

**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO  
EM ENGENHARIA ELÉTRICA**

**Configuração de Sistemas de Distribuição  
usando um Algoritmo Genético Sequencial**

**Tese de Doutorado**

**Helon David de Macêdo Braz**

**Campina Grande – Paraíba – Brasil**

**Novembro de 2010**

Helon David de Macêdo Braz

# Configuração de Sistemas de Distribuição usando um Algoritmo Genético Sequencial

*Tese apresentada à Coordenação do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, em cumprimento às exigências para obtenção do Grau de Doutor em Ciências no Domínio da Engenharia Elétrica.*

Área de concentração: Processamento da Energia

Benemar Alencar de Souza, D. Sc.

orientador

Campina Grande – Paraíba – Brasil

Novembro de 2010



FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCC

B827c Braz, Helon David de Macêdo  
Configuração de sistemas de distribuição usando um algoritmo genético sequencial / Helon David de Macêdo Braz. – Campina Grande, 2010.  
236f. : il. ; col.

Tese (Doutorado em Engenharia Elétrica) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática.

Orientador: Prof. Dr. Benemar Alencar de Souza

Referências.

1. Configuração Ótima de Redes de Distribuição.
2. Algoritmos Genéticos.
3. Estratégias de Codificação.
4. Algoritmo Genético Sequencial.
5. Teoria dos Grafos. I. Título.

CDU 621.3 (043)

UFCC - BIBLIOTECA - CAMPUS I	
20-08-12	1732-2012
doação	

CONFIGURAÇÃO DE SISTEMAS DE DISTRIBUIÇÃO USANDO UM  
ALGORÍTMO GENÉTICO SEQUÊNCIAL

HELON DAVID DE MACEDO BRAZ

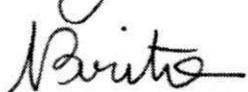
Tese Aprovada em 26.11.2010

  
BENEMAR ALENCAR DE SOUZA, D.Sc., UFCG  
Orientador

JOSÉ ROBERTO SANCHES MANTOVANI, Dr., UNESP  
Componente da Banca (Ausência Justificada)

  
ALEXANDRE CLAUDIO BOTELHO DELBEM, Dr., USP-ICMC  
Componente da Banca

  
ANTONIO MARCUS NOGUEIRA LIMA, Dr., UFCG  
Componente da Banca

  
NÚBIA SILVA DANTAS BRITO, D.Sc., UFCG  
Componente da Banca

CAMPINA GRANDE - PB  
NOVEMBRO - 2010

Aos meus pais, Severina e Walter,  
e às minhas irmãs, Helma e Érika,  
DEDICO.

## Agradecimentos

Primeiramente a Deus, pelo discernimento em perseverar neste trabalho de tese.

À família, pelo apoio contínuo e incondicional, sobretudo de minha mãe Severina.

À minha querida namorada Agnes, pelo incentivo, paciência e ajuda na revisão do texto.

Ao professor Benemar, pela orientação que teve início ainda na minha graduação.

Aos membros da banca examinadora, pelas valiosas contribuições ao trabalho.

À companheira de pesquisa Djane, pela companhia no tema e contribuições valiosas que resultaram no aperfeiçoamento deste trabalho.

Aos amigos da UFCG Franklin Pamplona, Kleber Silva, Ângelo Almeida, José Antonio, Gustavo e Luciano Sales, pelo companheirismo e amizade.

Aos colegas da CHESF Gustavo Melo, André, Fabiana, Josivan, Methodio, Oswaldo, Fernando, Leôncio, Hugo, Wilton, Glebson, Pablo, Karina, Aparecida, Luis Carlos, Marcelo Maia, Manfredo e Valéria, pela ajuda indispensável para superar o desafio de administrar tese e trabalho sobre temas disjuntos.

Aos amigos Paulo Coutinho, Márcio Araújo, Rodrigo Nóbrega, Wilson Rosas, Andrey Elísio, Luciano Rodrigues e Ronimack, pelo incentivo, confiança e amizade.

Aos funcionários Ângela, Suênia e Pedro da Coordenação de Pós-Graduação em Engenharia Elétrica da UFCG, pelo excelente serviço prestado.

E a tantos outros que participaram da realização deste trabalho, digo sinceramente,

MUITO OBRIGADO.

# Sumário

<b>LISTA DE FIGURAS.....</b>	<b>VIII</b>
<b>LISTA DE TABELAS.....</b>	<b>XIV</b>
<b>LISTA DE SÍMBOLOS .....</b>	<b>XVIII</b>
<b>RESUMO .....</b>	<b>XXVI</b>
<b>ABSTRACT.....</b>	<b>XXVII</b>
<b>INTRODUÇÃO.....</b>	<b>1</b>
<b>1.1 PRELIMINARES .....</b>	<b>1</b>
<b>1.2 APRESENTAÇÃO DO PROBLEMA .....</b>	<b>3</b>
<b>1.3 OBJETIVOS E CONTRIBUIÇÕES DA TESE .....</b>	<b>10</b>
<b>1.4 ORGANIZAÇÃO DO TEXTO .....</b>	<b>14</b>
<b>REVISÃO BIBLIOGRÁFICA.....</b>	<b>16</b>
<b>2.1 RECONFIGURAÇÃO DE REDES .....</b>	<b>17</b>
<b>2.2 RESTABELECIMENTO DE SERVIÇO.....</b>	<b>27</b>
<b>2.3 PLANEJAMENTO DE EXPANSÃO .....</b>	<b>30</b>
<b>2.4 SÍNTESE DOS TRÊS ENFOQUES DA CONFIGURAÇÃO.....</b>	<b>33</b>
<b>AGs PARA CONFIGURAÇÃO DE REDES.....</b>	<b>38</b>
<b>3.1 FUNDAMENTOS SOBRE AGs.....</b>	<b>38</b>
<b>3.2 CODIFICAÇÕES .....</b>	<b>41</b>
<b>3.2.1 Grafo de Setores.....</b>	<b>42</b>
<b>3.2.2 Um Exemplo Ilustrativo .....</b>	<b>45</b>
<b>3.2.3 Métricas de Comparação.....</b>	<b>48</b>
<b>3.2.4 Estado da Arte.....</b>	<b>51</b>
<b>3.3 CÁLCULO DE APTIDÃO .....</b>	<b>67</b>
<b>3.3.1 Horizonte de estudo.....</b>	<b>67</b>
<b>3.3.2 Formulação.....</b>	<b>68</b>

3.3.3 <i>Seleção dos Parâmetros</i> .....	73
3.3.4 <i>Memória Genética</i> .....	74
3.4 FLUXOGRAMA DO AG PROPOSTO .....	76
<b>CODIFICAÇÃO SEQUENCIAL SUBTRATIVA .....</b>	<b>79</b>
4.1 ALGORITMO PARA DETERMINAÇÃO DE MALHAS FUNDAMENTAIS.....	79
4.2 MATEMÁTICA DE MALHAS FUNDAMENTAIS.....	80
4.3 CODIFICAÇÃO.....	85
4.4 DECODIFICAÇÃO .....	86
4.5 OPERADORES GENÉTICOS .....	90
4.6 OTIMIZANDO O ALGORITMO E AS ESTRUTURAS .....	93
4.6.1 <i>Reduzindo o Mapa das Malhas</i> .....	93
4.6.2 <i>Uma versão vetorial para L</i> .....	95
4.7 MÉTRICAS DO CÓDIGO.....	97
4.7.1 <i>Espaço</i> .....	97
4.7.2 <i>Tempo</i> .....	97
4.7.3 <i>Factibilidade</i> .....	99
4.7.4 <i>Cobertura</i> .....	99
4.7.5 <i>Tendência</i> .....	99
4.7.6 <i>Localidade</i> .....	108
4.7.7 <i>Hereditariedade</i> .....	110
4.7.8 <i>Restrição</i> .....	112
4.7.9 <i>Hibridização</i> .....	113
<b>CODIFICAÇÃO SEQUENCIAL ADITIVA.....</b>	<b>114</b>
5.1 MATEMÁTICA DE SETORES.....	114
5.2 CODIFICAÇÃO.....	117
5.3 DECODIFICAÇÃO .....	118
5.4 OPERADORES GENÉTICOS .....	122
5.5 OTIMIZANDO O ALGORITMO E AS ESTRUTURAS .....	123
5.5.1 <i>Reduzindo o Mapa dos Setores</i> .....	123
5.5.2 <i>Uma versão vetorial para R</i> .....	125
5.6 MÉTRICAS DO CÓDIGO.....	127
5.6.1 <i>Espaço</i> .....	127
5.6.2 <i>Tempo</i> .....	127

5.6.3 <i>Factibilidade</i> .....	128
5.6.4 <i>Cobertura</i> .....	128
5.6.5 <i>Tendência</i> .....	129
5.6.6 <i>Localidade</i> .....	137
5.6.7 <i>Hereditariedade</i> .....	139
5.6.8 <i>Restrição</i> .....	140
5.6.9 <i>Hibridização</i> .....	141
<b>ALGORITMO GENÉTICO SEQUENCIAL.....</b>	<b>142</b>
6.1 <b>MOTIVAÇÃO E FUNDAMENTOS.....</b>	<b>142</b>
6.2 <b>FORMULAÇÃO DO ALGORITMO .....</b>	<b>144</b>
6.2.1 <i>Controle do Fator de Busca N</i> .....	145
6.2.2 <i>Repercussão nos algoritmos de decodificação</i> .....	147
6.2.3 <i>Heurísticas</i> .....	148
6.3 <b>MÉTRICAS DO ALGORITMO .....</b>	<b>151</b>
<b>RESULTADOS.....</b>	<b>156</b>
7.1 <b>DESCRIÇÃO DOS SISTEMAS TESTE.....</b>	<b>156</b>
7.2 <b>METODOLOGIA DO ESTUDO .....</b>	<b>158</b>
7.3 <b>TESTES COM O AG .....</b>	<b>162</b>
7.3.1 <i>Sistema de 16-nós</i> .....	163
7.3.2 <i>Sistema de 33-nós</i> .....	166
7.3.3 <i>Sistema de 70-nós</i> .....	168
7.3.4 <i>Sistema de 94-nós</i> .....	172
7.3.5 <i>Sistema de 137-nós</i> .....	177
7.3.6 <i>Síntese dos resultados - AG</i> .....	182
7.4 <b>TESTES COM O AGS .....</b>	<b>186</b>
7.4.1 <i>Heurísticas: N+, N-, I+ e I-</i> .....	187
7.4.2 <i>Controle do Fator de Busca N: Larguras Iguais e Proporcionais</i> .....	191
7.4.3 <i>Número de Patamares de N: 2, 3, 4 e 5</i> .....	194
7.4.4 <i>Síntese dos resultados - AGS</i> .....	199
<b>CONCLUSÕES E SUGESTÕES.....</b>	<b>201</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>205</b>
<b>APÊNDICE A.....</b>	<b>218</b>
A.1 <b>SISTEMA DE CINVANLAR ET AL (1988).....</b>	<b>218</b>

---

<b>A.2 SISTEMA DE BARAN &amp; WU (1989)</b> .....	219
<b>A.3 SISTEMA DE DAS (2006)</b> .....	220
<b>A.4 SISTEMA DE SU &amp; LEE (2003)</b> .....	223
<b>A.5 SISTEMA DE MANTOVANI ET AL (2000)</b> .....	225

## Lista de Figuras

Figura 1.1 – Resumo dos modelos elétricos adotados para Subestações, Barras e Ramos. .....	4
Figura 1.2 – Rede de Distribuição adotada como exemplo. ....	5
Figura 1.3 – Rede da Figura 1.2 após Reconfiguração: redução de perdas de 202.7kW para 153.5kW.....	5
Figura 1.4 – Rede da Figura 1.2 após atuação da proteção em decorrência de uma falta no nó 26: barras 26, 27, 28, 29, 30, 31 e 32 desenergizadas. ....	7
Figura 1.5 – Configuração obtida a partir da execução de um plano de Restabelecimento tomando a rede da Figura 1.4 como configuração inicial. ....	8
Figura 1.6 – Rede da Figura 1.2 após Planejamento: a) rede original, novos nós e arestas candidatas modeladas como ramos com chave e b) configuração proposta pelo Planejamento. ....	10
Figura 3.1 – Rede de Distribuição fictícia adotada como exemplo. ....	44
Figura 3.2 – Grafo de Setores: a) particionamento do sistema da Figura 3.1 e b) $G_S$ resultante. ....	45
Figura 3.3 – Grafo de Setores $G_S$ a ser usado nos exemplos da subseção 3.2.4.....	52
Figura 3.4 – Operador de cruzamento da codificação por vetor característico melhorado: acima os pais e os respectivos cromossomos; abaixo os filhos gerados, sendo um deles infactível. ....	60
Figura 3.5 – Exemplo dos operadores genéticos da codificação por Lista de Arestas: a) Cruzamento de $T_1$ e $T_2$ ; b) Mutação sobre $T_1$ .....	64
Figura 3.6 – Redução no tempo total de um AG no Planejamento de uma rede fictícia de 63 barras para diferentes ajustes de tamanho da memória genética.....	77
Figura 3.7 – Fluxograma do AG implementado. ....	78
Figura 4.1 – Malhas fundamentais obtidas pelo algoritmo descrito na Seção 4.1: a) rede malhada inicial; b-f) malhas fundamentais de 0 até 4.....	82

Figura 4.2 – Rede obtida pelo Algoritmo 4.1 decodificando o cromossomo exemplo de Subtrativa.....	90
Figura 4.3 – Comparação entre 5000 indivíduos gerados de forma aleatória utilizando as representações Vetor Característico – VC e Subtrativa – SUB: percentual em que cada aresta comparece com estado fechado nos indivíduos gerados.....	102
Figura 4.4 – Comparação entre 5000 indivíduos gerados de forma aleatória utilizando as representações Vetor Característico – VC e Subtrativa – SUB: percentual de soluções por profundidade máxima. ....	102
Figura 4.5 – Comparação entre 5000 indivíduos gerados de forma aleatória utilizando as representações Vetor Característico – VC e Subtrativa – SUB: percentual de soluções por faixa de aptidão. ....	103
Figura 4.6 – Evolução por faixa de aptidão para 1000 indivíduos com codificação Subtrativa implementando apenas cruzamento convencional, após as gerações: 1 (SUB_001), 50 (SUB_050), 100 (SUB_100) e 150 (SUB_150). Índices de Tendência no canto superior direito.....	104
Figura 4.7 – Evolução por profundidade máxima para 1000 indivíduos com codificação Subtrativa adotando cruzamento convencional, após as gerações: 1 (SUB_001), 50 (SUB_050), 100 (SUB_100) e 150 (SUB_150). Índices de Tendência no canto superior esquerdo.....	105
Figura 4.8 – Evolução por faixa de aptidão para 1000 indivíduos com codificação Subtrativa implementando apenas cruzamento na calda, após as gerações: 1 (SUB_001), 50 (SUB_050), 100 (SUB_100) e 150 (SUB_150). Índices de Tendência no centro da figura. ....	106
Figura 4.9 – Evolução por profundidade máxima para 1000 indivíduos com codificação Subtrativa adotando apenas cruzamento na calda, após as gerações: 1 (SUB_001), 50 (SUB_050), 100 (SUB_100) e 150 (SUB_150). Índices de Tendência no canto superior esquerdo.....	107
Figura 4.10 – Evolução por faixa de aptidão para 1000 indivíduos com codificação Subtrativa implementando apenas mutação convencional, após as gerações: 1 (SUB_001), 50 (SUB_050), 100 (SUB_100) e 150 (SUB_150). Índices de Tendência no canto superior direito.....	107
Figura 4.11 – Evolução por profundidade máxima para 1000 indivíduos com codificação Subtrativa adotando mutação convencional, após as gerações: 1 (SUB_001), 50 (SUB_050), 100 (SUB_100) e 150 (SUB_150). Índices de Tendência no canto superior esquerdo.....	109
Figura 4.12 – Evolução por faixa de aptidão para 1000 indivíduos com codificação Subtrativa implementando apenas mutação na calda, após as gerações: 1 (SUB_001), 50 (SUB_050), 100 (SUB_100) e 150 (SUB_150). Índices de Tendência no canto superior direito.....	109

- Figura 4.13 – Evolução por profundidade máxima para 1000 indivíduos com codificação Subtrativa adotando mutação na calda, após as gerações: 1 (SUB\_001), 50 (SUB\_050), 100 (SUB\_100) e 150 (SUB\_150). Índices de Tendência no canto superior esquerdo..... 110
- Figura 4.14 – Taxas de ocorrência observadas para distâncias entre pai e filho (d1f) após simulação com 5000 mutações convencionais com código Subtrativo. .... 111
- Figura 4.15 – Taxas de ocorrência observadas para distâncias entre pai e filho (d1f) após simulação com 5000 mutações na calda com código Subtrativo. .... 111
- Figura 5.1 – Rede obtida pelo Algoritmo 5.1 decodificando o cromossomo exemplo de Aditiva..... 123
- Figura 5.2 – Comparação entre 5000 indivíduos gerados de forma aleatória utilizando as representações Vetor Característico – VC e Aditiva – ADT: percentual de soluções por profundidade máxima. .... 130
- Figura 5.3 – Comparação entre 5000 indivíduos gerados de forma aleatória utilizando as representações Vetor Característico – VC e Aditiva – ADT: percentual de soluções por faixa de aptidão..... 131
- Figura 5.4 – Evolução por faixa de aptidão para 1000 indivíduos com codificação Aditiva implementando apenas cruzamento convencional, após as gerações: 1 (ADT\_001), 50 (ADT\_050), 100 (ADT\_100) e 150 (ADT\_150). Índices de Tendência no canto superior direito..... 132
- Figura 5.5 – Evolução por profundidade máxima para 1000 indivíduos com codificação Aditiva adotando cruzamento convencional, após as gerações: 1 (ADT\_001), 50 (ADT\_050), 100 (ADT\_100) e 150 (ADT\_150). Índices de Tendência no canto superior esquerdo..... 132
- Figura 5.6 – Evolução por faixa de aptidão para 1000 indivíduos com codificação Aditiva implementando apenas cruzamento na calda, após as gerações: 1 (ADT\_001), 50 (ADT\_050), 100 (ADT\_100) e 150 (ADT\_150). Índices de Tendência no canto superior esquerdo..... 133
- Figura 5.7 – Evolução por profundidade máxima para 1000 indivíduos com codificação Aditiva adotando apenas cruzamento na calda, após as gerações: 1 (ADT\_001), 50 (ADT\_050), 100 (ADT\_100) e 150 (ADT\_150). Índices de Tendência no canto superior esquerdo..... 134
- Figura 5.8 – Evolução por faixa de aptidão para 1000 indivíduos com codificação Aditiva implementando apenas mutação convencional, após as gerações: 1 (ADT\_001), 50 (ADT\_050), 100 (ADT\_100) e 150 (ADT\_150). Índices de Tendência no canto superior direito..... 135
- Figura 5.9 – Evolução por profundidade máxima para 1000 indivíduos com codificação Aditiva adotando mutação convencional, após as gerações: 1 (ADT\_001), 50 (ADT\_050), 100 (ADT\_100) e 150 (ADT\_150). Índices de Tendência no canto superior esquerdo..... 135

Figura 5.10 – Evolução por faixa de aptidão para 1000 indivíduos com codificação Aditiva implementando apenas mutação na calda, após as gerações: 1 (ADT_001), 50 (ADT_050), 100 (ADT_100) e 150 (ADT_150). Índices de Tendência no canto superior direito.....	136
Figura 5.11 – Evolução por profundidade máxima para 1000 indivíduos com codificação Aditiva adotando mutação na calda, após as gerações: 1 (ADT_001), 50 (ADT_050), 100 (ADT_100) e 150 (ADT_150). Índices de Tendência no canto superior esquerdo.....	137
Figura 5.12 – Taxas de ocorrência observadas para distâncias entre pai e filho (d1f) após simulação com 5000 mutações convencionais com código Aditivo. ....	138
Figura 5.13 – Taxas de ocorrência observadas para distâncias entre pai e filho (d1f) após simulação com 5000 mutações na calda com código Aditivo.....	138
Figura 6.1 – Fluxograma do AGS implementado. ....	144
Figura 6.2 – Exemplo das estratégias de controle de $N$ para 3 e 4 patamares: a) Larguras Iguais e b) Larguras Proporcionais. ....	146
Figura 6.3 – Cálculo dos níveis para rede da Figura 3.3 considerando todas as chaves fechadas: (a) níveis dos nós e (b) níveis das arestas. ....	150
Figura 7.1 – Diagrama do sistema 16-nós (CINVANLAR et al, 1988). ....	157
Figura 7.2 – Diagrama do sistema 33-nós (BARAN & WU, 1989). ....	158
Figura 7.3 – Diagrama do sistema 70-nós (DAS, 2006). ....	158
Figura 7.4 – Diagrama do sistema 94-nós (SU & LEE, 2003). ....	159
Figura 7.5 – Diagrama do sistema 137-nós (MANTOVANI et al, 2000). ....	159
Figura 7.6 – Melhor configuração obtida pelo AG para o sistema 16-nós.....	165
Figura 7.7 – Melhor configuração obtida pelo AG para o sistema 33-nós.....	168
Figura 7.8 – Melhor configuração obtida pelo AG para o sistema 70-nós.....	170
Figura 7.9 – Comparativo das melhores execuções no sistema 70-nós por geração: Subtrativa contra VC e Melhorado. ....	171
Figura 7.10 – Comparativo das melhores execuções no sistema 70-nós por geração: Aditiva contra VC e Melhorado. ....	171
Figura 7.11 – Comparativo das melhores execuções no sistema 70-nós por tempo de execução: Subtrativa contra VC e Melhorado.....	173
Figura 7.12 – Comparativo das melhores execuções no sistema 70-nós por tempo de execução: Aditiva contra VC e Melhorado. ....	173

