitas caixas terminais instaladas, e com pouca demanda. Pela tabela 5.3, podemos ver que esta instância possui 96 caixas instaladas que atendem uma demanda de somente 424.

A tabela 5.6 apresenta os resultados obtidos realizando os testes da tabela 5.4 para esta instância.

A tabela mostra que o teste, dentre os que não consideram a rede existente, que obteve o melhor resultado também foi o teste 3, aplicando uma taxa de mutação baixa. Este teste foi o que resultou em mais caixas terminais, isto porque não compensa remover caixas existentes e sim aproveitá-las. O algoritmo alocou 9 caixas novas, sendo 8 caixas de capacidade 10 e 1 caixa de capacidade 20. Podemos observar que a solução obtida por este teste é muito boa, visto

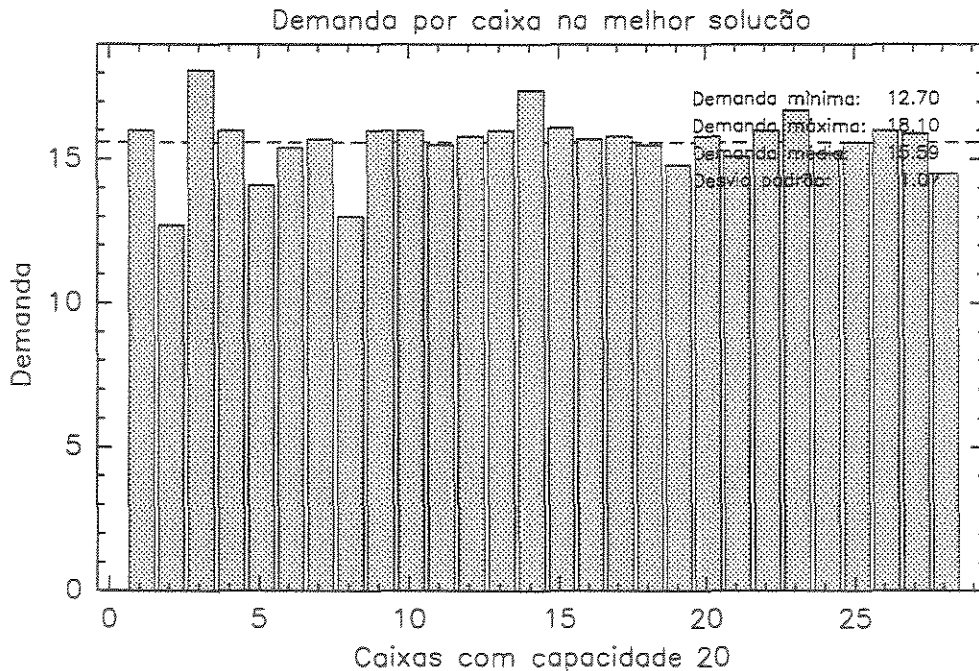


Figura 5.9: Instância 1: Distribuição de demanda nas caixas com capacidade 20.

que o seu custo está a menos do que 7% do custo ótimo. O teste 5 mostra que aumentando a taxa de penalidade o custo aumenta um pouco, mas nenhuma caixa viola a restrição de capacidade máxima. O custo aumenta, porque às vezes é necessário fazer transferência de pontos de demanda.

Novamente, o teste 6, que não considera a rede existente, mostrou que o número de caixas necessário para atender a demanda desta área é bem menor do que o existente. As figuras 5.15 e 5.16 ilustram a distribuição de demanda entre as caixas de capacidade 10 e 20 respectivamente, de uma solução obtida com o teste 6.

A seguir apresentamos os gráficos da melhor solução obtida pelo teste 3. O gráfico de evolução do AGPC está ilustrado na figura 5.17. Este gráfico mostra que a melhor solução encontrada foi antes da geração de número 1000.

As figuras 5.18 e 5.19 ilustram a distribuição de demanda entre as caixas, com capacidade 10 e 20, respectivamente. Estes gráficos mostram que a demanda não está muito bem distribuída entre as caixas, isto é devido ao grande número de caixas existentes.

A figura 5.20 ilustra a associação dos pontos em caixas terminais, e o posicionamento de caixas.

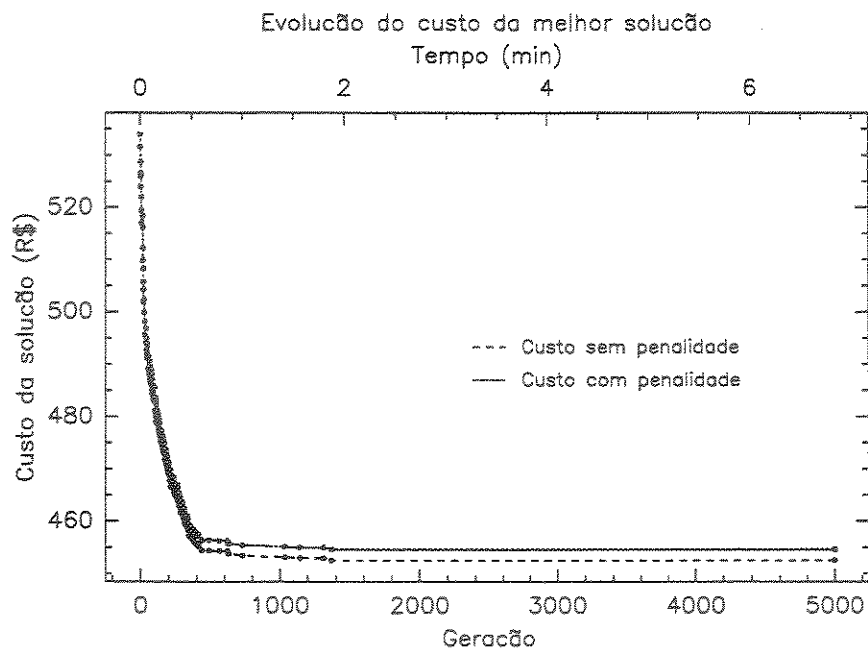


Figura 5.10: Instância 1: Gráfico de evolução do AGPC. Custo x Geração.

teste	tempo (min)	num. caixas	caixas novas	cx. 10	cx. 20	custo total	limit. inf.	%	demanda			
									min10	max10	min20	max20
1	12	94	9	49	45	916,53	779,83	117,52	1,00	9,38	1,26	17,40
2	16	94	9	49	45	953,37	779,83	122,25	1,00	9,30	1,18	17,66
3	9	98	9	51	47	830,42	779,83	106,49	1,00	9,50	1,34	17,00
4	9	97	9	51	46	863,21	779,83	110,70	1,00	9,50	1,26	18,10
5	11	98	9	51	47	914,72	779,83	117,30	1,00	7,90	1,54	15,82
6	18	47	47	20	27	4.947,34	4.645,64	106,50	1,60	8,00	11,38	18,24

Tabela 5.6: Resultados para a instância 2.

### Resultado dos testes para a instância 3

Para esta instância o AGPC tem maior liberdade para agir, pois ela possui poucas caixas terminais (9) atendendo poucos pontos.

A tabela 5.7 apresenta os resultados obtidos na execução dos testes da tabela 5.4 para esta instância.