Figura 7.13 – Melhor configuração obtida pelo AG para o sistema 94-nós.....	175
Figura 7.14 – Comparativo das melhores execuções no sistema 94-nós por geração: Subtrativa contra VC e Melhorado. ....	176
Figura 7.15 – Comparativo das melhores execuções no sistema 94-nós por geração: Aditiva contra VC e Melhorado. ....	176
Figura 7.16 – Comparativo das melhores execuções no sistema 94-nós por tempo de execução: Subtrativa contra VC e Melhorado.....	177
Figura 7.17 – Comparativo das melhores execuções no sistema 94-nós por tempo de execução: Aditiva contra VC e Melhorado. ....	177
Figura 7.18 – Melhor configuração obtida pelo AG para o sistema 137-nós.....	179
Figura 7.19 – Comparativo das melhores execuções no sistema 137-nós por geração: Subtrativa contra VC e Melhorado. ....	180
Figura 7.20 – Comparativo das melhores execuções no sistema 137-nós por geração: Aditiva contra VC e Melhorado. ....	180
Figura 7.21 – Comparativo das melhores execuções no sistema 137-nós por tempo de execução: Subtrativa contra VC e Melhorado.....	181
Figura 7.22 – Comparativo das melhores execuções no sistema 137-nós por tempo de execução: Aditiva contra VC e Melhorado. ....	181
Figura 7.23 – Evolução observada no tempo de avaliação por indivíduo ( $T_A$ ) em função do número de nós do sistema testado.....	186
Figura 7.24 – Comparativo das melhores execuções no sistema 137-nós por geração: SUB-CALDA com AG (Seção 7.3) contra AGS e heurísticas indicadas para SUB (N- e I+). ....	191
Figura 7.25 – Comparativo das melhores execuções no sistema 137-nós por geração: SUB-CALDA com AG (Seção 7.3) contra AGS e heurísticas não indicadas para SUB (N+ e I-). ....	191
Figura 7.26 – Comparativo das melhores execuções no sistema 137-nós por geração: ADT-CALDA com AG (Seção 7.3) contra AGS e heurísticas indicadas para ADT (N+ e I-). ....	192
Figura 7.27 – Comparativo das melhores execuções no sistema 137-nós por geração: ADT-CALDA com AG (Seção 7.3) contra AGS e heurísticas não indicadas para ADT (N- e I+). ....	192
Figura 7.28 – Comparativo por geração das melhores execuções de SUB-CALDA/I+ (SUBI+) no sistema 137-nós: larguras iguais (IGUAL) contra larguras proporcionais (PROP). ....	195

- Figura 7.29 – Comparativo por geração das melhores execuções de ADT-CALDA/I– (ADTI–) no sistema 137-nós: larguras iguais (IGUAL) contra larguras proporcionais (PROP).....195
- Figura 7.30 – Evolução por geração das melhores execuções de SUB-CALDA/I+ com largura proporcional para o sistema 137-nós: 2, 3, 4 e 5 patamares do fator de busca  $N$ .....198
- Figura 7.31 – Evolução por geração das melhores execuções de ADT-CALDA/I– com largura proporcional para o sistema 137-nós: 2, 3, 4 e 5 patamares do fator de busca  $N$ .....199

## Lista de Tabelas

Tabela 2.1 – Resumo da revisão bibliográfica sobre Configuração de Redes de Distribuição com ênfase nas técnicas de representação das redes.....	35
Tabela 3.1 – Descrição geral dos módulos de um AG tomando como base o Algoritmo 3.1.....	40
Tabela 3.2 – Tabela de Adjacências usada para guiar a representação Predecessor 2....	54
Tabela 3.3 – Detalhamento das etapas da codificação usando Número Prufer para a rede da Figura 3.3. ....	56
Tabela 3.4 – Detalhamento das etapas da decodificação usando o Número Prufer obtido na Tabela 3.3.....	56
Tabela 3.5 – Resultado do pré-processamento para as codificações por Malhas Fundamentais Com e Sem Corte. ....	62
Tabela 3.6 – Resumo quantitativo das codificações da Configuração de Redes no estado da arte: específico para o exemplo adotado.....	66
Tabela 3.7 – Resumo dos objetivos e restrições mais comuns em estudos de Configuração de Redes (legenda abaixo).....	69
Tabela 3.8 – Resumo da metodologia proposta para ajuste automático dos parâmetros da Equação (3.14), considerando os enfoques: Reconfiguração, Restabelecimento e Planejamento. ....	75
Tabela 4.1 – Etapas do algoritmo de obtenção de Malhas Fundamentais para o sistema da Figura 3.3 (Capítulo 3). ....	80
Tabela 4.2 – Taxas de ocorrência obtidas no estudo de hereditariedade: distância entre pais (d12) versus maior distância observada entre pais e filhos (d1f ou d2f) após operação de cruzamento convencional em indivíduos com codificação Subtrativa. ....	112
Tabela 4.3 – Taxas de ocorrência obtidas no estudo de hereditariedade: distância entre pais (d12) versus maior distância observada entre pais e filhos (d1f ou d2f) após operação de cruzamento na calda em indivíduos com codificação Subtrativa. ....	113

Tabela 5.1 – Taxas de ocorrência obtidas no estudo de hereditariedade: distância entre pais (d12) versus maior distância observada entre pais e filhos (d1f ou d2f) após operação de cruzamento convencional em indivíduos com codificação Aditiva... 140	140
Tabela 5.2 – Taxas de ocorrência obtidas no estudo de hereditariedade: distância entre pais (d12) versus maior distância observada entre pais e filhos (d1f ou d2f) após operação de cruzamento na calda em indivíduos com codificação Aditiva..... 140	140
Tabela 6.1 – Comparativo das Codificações Sequenciais Subtrativa e Aditiva. .... 143	143
Tabela 6.2 – Taxas de ocorrência por faixa de aptidão para um conjunto de 5000 indivíduos gerados aleatoriamente com código Subtrativo, fatores de busca N de 33.3%, 66.6% e 100%, adotando as seguintes heurísticas de ordenação das arestas: Randômica (RD), Níveis crescente (N+), Níveis decrescente (N-), Correntes crescente (I+) e Correntes decrescente (I-). .... 152	152
Tabela 6.3 – Taxas de ocorrência por faixa de aptidão para um conjunto de 5000 indivíduos gerados aleatoriamente com código Aditivo, fatores de busca N de 33.3%, 66.6% e 100%, adotando as seguintes heurísticas de ordenação das arestas: Randômica (RD), Níveis crescente (N+), Níveis decrescente (N-), Correntes crescente (I+) e Correntes decrescente (I-). .... 153	153
Tabela 6.4 – Taxas de ocorrência por profundidade máxima para um conjunto de 5000 indivíduos gerados aleatoriamente com código Subtrativo, fatores de busca N de 33.3%, 66.6% e 100%, adotando as seguintes heurísticas de ordenação das arestas: Randômica (RD), Níveis crescente (N+), Níveis decrescente (N-), Correntes crescente (I+) e Correntes decrescente (I-). .... 154	154
Tabela 6.5 – Taxas de ocorrência por profundidade máxima para um conjunto de 5000 indivíduos gerados aleatoriamente com código Aditivo, fatores de busca N de 33.3%, 66.6% e 100%, adotando as seguintes heurísticas de ordenação das arestas: Randômica (RD), Níveis crescente (N+), Níveis decrescente (N-), Correntes crescente (I+) e Correntes decrescente (I-). .... 155	155
Tabela 7.1 – Dados referentes às configurações iniciais dos três sistemas usados nos testes. .... 157	157
Tabela 7.2 – Aptidões médias das populações iniciais do AG, para as 10 execuções realizadas no sistema de 16-nós. .... 164	164
Tabela 7.3 – Resultados obtidos para o sistema de 16-nós em 10 execuções do AG: melhor execução e dados médios. .... 164	164
Tabela 7.4 – Aptidões médias das populações iniciais do AG, para as 10 execuções realizadas no sistema de 33-nós. .... 167	167
Tabela 7.5 – Resultados obtidos para o sistema de 33-nós em 10 execuções do AG: melhor execução e dados médios. .... 167	167
Tabela 7.6 – Aptidões médias das populações iniciais do AG, para as 10 execuções realizadas no sistema de 70-nós. .... 169	169

Tabela 7.7 – Resultados obtidos para o sistema de 70-nós em 10 execuções do AG: melhor execução e dados médios. ....	169
Tabela 7.8 – Aptidões médias das populações iniciais do AG, para as 10 execuções realizadas no sistema de 94-nós.....	174
Tabela 7.9 – Resultados obtidos para o sistema de 94-nós em 10 execuções do AG: melhor execução e dados médios. ....	174
Tabela 7.10 – Aptidões médias das populações iniciais do AG, para as 10 execuções realizadas no sistema de 137-nós.....	178
Tabela 7.11 – Resultados obtidos para o sistema de 137-nós em 10 execuções do AG: melhor execução e dados médios. ....	178
Tabela 7.12 – Resumo das aptidões médias das populações iniciais dos cinco sistemas teste: simulações com o AG Básico.....	183
Tabela 7.13 – Resumo das melhores aptidões obtidas nos testes do AG Básico para os cinco sistemas testados.....	185
Tabela 7.14 – Resumo das aptidões médias obtidas nos testes do AG Básico para os cinco sistemas testados.....	185
Tabela 7.15 – Resultados obtidos com o AGS combinando SUB-CALDA e ADT-CALDA às heurísticas N+, N-, I+ e I- para o sistema de 70-nós: $N$ com larguras iguais e $n_{pat} = 3$ . ....	188
Tabela 7.16 – Resultados obtidos com o AGS combinando SUB-CALDA e ADT-CALDA às heurísticas N+, N-, I+ e I- para o sistema de 94-nós: $N$ com larguras iguais e $n_{pat} = 3$ . ....	188
Tabela 7.17 – Resultados obtidos com o AGS combinando SUB-CALDA e ADT-CALDA às heurísticas N+, N-, I+ e I- para o sistema de 137-nós: $N$ com larguras iguais e $n_{pat} = 3$ . ....	189
Tabela 7.18 – Resultados obtidos com o AGS combinando SUB-CALDA/I+ e ADT-CALDA/I- às lógicas de controle de $N$ por larguras iguais e proporcionais: sistema de 70-nós e $n_{pat} = 3$ .....	193
Tabela 7.19 – Resultados obtidos com o AGS combinando SUB-CALDA/I+ e ADT-CALDA/I- às lógicas de controle de $N$ por larguras iguais e proporcionais: sistema de 94-nós e $n_{pat} = 3$ .....	193
Tabela 7.20 – Resultados obtidos com o AGS combinando SUB-CALDA/I+ e ADT-CALDA/I- às lógicas de controle de $N$ por larguras iguais e proporcionais: sistema de 137-nós e $n_{pat} = 3$ .....	193
Tabela 7.21 – Resultados obtidos com o AGS combinando SUB-CALDA/I+ e ADT-CALDA/I-, controle de $N$ por larguras proporcionais e variando o número de patamares 2-5: sistema 70-nós.....	196

---

Tabela 7.22 – Resultados obtidos com o AGS combinando SUB-CALDA/I+ e ADT-CALDA/I-, controle de $N$ por larguras proporcionais e variando o número de patamares 2-5: sistema 94-nós.....	196
Tabela 7.23 – Resultados obtidos com o AGS combinando SUB-CALDA/I+ e ADT-CALDA/I-, controle de $N$ por larguras proporcionais e variando o número de patamares 2-5: sistema 134-nós.....	197
Tabela 7.24 – Resumo dos resultados obtidos pelo AGS nos testes da subseção 7.4.1 para os sistemas 70-nós, 94-nós e 137-nós: melhor aptidão e média em 10 execuções.....	200
Tabela A.1 – Dados de entrada referentes ao sistema 16-nós. ....	218
Tabela A.2 – Dados de entrada referentes ao sistema 33-nós. ....	219
Tabela A.3 – Dados de entrada referentes ao sistema 70-nós. ....	220
Tabela A.4 – Dados de entrada referentes ao sistema 94-nós. ....	223
Tabela A.5 – Dados de entrada referentes ao sistema 137-nós.....	226

# Lista de Símbolos

## Siglas e Modelo do Sistema Elétrico

SEP	Sistema Elétrico de Potência
PRODIST	Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional
ANEEL	Agência Nacional de Energia Elétrica
AT	Alta Tensão, definida no PRODIST
MT	Média Tensão, definida no PRODIST
BT	Baixa Tensão, definida no PRODIST
NF	Chaves que operam Normalmente Fechadas
NA	Chaves que operam Normalmente Abertas
AG	Algoritmo Genético
AGS	Algoritmo Genético Sequencial
NSGA	Tipo de AG multiobjetivo cujo nome deriva de: <i>Nondominated Sorting Genetic Algorithm</i>
SPEA	Tipo de AG multiobjetivo cujo nome deriva de: <i>Strength Pareto Evolutionary Algorithm</i>
SIG	Sistema de Informações Geográficas
MG	Memória Genética
VC	Representação de rede por Vetor Característico

Melhorado	Representação de rede por Vetor Característico Melhorado
$S_{nom}$	Capacidade de uma subestação, em MVA
$V_{nom}$	Tensão nominal de uma subestação, em kV
$V_{min}$	Tensão mínima tolerável em uma barra, em kV
$V_{max}$	Tensão máxima tolerável em uma barra, em kV
$SL$	Carga complexa em uma barra, em MVA
$Z_{BC}$	Impedância total de um Banco de Capacitores em uma barra, em $\Omega$
$prog$	Programação diária de um Banco de Capacitores Chaveado
$V_{alvo}$	Tensão estipulada de um gerador instalado em uma barra, em pu
$P_{inj}$	Potência Ativa injetada pelo gerador, em MW
$fp_{min}$	Fator de potência mínimo para operação segura do gerador
$fp_{max}$	Fator de potência máximo para operação segura do gerador
$I_{max}$	Corrente máxima admissível em um ramo, em A
$Z_S$	Impedância de um ramo, em $\Omega$

### Dados Gerais

$K_x$	Grafo completo com $x$ nós
$n(\mathbf{X})$	Número de elementos do conjunto $\mathbf{X}$ , seja qual for o tipo
$P_x$	Número total de permutações com $x$ elementos: $P_x = x!$
$C_{x,y}$	Número total de combinações de $x$ em grupos de $y$ : $C_{x,y} = x!/y!(x-y)!$

### Teoria dos Grafos

$G$	Representa um grafo genérico, notação $G(X, A)$
$X$	Conjunto de nós de $G$
$A$	Conjunto de arestas de $G$

### Grafo de Setores

$G_S$	Grafo de Setores que representa determinada rede, notação $G_S(X_S, A_S)$
$X_S$	Conjunto formado pelos setores, considerando-os como nós de $G_S$
$A_S$	Conjunto formado pelos ramos cuja chave está habilitada na rede base, considerando-os como arestas de $G_S$
$T_S$	Árvore ou floresta definida em $G_S$ , notação $T_S(X_{ST}, A_{ST})$
$X_{ST}$	Conjunto de nós de $T_S$ . Por definição, $X_{ST} = X_S = X_F \cup X_L$ . Sendo, $X_F$ e $X_L$ disjuntos
$X_F$	Conjunto de nós fonte, ou seja, criados a partir de um setor que contém alguma subestação
$X_L$	Conjunto formado a partir de $X_{ST}$ por exclusão de $X_F$ , logo compreende os setores que não possuem qualquer subestação
$A_{ST}$	Conjunto de ramos de $T_S$ . Por definição, $A_{ST} \subset A_S$ e $A_S = A_{ST} \cup A_{NA}$ . Sendo, $A_{ST}$ e $A_{NA}$ disjuntos
$A_{NA}$	Conjunto de arestas de $G_S$ que não pertencem à $T_S$ , chamadas de cordas.
$M_i$	Malha fundamental definida pela corda $i$ em $T_S$

### Algoritmos Genéticos

$\sigma$	Número de soluções em um espaço de busca
$\alpha_i$	Número de símbolos do alfabeto do gene $i$ de um cromossomo
$\gamma$	Número total de genes em um cromossomo
d1f	Número de arestas diferentes entre o pai (1) e o filho (f), após uma operação de mutação ou cruzamento
d2f	Número de arestas diferentes entre o pai (2) e o filho (f), após uma operação de cruzamento

$d_{12}$	Número de arestas diferentes entre os pais 1 e 2 selecionados para operação de cruzamento
$g_{\max}$	Número máximo de gerações adotado como critério de parada do AG ou AGS
$t_{\max}$	Tempo de execução máximo adotado como critério de parada do AG ou AGS
$\Phi_g$	Espaço de Busca dos Genótipos
$d_g$	Distância entre genótipos, medida no espaço $\Phi_g$
$\Phi_f$	Espaço de Busca dos Fenótipos
$d_f$	Distância entre fenótipos, medida no espaço $\Phi_f$

### **Funções Objetivo e de Aptidão**

$D$	Número de dias até atingir o horizonte de estudo escolhido
$f_{ap}$	Função de Aptidão do AG
$f(T_s, D)$	Função Objetivo da solução
$H(T_s)$	Restrições de igualdade impostas pelo fluxo de carga à $T_s$
$I_b$	Corrente no ramo $b$ , em A
$V_i$	Tensão no nó $i$ , em kV
$S_j$	Fluxo de potência no nó fonte $j$ , em MVA
$w_x$	Peso associado ao objetivo $x$ em $f(T_s, D)$
$f_x(T_s, D)$	Parcela associada ao objetivo $x$ em $f(T_s, D)$
$\Delta E_P$	Energia perdida no intervalo de tempo $D$
$\Delta E_E$	Energia entregue no intervalo de tempo $D$
$T_S^0$	Árvore ou floresta correspondente a configuração inicial

$n_{op}$	Número de manobras realizadas para obter $T_s$ de $T_s^0$ desconsiderando as chaves manobradas para isolar o defeito, em um estudo de Restabelecimento
$n_{tt}$	Número total de chaves de $G_s$ desconsiderando as chaves manobradas para isolar o defeito, em um estudo de Restabelecimento
$c_k$	Nível de prioridade de atendimento da carga $k$
$P^{nom}_k$	Carga ativa nominal do nó de carga $k$ , em MW
<b>A</b>	Conjunto formado pelas arestas candidatas que foram habilitadas para obter $T_s$ no eventual estudo de Planejamento ( <b>A</b> $\subset$ <b>B</b> )
<b>B</b>	Conjunto formado por todas as arestas candidatas no eventual estudo de Planejamento
$\$k$	Custo de construção da aresta candidata $k$
$k_i$	Multiplicador associado a penalização da função $f_i$
$x_i$	Parâmetro negativo que controla a intensidade da penalização $i$
$nv_{radial}$	Soma dos nós desconectados mais o número de malhas fechadas na configuração atual do sistema
$nv_v$	Número de nós em que houve violação da restrição de tensão nodal
$nv_l$	Número de ramos em que houve violação da restrição de suportabilidade
$nv_s$	Número de nós fonte em que houve violação da restrição de capacidade
$p_o$	Valor percentual associado a determinado objetivo no ajuste automático dos parâmetros da função de aptidão $f_{ap}$ (Equação 3.14)
$p_r$	Valor percentual associado à determinada restrição no ajuste automático dos parâmetros da função de aptidão $f_{ap}$ (Equação 3.14)
$b$	Número de manobras de chaves a serem consideradas no ajuste automático dos parâmetros da função de aptidão $f_{ap}$ (Equação 3.14)

$\$_{MWh}$  Custo de 1MWh em unidades monetárias

### **Codificação Subtrativa**

$L$  Mapa das Malhas

$\alpha$  Vetor inteiro auxiliar de comprimento  $n(A_{NA})$  que opera conjuntamente com  $L$  na Decodificação Subtrativa

$XOR$  Operador especial definido para auxiliar a execução de expansões e reduções de malhas em  $L$

$L'$  Versão vetorial do Mapa das Malhas

$XOR'$  Operador especial definido para auxiliar a execução de expansões e reduções de malhas em  $L'$

### **Codificação Aditiva**

$R$  Mapa dos Setores

$atual$  Vetor com lista de arestas de fronteira

$acc$  Número de elementos não-negativos de  $atual$

$IXOR$  Operador especial definido para facilitar o controle das arestas de fronteira com base em  $R$

$R'$  Versão vetorial do Mapa dos Setores

$IXOR'$  Operador especial definido para facilitar o controle das arestas de fronteira com base em  $R'$

### **Operadores Genéticos na Calda**

$N_c$  Alcance do operador de cruzamento na calda, calculado com base na aptidão do indivíduo e na média da população

$N_m$  Alcance do operador de mutação na calda, calculado com base na aptidão do indivíduo e na média da população

---

$C_p$	Local em que efetivamente ocorre o cruzamento na calda, calculado em função de $N_c$
$M_p$	Local em que efetivamente ocorre a mutação na calda, calculado em função de $N_m$
$c_1, c_2$	Coefficientes associados ao alcance do cruzamento na calda
$m_1, m_2$	Coefficientes associados ao alcance da mutação na calda
$f_{\max}$	Maior aptidão dentre os indivíduos da população
$f_i'$	Maior aptidão entre os dois cromossomos selecionados para cruzamento na calda
$f_i$	Aptidão do cromossomo selecionado para mutação na calda
$\bar{f}$	Aptidão média da população

### **Algoritmo Genético Sequencial**

$N$	Fator de busca do AGS
$n_{\text{pat}}$	Número de patamares de $N$ utilizados pelo AGS
$N_0$	Valor inicial de $N$
$N_{\text{final}}$	Valor final de $N$
N+	Heurística proposta para o AGS cuja ordenação ocorre em função dos níveis das arestas em ordem crescente
N-	Heurística proposta para o AGS cuja ordenação ocorre em função dos níveis das arestas em ordem decrescente
I+	Heurística proposta para o AGS cuja ordenação ocorre em função das correntes nas arestas na rede malhada obtida com todas as chaves fechadas em ordem crescente

- I– Heurística proposta para o AGS cuja ordenação ocorre em função das correntes nas arestas na rede malhada obtida com todas as chaves fechadas em ordem decrescente

### **Resultados**

$T_C$	Tempo médio de Obtenção da solução final em um estudo usando AG
$T_A$	Tempo médio de avaliação por indivíduo em um estudo usando AG
$T_M$	Tempo médio total de execução do AG
$g'$	Número médio de gerações do AG para que seja obtida a solução final
$n_{POP}$	Número de indivíduos na população do AG ou AGS

## Resumo

O problema da Configuração de Redes de Distribuição foi formulado para ser resolvido empregando-se Algoritmos Genéticos (AGs). A formulação do problema é geral, englobando os três enfoques do problema: Reconfiguração, Restabelecimento e Planejamento. Apresentam-se uma função de aptidão que engloba os três enfoques, um procedimento para ajuste automático de seus parâmetros e uma memória genética, que mitiga as chamadas à função. Contudo, apesar da formulação generalizada, o enfoque principal do trabalho é no desenvolvimento de novas técnicas que permitam aos AGs configurar redes de grande porte em um tempo factível. Nesse sentido, duas codificações são apresentadas: Subtrativa e Aditiva. Elas têm em comum a vantagem de codificarem exclusivamente configurações radiais conexas, o fato de utilizarem uma lógica cuja interpretação é sequencial e não demandarem operadores genéticos específicos. Sua principal desvantagem reside na maior complexidade de seus algoritmos de decodificação, porém estruturas de dados e operadores que simplificam a decodificação foram propostos:  $L$ ,  $L'$ ,  $XOR$ ,  $XOR'$ ,  $R$ ,  $R'$ ,  $IXOR$  e  $IXOR'$ . Os resultados obtidos em cinco sistemas teste disponíveis na literatura demonstram que as referidas codificações são efetivas na melhoria do desempenho dos AGs. Embora Subtrativa e Aditiva caracterizem uma contribuição ao estado da arte, considera-se como principal inovação do trabalho a formulação de um AG Sequencial. Esse algoritmo tem como pré-requisito a utilização de um código sequencial e se diferencia do AG convencional por combinar espaço de busca com tamanho controlado a um critério de otimização heurístico. Para testá-lo foram utilizados os três maiores sistemas dentre os cinco usados anteriormente, tendo sido observados ganhos consistentes de desempenho, mesmo tomando como referência os resultados obtidos por Subtrativa e Aditiva com o AG.