Esta tabela mostra que o melhor resultado foi obtido com o teste número 4, usando uma quantidade pequena de elite. A solução obtida com a execução deste teste possui um custo que é menos do que 5% do custo de um limitante para o ótimo. Nesta instância observamos também que demanda mínima e a média das caixas aumentou. O teste 6, que não considera a rede existente, obteve resultados semelhantes aos dos outros testes, quanto ao número de caixas



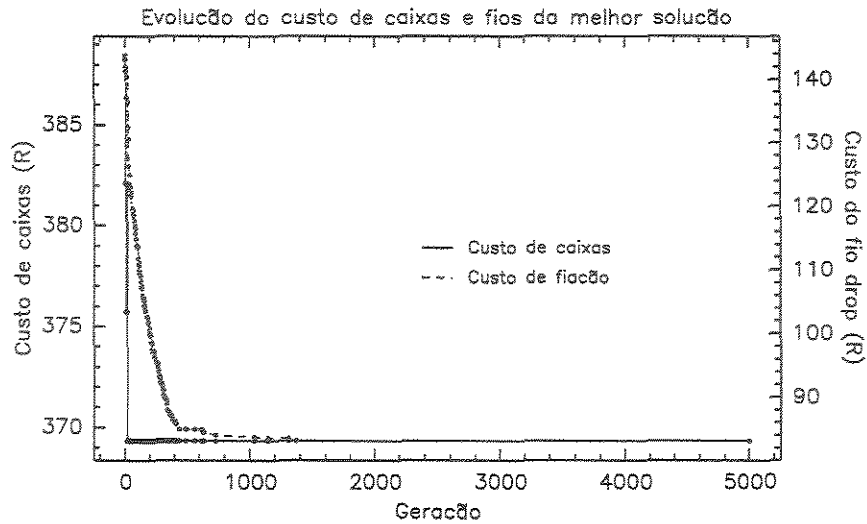


Figura 5.11: Instância 1: Custo de caixas e fiação.

teste	tempo (min)	num. caixas	caixas novas	cx. 10	cx. 20	custo total	limit. inf.	% limit.	demanda			
									min10	max10	min20	max20
1	8	34	25	11	23	2.664,50	2428,17	109,73	4,56	7,78	5,10	19,20
2	7	33	25	10	23	2.686,72	2428,17	110,65	4,14	7,94	5,90	18,82
3	7	33	24	10	23	2.569,92	2428,17	105,83	3,92	7,74	7,46	19,20
4	8	33	24	10	23	2.547,11	2428,17	104,90	4,22	8,00	7,86	19,02
6	11	33	33	11	22	3.603,00	3356,18	107,35	5,06	7,90	9,80	17,74

Tabela 5.7: Resultados para a instância 3.

terminais instaladas, porém conseguiu uma distribuição de demanda entre as caixas um pouco melhor, como pode ser observado nas figuras 5.21 e 5.22.

Para esta instância apresentamos os gráficos obtidos na execução do teste 4, que obteve o melhor resultado, lembrando que o resultado apresentado na tabela 5.7 é a média de 5 execuções de cada teste.

A figura 5.23 ilustra o gráfico de evolução do AGPC durante as gerações. A melhor solução para esta execução foi encontrada por volta da geração 3000.

As figuras 5.24 e 5.25 ilustram a distribuição de demanda entre as caixas, com capacidade 10 e 20, respectivamente. Nestes gráficos verificamos que nenhuma caixa viola a restrição de capacidade mínima, e as caixas de 10 não violam a capacidade máxima. A distribuição de demanda entre as caixas está muito mais balanceada do que nos testes apresentados para as instâncias anteriores.

A figura 5.26 ilustra a associação dos pontos em caixas terminais, e o posicionamento dessas caixas.

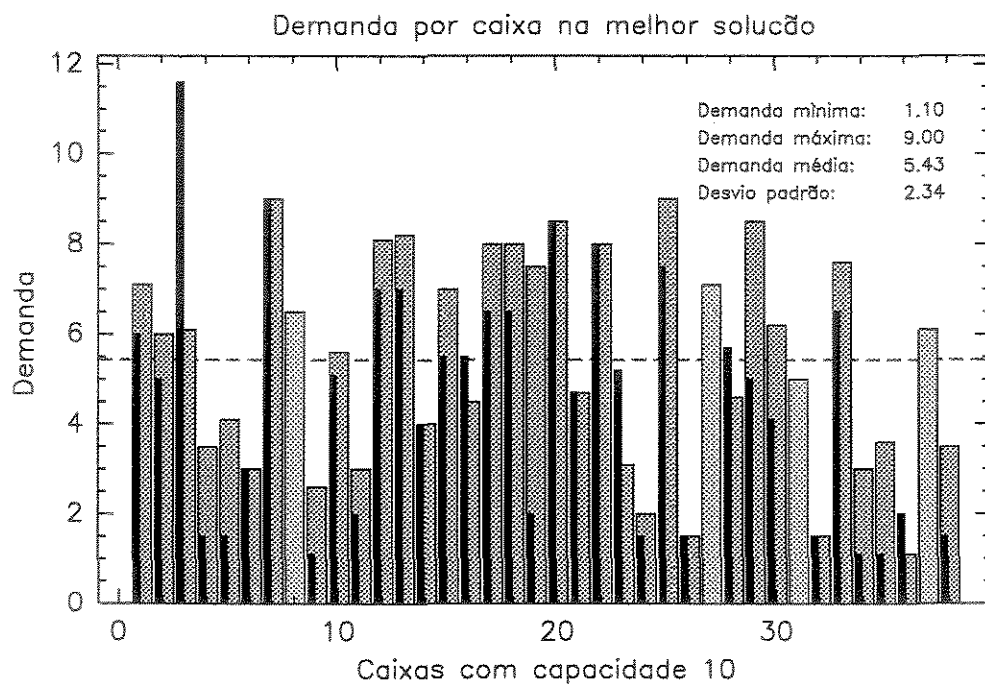


Figura 5.12: Instância 1: Distribuição de demanda nas caixas com capacidade 10.

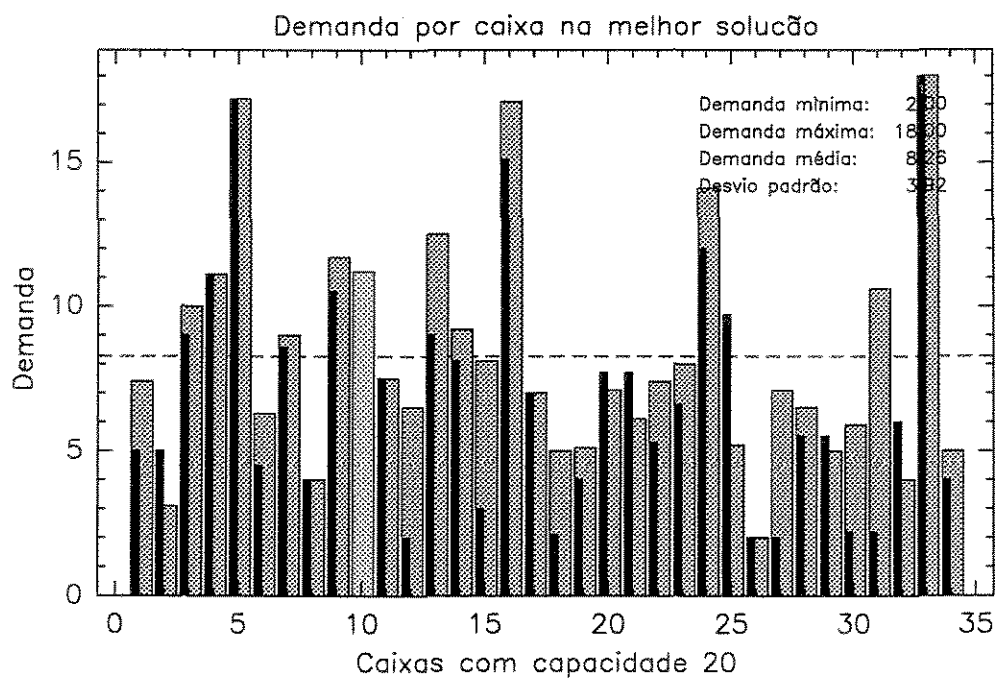
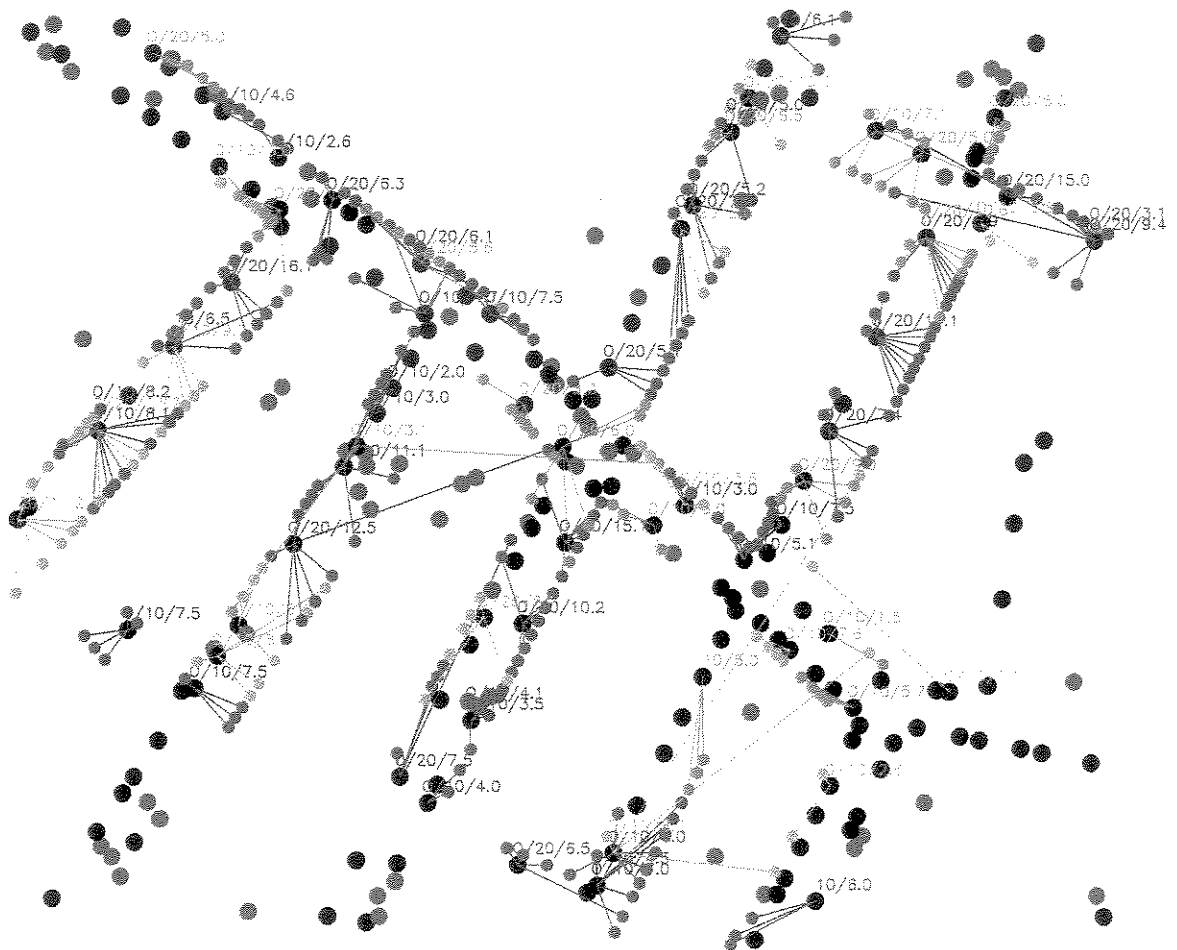


Figura 5.13: Instância 1: Distribuição de demanda nas caixas com capacidade 20.



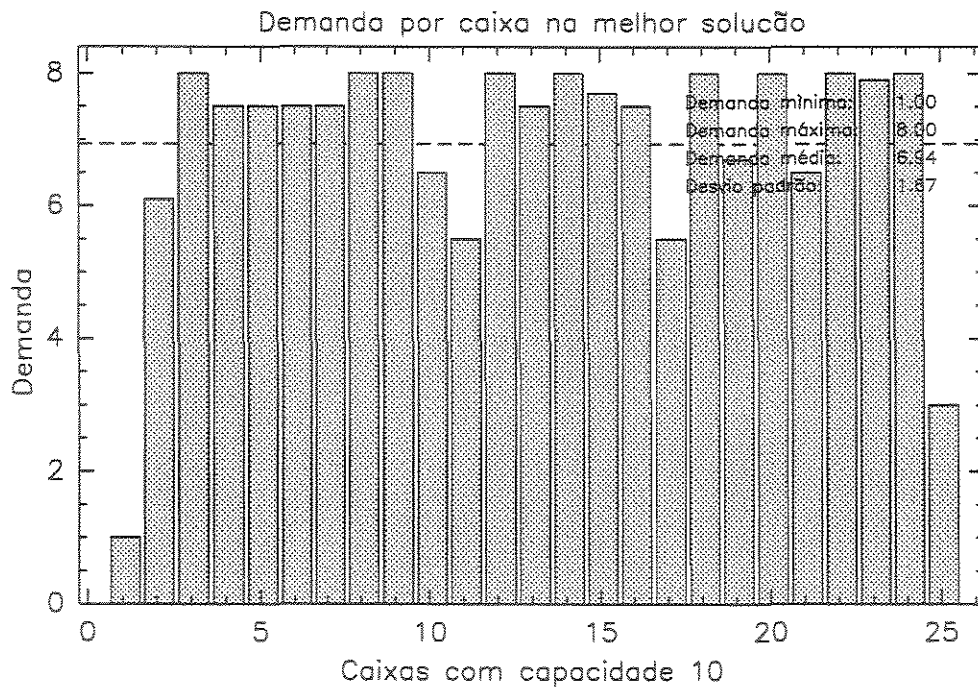


Figura 5.15: Instância 2: Distribuição de demanda nas caixas com capacidade 10.

#### Resultado dos testes para a instância 4

Esta instância também proporciona ao AGPC uma liberdade maior para agir, por possuir um número relativamente pequeno de caixas (21) já instaladas.

A tabela 5.8 apresenta os resultados obtidos na execução dos testes da tabela 5.4 para esta instância.

teste	tempo (min)	num. caixas	caixas novas	cx. 10	cx. 20	custo total	limit. inf.	%	demanda			
									min10	max10	min20	max20
1	10	43	22	16	27	2.542,33	2243,18	113,33	5,72	9,38	3,20	19,12
2	8	43	22	15	28	2.597,17	2243,18	115,78	5,02	9,62	3,42	17,88
3	10	42	21	16	26	2.442,91	2243,18	108,90	5,40	9,66	2,58	19,62
4	10	41	21	15	26	2.463,63	2243,18	109,82	5,68	9,90	3,26	18,96
6	13	35	35	7	28	4.209,38	3974,90	105,90	6,26	7,96	13,40	17,06

Tabela 5.8: Resultados para a instância 4.

Esta tabela mostra que o melhor resultado foi obtido com o teste número 3, usando probabilidade de mutação pequena. Porém a diferença nas soluções obtidas por cada teste não é muito significativa. O teste 6, que não considera a rede existente, obteve um resultado um pouco diferente, com menos caixas terminais. A distribuição de demanda entre as caixas, para uma solução obtida com este teste, pode ser visualizada nas figuras 5.27 e 5.28.