# Abstract

The Distribution Network Configuration Problem is stated to be solved by using Genetic Algorithms (GAs). The formulation includes network reconfiguration, service restoration, and expansion planning. A fitness function comprising those three approaches, a procedure to automatically adjust its parameters and a genetic memory which mitigates the overall fitness calls are presented. However, despite this general formulation, the main focus of the thesis is in developing new techniques so as to configure large networks using GAs in a feasible time. Accordingly, two encodings are presented: Subtractive and Additive. They share the advantage of representing only radial and connected networks, the fact that adopt a logic whose interpretation is sequential, and do not demanding any specific genetic operator. Their main drawback is the complexity of decodification algorithms, so specific data structures and operators were proposed in order to simplify the decodification process:  $L$ ,  $L'$ ,  $XOR$ ,  $XOR'$ ,  $R$ ,  $R'$ ,  $IXOR$  and  $IXOR'$ . The results on five test systems available in the configuration literature show that these encodings are effective in improving the performance of GAs. Although Subtractive and Additive characterize a contribution to the state of the art, it is considered as the main innovation of the thesis the formulation of a Sequential GA. This algorithm shall use some sequential encoding and differs from the conventional GA combining the control of search space size with a heuristic optimization criterion. In order to validate the Sequential GA, it was used the largest three systems of the five used previously. Consistent improvements in performance were observed, taking as reference even the results obtained by Subtractive and Additive with the GA.

# Capítulo 1

## Introdução

---

Este capítulo tem o propósito de formular o problema da *Configuração de Redes de Distribuição*, incluindo seus três enfoques principais: Reconfiguração para Minimização de Perdas, Restabelecimento de Serviço em Condições de Falta e Planejamento da Expansão; e apresentar os objetivos e contribuições deste trabalho ao estado da arte. Para melhorar o encadeamento do texto, duas seções auxiliares foram incluídas: uma seção inicial chamada de Preliminares, cujas metas são delimitar esta pesquisa dentro dos Sistemas Elétricos de Potência (SEPs) e descrever os modelos elétricos adotados; e uma seção final, denominada Organização, que delinea como o texto dos próximos capítulos está organizado.

### 1.1 Preliminares

O termo SEPs descreve um sistema complexo que engloba geração, transmissão, distribuição e utilização de energia elétrica (SOUZA, 1997). Sua função primordial é garantir o suprimento de energia elétrica aos consumidores, grandes ou pequenos, no local em que eles se encontram e de acordo com critérios de qualidade e continuidade de serviço. O sistema opera como se fosse uma cadeia: nas unidades de geração, diferentes formas de energia são continuamente convertidas em energia elétrica, que é entregue às redes de transmissão; as quais têm a função de conduzi-la até os grandes

centros consumidores, onde os sistemas de distribuição procedem à entrega aos consumidores para pronta utilização.

O sistema de distribuição ocupa uma posição de destaque, por estar localizado na extremidade final da cadeia – em contato direto com os consumidores. De acordo com os Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional – PRODIST (ANEEL, 2010), ele é subdividido de acordo com o nível de tensão nominal de operação em:

- Alta Tensão (AT) – Tensão entre fases cujo valor eficaz é igual ou superior a 69 kV e inferior a 230 kV, ou instalações em tensão igual ou superior a 230 kV quando especificamente definidas pela Agência Nacional de Energia Elétrica (ANEEL);
- Média Tensão (MT) – Tensão entre fases cujo valor eficaz é superior a 1 kV e inferior a 69 kV; e
- Baixa Tensão (BT) – Tensão entre fases cujo valor eficaz é igual ou inferior a 1 kV.

Neste trabalho de tese, o enfoque é na distribuição de energia em MT, com tensões nominais usualmente de 13.8 kV e 34.5 kV. Para efeito do estudo, as cargas alimentadas em BT são consideradas como concentradas nos transformadores MT/BT e em cada uma destas transformações, diz-se que há uma *barra* e interligando-as que existem *ramos*.

Embora existam várias configurações possíveis para operação das redes de distribuição em MT, a configuração radial é predominante. Configuração radial significa que há um único caminho energizado desde uma subestação até qualquer barra em qualquer instante. A principal vantagem desta topologia é a simplicidade de manter os esquemas de proteção efetivos quando o sistema passa a atender novos clientes, além de minimizar o investimento com tais expansões. Contudo, o custo decorrente é uma rede com baixa confiabilidade e sem flexibilidade operacional. Na prática, estes

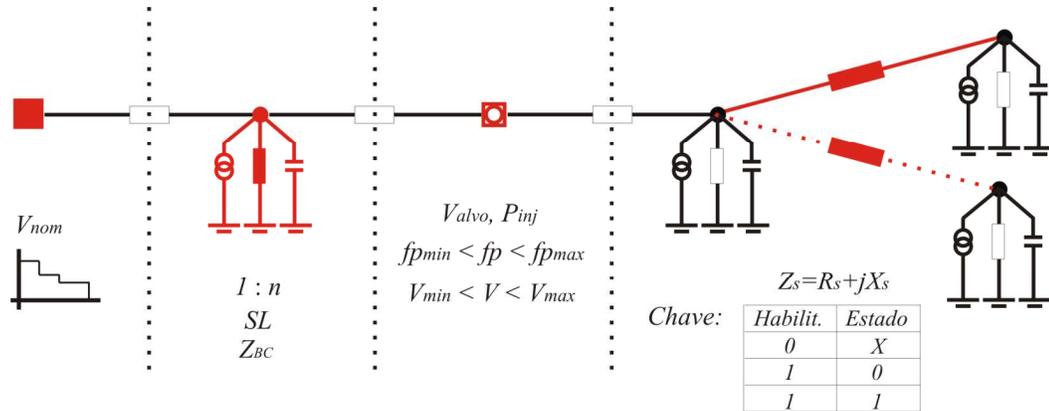
sistemas são construídos com algumas chaves seccionadoras e de interligação, dispostas estrategicamente na rede para dotá-la da possibilidade de reconfiguração. Estas chaves operam, respectivamente, na condição *normalmente fechadas* (NF) e *normalmente abertas* (NA). Desse modo, há mais de um caminho entre subestações e barras, porém a restrição de radialidade garante que somente um opera energizado por vez.

O presente trabalho parte da premissa que a distribuição em MT opera radialmente e para representá-la são utilizados três elementos construtivos: subestações, barras e ramos.

De acordo com o modelo elétrico proposto, uma subestação é composta por sua capacidade  $S_{nom}$ , tensão terminal  $V_{nom}$ , e curva de variação diária de demanda com até quatro níveis normalizados. As barras têm limites de tensão de operação ( $V_{min}$ ,  $V_{max}$ ) e podem modelar vários componentes que operam em derivação: cargas  $SL$ , modeladas como uma composição de impedância, corrente e potência constantes; bancos de capacitores, fixos ou chaveados, modelados como uma impedância  $Z_{BC}$  e uma programação diária *prog*; reguladores de tensão, com relação 1:n; e geradores, modelados como barras de tensão controlada  $V_{alvo}$ , com injeção ativa estipulada  $P_{inj}$  e limites de fator de potência ( $fp_{min}$ ,  $fp_{max}$ ). Por fim, ramos consistem em elementos que permitem interligar subestações e barras para formar a rede de distribuição. Eles são modelados por uma corrente máxima  $I_{max}$  que define sua suportabilidade, impedância série  $Z_S$  e uma chave que pode ou não ser habilitada. Na Figura 1.1 apresenta-se um diagrama ilustrativo dos modelos adotados.

## 1.2 Apresentação do problema

O termo estudo de Configuração de Redes engloba principalmente três enfoques: Reconfiguração de Redes, Restabelecimento de Serviço em Condições de Falhas e



**Figura 1.1 – Resumo dos modelos elétricos adotados para Subestações, Barras e Ramos.**

Planejamento da Expansão. Por questão de simplicidade, eles são referenciados no texto, respectivamente, como problemas de Reconfiguração, Restabelecimento e Planejamento. Em todos os casos, o objetivo consiste em descobrir uma topologia de rede que atenda às restrições do problema e optimize vários objetivos simultaneamente; porém, as formulações de cada um destes tipos de problema se diferenciam em função de seus objetivos e restrições.

Geralmente, os estudos de Reconfiguração consistem em determinar o estado das chaves (fechada/aberta) de secção e interligação do sistema, visando obter uma configuração topológica que mitigue as perdas globais. Outros objetivos podem ser considerados, como: melhoria do perfil de tensão, redução da sobrecarga nos equipamentos e balanceamento; todavia, a configuração com perdas mínimas tende a atender satisfatoriamente tais objetivos e, por isto, tem sido o critério escolhido na maioria dos trabalhos que abordam o tema. Minimizar o número de manobras pode ser visto como um objetivo secundário da Reconfiguração, com efeito na vida útil dos equipamentos. Seja qual for a solução, ela deve atender às restrições de radialidade, conexão, capacidade das subestações, limites de tensão nas barras e suportabilidade dos ramos.

Como um exemplo, considere as configurações das Figuras 1.2 e 1.3, em

que um quadrado maior representa a única subestação e círculos representam as cargas. Quadrados menores diferenciam as arestas com chave das demais.

A configuração de rede ilustrada na Figura 1.3 resulta da configuração da Figura 1.2 mediante a abertura das chaves  $S_7$ ,  $S_8$  e  $S_{13}$ ; e pelo fechamento das chaves  $S_3$ ,  $S_9$  e  $S_{16}$ . Esta configuração implica em uma redução de aproximadamente 25% nas perdas elétricas, de 202.7 para 153.5 kW, e em um correspondente aumento na tensão

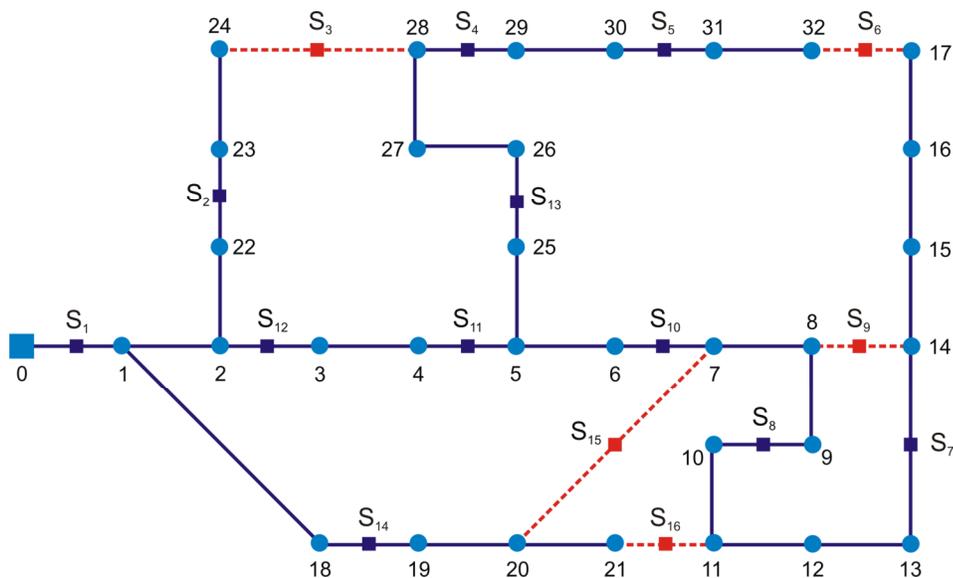


Figura 1.2 – Rede de Distribuição adotada como exemplo.

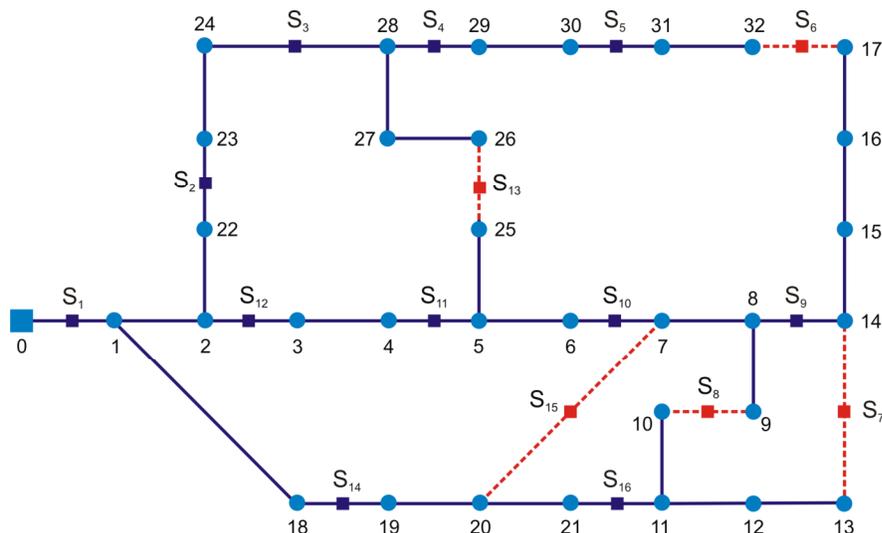


Figura 1.3 – Rede da Figura 1.2 após Reconfiguração: redução de perdas de 202.7kW para 153.5kW.

mínima verificada de 11.56 kV para 11.80 kV. Ela também demonstra uma propriedade importante da Reconfiguração: o número de manobras de abertura é igual ao de fechamentos.

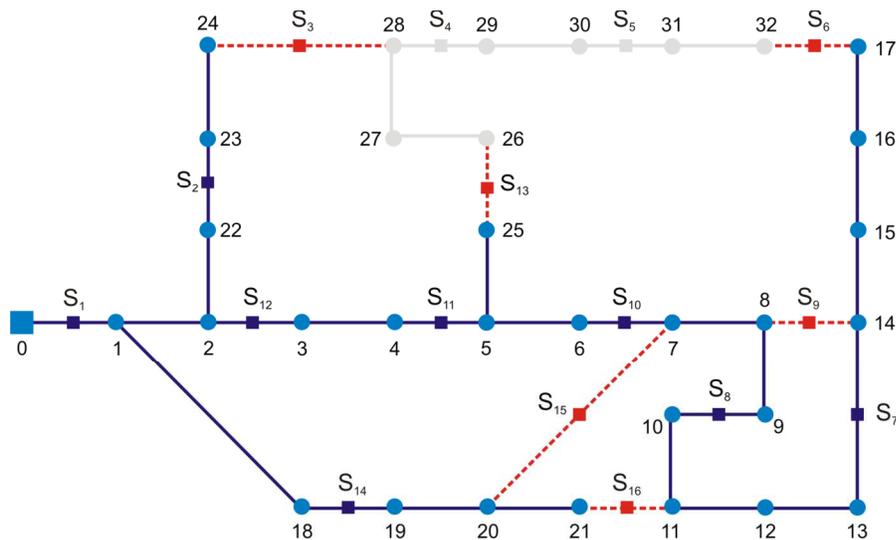
O Restabelecimento é semelhante à Reconfiguração no sentido em que o procedimento, ora abre chaves NF ora fecha chaves NA. Apesar disso, seu objetivo não é a redução de custos operacionais, mas sim reenergizar áreas em que, embora não haja defeito, o suprimento de energia está interrompido devido à operação do sistema de proteção após uma falta permanente. Entende-se como falta permanente, tanto os desligamentos permanentes da rede devido a curtos-circuitos, quanto os desligamentos programados que visam realizar tarefas rotineiras de manutenção.

Antes de definir um eventual plano de Restabelecimento, o defeito deve ser isolado da rede de distribuição pela abertura e travamento de todas as chaves pertencentes ao setor sob contingência. Após o isolamento, o plano de Restabelecimento pode ser implantado visando principalmente minimizar a carga fora de serviço e o tempo total da interrupção. Quando não é possível garantir o suprimento a todas as cargas, uma lista de prioridades deve ser atendida. Instalações como hospitais, polícia, bombeiros e serviços, como de telecomunicações, devem ser atendidos com prioridade máxima. Seguidos por bancos, empresas de alta tecnologia, algumas indústrias, escolas, supermercados e outros locais públicos. Por fim, consumidores normais.

As restrições operacionais citadas para a Reconfiguração de rede também se aplicam ao Restabelecimento. Porém, a restrição de rede conexa deve ser relaxada; uma vez que, dependendo da contingência, pode ser impossível restabelecer algumas áreas. Adicionalmente, há uma forte restrição de tempo: o plano precisa ser elaborado em um curto prazo e deve implicar em um número reduzido de chaveamentos para minimizar o transtorno.

Para exemplificar, considere que houve um defeito no nó 26 do sistema apresentado na Figura 1.2, o qual provocou a abertura da chave  $S_{13}$ . Na Figura 1.4 ilustra-se a configuração de rede resultante. Antes de qualquer outra ação, as chaves  $S_3$ ,  $S_4$  e  $S_{13}$  devem ser abertas e travadas para garantir o isolamento do defeito. Um plano viável para a reenergização é abrir a chave  $S_8$  e fechar as chaves  $S_6$  e  $S_9$ , como ilustrado na Figura 1.5. Neste caso, apenas a carga do setor defeituoso permanece fora de serviço e foram necessárias cinco manobras de chaves no total, sendo duas para isolar o defeito e três para o restabelecimento do serviço ( $S_3$  já estava aberta na configuração original). O custo foi uma redução da tensão mínima da rede, que saiu de 11.56 kV para 10.54 kV.

Observe que o estudo de Restabelecimento pode ser visto como um caso particular da Reconfiguração; ambos os processos de busca consistem em operar chaves já existentes no sistema procurando a configuração que melhor atenda aos objetivos do problema. Contudo, para esta condição o número de manobras de fechamento é maior que o de aberturas. A razão é que há setores desligados na rede inicial, sendo necessária



**Figura 1.4 – Rede da Figura 1.2 após atuação da proteção em decorrência de uma falta no nó 26: barras 26, 27, 28, 29, 30, 31 e 32 desenergizadas.**



tais parcelas. As restrições do problema são similares as da Reconfiguração, acrescidas de eventuais restrições geográficas ou urbanísticas relacionadas à construção de novos ramos.

Do ponto de vista topológico, o Planejamento de expansão se diferencia da Reconfiguração e do Restabelecimento porque não se restringe à manobra de chaves. Notadamente, sua solução inclui ramos novos a serem efetivamente incluídos ao sistema. Entretanto, é possível tratá-lo da mesma maneira adotando alguns artifícios. Por exemplo, basta definir um ramo com chave para cada alternativa candidata a ser pesquisada e assumir que a mesma foi incluída pelo Planejamento se, na configuração final, sua chave estiver fechada e que não foi selecionada, em caso contrário. Um exemplo simplificado de expansão de rede é apresentado na Figura 1.6. Primeiro, a configuração original com os novos nós e os ramos candidatos modelados como arestas com chave; em seguida, a rede resultante da expansão. Observe que apenas as arestas que contém as chaves  $S_7$ ,  $S_8$ ,  $S_{15}$  e  $S_{23}$  não constam na rede final; estas foram as chaves que permaneceram abertas na configuração proposta pelo Planejamento. Mesmo na rede original, duas chaves foram operadas:  $S_3$  e  $S_{13}$ .

Com base na descrição de cada um dos enfoques, é possível formular o problema geral da Configuração de Redes como: *determinar uma configuração de rede radial conexa que, concorrentemente, permita atender às várias restrições e aos objetivos do problema*. São muitos os complicadores: há restrições e objetivos modelados por variáveis contínuas e não-lineares, enquanto o estado das chaves estabelece um conjunto de variáveis discretas; a restrição de radialidade é difícil de modelar matematicamente e o problema é inerentemente multiobjetivo. Além disso, o número de configurações candidatas sofre explosão combinatória para redes de maior



Reconfiguração, Restabelecimento e Planejamento; e seu escopo são as redes de distribuição em MT de *grande porte*, com centenas ou até poucos milhares de barras.

O tema é de grande interesse, fato comprovado pelo grande número de publicações nas principais revistas técnicas especializadas nas últimas quatro décadas. É igualmente um desafio técnico atual, porque ainda não há uma solução consensual capaz de configurar redes de grande porte em um tempo factível sem que seja preciso lançar mão de fortes simplificações. O que se observa no estado da arte é que a complexidade do problema restringe as técnicas de busca factíveis para sua solução.

Uma das técnicas que tem sido cada vez mais utilizada são os algoritmos evolucionários, em particular os Algoritmos Genéticos (AGs). Sua maior vantagem é o fato de conviverem satisfatoriamente com a natureza discreta do problema, com a não-linearidade da função objetivo e, principalmente, por serem capazes de encontrar soluções ótimas, ou quase-ótimas, mesmo para problemas de alta complexidade computacional. Entretanto, sua principal desvantagem é o alto tempo de processamento.

No caso específico da Configuração, sabe-se que a melhoria no modo como as redes são codificadas incide diretamente no desempenho dos AGs, permitindo simultaneamente que melhores soluções sejam obtidas e que o tempo de processamento total seja mitigado. Com o intuito de resolver o problema tirando proveito deste fato, define-se como objetivo do presente trabalho *pesquisar novas estratégias de codificação aplicáveis à Configuração de Redes de Distribuição em sua formulação geral, visando estender a solução do problema para sistemas de distribuição de grande porte*. Nesse sentido, duas estratégias de *Codificação Sequencial* são propostas: uma *Subtrativa* e outra *Aditiva*.

A primeira codificação proposta é nomeada de *Subtrativa* porque parte da topologia malhada, obtida pelo fechamento de todas as chaves do sistema, e opera

abrindo chaves até que uma topologia radial seja obtida. A segunda é chamada de codificação *Aditiva*, visto que faz o oposto: supõe que todas as chaves no sistema estão abertas e procede ao fechamento sequencial das mesmas até obter uma topologia radial conexa. O termo *Sequencial* se refere ao modo como ambos os códigos são interpretados, em que o significado de um gene depende dele mesmo e daqueles que o precedem.