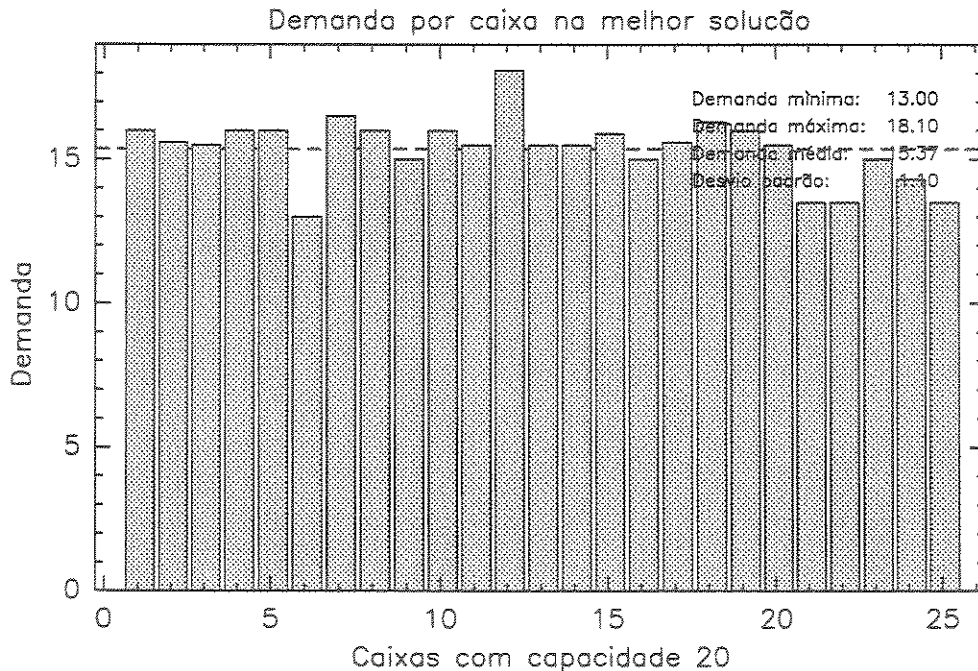


Figura 5.16: Instância 2: Distribuição de demanda nas caixas com capacidade 20.

Para esta instância apresentamos os gráficos obtidos na execução do teste 3, que obteve o melhor resultado, com um custo a 8,9% do ótimo. A figura 5.29 ilustra o gráfico de evolução do AGPC durante as gerações. A melhor solução para esta execução foi encontrada antes da geração 3000.

As figuras 5.30 e 5.31 ilustram a distribuição de demanda entre as caixas, com capacidade 10 e 20, respectivamente. A distribuição de demanda entre as caixas de capacidade 10 está bem balanceada, porém existem algumas caixas de capacidade 20 que atendem pouca demanda.

A figura 5.32 ilustra a associação dos pontos em caixas terminais, e o posicionamento de caixas.

### 5.3.5 Conclusão dos resultados

Com os testes aplicados a estes 4 tipos de instâncias, concluímos que quando existe um grande número de caixas terminais já instaladas, mais do que o necessário, o algoritmo não consegue obter soluções boas no sentido de balancear a demanda das caixas. Ele tenta aproveitar as caixas existentes, pois para obedecer a regra de balanceamento de caixas, várias caixas deveriam ser retiradas e com isso muitas transferências feitas. Por causa disso o custo do projeto fica muito caro, compensando deixar caixas subutilizadas.

Porém, para instâncias com poucas caixas o algoritmo conseguiu obter soluções melhores, com custos muito perto do limitante para o custo ótimo e com demanda balanceada entre caixas

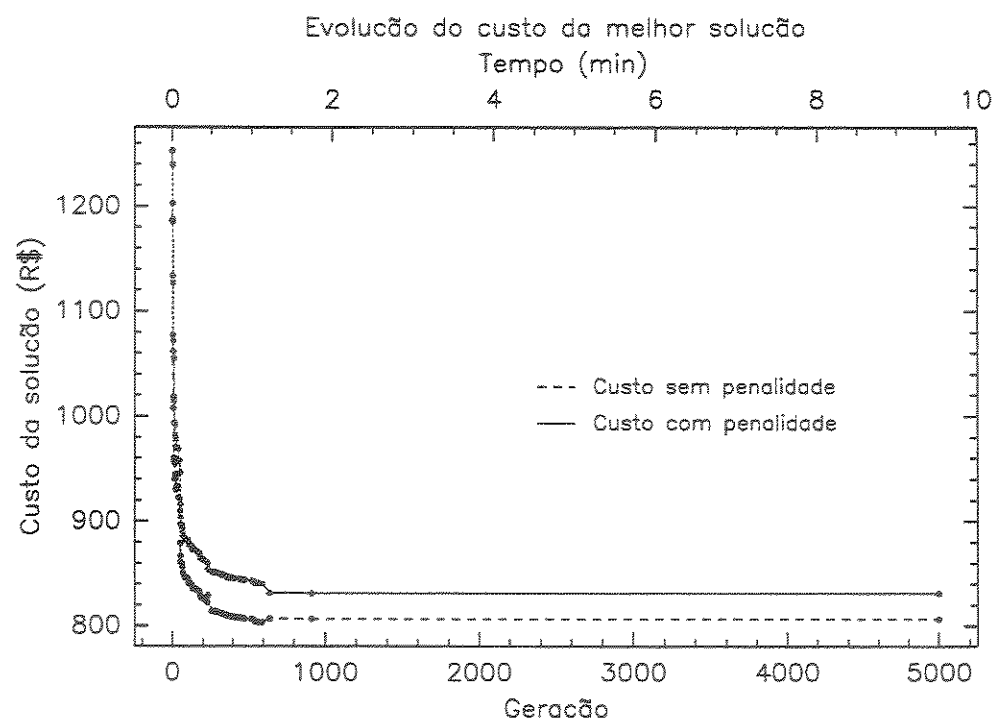


Figura 5.17: Instância 2: Gráfico de evolução do AGPC. Custo x Geração.

próximas.

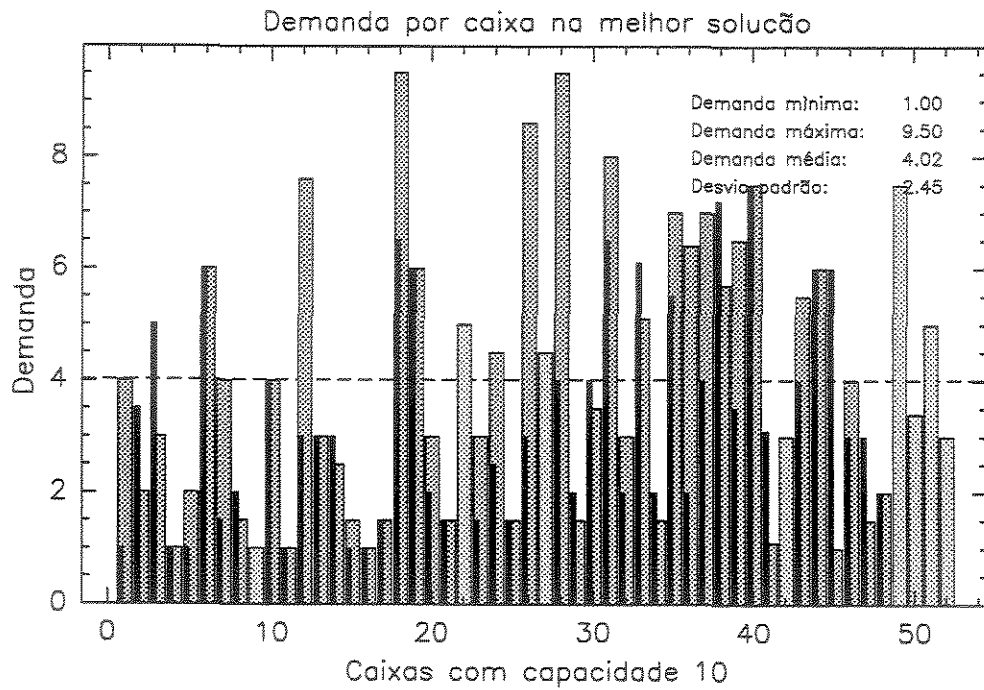


Figura 5.18: Instância 2: Distribuição de demanda nas caixas com capacidade 10.