As codificações elaboradas garantem que apenas configurações radiais conexas sejam representadas, mesmo quando operadores genéticos convencionais são utilizados. Esta característica confere de imediato às duas versões uma condição de destaque ao lado das principais representações de rede no estado da arte; principalmente porque ainda não há consenso sobre qual seria a melhor técnica. Desse modo, as codificações sequenciais caracterizam por si uma contribuição do presente trabalho ao estado da arte. Entretanto, somente no curso da pesquisa foi possível concluir que restringir a busca às configurações radiais conexas ainda pode ser insuficiente para redes de distribuição de grande porte, dado o número combinatório de soluções candidatas.

Como principal contribuição deste trabalho de tese, propõe-se uma variação do próprio AG, chamada de AG Sequencial (AGS). Esse algoritmo tem como pré-requisito a utilização de código sequencial e se diferencia do AG convencional por combinar um espaço de busca com tamanho ajustável a um critério de otimização heurístico. Esta combinação permite realizar uma busca nas imediações da solução quase-ótima que seria obtida por tal heurística, favorecendo o desempenho do algoritmo.

Como uma contribuição adicional, necessária para viabilizar uma formulação geral para a Configuração, apresenta-se uma função de aptidão com

objetivos e restrições dos três enfoques do problema modelados por termos normalizados no intervalo  $[0,1]$ . Uma função linear é utilizada para os objetivos, enquanto as restrições utilizam uma função exponencial. Deste modo, é possível obter uma mesma resolução para cada um dos objetivos na função e, mais importante, garantir uma penalização maior associada às restrições. Não obstante, uma metodologia para o ajuste automático dos parâmetros da função de aptidão é apresentada, sendo este algoritmo uma outra contribuição ao estado da arte.

À parte a solução da Configuração de Redes de MT, as codificações desenvolvidas implicam em uma quebra de paradigma em termos de algoritmos evolucionários; sua lógica de interpretação é sequencial, implicando na atualização dinâmica do alfabeto dos genes. O mesmo pode ser dito do AGS, pois na bibliografia consultada não foi observada nenhuma tentativa de manipular a tendência de uma técnica de codificação em gerar redes de um tipo particular para proporcionar um ganho de performance dos algoritmos evolucionários. Tratando-se do problema amplo de buscar uma árvore de cobertura ótima em um grafo, há uma expectativa de que seja possível generalizar as idéias que fundamentam o AGS como um procedimento geral para melhoria da performance em algoritmos evolucionários.

No que concerne à divulgação dos resultados obtidos nesta tese, enumeram-se as publicações relacionadas diretamente ao trabalho:

1. Distribution Network Reconfiguration Using Genetic Algorithms With Sequential Encoding: Subtractive and Additive Approaches, a ser publicado no IEEE Transactions on Power Systems.
2. Planejamento da Rede Coletora de um Parque de Geração Eólica Usando Algoritmos Genéticos com Dimensionamento de Cabos Baseado em Conductor

- Econômico, a ser apresentado no Encontro Regional Ibero-americano do CIGRÉ, Cidade do Leste – Paraguai em maio de 2011.
3. Planejamento da Rede Coletora de um Parque de Geração Eólica usando Algoritmos Genéticos, publicado no Simpósio Brasileiro de Sistemas Elétricos realizado em Belém – PA em 2010.
  4. Reconfiguração de Sistemas de Distribuição de Energia Elétrica Utilizando Algoritmos Genéticos, publicado no Simpósio Brasileiro de Sistemas Elétricos realizado em Belo Horizonte – MG em 2008.
  5. Reconfiguração de Redes de Distribuição: Uma Revisão de Três Estratégias de Codificação a Procura de Algoritmos Genéticos Mais Eficientes, publicado no Congresso Brasileiro de Automática realizado em Juiz de Fora – MG em 2008.
  6. Restabelecimento de Serviço em Redes de Distribuição utilizando Algoritmos Genéticos, publicado no Congresso Brasileiro de Automática realizado em Salvador – BA em 2006.
  7. Um Algoritmo Genético para Implantação de Redes de Distribuição, publicado no Congresso Brasileiro de Automática realizado em Gramado – RS em 2004.
  8. Computação evolucionária para planejamento da expansão de redes de distribuição, apresentado no IEEE/PES T&D Latin America realizado em São Paulo – SP em 2004.
  9. Um Algoritmo Genético para Configuração Ótima de Alimentadores de Energia Elétrica, publicado no Simpósio Brasileiro de Automação Inteligente realizado em Bauru – SP em 2003.

## **1.4 Organização do Texto**

O texto deste trabalho de tese foi organizado em oito capítulos: no Capítulo 2 apresenta-

---

se uma revisão da literatura técnica sobre a Configuração de Redes de Distribuição, com ênfase nas estratégias de representação de redes disponíveis no estado da arte; em seguida, os Capítulos 3, 4 e 5 descrevem, respectivamente, fundamentos sobre AGs dedicados à solução da Configuração de Redes e as técnicas de Codificação Sequenciais propostas: Subtrativa e Aditiva; no Capítulo 6, o AGS é formulado, caracterizando a principal contribuição do presente trabalho; posteriormente, apresentam-se no Capítulo 7 os resultados obtidos nos testes executados com o AG e AGS em cinco sistemas disponíveis na literatura sobre Configuração de Redes; e, por fim, o Capítulo 8 expõe algumas conclusões e limitações dos estudos realizados, terminando com a indicação de alguns desdobramentos possíveis deste trabalho de doutoramento.

## Capítulo 2

# Revisão Bibliográfica

---

Este capítulo relata os principais trabalhos publicados na literatura técnica sobre o problema da Configuração de Redes de Distribuição. Seu objetivo é fornecer uma perspectiva cronológica do estado da arte, sobretudo em relação às técnicas de representação de redes já estabelecidas. Este levantamento de representações será retomado no Capítulo 3, no entanto agrupando as técnicas por suas similaridades e, principalmente, fornecendo um exemplo prático com cada uma das codificações relatadas aqui.

O tema Configuração de Redes é considerado clássico na pesquisa sobre sistemas de distribuição, porém tal processo parece ter se acelerado nos últimos anos, motivado aparentemente pelo crescente interesse das empresas do ramo na automação de seus sistemas. Embora seja admissível em alguns casos resolver a Configuração na sua formulação genérica, a maioria dos trabalhos relacionados se propõe a resolver apenas um dos três enfoques do problema: Reconfiguração, Restabelecimento ou Planejamento. Isto decorre das diferenças nos objetivos e restrições de cada uma das formulações, implicando igualmente em diferenças nas técnicas de solução utilizadas. Ainda assim, há trabalhos que asseguram a resolução do problema genérico, como exemplificado por Morelato & Monticelli (1989) e Delbem et al (2005).

Considerando a segmentação da literatura sobre o tema em função dos diferentes enfoques do problema, foram incluídas neste capítulo seções dedicadas à

revisão bibliográfica de cada um deles. Após estas três seções, um sumário da pesquisa bibliográfica é apresentado.

## 2.1 Reconfiguração de Redes

A formulação mais comum da Configuração de Redes é a Reconfiguração, em especial, visando minimizar perdas. Em parte por esta razão e também por ser o enfoque com menor conjunto de restrições, esta é a linha de pesquisa na qual o maior número de artigos e representações de rede foram publicados.

Merlin & Back (1975) foram pioneiros neste tipo de estudo, pois propuseram ainda na década de 70 um método heurístico que partindo de uma variação da técnica de *branch-and-bound*, possibilita a determinação da configuração ótima, ou quase ótima, em termos de redução de perdas ativas. O método proposto parte da rede em malha obtida com o fechamento de todas as suas chaves e segue abrindo sequencialmente as chaves segundo o critério de minimização de perdas ativas, até a obtenção da primeira configuração radial. O algoritmo é do tipo guloso<sup>1</sup> e emprega um modelo resistivo linear como equivalente da rede, sendo referenciado aqui como método de *Abertura Sequencial*.

Em 1988, Civanlar et al (1988) propuseram um método de solução cuja configuração inicial era radial em vez de malhada. O processo de busca proposto consiste em tentativas de recombinar trechos interligados com o objetivo de minimizar perdas. Devido ao grande número de configurações candidatas, definiram um critério baseado na tensão terminal das chaves NA para restringir a pesquisa somente às configurações que resultam em redução das perdas. Também simplificaram o algoritmo,

---

<sup>1</sup> O termo guloso ou ganancioso se aplica aos algoritmos cuja operação é baseada em etapas, nas quais o algoritmo escolhe a melhor opção na etapa corrente sem considerar a solução como um todo.

desenvolvendo uma equação que permite aproximar a variação nas perdas ativas do sistema quando ocorre transferência de ramos. Esse algoritmo ficou conhecido como *Troca de Ramos* e inspirou vários trabalhos posteriores. Contudo, sua principal limitação é que apenas uma fração das configurações de rede possíveis está no espaço de busca, a qual depende ainda da configuração inicial do sistema.

Um ano depois, Shirmohammadi & Hong (1989) propuseram uma variação da abertura sequencial que emprega um algoritmo de fluxo de carga específico para redes fracamente malhadas e uma heurística para abertura das chaves baseada em correntes ótimas.

Baran & Wu (1989) aprimoraram a troca de ramos formulando dois métodos de cálculo de fluxo de carga específicos das redes radiais, que foram utilizados para acelerar a busca da solução ótima com diferentes graus de precisão. Além das perdas, contemplaram o balanceamento de cargas. A partir deste trabalho, o problema de Reconfiguração passou a ser reconhecido como sendo de natureza combinatória.

Ainda em 1989, Liu et al (1989) desenvolveram dois algoritmos heurísticos para solução do problema. O primeiro parte da premissa de que as cargas são distribuídas ao longo dos ramos da rede e o outro supõe cargas concentradas nas barras. O procedimento de busca consiste em calcular pontos ótimos de abertura das malhas geradas pelo fechamento de chaves NA. Quatro sistemas de teste são utilizados para validação, com tamanhos variando de 18 a 168 chaves.

Nara et al (1992) foram os primeiros a empregar AGs para solução do problema. Modelaram a rede considerando que ramos podem ter mais de uma chave e codificaram sua topologia usando uma palavra com comprimento proporcional ao número de chaves NA da rede. Cada gene é formado por um identificador de ramo e um identificador de chave, válido no ramo selecionado. O algoritmo proposto usa seleção

por roleta, cruzamento e mutação convencionais, considerando na função de aptidão as restrições de violação de tensão nodal e de suportabilidade dos ramos. Considera-se fator de demanda unitário, cargas modeladas como de corrente constante e cálculo da variação nas perdas simplificado, adotando uma expressão desenvolvida no artigo.

Mais tarde, Sarfi et al (1994a) apresentaram uma heurística de pré-processamento que classifica as chaves em uma rede em termos de seu potencial para redução de perdas. Para tanto, vários índices foram propostos; relativos a suportabilidade e impedância dos ramos, queda de tensão nas barras e até distâncias físicas. O resultado foi uma redução no número de configurações candidatas, entretanto, sem garantias de permanência da solução ótima global no espaço de busca. Um ano depois, os mesmos autores propuseram um algoritmo de Reconfiguração que aproveita parte da idéia de pré-processamento por meio de um índice de eficiência na transmissão, definido como a razão do fluxo de potência em um ramo pela sua perda ativa (SARFI et al, 1995). Contudo, em vez de restringir o espaço de busca pela retirada de algumas chaves, propõem agrupar partes da rede conectadas por ramos de alta eficiência assumindo que estes chaveamentos seriam indesejáveis. A técnica de troca de ramos foi empregada na interligação dos grupos resultantes.

Peponis et al (1995) elaboraram um método para Reconfiguração de rede e consequente realocação de bancos de capacitores fora de serviço, empregando a técnica de programação dinâmica. Utilizaram troca de ramos e abertura sequencial na parte de reconfiguração do algoritmo e sugeriram melhorias às duas técnicas: na troca de ramos, simplificaram a equação que permite calcular a variação nas perdas em virtude de troca de ramos; e na abertura sequencial estenderam a técnica para vários níveis de demanda em vez de um único.

Em 1997, Borozan et al (1997a) estenderam o método da abertura

sequencial para sistemas trifásicos desbalanceados. No mesmo ano, Borozan et al (1997b) apresentaram uma ferramenta integrada composta por três módulos: estimação de demanda, Reconfiguração de rede usando abertura sequencial e avaliação custo/benefício.

Rudnick et al (1997) sugeriram uma técnica heurística baseada na troca de ramos, através da qual diferentes configurações radiais são geradas visando otimizar uma função definida em termos de perdas e balanceamento de carga. O fluxo de carga é realizado através de uma versão simplificada e não iterativa do método da soma de potências (CESPEDES, 1990).

Ainda em 1997, Fudou et al (1997) adotaram uma abordagem diferente e retomaram os AGs para solução do problema. Os autores propuseram representar a rede por meio de um vetor de inteiros com comprimento igual ao número de barras; cada gene identifica a origem de determinado nó. Essa codificação é referenciada como *Predecessor I*.

Outro trabalho inspirado na troca de ramos foi publicado por Kashem et al (1999). Uma equação similar àquela proposta por Baran & Wu (1989) foi elaborada, mas, ao invés de variação de perdas, consideraram o balanceamento de carga. O processo de busca testa qual das malhas da rede é a mais promissora em termos de melhoria no balanceamento, fecha a malha e usa a equação proposta no artigo para decidir que chave deve ser aberta para restabelecer a radialidade. O processo continua enquanto houver malhas com potencial de melhoria no balanceamento.

Morton & Mareels (2000) apresentaram um método designado como de força bruta para solução do problema de Reconfiguração. Sua vantagem é a garantia de se encontrar a solução ótima global. Contudo, o esforço computacional associado é proibitivo para sistemas com dimensões reais. Para amenizar este esforço, foi utilizado o

modelo da corrente constante para representar a carga.

Mantovani et al (2000) propuseram um método de busca na árvore de possibilidades semelhante ao proposto por Morelato & Monticelli (1989) no sentido em que ramos completos são podados visando redução no número de soluções pesquisadas. O critério de poda é baseado no limite máximo de queda de tensão nos alimentadores e os objetivos englobam perdas elétricas e balanceamento de carga. O método de geração da árvore de possibilidades pode ser visto como uma variação da troca de ramos, uma vez que as configurações são geradas fechando uma chave NA por vez e abrindo sequencialmente as outras chaves na malha resultante. Cada chave aberta estabelece uma configuração candidata, entretanto, se esta implica em uma queda de tensão maior que um limiar permitido, todas as configurações geradas a partir dela são desprezadas.

O método de Mantovani et al (2000) requer cuidado especial na definição do limiar de queda de tensão, já que sua opção é subjetiva e depende da perícia do especialista. Para validação, utilizaram três sistemas disponíveis na literatura sobre configuração de redes e um quarto sistema de 137 barras, cujos dados constam no próprio artigo.

No mesmo ano, Lin et al (2000) apresentaram uma técnica refinada que combina AGs, busca tabu e abertura sequencial. A abertura sequencial é usada para inicializar metade da população do AG e a busca tabu fornece uma lista de configurações que não devem ser visitadas; ou por serem ineficazes ou porque já foram visitadas. O trabalho também é o primeiro na bibliografia consultada a utilizar codificação com base em malhas fundamentais, denominada de *Malhas Fundamentais Sem Corte*. Não obstante, propõe um artifício engenhoso para ajuste dinâmico das taxas de mutação e cruzamento, partindo da premissa que a soma de tais taxas é sempre igual a 100%. O algoritmo foi validado usando como referência os algoritmos de Baran &

Wu (1989) e Goswami & Basu (1992).

Ramos et al (2001) usaram programação linear inteira mista para resolver o problema da Reconfiguração. Propuseram uma representação de rede baseada na lista de percursos possíveis desde uma subestação até os nós terminais da rede, que nomearam como *Path-Based*. Visando reduzir a complexidade do problema, um perfil de tensão plano e perdas nulas foram considerados, o que permitiu dispensar o cálculo do fluxo de carga.

A partir de 2002, observa-se uma predominância dos AGs como ferramenta de busca nos trabalhos que abordam o tema. Mais precisamente, de variações de AGs. Neste ano, Huang (2002) propôs um AG cuja principal contribuição é o uso de lógica nebulosa para ponderação dos objetivos a serem minimizados: perdas, violações de tensão nodal, sobrecarga nos ramos e número de chaveamentos. Representou a rede por um vetor binário com comprimento igual ao número de chaves, no qual os valores 1 e 0 representam, respectivamente, chaves fechadas e abertas. Tal representação de rede é referenciada por *Vetor Característico*.

No mesmo ano, Zhu (2002) apresenta um AG com mutação dinâmica: sua taxa começa em 100% e diminui no curso das gerações até chegar ao valor mínimo de 5%. A codificação adotada é uma simplificação daquela proposta por Nara et al (1992). A diferença é que não se faz necessário incluir no código a identificação da chave e do ramo, pois admite-se que há apenas uma chave por ramo. Esta codificação é denominada de *Lista de Chaves Abertas*.

King et al (2003) retomaram a idéia de codificação baseada nas malhas fundamentais da rede e apresentaram um AG inspirado no trabalho de Zhu (2002). O artigo propõe montar uma tabela, na qual cada linha armazena as chaves pertencentes a uma malha fundamental. As repetições verificadas entre linhas devem ser então

retiradas, de modo que nenhuma chave ocorra mais de uma vez na tabela. Define-se então, um indivíduo cujo número de genes é igual ao de malhas fundamentais e onde o valor armazenado no gene  $i$  denota a chave a ser aberta na malha fundamental  $i$ .

A técnica proposta por King et al (2003) melhora a codificação por *Lista de Chaves Abertas* porque deixa de representar soluções onde uma mesma malha fundamental é aberta mais de uma vez. No entanto, ela exclui algumas configurações factíveis ao retirar as duplicidades ainda na fase de construção da tabela. Tal técnica é referenciada neste texto como codificação por *Malhas Fundamentais Com Corte*. Vale ressaltar que a opção oposta, de não retirar as duplicidades no início do algoritmo, implica em um espaço de busca maior e com várias configurações infactíveis. Por exemplo, soluções em que uma chave é aberta mais de uma vez integram o espaço de busca.

Um ano depois, King et al (2004) propuseram outro AG usando codificação por malhas fundamentais. Desta vez, usaram taxas de cruzamento e de mutação ajustadas com base em critérios calculados com base em lógica nebulosa. Basicamente, o algoritmo usa controladores que fundamentados nas aptidões médias de duas gerações anteriores ajustam as taxas de mutação e cruzamento da geração atual.

Prasad et al (2005) retomaram a codificação proposta em Lin et al (2000), contornando o problema de abrir uma mesma chave duas vezes pela escolha aleatória de outra chave da malha fundamental para abrir sempre que tal condição ocorre; porém, não resolveram o problema de abrir duas chaves distintas de uma mesma malha. Logo, operam em um espaço de busca com algumas configurações de rede infactíveis. O AG tem uma taxa de mutação variável que é controlada por lógica nebulosa em função da variação na aptidão média da população e do seu desvio padrão.

No mesmo ano, outras duas codificações de rede foram propostas por

Hong & Ho (2005) e Ramos et al (2005). O primeiro trabalho aplica uma representação chamada de Número *Prufer* ao problema da Reconfiguração de rede. Segundo sua codificação, uma rede radial com uma única subestação é convertida em uma lista de inteiros com exatamente o número de nós da rede menos dois. Trata-se de uma codificação nodal, capaz de representar todas as configurações radiais de um grafo completo com o número de nós da rede. Além disso, o artigo usa lógica nebulosa para ponderar mais precisamente os objetivos do problema e define um artifício engenhoso para tornar a penalização de soluções infactíveis fraca no início do algoritmo e forte nas últimas gerações. Ramos et al (2005) por sua vez apresentaram um AG que utiliza a codificação *Path-Based* proposta em Ramos et al (2001). No final do artigo forneceram um algoritmo para determinação dos percursos exigidos pela codificação, contudo, utilizaram um critério subjetivo de distância elétrica para podar a árvore de caminhos.

Ainda em 2005, Delbem et al (2005) apresentaram outra representação de rede baseada em percursos. Na solução proposta, apenas um caminho precisa ser armazenado para cada nó terminal da rede – em termos de grafos, um caminho para cada folha na árvore. Nomearam sua codificação de representação por *Cadeia Principal* e definiram operadores genéticos específicos que geram apenas descendentes radiais conexos. O artigo tem idéias interessantes, como uma formulação geral para o problema da Configuração de redes e uma tabela onde são armazenadas as melhores soluções para cada um dos objetivos da solução. Aparentemente, as desvantagens da representação proposta são o tamanho dos indivíduos e a dificuldade de implementar os operadores genéticos.

Já em 2006, Das (2006) propôs um algoritmo com um processo de busca semelhante ao de Kashem et al (1999): fecha a malha mais promissora da rede em termos da diferença de potencial na sua corda, testa cada uma das configurações obtidas

abrindo tal malha e, então, escolhe a melhor. O processo continua enquanto houver malhas com potencial de melhoria do objetivo definido. A diferença é que em vez de conduzir a busca pelo objetivo balanceamento, optou por considerar vários objetivos calculados por meio de lógica nebulosa: perdas, violação de tensão nodal, suportabilidade e balanceamento. A melhor opção é determinada pelo critério Máx-Min. Embora o algoritmo seja engenhoso, seu processamento é do tipo guloso, podendo conduzir a uma solução final indesejada.

Mendoza et al (2006a) retomaram os AGs com codificação por Malhas Fundamentais Sem Corte, mas se diferenciaram dos trabalhos anteriores por terem incluído no algoritmo proposto operadores especiais que reduzem a geração de descendentes que violam as restrições de radialidade e conexão. No entanto, não conseguiram zerar a geração destes descendentes infactíveis.

No ano seguinte, Shin et al (2007) propuseram mais um algoritmo baseado na Troca de Ramos. O algoritmo proposto busca uma configuração de rede que atenda simultaneamente aos objetivos de minimização de perdas e maximização da estabilidade e usa um AG apenas para iniciar a troca de ramos, caso não haja uma configuração factível.

No início de 2008, três artigos tratando da Configuração de redes foram publicados na revista *IEEE Transactions on Power Systems*: Wang & Cheng (2008), Zhang et al (2008) e Enacheanu et al (2008).

Wang & Cheng (2008) apresentaram um algoritmo cuja metáfora é a Simulação do Crescimento de Plantas e modelaram a rede de distribuição usando a idéia de malhas fundamentais. De acordo com o algoritmo, configurações factíveis geradas a partir da configuração atual e que possuam aptidão melhor que a da configuração inicial, são definidas como folhas. A geração das folhas é feita por um algoritmo que,

para cada malha  $i$ , testa as configurações geradas variando o local de abertura de  $i$ , enquanto as outras malhas permanecem em sua condição base. O procedimento é iterativo e consiste em selecionar uma folha dentre as disponíveis em cada iteração para tomar o lugar da configuração atual. Embora o artigo forneça uma descrição dos estados possíveis das chaves em uma configuração radial qualquer, apresentando a idéia de chave temporariamente fechada como um recurso para garantir a geração apenas de configurações radiais, não fica claro como tal conceito influencia no algoritmo de geração das configurações folha.

Zhang et al (2008) combinaram troca de ramos e AGs para realizar, respectivamente, Reconfiguração de rede e programação de bancos de capacitores chaveados. O algoritmo proposto é adaptativo, no sentido que os operadores genéticos aplicam mudanças mais significativas quando os pais têm uma aptidão baixa em relação à média da população e menos significativas em caso contrário. Propõem alterar a troca de ramos usando uma equação baseada na premissa de que a configuração com perda total mínima é aquela cujas perdas nas malhas são mínimas. Contudo, supõem que a eventual carga transferida entre ramos é uma variável contínua.

Enacheanu et al (2008) apresentam um AG com uma nova representação de rede baseada na teoria de *Matroids*. Esta codificação possui duas características extremamente importantes: garante que apenas configurações radiais pertencem ao estado de busca e possui operadores genéticos que geram somente descendentes radiais conexos. Ela ainda pode ser usada igualmente em grafos planares e não planares. Sua desvantagem aparente é a complexidade computacional, visto que a todo momento é preciso operar em árvores e co-árvores, determinar malhas ou obter conjuntos união e interseção entre ramos.

Ainda em 2008, Carreno et al (2008) propuseram uma nova estratégia de

codificação que consiste em representar uma rede radial conexa por dois conjuntos: ramos fechados e ramos abertos. O código independe da posição relativa dos elementos em cada um destes conjuntos, de forma similar a *Messy Representation* proposta por Goldberg (1989). Definiram um operador de recombinação específico, que garante descendentes radiais conexos, bem como uma variação do AG que nomearam de *Melhoria Local*, inspirada na heurística Troca de Ramos. Essa codificação é nomeada de *Conjunto de Arestas NF-NA*.

Fontan (2008) propôs em seu trabalho de mestrado a codificação por *Vetor Característico Melhorado* visando atender ao compromisso entre complexidade computacional e redução no espaço de busca. O código é essencialmente o mesmo do vetor característico original, mas foram apresentados um algoritmo de inicialização e operadores genéticos específicos que permitem obter uma redução significativa no espaço de busca com um pequeno acréscimo na complexidade do AG.

Por fim, Queiroz et al (2009) apresentaram um AG híbrido e adaptativo para solução da Configuração considerando vários níveis de carregamento. A porção híbrida do AG se deve a realização de uma busca local inspirada na troca de ramos, que no artigo ocorre a cada 50 gerações. O termo adaptativo se deve a variação da taxa de mutação no curso das gerações em função da diversidade da população, estimada pela diferença entre melhor aptidão e a média das aptidões. Adotaram uma representação de rede chamada de *Network Random Keys*, proposta por Rothlauf et al (2002) adaptando a representação *Random Keys* que fora proposta por Bean (1993) visando resolver problemas de sequenciamento.

## **2.2 Restabelecimento de Serviço**

Outra abordagem do problema da Configuração de redes consiste em determinar a

configuração ótima para restabelecer o serviço quando ocorre uma falta. O problema difere da Reconfiguração principalmente porque possui fortes restrições de tempo. A urgência em obter uma resposta repercute nas técnicas e algoritmos propostos para resolver o problema. Predominam os sistemas especialistas como ferramenta de solução, algoritmos de fluxo de carga aproximados para o cálculo do estado da rede de distribuição e modelos simplificados para representar equipamentos. Mesmo assim, há algumas exceções à regra, principalmente AGs.

De forma geral, sistemas especialistas são algoritmos formados por listas de regras específicas de determinado assunto, que devem ser seguidas passo-a-passo. Liu et al (1988), Hsu et al (1992), Sarma et al (1994) e Miu et al (1998) são exemplos de trabalhos que propõem diferentes sistemas especialistas para resolver o Restabelecimento de serviço. Usualmente, estas soluções são dependentes da configuração inicial do sistema, consideram apenas alguns níveis de transferência de carga entre alimentadores e, embora tenham menor complexidade computacional que outras técnicas, apresentam dificuldades para realizar estudos em grandes redes ou em sistemas operando perto de seu limite de carregamento. Por nível de transferência de carga considere a seguinte definição recursiva: nível 0 é o próprio alimentador; e nível  $k \in N^*$  compreende qualquer alimentador na vizinhança do nível  $k-1$ .

Além dos sistemas especialistas mencionados, foram propostas combinações desta técnica com diferentes algoritmos de busca visando melhoria de desempenho. Aoki et al (1989), Zhou et al (1997), Siqing et al (1998), Nagata et al (2002) e Chen et al (2002) apresentaram combinações de sistemas especialistas, respectivamente, com o método do gradiente efetivo, programação linear, AGs, teoria de agentes e redes de petri. Destaca-se na lista o trabalho de Siqing et al (1998), porque propuseram uma variação do vetor característico na qual apenas as chaves posicionadas

no interior e na fronteira da área desenergizada são representadas. Ainda sobre algoritmos híbridos, Carvalho et al (2007) combinaram um algoritmo evolucionário com programação dinâmica para solução conjunta do problema do Restabelecimento de serviço e do sequenciamento de manobras.

Há também trabalhos que adotam critérios heurísticos, dos quais merecem destaque Morelato & Monticelli (1989) e Shirmohammadi (1992). O primeiro trabalho apresenta um método de busca que permite podar a árvore de decisão do problema com base em conhecimento do especialista, enquanto o segundo é uma extensão do trabalho proposto por Shirmohammadi & Hong (1989) para Reconfiguração de rede. Para Shirmohammadi (1992), o estado inicial é a rede malhada com todas as chaves fechadas e o procedimento de busca consiste em abri-las sucessivamente de acordo com um critério heurístico – uma abertura sequencial.

Mais recentemente, cada vez mais trabalhos têm adotado metaheurísticas para resolução do problema (FUKUYAMA & CHIANG, 1995; BRETAS et al, 1998; MATOS & MELO, 1999; TOUNE et al, 2002; WATANABE & NODU, 2004; CHEN et al, 2005; WATANABE et al, 2006; MANJUNATH & MOHAN, 2007; LIU & GU, 2007; SANTOS et al, 2008). Esta abordagem permite dotar as soluções de independência com relação ao estado inicial da rede e estende a busca realizada a qualquer nível de transferência de carga entre alimentadores. Embora implique maior complexidade computacional, contrapondo a restrição de tempo inerente ao Restabelecimento de serviço. Para atender tal restrição, artifícios têm sido usados visando reduzir a complexidade dos algoritmos; seja pela simplificação no cálculo da função objetivo, pela utilização de algoritmos com ajuste dinâmico dos parâmetros ou mesmo, implementando regras para reduzir o espaço de busca da solução.

Restringindo a análise ao tema representação de redes, alguns trabalhos

merecem destaque:

- Fukuyama & Chiang (1995) apresentaram um AG com processamento paralelo, codificação Predecessor 1 e reparação nos operadores genéticos;
- Bretas et al (1998) foram os primeiros a usar representação por cadeia principal, todavia sem detalhar os operadores genéticos associados;
- Watanabe et al (2006) aprimoraram o algoritmo proposto em Watanabe & Nodu (2004) visando melhorar seu desempenho em termos de tempo de processamento. Em particular, propõem usar uma representação proposta por Raidl & Julstrom (2003) e referenciada como *Conjunto de Arestas*, cujos operadores genéticos são específicos e há garantia de geração apenas de configurações radiais conexas; e
- Santos et al (2008) utilizaram a representação *Nó-Profundidade*, que pode ser vista como a segunda geração da codificação por Cadeia Principal, sendo mais compacta e implementando operações genéticas com menor complexidade.

### **2.3 Planejamento de Expansão**

O Planejamento é outra versão do problema da Configuração de Rede frequentemente abordado na literatura. Ele difere dos outros enfoques principalmente por apresentar menos restrições de tempo e envolver maiores custos. Por este motivo, a literatura sobre o tema apresenta em geral formulações mais rigorosas, algoritmos completos de fluxo de carga e considera períodos de tempo da ordem de anos para guiar com a maior precisão possível a busca por uma solução ótima. Alguns trabalhos de destaque são relacionados nesta seção.

Miranda et al (1994) empregaram AGs para o Planejamento dinâmico de novos alimentadores e expansão/construção de subestações, considerando os objetivos de minimização de custos e melhoria da confiabilidade das soluções como critérios da

otimização combinatória. Representaram as eventuais alocações de novas subestações ou expansões como arestas candidatas fictícias no AG; se a chave de uma aresta fictícia estiver fechada na configuração final significa que o equipamento foi efetivamente incluído no Planejamento. Desse modo, obtiveram uma equivalência entre Planejamento e Reconfiguração. Uma variação da codificação predecessor, nomeada de *Predecessor 2*, foi adotada. Nesta representação, o gene informa qual dentre as arestas incidentes em um determinado nó deve ser habilitada.

Jonnavithula & Billinton (1996) e Tang (1996) utilizaram, respectivamente, resfriamento simulado e programação inteira mista para o Planejamento de sistemas de distribuição. Em comum, assumiram a confiabilidade da rede como objetivo principal. Peponis & Papadopoulos (1997) propuseram um algoritmo que combina programação dinâmica e troca de ramos para o Planejamento de expansão integrado à alocação de vários equipamentos como: reguladores de tensão, bancos de capacitores e ampliação de subestações.

No ano seguinte, Lin et al (1998) apresentaram um AG com representação por vetor característico que engloba o Planejamento e a alocação de novas subestações de forma semelhante a Miranda et al (1994). Consideraram como objetivos a minimização do investimento inicial e das perdas ativas resultantes.

Ferreira et al (2001) apresentaram outro algoritmo evolucionário com codificação por vetor característico. Embora, cientes de que a maior desvantagem desta técnica é a criação de descendentes ineficazes, propuseram um operador genético específico que segundo os autores garante a geração de descendentes eficazes e o nomearam de *Path-Interchange Algorithm* (PIA). Tal operador é descrito da seguinte maneira: i) selecione um par de soluções radiais conexas  $u^A$  e  $u^B$ ; ii) selecione dois nós  $n_1$  e  $n_2$ ; iii) descubra o caminho  $p^A$  entre  $n_1$  e  $n_2$  em  $u^A$ , de modo que  $u^A = p^A + \sim p^A$  ( $\sim p^A$

o grafo complementar de  $p^A$  em  $u^A$ ); iv) descubra o caminho  $p^B$  entre  $n_1$  e  $n_2$  em  $u^B$ , de modo que  $u^B = p^B + \sim p^B$  ( $\sim p^B$  o grafo complementar de  $p^B$  em  $u^B$ ); v) troque os caminhos:  $u^A = p^B + \sim p^A$  e  $u^B = p^A + \sim p^B$ . O problema é que os caminhos via de regra têm um número diferente de ramos, conduzindo a configurações que violam a regra de número de ramos e cordas constante.

No ano seguinte, o Planejamento de alimentadores foi abordado por Boulaxis & Papadopoulos (2002), que formularam um método baseado em programação dinâmica que incorpora dados de sistemas de informações geográficas (SIGs) ao problema, permitindo contabilizar restrições físicas associadas à geografia da área.

Souza et al (2003) apresentaram um AG com uma codificação empilhada, ao qual adicionaram uma memória genética que evita o cálculo da função de aptidão sempre que possível. Assim, mitigaram expressivamente o esforço computacional. Esta técnica empilhada serviu como inspiração para os códigos sequenciais apresentados neste trabalho de tese.

Já em 2006, duas publicações adotaram AGs multiobjetivo para o Planejamento de expansão. Carrano et al (2006) propuseram um NSGA-II com os objetivos de reduzir investimento, custo operacional e energia não-suprida em caso de falha. Para automatizar a criação do conjunto de arestas candidatas, apenas as arestas conectando um determinado nó aos seus vizinhos mais próximos são consideradas. O número de arestas incidentes consideradas para um determinado nó é calculado em função de sua distância média em relação à rede: são consideradas  $mxm$  arestas para o nó com menor distância média,  $mnn$  para o nó com maior distância média ( $mxm > mnn$ ) e um valor interpolado dentro deste intervalo para os demais. Mendoza et al (2006b) por sua vez, apresentaram um comparativo entre os dois principais algoritmos do tipo

multiobjetivo: NSGA e SPEA. Tendo formulado o problema como de minimização de custos e de um índice de confiabilidade definido no artigo.

A principal vantagem dos algoritmos multiobjetivo em relação à formulação convencional é o tratamento preciso de cada um dos objetivos do problema de forma independente; isto é, sem a necessidade de definir uma função composta por parcelas relativas a cada um dos objetivos. Contudo, ainda não há consenso sobre em que condições e número de objetivos a melhoria de desempenho supera o aumento de complexidade inerente a tais algoritmos. Restringindo a análise ao tema codificação de redes, o fato é que uma codificação que representa somente configurações radiais conexas e com complexidade mínima é bastante desejável e pode melhorar as soluções obtidas, seja qual for o tipo de AG.

## **2.4 Síntese dos Três Enfoques da Configuração**

Como é possível observar nas Seções 2.1, 2.2 e 2.3, há um grande número de publicações sobre a Configuração de Redes e, igualmente, um número significativo de técnicas utilizadas. Sarfi et al (1994b), Nara (1997), Nara & Song (2002) e Rivas-Dávalos et al (2007) reportam revisões bibliográficas que, direta ou indiretamente, contemplam outros trabalhos sobre o tema.

Com base na revisão realizada, observa-se que mesmo tendo sido propostas, respectivamente, nas décadas de setenta e oitenta, a abertura sequencial e a troca de ramos são as técnicas mais utilizadas na literatura sobre Configuração de redes. Sua principal vantagem é a simplicidade. No caso da abertura sequencial é possível definir uma infinidade de critérios para escolher a melhor chave para ser aberta em um determinado passo do algoritmo, considerando diferentes objetivos. A troca de ramos é igualmente suscetível aos diferentes critérios para seleção da melhor transferência de

carga em cada instante. Entretanto, a restrição é que ambos são essencialmente algoritmos gulosos e, em cada iteração, fazem a melhor escolha possível sem considerar a solução como um todo. Tal característica conduz a uma pesquisa restrita a uma fração do espaço de busca e pode acarretar em soluções indesejadas.

Visando contornar as deficiências dos algoritmos gulosos mencionados e, principalmente, conduzir uma busca considerando todas as configurações de rede possíveis, outras técnicas de busca têm sido propostas. Dentre estas alternativas, a mais comum são os AGs; em particular, algoritmos modificados visando resolver sua maior limitação: tempo de execução. Embora tenham sido obtidos alguns resultados promissores com AGs para redes de médio porte, em sistemas de grande porte o espaço de busca pode ser excessivamente extenso.

A impressão é que mesmo passados vinte anos, a idéia de combinar técnicas heurísticas com algoritmos de otimização, apresentada na seção Discussão de Liu et al (1988), ainda é factível. Precisamente, associar heurísticas e algoritmos evolucionários pode ser uma alternativa para transpor as limitações atuais. Essa tendência já pode ser observada nesta tese e nos trabalhos de Carreno et al (2008) e Queiroz et al (2009), que inseriram no algoritmo evolutivo uma rotina de busca local baseada em heurísticas específicas dos sistemas de distribuição – *hibridização*.

Na Tabela 2.1, apresenta-se um resumo da revisão bibliográfica realizada nas seções anteriores, enfatizando alguns artigos clássicos sobre o tema e apresentando uma listagem com várias técnicas de representação de redes. As principais representações nesta lista foram incluídas em um *Conjunto de Codificações Ótimas*, tendo em comum a garantia de que somente configurações radiais conexas são codificadas. Logo, elas definem essencialmente a referência para o trabalho proposto.

**Tabela 2.1 – Resumo da revisão bibliográfica sobre Configuração de Redes de Distribuição com ênfase nas técnicas de representação das redes.**

Referência	Enfoque	Técnica	Limitações
Merlin & Back, 1988	Reconfiguração	Propõe Abertura Sequencial de Chaves	Algoritmo Guloso
Cinvanlar et al, 1988	Reconfiguração	Propõe Troca de Ramos	Algoritmo Guloso
Baran & Wu, 1989	Reconfiguração	Aprimora Troca de Ramos	Algoritmo Guloso
Morelato & Monticelli, 1989	Formulação Geral	Realiza uma busca na árvore de possibilidades definida pelo Vetor Característico da Rede e faz podas para reduzir o espaço de busca	Requer a criação da árvore de possibilidade, o que mesmo com podas pode ser inviável para sistemas maiores
Shirmohammadi & Hong, 1989	Reconfiguração	Aprimora Abertura Sequencial	Algoritmo Guloso
Nara et al, 1992	Reconfiguração	AG com representação similar a Lista de Chaves Abertas. A diferença é que considera ramos com mais de uma chave	A codificação utilizada não garante que somente configurações radiais conexas estejam no espaço de busca
Shirmohammadi, 1992	Restabelecimento	Adapta o artigo de Shirmohammadi & Hong (1989)	Algoritmo Guloso
Miranda et al, 1994	Planejamento	Adotaram uma variação da codificação Predecessor 1 que foi nomeada de Predecessor 2, na qual o gene informa qual dentre as arestas incidentes em um determinado nó deve ser habilitada	A codificação utilizada não garante que somente configurações radiais conexas estejam no espaço de busca
Sarfi et al, 1995	Reconfiguração	Utilizam Troca de Ramos, mas antes reduzem a rede usando um algoritmo de partição	Algoritmo Guloso
Fukuyama & Chiang, 1995	Restabelecimento	Propõem um AG Paralelo e codificam a rede usando Predecessor 1 com reparação de inactiváveis	Resolve o problema da geração de inactiváveis, criando outro: reparação em geral diminui a performance
Lin et al, 1998	Planejamento	Utilizaram um AG com codificação por Vetor Característico	A codificação utilizada não garante que somente configurações radiais conexas estejam no espaço de busca
Siqing et al, 1998	Restabelecimento	Combinam sistemas especialistas e AGs, mas apenas as chaves internas ou na fronteira da área desenergizada foram incluídas no Vetor Característico	A codificação utilizada não garante que somente configurações radiais conexas estejam no espaço de busca
Kashem et al, 1999	Reconfiguração	Modifica Troca de Ramos, mas considerando o	Algoritmo Guloso

Referência	Enfoque	Técnica	Limitações
		balanceamento de carga	
Lin et al, 2000	Reconfiguração	Combinaram abertura sequencial, busca tabu e AG com codificação por Malhas Fundamentais Sem Corte	A codificação utilizada não garante que somente configurações radiais conexas estejam no espaço de busca
Mantovani et al, 2000	Reconfiguração	Desenvolveram um algoritmo de busca na árvore de possibilidades com podas determinadas por um critério associado à queda de tensão	Requer a criação da árvore de possibilidade, o que mesmo com podas pode ser inviável para sistemas maiores.
Ferreira et al, 2001	Planejamento	Algoritmo Evolucionário com codificação por Vetor Característico e operador genético específico que evita a geração de infactíveis	O operador nomeado de PIA como descrito no artigo não garante a manutenção de configurações radiais conexas.
Zhu, 2002	Reconfiguração	Propôs um AG com codificação por Lista de Chaves Abertas	A codificação utilizada não garante que somente configurações radiais conexas estejam no espaço de busca
King et al, 2004	Reconfiguração	AG com taxas de mutação e cruzamento controladas por Lógica Nebulosa e codificação por Malhas Fundamentais Com Corte	Junto com as soluções em que uma chave é aberta mais de uma vez, esta estratégia exclui algumas configurações factíveis
Watanabe & Nodu, 2004	Restabelecimento	AG com representação por Lista de Arestas e operadores específicos	Integra o <b>Conjunto de Codificações Ótimas</b>
Hong & Ho, 2005	Reconfiguração	AG com Lógica Nebulosa para cálculo dos objetivos do problema e codificação por Número Prufer	Codifica todas as representações radiais conexas do grafo completo, ignorando a lista de arestas candidatas real
Prasad et al, 2005	Reconfiguração	AG com codificação por Malhas Fundamentais Sem Corte, mas com tratamento para o caso de abrir uma chave mais de uma vez	A codificação utilizada não garante que somente configurações radiais conexas estejam no espaço de busca
Delbem et al, 2005	Formulação Geral	Algoritmo Evolucionário com representação por Cadeia Principal e operadores genéticos específicos	Integra o <b>Conjunto de Codificações Ótimas</b>
Ramos et al, 2005	Reconfiguração	AG com codificação Path-Based	Garante soluções radiais conexas, mas não representa todas as soluções. Requer checagem toda vez que um novo indivíduo é gerado
Das, 2006	Reconfiguração	Busca sequencial com diferença de tensão como critério para fechar chave	Algoritmo Guloso

Referência	Enfoque	Técnica	Limitações
		NA e vários objetivos calculados com Lógica Nebulosa para abrir NF	
Mendoza et al, 2006a	Reconfiguração	AG com representação por Malhas Fundamentais Sem Corte e operadores genéticos melhorados	A codificação utilizada não garante que somente configurações radiais conexas estejam no espaço de busca
Shin et al, 2007	Reconfiguração	Melhora troca de ramos e usa um AG para iniciá-lo	Algoritmo Guloso
Enacheanu et al, 2008	Reconfiguração	AG com representação usando teoria dos Matroids e operadores genéticos que não geram soluções infactíveis	Integra o <b>Conjunto de Codificações Ótimas</b>
Fontan, 2008	Reconfiguração	AG com representação por Vetor Característico Melhorado, incluindo inicialização e operadores genéticos específicos.	A codificação utilizada não garante que somente configurações radiais conexas estejam no espaço de busca
Carreno et al, 2008	Reconfiguração	AG proposto por Chu & Beasley (1997) com representação por Conjunto de Arestas NF-NA, incluindo inicialização, operador de recombinação específico e rotina de busca local.	Integra o <b>Conjunto de Codificações Ótimas</b>
Santos et al, 2008	Restabelecimento	Algoritmo Evolucionário com representação por Nó-Profundidade, incluindo inicialização e operadores genéticos específicos.	Integra o <b>Conjunto de Codificações Ótimas</b>
Queiroz et al, 2009	Reconfiguração	AG Híbrido e Adaptativo com representação por <i>Network Random Keys</i> , não demanda operadores genéticos específicos.	Integra o <b>Conjunto de Codificações Ótimas</b>

## Capítulo 3

# AGs para Configuração de Redes

---

Este capítulo está organizado em quatro seções: Fundamentos sobre AGs, Estratégias de Codificação, Cálculo de Aptidão e Fluxograma do AG Proposto. O propósito é apresentar os conceitos básicos sobre AGs; expor com clareza o estado da arte no que tange a representação de redes e suas métricas de comparação; formular a função objetivo do problema como tratado nesta tese para, partindo dela, obter a função de aptidão a ser efetivamente usada no AG; e proporcionar uma visão geral do AG desenvolvido, incluindo alguns módulos e opções que não serão explorados nos resultados da tese (Capítulo 7) por não pertencerem ao foco principal do trabalho.