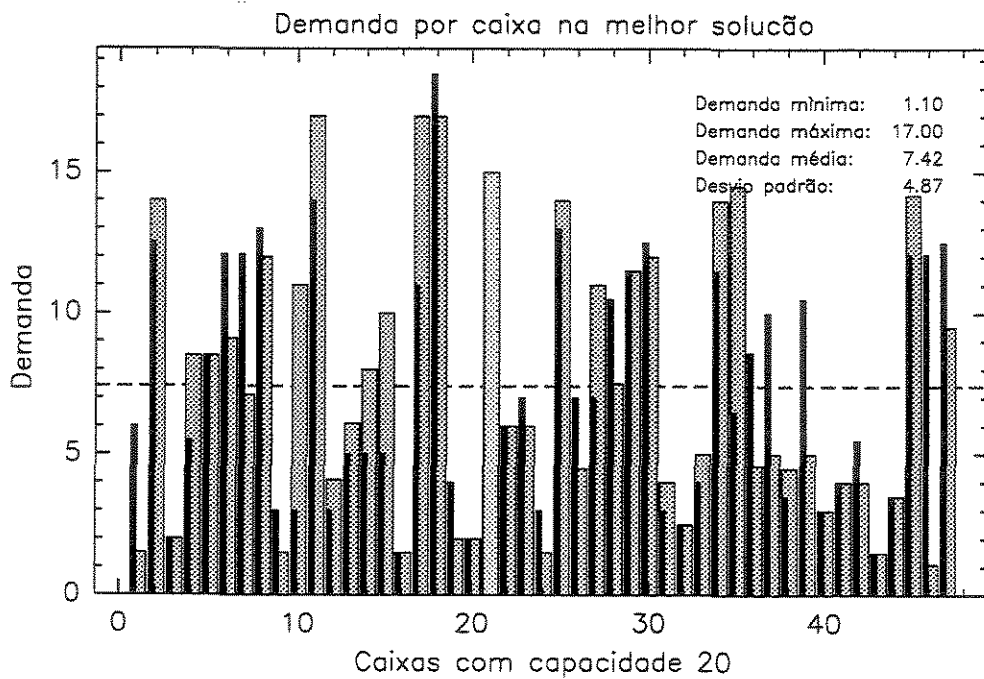


Figura 5.19: Instância 2: Distribuição de demanda nas caixas com capacidade 20.





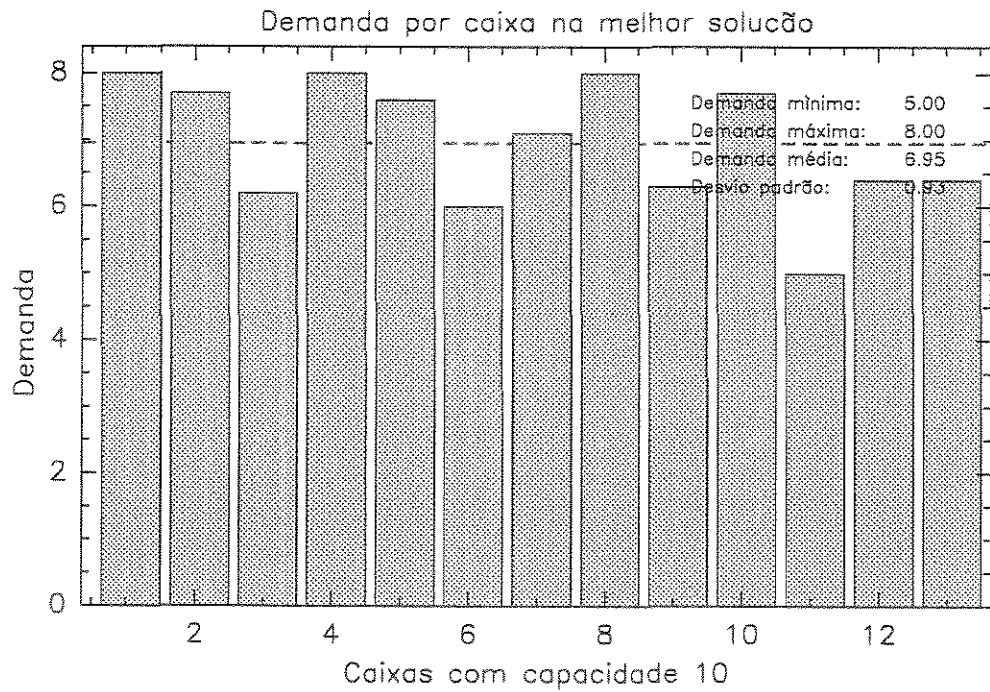


Figura 5.21: Instância 3: Distribuição de demanda nas caixas com capacidade 10.

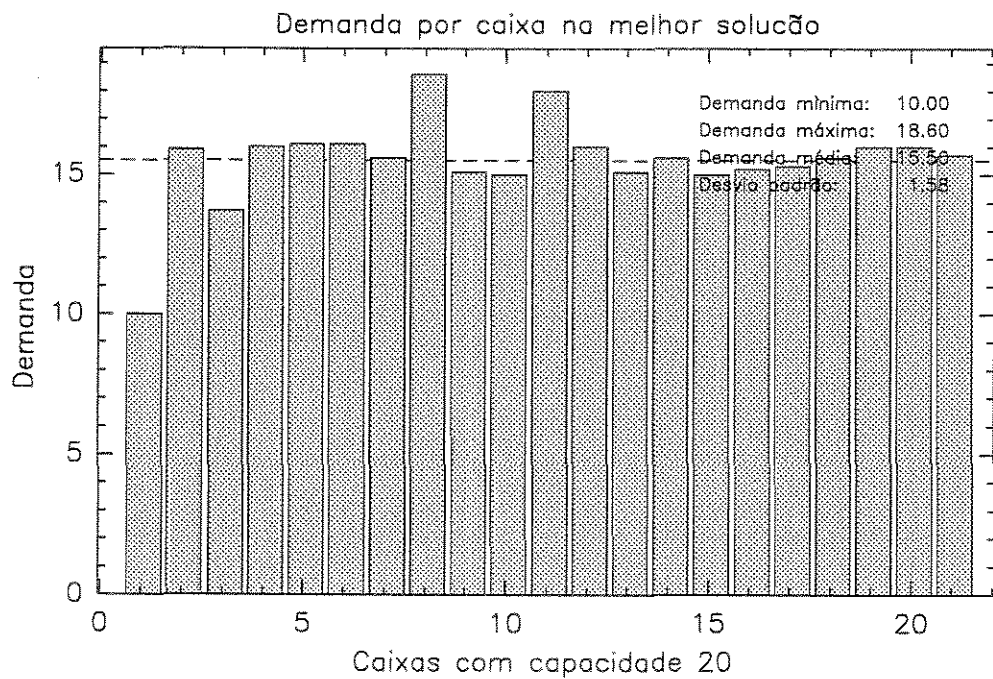


Figura 5.22: Instância 3: Distribuição de demanda nas caixas com capacidade 20.