### 3.1 Fundamentos sobre AGs

Os AGs são métodos probabilísticos de otimização e busca, inspirados nos mecanismos de evolução natural como ocorrem na natureza. Foram introduzidos por John Holland em parceria com alguns de seus colegas e por estudantes da Universidade de Michigan durante a década de 60, porém realmente se popularizaram a partir da publicação do livro *Genetic Algorithms in Search, Optimization and Machine Learning*, de autoria de um de seus alunos, David Goldberg (HOLLAND, 1975; GOLDBERG, 1989).

O princípio de funcionamento do algoritmo reside na sobrevivência do mais apto, da forma como o naturalista e fisiologista Charles Darwin enunciou em seu

livro *A origem das espécies* (DARWIN, 1859). De acordo com Darwin, “Quanto melhor um indivíduo se adaptar ao seu meio ambiente, maior será sua chance de sobreviver e gerar descendentes”.

Os AGs emulam o processo de evolução natural fornecendo um ambiente computacional no qual soluções codificadas de um problema, denominadas *cromossomos*, representam espécimes; uma *função de aptidão*, cujo argumento é um cromossomo, determina a chance do espécime de gerar descendentes; operações matemáticas entre cromossomos simulam o processo de reprodução como ele ocorre na natureza; e iterações do algoritmo, chamadas de *gerações*, modelam a passagem do tempo. A terminologia própria dos AGs, ainda define como *genes*, as unidades de informação de um cromossomo e o par formado por um cromossomo e sua aptidão como um *indivíduo*.

Durante cada geração, uma população de indivíduos compete pela oportunidade de gerar descendentes em um processo chamado de *seleção*. O algoritmo garante que os mais aptos levem vantagem em média, mas o processo de reprodução é estocástico e eventualmente indivíduos menos aptos conseguem participar. Concluída a etapa de seleção de indivíduos para reprodução, começa a simulação da reprodução propriamente dita. Uma estratégia de combinação foi definida para emular a recombinação cromossômica que ocorre na fecundação (operador de *cruzamento*). O processo consiste em efetivamente trocar partes de dois cromossomos nomeados de *pais* para gerar novos cromossomos chamados de *filhos*. Adicionalmente, uma perturbação aleatória foi estabelecida com o intuito de simular as falhas de transmissão de material genético de uma geração para outra (operador de *mutação*). Tais operações ocorrem com probabilidades definidas, havendo casos em que um indivíduo passa para a geração seguinte sem sofrer nenhuma *operação genética*.

O AG Básico, considerando  $S(t)$  como a população na geração  $t$  e que  $S(t)$  não compete com  $S(t-1)$ , tem a seguinte estrutura:

Algoritmo 3.1 – AG Básico

```

1   t ← 0
2   Iniciar S(t)
3   Avaliar S(t)
4   Enquanto(testa critério de parada)
5       t ← t+1
6       Selecionar S(t) a partir de S(t-1)
7       Aplicar Cruzamento e Mutação sobre S(t)
8       Avaliar S(t)

```

Para que seja possível usar o Algoritmo 3.1 ou alguma variação dele para efetivamente resolver determinado problema, alguns parâmetros de entrada e uma série de algoritmos específicos precisam ser definidos (Tabela 3.1). Devido ao grande número de aplicações de AGs para solução dos mais diversos problemas é possível afirmar que existem na literatura algoritmos que se dispõem a melhorar definição e controle de todos os parâmetros de entrada enumerados; e igualmente, dezenas de

**Tabela 3.1 – Descrição geral dos módulos de um AG tomando como base o Algoritmo 3.1.**

<b>Módulo</b>	<b>Requer</b>	<b>Tipo</b>	<b>Linha no Algoritmo 3.1</b>
Criação da População	Número de indivíduos	Parâmetro	2
	Estratégia de Codificação	Dado de projeto (dependente do problema)	
Avaliação	Algoritmo de Decodificação	Depende da Codificação	3 e 8
	Função de Aptidão	Dado de projeto (dependente do problema)	
Critério de Parada	Critério de Parada	Dado de projeto (independente do problema)	4
	Parâmetros de Controle	Parâmetro	
Seleção	Algoritmo de Seleção	Dado de projeto (independente do problema)	6
	Número de indivíduos	Parâmetro	
Cruzamento	Algoritmo	Dado de projeto (dependente da Codificação)	7
	Probabilidade ou taxa	Parâmetro	
Mutação	Algoritmo	Dado de projeto (dependente da Codificação)	7
	Probabilidade ou taxa	Parâmetro	

variações para implementar cada um dos módulos. Isso, sem discutir as variantes do próprio AG em relação ao Algoritmo 3.1.

A constante em todo o arcabouço de possibilidades quando se trabalha com AGs é que a estratégia de codificação e a função de aptidão são as partes do algoritmo inerentemente dependentes da aplicação. Em que, o termo codificação consiste na forma como as soluções de um problema são traduzidas em cromossomos, com impacto no algoritmo de decodificação e, para alguns casos particulares, nos operadores genéticos – chamados de *operadores específicos*. Com relação à função de aptidão, há uma dependência da aplicação ainda mais significativa, visto que cabe unicamente a ela informar o AG sobre o problema a otimizar. Se esta função é inapropriada, a solução ótima na perspectiva do AG fica diferente da solução ótima do problema real. É impossível para o AG obter uma resposta satisfatória porque, popularmente, o algoritmo está *míope*.

Com base nos comentários acima, as próximas seções são restritas à fundamentação das estratégias de codificação, a formulação da função de aptidão e ao resumo do AG como um todo. Detalhes adicionais sobre os demais módulos, podem ser consultados na literatura específica sobre o tema (GOLDBERG, 1989; MICHALEWICZ, 1996; SAKAWA, 2002).

### **3.2 Codificações**

De acordo com o jargão próprio dos AGs, obtido majoritariamente mediante apropriação da genética, é dado o nome de *fenótipo* à solução propriamente dita de um problema, enquanto sua versão codificada correspondente é chamada de *genótipo*. Desse modo, podemos definir formalmente *codificação* como a transformação fenótipo-genótipo e *decodificação* como a transformação no sentido contrário: genótipo-fenótipo.

Ao conjunto formado por todos os fenótipos é dado o nome de *Espaço de Busca dos Fenótipos* ( $\Phi_f$ ). De forma equivalente, define-se o *Espaço de Busca dos Genótipos* ( $\Phi_g$ ) como sendo formado por todos os genótipos possíveis (ROTHLAUF, 2006).

Apresentam-se nesta seção alguns conceitos sobre o tema codificações, organizados em quatro subseções: na primeira, são abordados fundamentos de teoria dos grafos e uma abstração da rede de distribuição que permite considerá-la como um grafo de setores; a segunda, denominada de Um exemplo Ilustrativo, apresenta um exemplo prático da codificação de variáveis binárias, inteiras e reais, discutindo fundamentos sobre o procedimento de decodificação e o espaço de busca resultante; a seção seguinte aprofunda a discussão sobre codificações, especificamente sobre métricas utilizadas na comparação de representações; na quarta seção, tomando como base uma rede de exemplo, apresenta-se uma síntese do estado da arte em termos de representações de rede.

### 3.2.1 Grafo de Setores

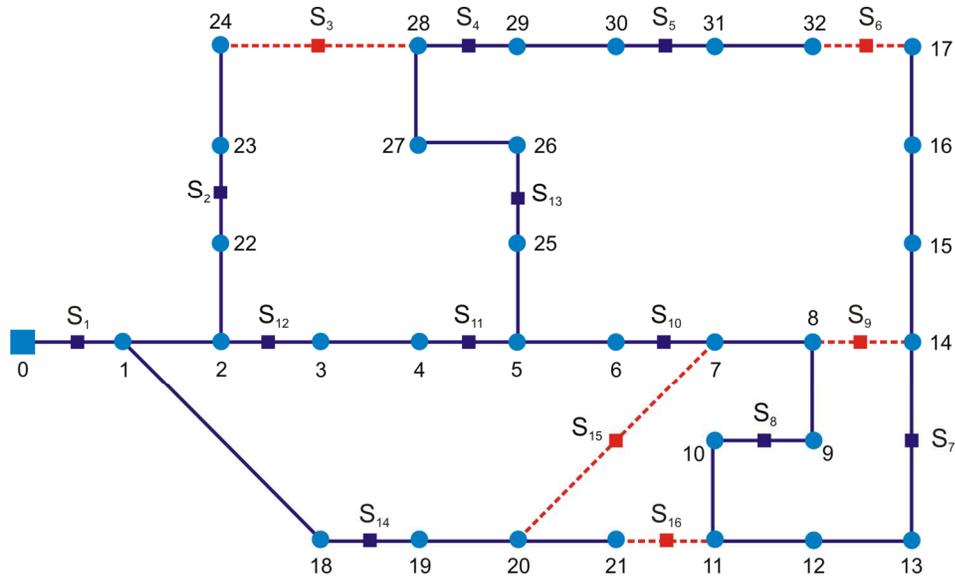
Sob o ponto de vista topológico, os sistemas de distribuição podem ser modelados por grafos. O principal benefício desta formulação é trazer para o estudo de Configuração de redes conceitos, técnicas e teoremas definidos com rigor matemático e já estabelecidos pelo uso em outras áreas de pesquisa. Por definição, um grafo  $G$  é formado por um conjunto  $X$  de nós e um conjunto  $A$  de arestas – notação  $G(X, A)$ ; cada aresta  $a_{ij}$  representa uma relação não-orientada entre dois elementos de  $X$ , e se diz que estes elementos são adjacentes.  $n(X)$  e  $n(A)$  são o número de elementos de  $X$  e  $A$ , respectivamente.

Tomando-se *setor* como sendo uma sub-rede composta internamente por subestações, barras e ramos sem chave; e delimitada externamente por ramos com

chave, é possível construir um grafo representativo de uma rede de distribuição, denominado Grafo de Setores – notação  $G_S(X_S, A_S)$  (BRETAS et al, 1998). O conjunto  $X_S$  é formado pelos setores que compõem a rede e  $A_S$  pelos ramos que possuem chave. Em uma rede com uma única subestação, as configuração radiais conexas estabelecem árvores em  $G_S$ . Para mais de uma subestação, há várias árvores resultando em uma floresta  $T_S(X_{ST}, A_{ST})$ . Considere as seguintes definições sobre  $T_S$  e  $G_S$ :

- $X_{ST} = X_S = X_F \cup X_L$ , sendo  $X_F$  e  $X_L$  conjuntos disjuntos formados, respectivamente, por *nós fonte* e por *nós de carga*. Um nó fonte é todo aquele criado a partir de um setor que contém alguma subestação; nós de carga são criados por exclusão, compreendendo todos os demais;
- $A_S = A_{ST} \cup A_{NA}$ , sendo  $A_{ST}$  e  $A_{NA}$  conjuntos disjuntos formados, respectivamente, pelas arestas *habilitadas* e *desabilitadas* por  $T_S$  em  $G_S$ . O termo *habilitada* indica que a aresta pertence a  $T_S$ , o que equivale a dizer que sua chave está fechada. De modo equivalente, se a aresta está *desabilitada*, sua chave está aberta e ela é definida como uma *corda*. Devido a  $X_{ST} = X_S$ , toda corda incide em dois nós pertencentes à  $T_S$ . Assim, cada corda estabelece um caminho fechado em  $G_S$  nomeado de *Malha Fundamental*. O que se observa é que, seja qual for a floresta definida em  $G_S$ , o número de cordas é constante e, por conseguinte, seu número de ramos e malhas fundamentais; e
- Por definição, se  $T_S$  existe:  $n(A_{ST}) = n(X_L)$ . Ou seja, há uma aresta habilitada em  $T_S$  para cada nó de carga. Por fim, se apenas uma aresta incide em um nó de carga, ele é denominado *folha* de  $T_S$ .

Na Figura 3.1 ilustra-se uma rede de distribuição fictícia. Como adotado no Capítulo 1, um quadrado maior representa a única subestação existente e círculos representam as cargas. Quadrados menores diferenciam as arestas com chave das demais. Na Figura 3.2



**Figura 3.1 – Rede de Distribuição fictícia adotada como exemplo.**

apresenta-se o processo de criação de  $G_S$  partindo da Figura 3.1.

A vantagem de utilizar a abstração Grafo de Setores é a possibilidade de realizar a Configuração em uma rede com menos nós e arestas do que se fosse adotada diretamente a rede de distribuição. Por exemplo: na rede da Figura 3.1 há 33 barras e 37 ramos, enquanto no grafo de setores da Figura 3.2b há 11 nós e 16 arestas. Contudo, se faz necessário definir como as seguintes restrições elétricas inerentes à Configuração podem ser consideradas no modelo baseado em  $G_S$ : a) suportabilidade dos ramos, b) limites de tensão nas barras e c) capacidade das subestações.

A idéia é mapear as restrições da rede real para  $G_S$ , de modo que um setor viola uma restrição se qualquer um de seus elementos viola tal restrição. As restrições a) e b) se tornam aplicáveis a qualquer setor, pois setores podem conter barras e ramos; a) vale para todas as arestas habilitadas, pois cada aresta em  $G_S$  deriva de um ramo com chave; e c) vale somente nos setores fonte, pois somente neles há subestações. Desse modo, a abstração fica completa.

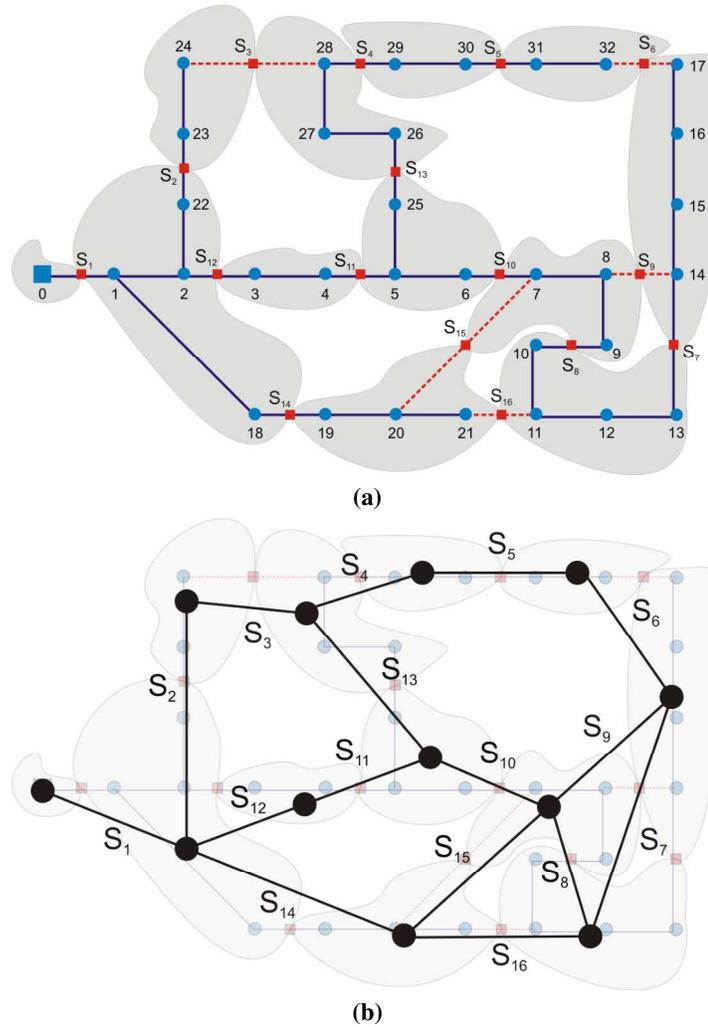


Figura 3.2 – Grafo de Setores: a) particionamento do sistema da Figura 3.1 e b)  $G_S$  resultante.

### 3.2.2 Um Exemplo Ilustrativo

Formulou-se nesta seção um exemplo ilustrativo sobre o processo de codificação e decodificação de grandezas binárias, inteiras e reais. Primeiramente, apresenta-se o exemplo supondo que foram utilizados alfabeto binário e, então, o mesmo exemplo é repetido adotando código inteiro.

Suponha que um vetor  $\mathbf{v}$  codifica três parâmetros de uma função que se deseja maximizar da seguinte maneira: os genes de 0 até 7 (numerados da direita para esquerda) representam uma variável real  $x$  definida no intervalo real  $[0, 10]$ ; os genes de

8 a 12 representam uma variável inteira  $y$  definida no intervalo  $[0, 25]$ ; e o gene 13 representa uma variável binária  $z$ . O cromossomo em questão tem a seguinte aparência:

13	12	11	10	9	8	7	6	5	4	3	2	1	0	← genes
2	2	2	2	2	2	2	2	2	2	2	2	2	2	← limite do alfabeto
1	1	0	0	1	0	1	0	0	0	1	1	0	1	← cromossomo
$z$	$y$					$x$								← segmentação.

A notação usada para  $\mathbf{v}$  é a seguinte: a primeira linha fornece a identificação dos genes, a segunda relaciona os valores máximos do alfabeto de cada um deles, a terceira é destacada em cinza porque armazena o cromossomo propriamente dito e a quarta linha fornece a segmentação dos bits conforme estabelece a lógica de codificação. Essa formatação é adotada nos exemplos das seções seguintes.

Como  $n(\mathbf{v}) = 14$  e o alfabeto é binário,  $\Phi_g$  estabelecido pela codificação proposta é formado por  $2^{14} = 16384$  palavras. Observe que o valor finito  $2^{14}$  contraria o senso comum, uma vez que  $x$  é uma grandeza real. O que ocorre de fato é que a estratégia de codificação proposta discretiza  $x$  em exatos  $2^8 = 256$  valores, que é o número de palavras geradas por 8 bits.

Em uma eventual decodificação, o valor de  $x$  pode ser obtido em duas etapas: primeiro, o número binário relativo a ele em  $\mathbf{v}$  deve ser convertido para inteiro:

7	6	5	4	3	2	1	0
2	2	2	2	2	2	2	2
1	0	0	0	1	1	0	1
$x$							

$$= 141;$$

e, então, pode ser realizado o cálculo, considerando o intervalo definido para a variável:

$$x = \frac{141}{256}(10 - 0) = 5,51.$$

Prosseguindo, uma estratégia possível para decodificar o valor de  $y$  é converter diretamente o valor armazenado em  $\mathbf{v}$  para inteiro:

12	11	10	9	8
2	2	2	2	2
1	0	0	1	0
$y$				

$$= 18.$$

Entretanto, como a faixa de valores gerada com 5 bits (32 valores) é maior que o intervalo possível para  $y$ , alguns valores inactiváveis são codificados. Um artifício possível para contornar o problema seria atribuir a  $y$  o resto da divisão do valor codificado em  $\mathbf{v}$  por 26, que é o valor limite permitido. O custo neste caso é representar algumas soluções candidatas duas vezes no espaço de busca, enquanto outras continuariam com uma única representação. Por fim, a interpretação de  $z$  é direta:  $z = 1$  para o exemplo.

Uma forma alternativa de codificação é utilizar genes inteiros em vez de binários. Neste caso, é preciso cuidado especial com a definição dos alfabetos: a mesma solução do exemplo anterior seria codificada por

$\mathbf{v} =$	2	1	0	← genes
	2	26	256	← limite do alfabeto
	1	18	141	← cromossomo
	$z$	$y$	$x$	← segmentação.

A decodificação fica simplificada para  $x$ , visto que é possível partir diretamente para a segunda etapa descrita anteriormente. Para  $y$  e  $z$ , a decodificação é direta. Uma vantagem é que nenhum valor fora do intervalo permitido de  $y$  é gerado. O tamanho de  $\Phi_g$  é calculado pela expressão geral

$$\sigma = \prod_{i=0}^{\gamma-1} \alpha_i = 2 \cdot 26 \cdot 256 = 13312. \quad (3.1)$$

Em que:  $\sigma$  é o número de soluções codificadas,  $\gamma$  é o comprimento do cromossomo e  $\alpha_i$  é o valor que define o alfabeto do gene  $i$ . Para o caso exemplo:  $\gamma = 3$ ,  $\alpha_2 = 2$ ,  $\alpha_1 = 26$  e  $\alpha_0 = 256$ .

A opção por uma ou outra codificação estabelece espaços de busca com, respectivamente, 16384 e 13312 possibilidades. A diferença é gerada em virtude da representação de valores de  $y$  fora do limite permitido quando a codificação binária foi utilizada. Tomando como base este exemplo, observe que diferentes representações são

possíveis para um mesmo problema, cada uma com características distintas.

Tendo concluído este exemplo genérico, é possível avançar para conceitos mais específicos das codificações. Na seção 3.2.3 apresentam-se alguns conceitos relevantes quando se deseja prever a performance de uma determinada estratégia de codificação ou comparar diferentes técnicas.

### 3.2.3 Métricas de Comparação

Como descrito na Seção 3.1, o conjunto dos chamados Algoritmos Evolucionários – entre eles os AGs – é extremamente numeroso, tanto no número de variantes do próprio algoritmo como na diversidade das aplicações elaboradas com sucesso nas últimas décadas. Mesmo assim, prossegue uma discussão técnica sobre como e por que estes algoritmos são eficazes na solução de tantos problemas diferentes; e também por que eles não se mostram adequados para outros tantos?

Raidl & Julstrom (2003) atribuem às codificações e seus respectivos operadores genéticos papel determinante para o sucesso dos algoritmos evolutivos. Nesse sentido, apresentam uma lista de características reconhecidas por diversos pesquisadores como relevantes quando se discute sobre o tema codificações. A lista a seguir está organizada definindo essas métricas e, em seguida, apresentando alguns comentários sobre o entendimento adotado em relação a cada uma delas neste trabalho:

- *Espaço* – a representação das soluções candidatas não deve ocupar grandes quantidades de memória;

Se o espaço na memória deve ser minimizado devido a capacidade de armazenamento, esta métrica impõe uma restrição ao hardware e não um requisito para o sucesso da otimização. Contudo, se minimizar espaço significa evitar representações de rede redundantes, define-se efetivamente uma métrica para o

sucesso da Configuração. Neste caso, contrariando Rothlauf (2006) que analisou com profundidade o tema e concluiu que uma representação mesmo redundante que tenha as propriedades de sinonimidade e uniformidade não impõe perda de performance;

- *Tempo* – é desejável que os algoritmos de decodificação, cruzamento e mutação tenham complexidade computacional mínima;

É importante observar que o tempo neste caso indica complexidade computacional, uma vez que tempos de execução de algoritmos são influenciados por fatores externos como hardware, carregamento ou implementação. A medida precisa geralmente utilizada, por exemplo para descrever um algoritmo que no pior caso executa segundo uma função quadrática, é  $O(n^2)$  – lê-se “o de  $n^2$ ” (CORMEN et al, 2002);

- *Factibilidade* – a representação deve codificar apenas soluções candidatas factíveis, mesmo após operações de cruzamento e mutação;

O conceito de solução candidata factível é abstrato. Por exemplo: uma das restrições elétricas da Configuração é a tensão mínima admissível nas barras; seria possível definir uma codificação que só represente redes radiais conexas que atendem à restrição de nível de tensão? A resposta mais provável é que tal técnica seria inviável. No presente trabalho, entende-se a métrica factibilidade como uma diretiva: tente incluir no código o maior número de restrições do problema, pois reduzir espaço de busca favorece performance;

- *Cobertura* – operações genéticas sucessivas devem ser capazes de gerar qualquer solução factível partindo de qualquer solução factível;

A possibilidade de alcançar qualquer solução factível a partir de qualquer solução factível é uma medida da qualidade dos operadores genéticos, sobretudo do

operador de mutação. No entanto, chegar em determinada solução não parece suficiente. O percurso é fator determinante para a performance, como indicam as métricas Localidade e Hereditariedade descritas adiante;

- *Tendência* – uma tendência de uma representação em gerar um certo tipo de solução só significa uma vantagem se as soluções favorecidas estão próximas da solução ótima. Caso contrário, codificações sem nenhuma tendência são preferíveis;

De acordo com Rothlauf (2006), o conceito de proximidade entre indivíduos está relacionado aos operadores genéticos, sendo nomeado de *distância*. O mesmo conceito se aplica aos fenótipos, porém neste caso influenciado pelo tipo de problema e não pelos operadores. Sempre que a distância entre genótipos ou fenótipos é mínima, se diz que eles são *vizinhos*. Por exemplo: o número 4 representado por um código binário de 3 bits é 100; seus vizinhos supondo mutação por inversão de um bit seriam: 101, 110 e 000, logo os valores inteiros 5, 6 e 0. A distância entre 4 e 6 em  $\Phi_g$  é mínima  $d_g = 1$ , porém em  $\Phi_f$  é  $d_f = 2$ ;

- *Localidade* – é definida como “*uma medida de quão bem genótipos vizinhos correspondem a fenótipos vizinhos*” por Rothlauf (2006) e como “*um indivíduo gerado por mutação deve representar uma solução semelhante a de seu pai*” por Raidl & Julstrom (2003);

Considerando que a operação genética capaz de gerar genótipos vizinhos é a mutação, as definições são semelhantes. No entanto, observe que a 1º formulação tem um tom quantitativo, enquanto a 2º é qualitativa. Ao utilizar o termo vizinhos, Rothlauf (2006) associa localidade máxima a condição em que distâncias mínimas no espaço dos genótipos resultam em distâncias mínimas no espaço dos fenótipos.

- *Hereditariedade* – um indivíduo gerado por cruzamento deve representar uma solução que combina majoritariamente partes das soluções associadas aos seus pais;

Essa propriedade é a base do processo de busca de um AG, porém para alguns problemas podem haver características excludentes nos pais em que se faz necessário flexibilizar a hereditariedade. Radcliffe (1992) aponta esta condição usando o termo em inglês *assortment*;

- *Restrição* – é desejável que o algoritmo de decodificação e os operadores genéticos possam considerar restrições específicas do problema que se deseja resolver;

Essa métrica se confunde com factibilidade, pois na prática significa tentar embutir no código e seus operadores restrições específicas do problema;

- *Hibridização* – os operadores genéticos devem permitir embutir heurísticas do problema que se deseja resolver.

Novamente, percebe-se intersecção entre métricas. Nesse caso, hibridização seria o caminho para gerar uma tendência, o que, como descrito anteriormente, só será uma vantagem se as soluções favorecidas são semelhantes à solução ótima.

A idéia é utilizar as métricas descritas nessa seção para avaliar as codificações sequenciais e o AGS propostos neste trabalho (Capítulos 4, 5 e 6), prevendo sua performance. No Capítulo 7, são realizados testes no sentido de validar as análises apresentadas.

### 3.2.4 Estado da Arte

Nesta seção, apresenta-se um detalhamento das técnicas de codificação já enumeradas na revisão bibliográfica, porém com um enfoque diferente: um exemplo prático ilustra operação e características das diferentes codificações. O encadeamento passa a ser definido pela afinidade entre técnicas e a cronologia é ignorada. Para todos os exemplos desta seção considere o grafo de setores da Figura 3.3. Este grafo é o mesmo que foi usado na subseção 3.2.1:  $X_S = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$  e  $A_S = \{S_1, S_2, S_3, S_4,$

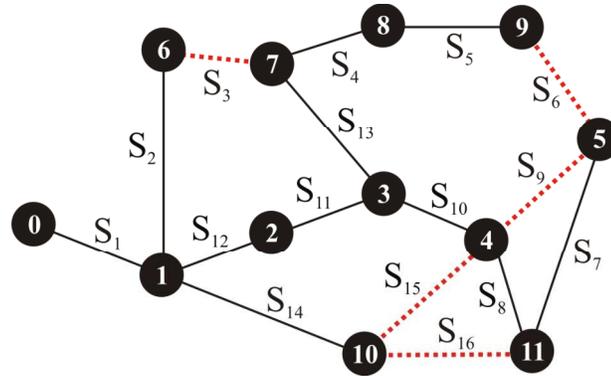


Figura 3.3 – Grafo de Setores  $G_S$  a ser usado nos exemplos da subseção 3.2.4.

$S_5, S_6, S_7, S_8, S_9, S_{10}, S_{11}, S_{12}, S_{13}, S_{14}, S_{15}, S_{16}$ . O único nó fonte é 0, sendo os demais de carga. O número de configurações radiais conexas foi obtido por busca exaustiva, sendo de 737 em um universo de 65536 combinações de estado das chaves.

Para favorecer o encadeamento do texto, as representações de rede pesquisadas foram agrupadas em Codificações de Nós e Codificações de Arestas; que, como seus nomes indicam, são aquelas cujos códigos são baseados, respectivamente, em nós e arestas do grafo de setores.

A estratégia mais simples para representar o grafo da Figura 3.3 utiliza um cromossomo inteiro com  $n(X_L)$  genes e um alfabeto único para todos, definido por  $n(X_S)$  símbolos. Para o exemplo da Figura 3.3,  $n(X_L) = 11$  e  $n(X_S) = 12$ , logo a codificação chamada de Predecessor 1 conduziria a

$$\mathbf{v} = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|} \hline 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ \hline 12 & 12 & 12 & 12 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \\ \hline 0 & 1 & 2 & 3 & 11 & 1 & 3 & 7 & 8 & 1 & 4 \\ \hline 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ \hline \end{array}$$

Como na subseção 3.2.2, a notação usada para  $\mathbf{v}$  é a seguinte: a primeira linha fornece a identificação dos genes, a segunda relaciona os valores máximos do alfabeto de cada um deles, a terceira é destacada em cinza porque armazena o cromossomo propriamente dito e a quarta linha fornece a segmentação dos bits conforme estabelece a lógica de codificação.

O algoritmo de decodificação neste caso é extremamente simples. Ele parte da configuração obtida com todas as chaves abertas e procede a interpretação dos genes em qualquer ordem, de modo que cada gene conduz ao fechamento de uma chave – desde que ela exista. Por exemplo, o gene 6 (primeira linha) codifica a origem do setor 5 (quarta linha); o valor 11 significa que a origem de 5 é 11, que equivale ao fechamento da chave  $S_7$ .

Predecessor 1 não garante que a configuração final seja radial, muito menos conexa; nem mesmo que ela seja possível, já que o código permite, por exemplo, que a origem do setor 11 seja 0, mesmo não existindo tal aresta no conjunto de candidatas. Como consequência, para o exemplo apresentado, o espaço de busca desta codificação é proibitivamente elevado, resultando em

$$\sigma = \prod_{i=0}^{10} 12 = 7.4 \cdot 10^{11}. \quad (3.2)$$

A representação Predecessor 1 é um exemplo de Codificação de Nós, pois tanto o comprimento dos cromossomos como seu conteúdo é definido em termos dos identificadores dos nós.

Partindo de Predecessor 1, mas restringindo o alfabeto de cada gene ao seu conjunto de nós adjacentes, é possível obter uma melhoria significativa em termos de representação. Tal codificação é nomeada de Predecessor 2 e requer um pré-processamento para construção da Tabela 3.2. Esta tabela permite não só definir o alfabeto de cada gene (coluna 4) como serve ao algoritmo de decodificação durante a interpretação dos cromossomos (colunas 2 e 3). Para o exemplo da Figura 3.3, a codificação Predecessor 2 conduz a

	10	9	8	7	6	5	4	3	2	1	0
	4	2	3	4	3	2	3	2	2	3	3
$\mathbf{v} =$	0	0	0	0	2	0	0	0	1	0	0
	1	2	3	4	5	6	7	8	9	10	11

**Tabela 3.2 – Tabela de Adjacências usada para guiar a representação Predecessor 2.**

Nó	Lista de Nós Adjacentes	Alfabeto do Gene	Limite Superior do Alfabeto
1	0, 2, 6, 10	0, 1, 2, 3	4
2	1, 3	0, 1	2
3	2, 4, 7	0, 1, 2	3
4	3, 5, 10, 11	0, 1, 2, 3	4
5	4, 9, 11	0, 1, 2	3
6	1, 7	0, 1	2
7	3, 6, 8	0, 1, 2	3
8	7, 9	0, 1	2
9	5, 8	0, 1	2
10	1, 4, 11	0, 1, 2	3
11	4, 5, 10	0, 1, 2	3

O novo espaço de busca pode ser calculado por (3.1) como

$$\sigma = \prod_{i=0}^{10} \alpha_i = 4^2 3^5 2^4 = 62208. \quad (3.3)$$

A redução é significativa e ocorre exclusivamente com a redução do alfabeto dos genes. Ainda há configurações inactíveis no espaço de busca, mas em todos os casos a interpretação dos genes leva a operação de chaves que realmente existem. Para cada gene avaliado, o algoritmo de decodificação consulta à Tabela 3.2. Por exemplo, para interpretar novamente o gene 6 é necessário fazer uma consulta à linha relativa ao nó 5 na Tabela 3.2. O valor 2 armazenado conduz a seleção do nó 11 como origem na coluna Lista de Nós Adjacentes.

As duas codificações do tipo Predecessor utilizam operadores genéticos convencionais com baixa complexidade e sem garantia de gerar configurações factíveis.

Outra Codificação de Nós que integra o estado da arte é nomeada de Número Prufer. Para  $n(X_F) = 1$ , ela utiliza um cromossomo inteiro com  $n(X_S)-2$  genes, todos com alfabeto de  $n(X_S)$  símbolos. Caso contrário, este procedimento é repetido para cada subsistema isolado. Seu algoritmo de codificação cria o cromossomo preenchendo os genes da esquerda para direita, utilizando uma árvore auxiliar  $T_{aux}$  que deve ser

iniciada com a configuração que se deseja codificar. O algoritmo pode ser descrito da seguinte maneira: i) selecione a folha de  $T_{aux}$  com o menor identificador, obtenha o identificador do seu nó de origem e inicie o gene atual com este valor; ii) exclua a folha selecionada e seu ramo de ligação com  $T_{aux}$ ; iii) avance para o próximo gene e repita i, enquanto o número de nós em  $T_{aux}$  for maior que dois.

O procedimento de decodificação de um Número Prufer supõe uma rede inicial com apenas nós e opera sobre os genes igualmente da esquerda para direita. Um conjunto  $C_f$  com todos os nós cujo identificador não ocorre no Número Prufer guia o processo. O algoritmo é o seguinte: i) selecione o gene atual  $g$  e o menor identificador  $m$  em  $C_f$ , então, crie uma aresta entre os nós identificados por  $g$  e  $m$ ; ii) exclua  $m$  em  $C_f$ ; iii) adicione  $g$  à  $C_f$ , se  $g$  não se repete no resto do número Prufer; iv) se ainda houver genes, retorna para i), caso contrário, adicione uma aresta entre os dois elementos restantes de  $C_f$ . As Tabelas 3.3 e 3.4 ilustram os procedimentos de codificação e decodificação usando Prufer para a rede de exemplo. O cromossomo neste caso é obtido da coluna Gene Prufer da Tabela 3.3, havendo uma aresta  $A_i$  associada a cada gene:

$$\mathbf{v} = \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ \hline 12 & 12 & 12 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \\ \hline 1 & 11 & 1 & 8 & 7 & 3 & 1 & 2 & 3 & 4 \\ \hline A_1 & A_2 & A_3 & A_4 & A_5 & A_6 & A_7 & A_8 & A_9 & A_{10} \\ \hline \end{array} .$$

O espaço de busca para a codificação por número Prufer pode ser calculado diretamente de (3.1):

$$\sigma = \prod_{i=0}^9 12 = 12^{10} = 6.2 \cdot 10^{10}. \quad (3.4)$$

Outra opção é utilizar a expressão que fornece o número total de configurações radiais conexas geradas em uma rede com  $n(X_F)$  nós fonte e  $n(X_L)$  nós de carga, supondo que todas as conexões são possíveis (BRAZ, 2003):

$$n(X_F)(n(X_F) + n(X_L))^{n(X_L)-1} = 1(1+11)^{11-1} = 12^{10} = 6.2 \cdot 10^{10}. \quad (3.5)$$

**Tabela 3.3 – Detalhamento das etapas da codificação usando Número Prufer para a rede da Figura 3.3.**

Etapa	Lista de Folhas Atual	Folha Escolhida	Gene Prufer
1	{0, 5, 6, 9, 10}	0	1
2	{5, 6, 9, 10}	5	11
3	{6, 9, 10, 11}	6	1
4	{9, 10, 11}	9	8
5	{8, 10, 11}	8	7
6	{7, 10, 11}	7	3
7	{10, 11}	10	1
8	{1, 11}	1	2
9	{2, 11}	2	3
10	{3, 11}	3	4
11	{4, 11}	-	-

**Tabela 3.4 – Detalhamento das etapas da decodificação usando o Número Prufer obtido na Tabela 3.3.**

Nó	Conjunto $C_f$	Gene Atual	Menor Elemento de $C_f$	Aresta Adicionada
1	{0, 5, 6, 9, 10}	1	0	0-1
2	{5, 6, 9, 10}	11	5	5-11
3	{6, 9, 10, 11}	1	6	1-6
4	{9, 10, 11}	8	9	8-9
5	{8, 10, 11}	7	8	7-8
6	{7, 10, 11}	3	7	3-7
7	{10, 11}	1	10	1-10
8	{1, 11}	2	1	1-2
9	{2, 11}	3	2	2-3
10	{3, 11}	4	3	3-4
11	{4, 11}	-	-	4-11

Ou seja, a codificação Prufer ignora as arestas candidatas da rede e supõe que  $G_S$  é um grafo completo  $K_{(n(X_F)+n(X_L))}$ , sendo esta sua maior desvantagem.

Os operadores genéticos são igualmente um problema para a codificação Prufer; mesmo considerando todas as arestas como possíveis, a combinação de dois números Prufer conduz em primeira análise a uma configuração significativamente diferente dos pais – baixa hereditariedade.

Ainda no grupo de Codificações de Nós, foram propostas duas codificações baseadas nos caminhos definidos por uma árvore no grafo de setores: Path-Based e Cadeia Principal. No primeiro caso, é realizado um pré-processamento no qual

são levantados vários caminhos iniciados nas subestações e seguindo até os nós. A idéia é habilitar um único caminho para cada nó, o que em última análise garante o atendimento às restrições de conexão e radialidade. O problema é que o número de caminhos cresce exponencialmente em função do tamanho da rede, logo uma estratégia de poda de caminhos em função da distância elétrica é utilizada. Além da poda, são definidas várias restrições em relação aos caminhos, do tipo “*habilitar dado caminho implica em habilitar outros caminhos...*”, e sabe-se que restrições deste tipo requerem checagem toda vez que um novo indivíduo é gerado, o que acaba por comprometer o desempenho.

A codificação por Cadeia Principal pode ser vista como uma versão compacta e melhorada da representação Path-Based. Seu cromossomo é formado por um conjunto de listas com caminhos de subestações até folhas da árvore (ou floresta) que representa a rede; portanto, o número de listas é igual ao de folhas. As listas podem ser implementadas por vetores ou listas duplamente encadeadas, entretanto, devem ser organizadas em uma ordem particular para viabilizar a execução de operadores específicos que garantem o atendimento às restrições de radialidade e conexão. Para o exemplo da Figura 3.3, a codificação por Cadeia Principal conduz ao seguinte cromossomo:

$$v = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 1 & 6 & & & & & \\ \hline 0 & 1 & 10 & & & & & \\ \hline 0 & 1 & 2 & 3 & 4 & 11 & 5 & \\ \hline 0 & 1 & 2 & 3 & 7 & 8 & 9 & \\ \hline \end{array} .$$

Em que, cada linha denota um caminho desde o nó fonte até uma folha na árvore codificada.

O espaço de busca resultante inclui todas as configurações radiais conexas e nenhuma que viole estas restrições. Por esta razão, tal codificação é

considerada ótima (Tabela 2.1). O algoritmo de decodificação parte do grafo de setores considerando apenas os nós e consiste em percorrer as listas que formam um cromossomo, habilitando os ramos. A única desvantagem aparente desta codificação é a complexidade dos operadores genéticos específicos, que exigem manipulação de várias listas com eventuais inversões.

A representação por Nó-Profundidade também integra o conjunto de codificações de Nós e pode ser vista como a segunda geração da codificação por Cadeia Principal. Nesta codificação, em vez de listas com caminhos desde folhas até subestações são utilizados vetores cujos elementos são pares do tipo {nó, profundidade}, havendo um vetor com comprimento igual ao número de nós para cada árvore. Usando representação por Nó-Profundidade o seguinte cromossomo seria obtido para a rede exemplo:

$$\mathbf{v} = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline 0 & 1 & 2 & 3 & 4 & 11 & 5 & 7 & 8 & 9 & 6 & 10 & \text{Nós} \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 4 & 5 & 6 & 2 & 2 & \text{Profundidade} \\ \hline \end{array} .$$

Para as representações Path-Based, Cadeia Principal e Nó-Profundidade, é complicado obter o número exato de configurações no espaço de busca, em virtude de (3.1) não se aplicar. No caso das representações por Cadeia Principal e Nó-Profundidade, por definição, apenas as configurações factíveis integram o espaço de busca. Consequentemente, seu espaço de busca é formado por 737 configurações.

Partindo para o conjunto de Codificações de Arestas, a técnica de representação mais simples e clássica é denominada de Vetor Característico, sendo seu cromossomo binário e com comprimento  $n(A_S)$ . Para o exemplo, usando o vetor característico da rede o código seria o seguinte:

$$\mathbf{v} = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline 15 & 14 & 13 & 12 & 11 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ \hline 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ \hline 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ \hline S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 & S_8 & S_9 & S_{10} & S_{11} & S_{12} & S_{13} & S_{14} & S_{15} & S_{16} \\ \hline \end{array} .$$

O espaço de busca neste caso é pouco maior que na codificação Predecessor 2. Ele compreende configurações com qualquer número de chaves fechadas, enquanto Predecessor 2 nem representa as configurações com mais de 11 chaves fechadas, nem outras tantas com poucas chaves fechadas. Assim, o espaço de busca calculado por meio de (3.1) é:

$$\sigma = \prod_{i=0}^{15} 2 = 65536. \quad (3.6)$$

A interpretação desta codificação é a mais simples possível; ela independe da condição inicial da rede e consiste em abrir toda chave cujo gene é 0 e fechar toda chave cujo gene é 1. Os operadores genéticos são igualmente simples e tradicionais.

Uma melhoria possível na representação por vetor característico é baseada no fato de toda configuração radial conexa em  $G_S$  ter um mesmo número de ramos e cordas. Isto é, garantindo que apenas vetores com  $n(A_{ST})$  valores 1 e  $n(A_{NA})$  valores 0 sejam representados, é possível excluir do espaço de busca um grande número de inactíveis. A diferença neste caso é que cabe primeiramente ao algoritmo de inicialização e depois aos operadores genéticos a garantia de que tal propriedade seja mantida, enquanto o código permanece essencialmente o mesmo. Observe que esta condição é necessária, mas não é suficiente para garantir radialidade e conexão. Geralmente, chaves fechadas de menos conduzem a configurações com setores de carga desconectados; chaves fechadas demais levam à configurações malhadas; e um número correto de chaves fechadas pode conduzir a configurações radiais conexas ou casos em que há setores de carga desconectados e malhas simultaneamente.

A estratégia de codificação referenciada neste texto como Vetor Característico Melhorado foi concebida com base na idéia de manter constante o

número de ramos e cordas. Seu algoritmo para mutação é o seguinte: selecionar aleatoriamente um gene de valor 0 e outro de valor 1 e permutá-los. Para o cruzamento são necessárias duas etapas: i) criar uma lista com todas as posições onde os genes dos pais possuem valores diferentes; ii) gerar os filhos a partir da cópia dos pais, porém selecionando duas diferenças cujos valores sejam alternados para fazer a troca. Nos dois casos, o número de ramos e cordas é mantido constante. Na Figura 3.4 apresenta-se um exemplo do cruzamento proposto, sendo o par escolhido formado pelos genes das posições 0 e 11.