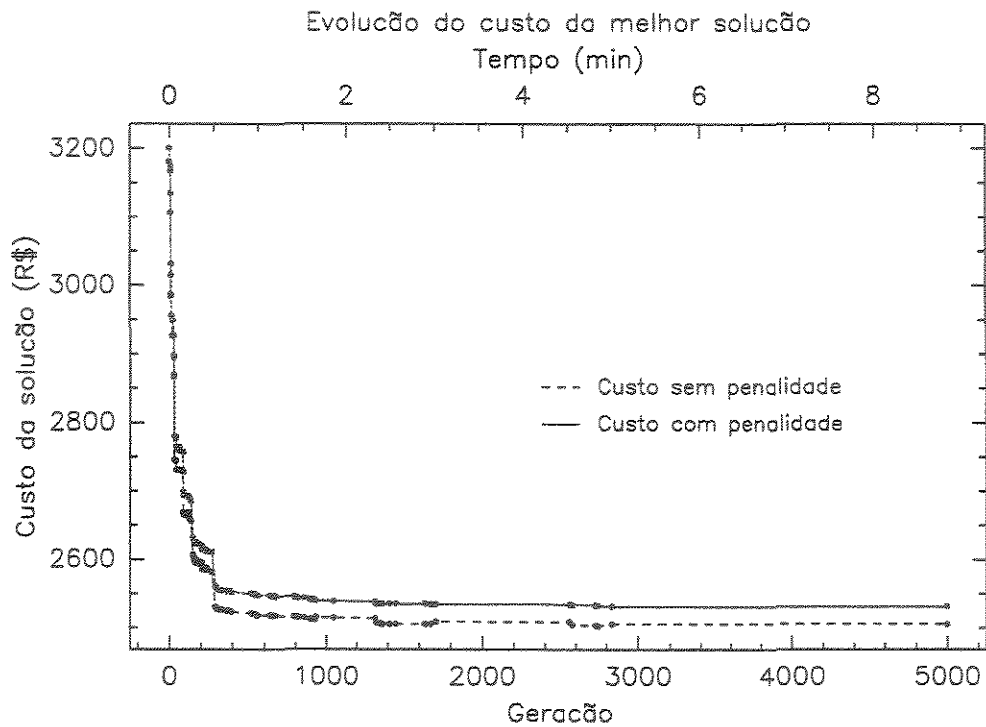


Figura 5.23: Instância 3: Gráfico de evolução do AGPC. Custo x Geração.

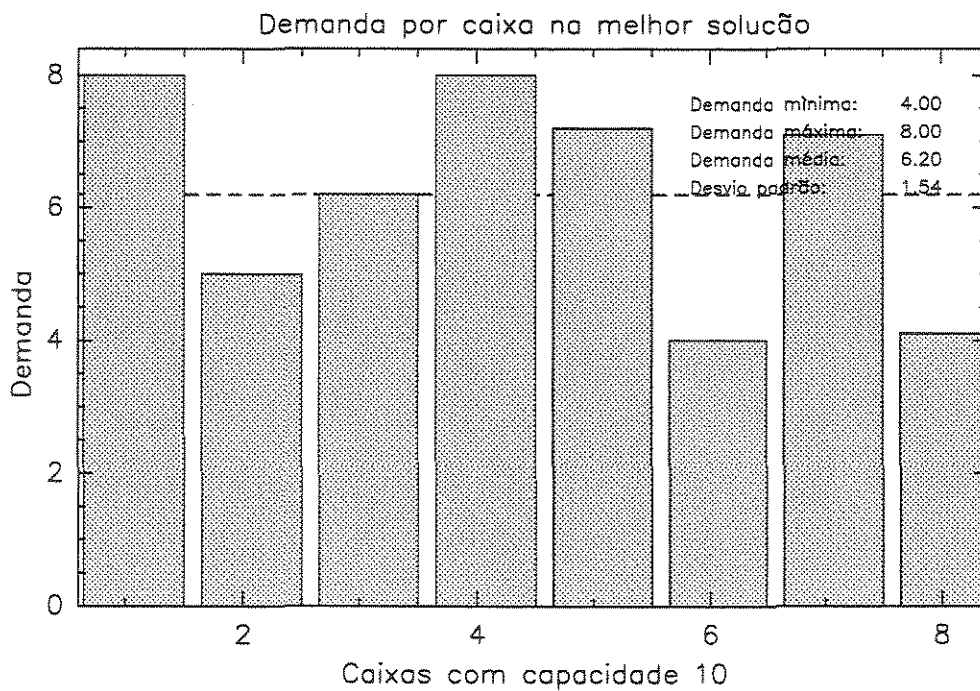


Figura 5.24: Instância 3: Distribuição de demanda nas caixas com capacidade 10.

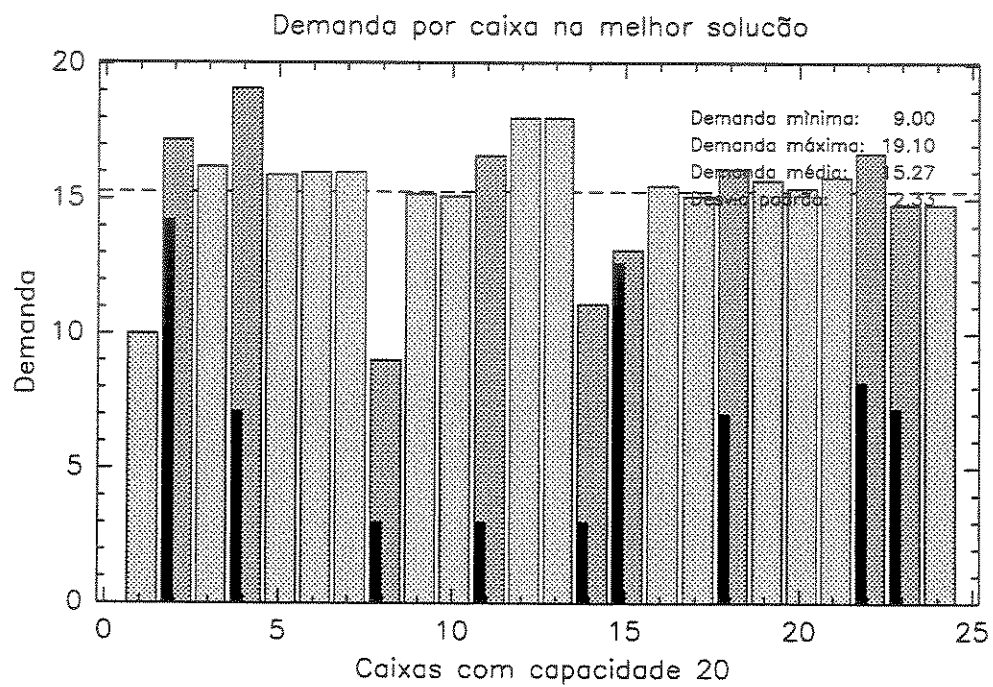


Figura 5.25: Instância 3: Distribuição de demanda nas caixas com capacidade 20.



Figura 5.26: Instância 3: Solução.

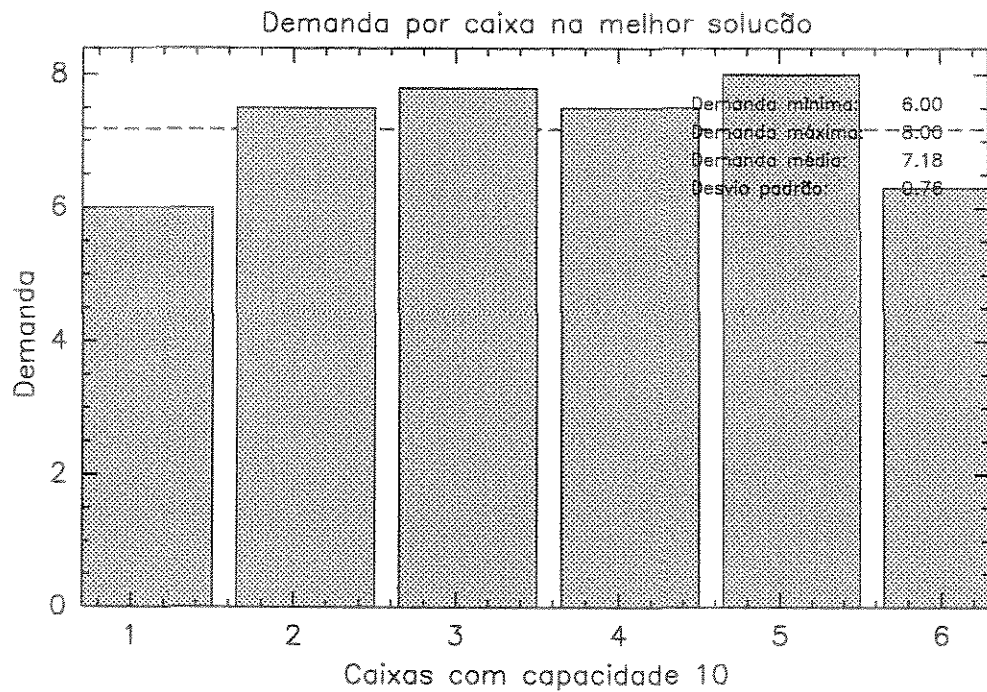


Figura 5.27: Instância 4: Distribuição de demanda nas caixas com capacidade 10.

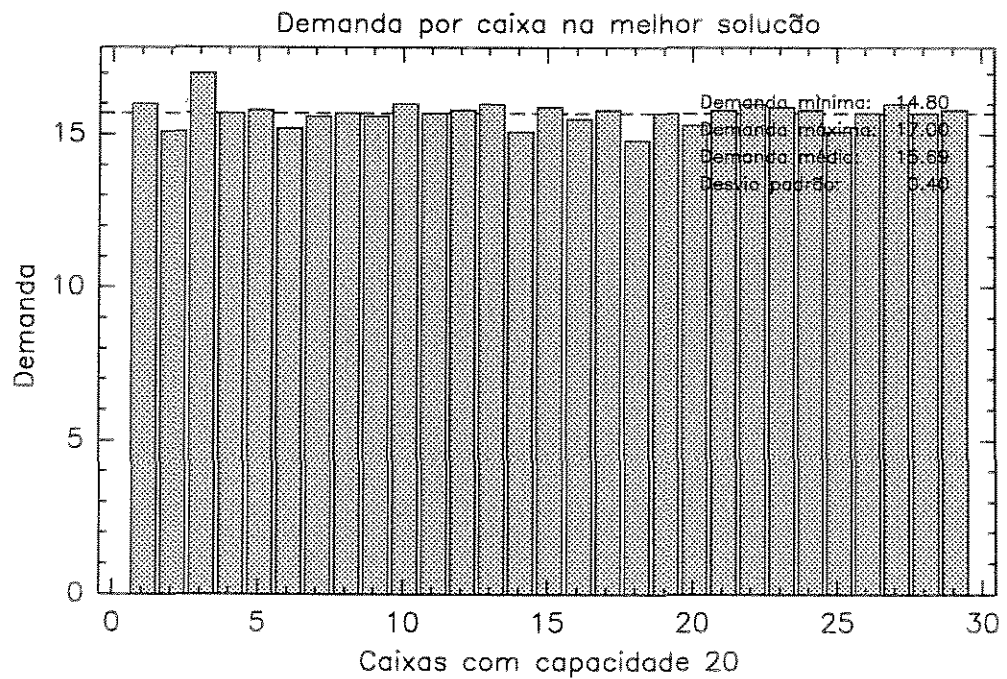


Figura 5.28: Instância 4: Distribuição de demanda nas caixas com capacidade 20.

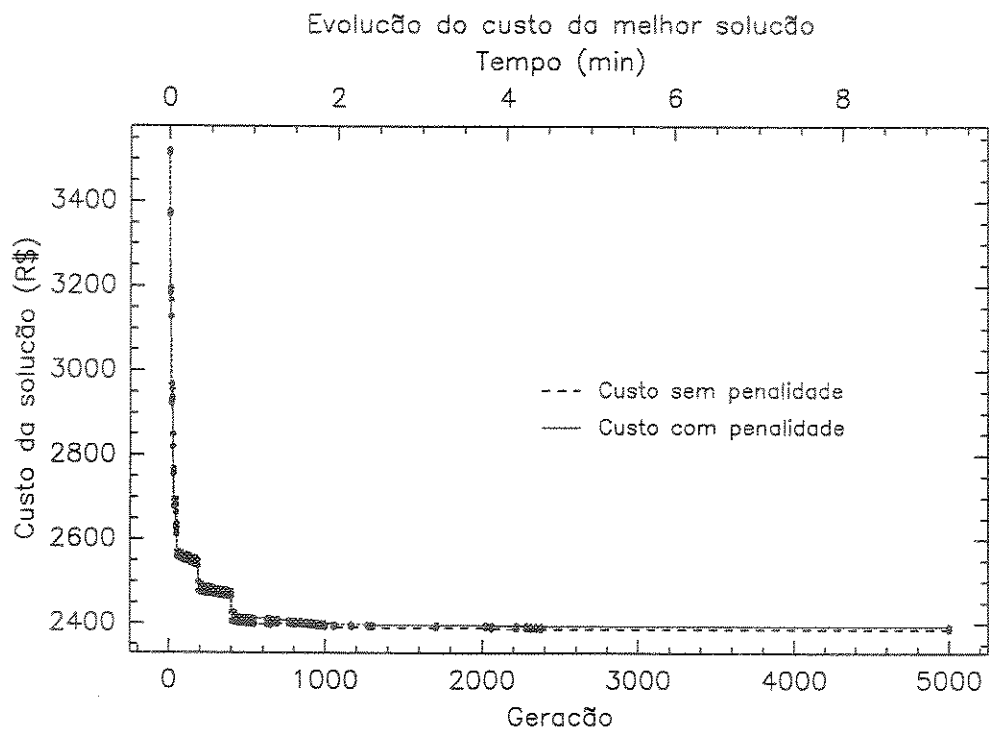


Figura 5.29: Instância 4: Gráfico de evolução do AGPC. Custo x Geração.

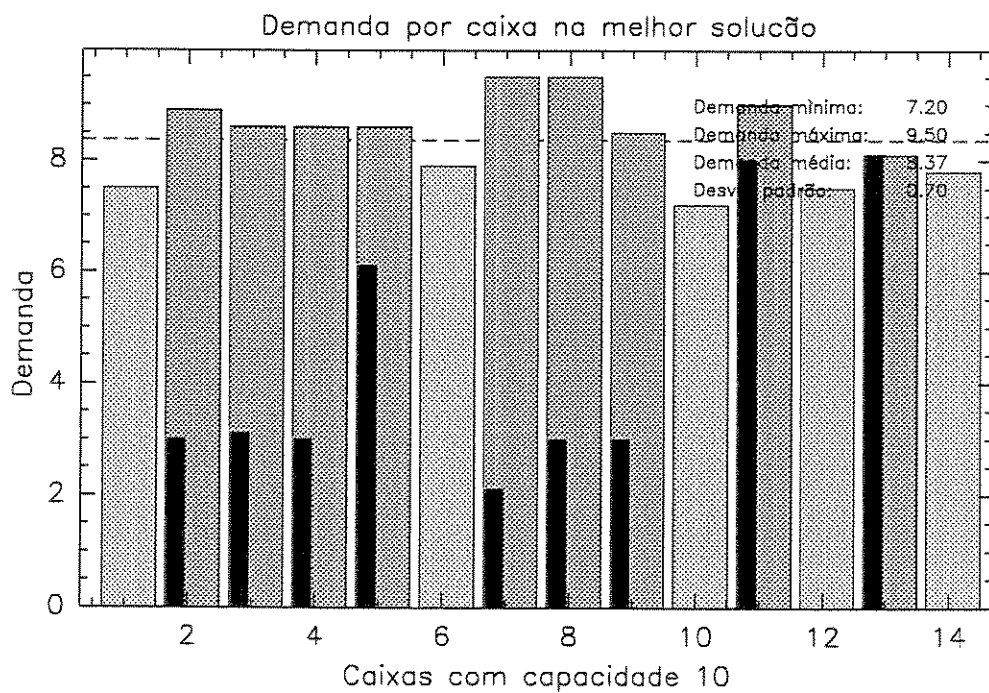


Figura 5.30: Instância 4: Distribuição de demanda nas caixas com capacidade 10.



## Capítulo 6

### Conclusão

Desenvolvemos AGs para resolver um problema de otimização complexo que está inserido no contexto de projeto de rede de telecomunicações.

O primeiro algoritmo foi desenvolvido para minimizar o custo total do projeto e produzir uma distribuição de demanda entre as várias seções de serviço tão uniforme quanto possível. É importante mencionar, que a princípio não são conhecidos o número de seções de serviço, nem o conjunto de pontos de demanda atendido por uma seção e nem a posição dos equipamentos. Outro aspecto importante do algoritmo é sua habilidade em lidar com informações geométricas que estão presentes nos dados de entrada. Com isto precisamos desenvolver novos operadores de cruzamento e mutação, enquanto outras aproximações para problemas similares não tratam este aspecto explicitamente. Todos os casos de teste são dados reais, tomados de vizinhanças urbanas de uma grande cidade. O algoritmo obteve bons resultados para todos os casos de teste, com tempos computacionais aceitáveis. A distribuição de demanda entre as seções foi bem uniforme e, o número de seções obtido pelo algoritmo foi sempre o mínimo.

Os resultados obtidos por este algoritmo também foram avaliados por engenheiros experientes no projeto de rede de telecomunicação. Eles também concluíram que o algoritmo pode ser uma técnica promissora para este problema, automatizando um trabalho que se feito por engenheiros de projeto pode levar cerca de seis meses para completar um projeto de definição de seções de serviço em uma área de projeto não muito grande.

O segundo algoritmo foi desenvolvido para remodelar a rede existente para atender demandas futuras que surgem ao longo do tempo, visando sempre, a minimização do custo das operações. Este algoritmo obteve bons resultados, como pode ser visto na comparação dos resultados com um limitante para o valor ótimo.

Para instâncias que possuem características semelhantes às restrições do projeto, o algoritmo obteve soluções muito boas, respeitando as restrições. Já para as instâncias onde o número de caixas instaladas é grande, e assim violando as restrições de capacidade mínima e balanceamento de demanda, o algoritmo minimizou o custo total de apenas atender aos pontos



de demanda ainda não instalados.

Os resultados obtidos por este algoritmo também foram apresentados e avaliados por engenheiros de projeto de rede, concluindo também que o algoritmo executou de forma adequada perante as situações testadas.

Este trabalho vem a ser uma ferramenta promissora na automatização de um pré-projeto de rede telefônica, pois até onde sabemos, não existe uma ferramenta para ajudar neste planejamento. Antigamente, um projeto de rede de telecomunicações era realizado e estimado por um tempo mínimo de 5 anos. Hoje com as constantes mudanças e crescimento da população, a rede de telefonia sofre freqüentes modificações. Para atender este crescimento de forma rápida, as empresas remodelam a rede existente da forma mais fácil. Esta remodelagem, muitas vezes não considera a situação global da rede, considerando apenas regiões bem próximas da posição do ponto que está necessitando de mais demanda.

Agora, com a automatização deste processo, as empresas poderão redimensionar a rede de forma a manter as especificações desejadas. Além disso, este processo pode ser feito num tempo menor do que um processo manual levaria.

## 6.1 Trabalhos Futuros

Muito ainda pode ser feito para melhorar, projetar, planejar e expandir uma rede de telecomunicações. Um trabalho imediato que surge, da conclusão deste trabalho, é considerar a rede existente para remodelar a parte de definição de seções de serviço. Ou seja, considerar os armários de distribuição já instalados, os pontos que já estão sendo atendidos por um armário, a definição de seções existentes e pontos de demanda que devem ser alocados em novas seções, e assim criar novas seções a partir das existentes.

Outro trabalho posterior a este, é considerar o cabeamento dentro de uma seção de serviço. Pares de fios telefônicos partem do armário de distribuição e chegam até as caixas terminais, atendendo as demandas ao longo do seu percurso. Esses pares telefônicos são agrupados em feixes, ou cabos, de maior capacidade.

Dados a posição do armário de distribuição que possui uma capacidade máxima, as posições das caixas terminais, consideradas um conjunto de destinos com demandas a serem atendidas e os postes, este problema consiste em instalar cabos, em postes, com capacidades para rotear o fluxo necessário do armário até cada caixa terminal.

# Referências Bibliográficas

- [1] C. D. Almeida, N. Amarilla, and B. Barán. Optimización multiobjetivo en la planificación de centrales telefónicas, 2003. [www.cnc.una.py/invest/paper2/Paperalm2.pdf](http://www.cnc.una.py/invest/paper2/Paperalm2.pdf).
- [2] D. Beasley, D. R. Bull, and R. R. Martin. An overview of genetic algorithms: Part 1, fundamentals. *University Computing*, 15(2):58–69, 1993.
- [3] D. Beasley, D. R. Bull, and R. R. Martin. An overview of genetic algorithms: Part ii, fundamentals. *University Computing*, 15(4):170–181, 1993.
- [4] J.E. Beasley and P.C. Chu. A genetic algorithm for the set covering problem. *European Journal of Operational Research*, pages 392–404, 94 (1996).
- [5] Regina Helena Bastos Cabral. Um sistema para planejamento de redes telefônicas urbanas. Master's thesis, Instituto de Ciências Exatas da Universidade Federal de Minas Gerais, 1985.
- [6] David W. Corne, Martin J. Oates, and George D. Smith, editors. *Telecommunications Optimization: Heuristic and Adaptive Techniques*. Wiley, 2000.
- [7] Rômelo Amarildo Vasconcelos Costa. Uma heurística para o problema de localização de centros de fios - microlocus. Master's thesis, Instituto de Ciências Exatas da Universidade Federal de Minas Gerais, 1991.
- [8] A. B. Sirihal G. R. Mateus, H. P. L. Luna. Heuristics for distribution network design in telecommunication. *Journal of Heuristics*, 6(1), 2000.
- [9] Michael R. Garey and David S. Johnson. *Computers and Intractability – A Guide to the Theory of NP-Completeness*. Freeman, 1979.
- [10] J. H. Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, 1975.
- [11] Sami Khuri and Teresa Chiu. Heuristic algorithms for the terminal assignment problem. In *Selected Areas in Cryptography*, pages 247–251, 1997.

- [12] S. Livramento, A. V. Moura, F. K. Miyazawa, M. M. H, and R. A. Miranda. A genetic algorithm for telecommunication network design. In *Proceedings of EvoComNet. Lecture Notes in Computer Science*, volume 3005, pages 140–149, 2004.
- [13] G. R. Mateus and R. V. L. Franqueira. Model and heuristic for a generalized access network design problem. In *Telecommunication System*, pages 257–271, 1998.
- [14] G. R. Mateus, H. P. L. Luna, A. B. Sirihal, and E. F. Cota. Especificação do sistema configurador de rede de distribuição (cordis). Technical report, Instituto de Ciências Exatas, Departamento de Ciência da Computação - UFMG, 1993.
- [15] G. R. Mateus and J.-M. Thizy. Exact sequential choice of locations in a network. In *Annals of Operations Research*, volume 86, pages 199–219, 1999.
- [16] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, third, revised and extended edition edition, 1996.
- [17] Z. Michalewicz and D. B. Fogel. *How to solve it : modern heuristics*. Springer, 1999.
- [18] A. V. Moura, F. K. Miyazawa, S. Livramento, A. A. Santos, and N. S. Volpato. Posicionamento de caixas terminais considerando a rede existente. Quarto Relatório Semestral do Projeto Localização de Recursos em Rede Telefônica., 2004.
- [19] A. V. Moura, F. K. Miyazawa, S. Livramento, A. A. Santos, N. S. Volpato, and D. Ferber. Definição de seções de serviço e posicionamento de armários. Terceiro Relatório Semestral do Projeto Localização de Recursos em Rede Telefônica., 2004.
- [20] A. V. Moura, F. K. Miyazawa, S. Livramento, A. A. Santos, N. S. Volpato, and A. M. Passos. Definição de seções de serviço e posicionamento de armários. Segundo Relatório Semestral do Projeto Localização de Recursos em Rede Telefônica., 2003.
- [21] P. Ross, D. Corne, and H. L. Fang. Timetabling by genetic algorithms: Issues and aproaches. Technical report, Department of Artificial Intelligence, University of Edinburgh, 1994.