A decodificação para o Vetor Característico Melhorado é idêntica àquela descrita para sua versão convencional, entretanto o espaço de busca sofre uma redução significativa. São codificadas apenas as palavras cujo número de zeros é igual a  $n(A_{NA})$ , resultando em:

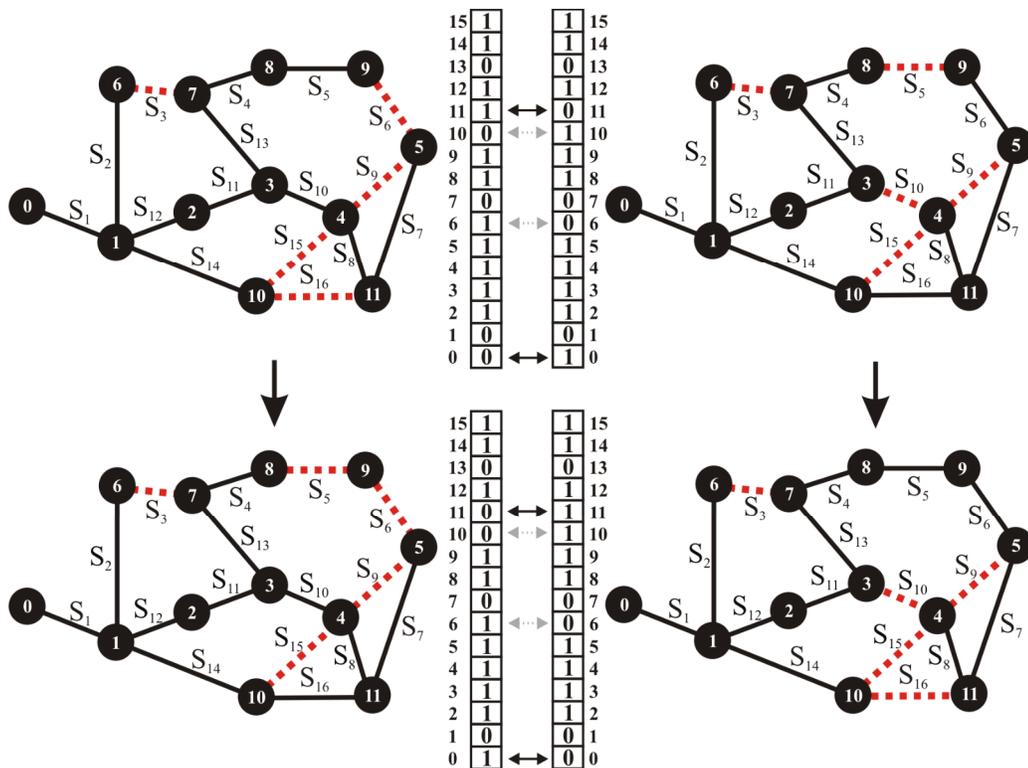


Figura 3.4 – Operador de cruzamento da codificação por vetor característico melhorado: acima os pais e os respectivos cromossomos; abaixo os filhos gerados, sendo um deles infactível.

$$C_{16,5} = \frac{16!}{1!5!} = 4368. \quad (3.7)$$

Outra estratégia de codificação que se propõe a tirar proveito do fato do número de cordas ser constante é referenciada como Lista de Chaves Abertas. Ela parte da rede malhada obtida com o fechamento de todas as chaves e usa uma representação inteira em vez de binária, conferindo um aspecto mais compacto aos cromossomos. São  $n(A_{NA})$  genes, cada um comandando a abertura de uma chave.

Para o exemplo, usando a mesma notação anterior e associando o valor inteiro  $i$  à chave  $S_{i+1}$  na rede, o código usando Lista de Chaves Abertas seria o seguinte:

$$\mathbf{v} = \begin{array}{|c|c|c|c|c|} \hline 4 & 3 & 2 & 1 & 0 \\ \hline 16 & 16 & 16 & 16 & 16 \\ \hline 15 & 14 & 2 & 5 & 8 \\ \hline 5 & 4 & 3 & 2 & 1 \\ \hline \end{array} .$$

A decodificação é direta e resulta na abertura das chaves  $S_{16}$ ,  $S_{15}$ ,  $S_3$ ,  $S_6$  e  $S_9$ , da esquerda para direita. O espaço de busca resultante é significativamente maior que aquele obtido pelo Vetor Característico Melhorado:

$$\sigma = \prod_{i=0}^4 16 = 1.05 \cdot 10^6. \quad (3.8)$$

Tal aumento ocorre porque neste código uma mesma solução com cinco chaves diferentes é representada  $P_5$  vezes. Além disso, palavras com uma mesma chave sendo aberta mais de uma vez são igualmente incluídas no espaço de busca. Embora operadores genéticos convencionais possam ser utilizados, uma melhoria simples é controlar a repetição de chaves.

As codificações por Vetor Característico Melhorado e por Lista de Chaves Abertas tentam tirar vantagem do número de cordas ser constante para todas as configurações radiais conexas, entretanto, desconsideram o fato de que uma única corda deve ser associada a cada uma das malhas fundamentais de  $G_5$ . Com o intuito de incluir este conhecimento no procedimento de codificação foram propostas as técnicas de

Codificação por Malhas Fundamentais *Sem Cortes* e *Com Cortes*. Ambas requerem um pré-processamento para obtenção de uma lista de malhas fundamentais como ilustrado na Tabela 3.5. Para cada corda no grafo da Figura 3.3 é obtida uma malha fundamental, não necessariamente a menor. A diferença é que, no caso nomeado como Com Corte, as duplicidades são retiradas, resultando em uma nova lista de malhas e em um novo tamanho do alfabeto.

Retomando o exemplo da Figura 3.3, o cromossomo obtido para a representação Sem Corte é o seguinte:

$$\mathbf{v} = \begin{array}{|c|c|c|c|c|} \hline 4 & 3 & 2 & 1 & 0 \\ \hline 5 & 7 & 3 & 5 & 6 \\ \hline 1 & 2 & 2 & 4 & 5 \\ \hline M_0 & M_1 & M_2 & M_3 & M_4 \\ \hline \end{array} .$$

Para a codificação Com Cortes:

$$\mathbf{v} = \begin{array}{|c|c|c|c|c|} \hline 4 & 3 & 2 & 1 & 0 \\ \hline 5 & 6 & 1 & 2 & 1 \\ \hline 1 & 2 & 0 & 1 & 0 \\ \hline M_0 & M_1 & M_2 & M_3 & M_4 \\ \hline \end{array} .$$

Seus espaços de busca são calculados a partir de (3.1) e conduzem, respectivamente, a

$$\sigma = \prod_{i=0}^4 \alpha_i = 5.7.3.5.6 = 3150 \quad (3.9)$$

e

$$\sigma = \prod_{i=0}^4 \alpha_i = 5.6.1.2.1 = 60 . \quad (3.10)$$

A codificação Com Corte conduz a uma vantagem significativa, sendo

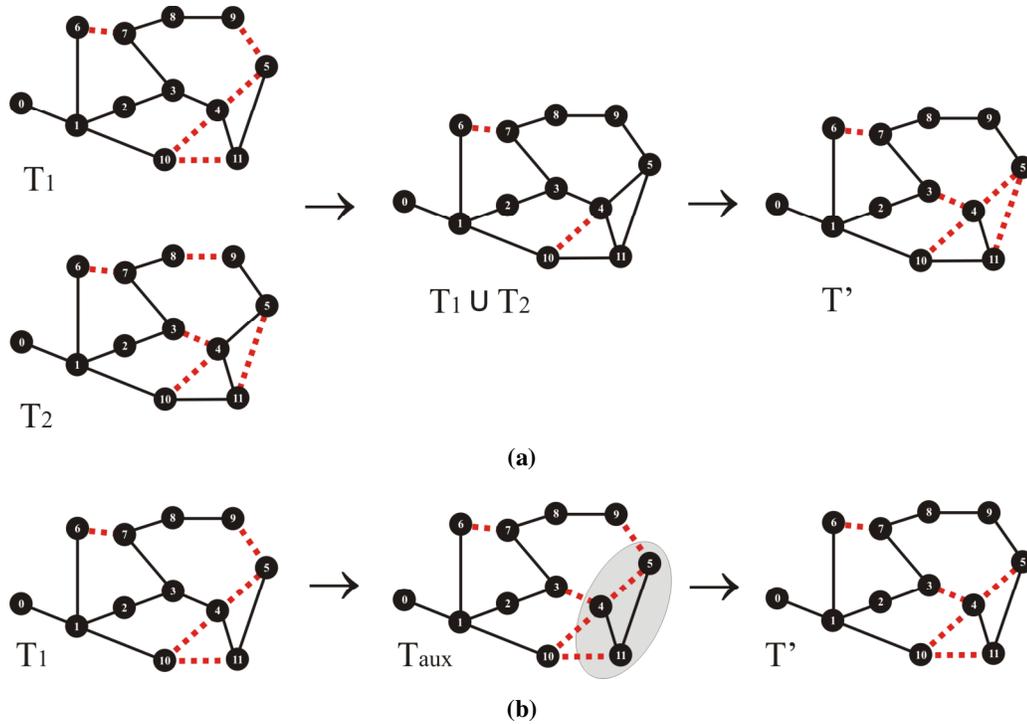
**Tabela 3.5 – Resultado do pré-processamento para as codificações por Malhas Fundamentais Com e Sem Corte.**

Malhas Fundamentais Sem Corte		Malhas Fundamentais Com Corte	
Malhas	Tamanho do Alfabeto	Malhas	Tamanho do Alfabeto
$S_2, S_3, S_{11}, S_{12}, S_{13}$	5	$S_2, S_3, S_{11}, S_{12}, S_{13}$	5
$S_4, S_5, S_6, S_7, S_8, S_{10}, S_{13}$	7	$S_4, S_5, S_6, S_7, S_8, S_{10}$	6
$S_7, S_8, S_9$	3	$S_9$	1
$S_{10}, S_{11}, S_{12}, S_{14}, S_{15}$	5	$S_{14}, S_{15}$	2
$S_8, S_{10}, S_{11}, S_{12}, S_{14}, S_{16}$	6	$S_{16}$	1

aproximadamente 98% menor que no caso Sem Corte. O problema da codificação Com Corte de duplicidades é que ela deixa de codificar configurações factíveis. Isto é, o corte implementado é severo demais. Por exemplo: a rede radial conexa formada abrindo as chaves  $S_3$ ,  $S_6$ ,  $S_9$ ,  $S_{10}$  e  $S_8$  não faz parte do espaço de busca. Por outro lado, a codificação Sem Cortes inclui no espaço de busca soluções em que uma chave é aberta mais de uma vez e mais grave, soluções em que as restrições de conexão e radialidade são violadas, mesmo não ocorrendo duplicidades na lista de chaves abertas. Mesmo assim, o espaço de busca é menor que aquele obtido com Vetor Característico Melhorado, o que lhe confere vantagem comparativa. Em relação aos operadores genéticos, mais uma vez é possível utilizar algoritmos convencionais.

Outra alternativa dentre as Codificações de Arestas, proposta originalmente para outros problemas da Teoria dos Grafos cujo objetivo é definir uma árvore ótima, consiste em simplesmente representar a rede pela lista de  $n(A_{ST})$  ramos de sua árvore e usar operadores genéticos específicos que garantem que os indivíduos gerados sempre atendem às restrições do problema – codificação por Lista de Arestas.

A idéia é implementar o cruzamento entre duas árvores ou florestas a partir de um grafo auxiliar criado por sua união. Uma nova árvore ou floresta é então criada a partir deste grafo por um algoritmo clássico de geração de árvore de cobertura mínima, mas escolhendo as novas arestas de modo aleatório. Para mutação, são propostas duas alternativas: adicionar aleatoriamente uma aresta a árvore, criando uma malha, e então excluir alguma aresta nesta malha para restabelecer a radialidade; ou excluir uma aresta qualquer aleatoriamente, separando a árvore em duas sub-árvores, e igualmente de modo randômico, escolher uma aresta entre elas para reconectá-las. Na Figura 3.5 ilustra-se o algoritmo de cruzamento e a segunda opção de mutação descritos. Trata-se de outra codificação ótima em termos de espaço de busca.



**Figura 3.5 – Exemplo dos operadores genéticos da codificação por Lista de Arestas:**  
 a) Cruzamento de  $T_1$  e  $T_2$ ; b) Mutação sobre  $T_1$ .

Recentemente, foi elaborada uma estratégia de Codificação de Arestas baseada no conceito de Matroids. O código proposto é essencialmente uma Lista de Chaves Abertas; o diferencial são os operadores genéticos que garantem um espaço de busca apenas com configurações radiais conexas, de modo que esta é mais uma codificação ótima sob o ponto de vista do espaço de busca. O operador de mutação é similar ao enunciado na codificação por Lista de Arestas: adicionar aleatoriamente uma aresta a árvore para criar uma malha, e então excluir alguma aresta na malha criada para restabelecer a radialidade. Já o algoritmo de cruzamento proposto é mais complexo e requer várias operações baseadas na Teoria dos Grafos.

A representação por Lista de Arestas NF-NA pode ser vista como uma variação da codificação por Lista de Arestas. A diferença é que ela utiliza um cromossomo mais longo, formado por um primeiro conjunto semelhante ao utilizado na representação por Lista de Arestas e um segundo conjunto, onde armazena a lista de

arestas desabilitadas. Para a rede exemplo, identificando a aresta  $S_{i+1}$  por  $i$  e a divisão entre os conjuntos de arestas habilitadas e desabilitadas por NF-NA, o seguinte cromossomo seria obtido:

$$\mathbf{v} = \left[ \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline 0 & 13 & 1 & 11 & 10 & 12 & 9 & 6 & 4 & 3 & 7 & \text{NF-NA} & 5 & 14 & 15 & 2 & 8 \\ \hline \end{array} \right] .$$

Um operador de recombinação específico é utilizado, de modo a manter conexão e radialidade das soluções. O algoritmo pode ser descrito da seguinte maneira:

i) selecione de forma randômica dois indivíduos para recombinação, sendo o primeiro deles chamado de *principal* e o outro de *secundário*; ii) gere de forma randômica um local de corte  $nr$  no conjunto das arestas habilitadas, dentro da faixa  $r_1\%$  e  $r_2\%$  de seu comprimento; iii) copie os  $nr$  primeiros elementos de principal para o descendente; iv) remova de secundário todos os elementos que já foram adicionados ao descendente; e v) para cada aresta ainda em secundário, independente do seu estado: se ela não gera um ciclo no descendente, adicione no conjunto de arestas habilitadas e, caso contrário, adicione no conjunto de arestas desabilitadas.

Não foi formulada a operação mutação em Lista de Arestas NF-NA, entretanto o mesmo procedimento da representação por Lista de Arestas poderia ser utilizado: habilitar uma aresta do conjunto de arestas desabilitadas de forma randômica; abrir uma aresta no ciclo criado, escolhida de forma aleatória, para restabelecer a radialidade.

Por fim, a última Codificação de Arestas descrita nesta revisão é chamada de Network Random Keys. Seu código consiste em associar de forma randômica um peso a cada uma das arestas de  $G_S$ , sendo a decodificação levada a efeito pela execução do algoritmo de geração de árvores de cobertura de Kruskal (1956). Não há necessidade de utilizar operadores genéticos específicos, pois mesmo com

operadores convencionais há garantia de manutenção da conexão e radialidade. Além disso, o código é redundante: há vários genótipos para um mesmo fenótipo. Um possível genótipo para a rede exemplo da Figura 3.3, considerando genes inteiros no intervalo  $[0, 80[$ , seria:

$$v = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline 15 & 14 & 13 & 12 & 11 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ \hline 80 & 80 & 80 & 80 & 80 & 80 & 80 & 80 & 80 & 80 & 80 & 80 & 80 & 80 & 80 & 80 \\ \hline 79 & 69 & 70 & 8 & 41 & 43 & 5 & 27 & 38 & 21 & 11 & 9 & 16 & 59 & 78 & 62 \\ \hline S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 & S_8 & S_9 & S_{10} & S_{11} & S_{12} & S_{13} & S_{14} & S_{15} & S_{16} \\ \hline \end{array} .$$

Na Tabela 3.6 apresenta-se um resumo das codificações descritas nesta seção. As principais codificações na lista são: Cadeia Principal, Nó-Profundidade, Lista de Arestas, Matroids, Lista de Arestas NF-NA e Network Random Keys, pois implicam na representação exclusivamente de redes radiais conexas. As codificações propostas

**Tabela 3.6 – Resumo quantitativo das codificações da Configuração de Redes no estado da arte: específico para o exemplo adotado.**

	Codificação	Espaço de Busca	Operadores Genéticos	Decodificação	Antes da Decodificação**
Codificação de Nós	Predecessor 1	$7.4 \cdot 10^{11}$	Convencionais	Direta	Aberto
	Predecessor 2	62208	Convencionais	Indireta	Aberto
	Prufer	$6.2 \cdot 10^{10}$	Convencionais	Indireta	Aberto
	Path-Based	*	Específicos	Indireta	Aberto
	Cadeia Principal	737	Específicos	Indireta	Aberto
	Nó-Profundidade	737	Específicos	Indireta	Aberto
Codificação de Arestas	Vetor Característico	65536	Convencionais	Direta	Qualquer
	Vetor Característico Melhorado	4368	Específicos	Direta	Qualquer
	Lista de Chaves Abertas	$1.05 \cdot 10^6$	Convencionais	Direta	Fechado
	Malha Fundamental Com Corte	60*	Convencionais	Indireta	Fechado
	Malha Fundamental Sem Corte	3150	Convencionais	Indireta	Fechado
	Lista de Arestas	737	Específicos	Direta	Aberto
	Matroids	737	Específicos	Direta	Fechado
	Lista de Arestas NF-NA	737	Específicos	Direta	Qualquer
	Network Random Keys	737	Convencionais	Indireta	Aberto

\* deixam de representar configurações factíveis.

\*\* condição das arestas do grafo básico antes de iniciar a decodificação.

neste trabalho concorrem com este grupo, sendo Network Random Keys a representação mais parecida: não exige operadores genéticos específicos e é redundante.

### 3.3 Cálculo de Aptidão

Como descrito na Seção 3.1, as partes de um AG que são inerentemente dependentes do problema a ser resolvido são a estratégia de codificação e a função de aptidão. Visto que a seção anterior foi inteiramente dedicada às codificações, nesta trata-se da formulação da função de aptidão.

O problema é que, por definição, a função de aptidão é fundamentada na função objetivo. Logo, antes de determiná-la, é preciso obter uma formulação geral para o problema da Configuração de Redes em termos de horizonte de estudo e seus objetivos/restrições. Esse é o tema das Seções 3.3.1, 3.3.2 e 3.3.3. Adicionalmente, apresenta-se na subseção 3.3.4 uma estrutura de dados simples que pode ser adicionada ao AG visando evitar sempre que possível o cálculo da função de aptidão: *memória genética*.

#### 3.3.1 Horizonte de estudo

O primeiro aspecto a ser considerado no sentido de uma formulação geral para a Configuração é o período de cada um dos estudos. Como regra, quanto maior é o horizonte de estudo, maior o número de perfis de carregamento a serem considerados e, conseqüentemente, maior é a complexidade computacional. Isto significa que, para um bom desempenho, o ajuste deve ser o menor possível mas respeitando as particularidades de cada estudo. Por questão de unicidade, a notação geral utilizada é um horizonte de  $D$  dias, definido da seguinte forma:

- No caso da Reconfiguração é imprescindível considerar diferentes níveis de

demanda do sistema, porque a configuração ótima para a demanda de pico não é necessariamente a mesma obtida avaliando todos os níveis de carregamento. Entretanto, não é preciso considerar crescimentos anuais, uma vez que as configurações obtidas geralmente têm efeito sazonal. Assim, a opção é usar um único dia cujo carregamento seja representativo do intervalo de estudo ( $D = 1$ );

- Para o Restabelecimento, o procedimento deve levar em consideração duas grandezas estimadas: tempo para implantação do plano de Restabelecimento  $t_1$  e tempo de reparo da rede  $t_2$ . Ambos, contados a partir do instante do defeito ( $t = 0$ ). Uma estimativa precisa de  $t_1$  e  $t_2$  permite determinar com maior segurança os níveis de carregamento que devem ser considerados para operação temporária da rede, uma vez que serão aqueles verificados no intervalo de tempo entre  $t_1$  e  $t_2$  ( $\Delta t = t_2 - t_1$ ). O intervalo  $\Delta t$  é uma fração de um dia, porém um artifício para manter o horizonte medido em dias consiste em criar um dia fictício representativo do Restabelecimento multiplicando as durações de cada patamar a ser considerado pela constante ( $24/\Delta t$ ). Na ausência de estimativas precisas, o mais conservativo é considerar um dia típico de carregamento válido para o período do estudo. Assim, em ambas as situações é possível fazer  $D = 1$ ; e
- Por fim, o horizonte de estudo para o Planejamento precisa ser da ordem de anos para incluir o eventual crescimento previsto das cargas e eventuais empreendimentos de geração. Caso contrário, seria impossível contabilizar adequadamente os custos operacionais do sistema ( $D = 365 \cdot \text{Número de Anos}$ ).

### 3.3.2 Formulação

Definido o período a ser considerado como  $D$  seja qual for o enfoque, resta estabelecer uma função geral que contemple conjuntamente todos os objetivos e restrições das três

variantes da Configuração. Na Tabela 3.7 os objetivos e restrições mais comuns de cada um dos enfoques são resumidos, de onde é possível delimitar a formulação apresentada nesta tese:

- 1) Reduzir Perdas, Melhorar Perfil de Tensão e Balancear Carga são considerados em todos os casos. Na prática, atender à redução de perdas implica igualmente em atender aos outros dois objetivos. Assim, optou-se por incluir na função objetivo apenas uma parcela relativa à energia perdida no período  $D$ , resultante da integração das perdas, considerando níveis de carregamento e crescimento de carga, quando se aplicam;
- 2) Reduzir Manobras, Atender a Carga com Prioridades e Reduzir Investimento são objetivos para um ou dois enfoques e não para todos. Logo, é necessário formulá-los como termos independentes da função objetivo para considerá-los adequadamente. Quando necessário, eles podem ser desabilitados tornando seus pesos nulos na função;
- 3) Os objetivos Maximizar Confiabilidade e Estabilidade são geralmente tratados por

**Tabela 3.7 – Resumo dos objetivos e restrições mais comuns em estudos de Configuração de Redes (legenda abaixo).**

Objetivo/Enfoque	Reconfiguração	Restabelecimento	Planejamento
Reduzir Perdas	OP	O	OP
Melhorar Perfil de Tensão	OP	O	OP
Balancear Carga	OP	O	OP
Reduzir Manobras	O	OP	N
Atender a Carga com Prioridades	R	OP	R
Reduzir Investimento	N	N	OP
Maximizar a Confiabilidade	N	N	O
Maximizar Estabilidade	O	N	OP
Radialidade	R	R	R
Tensão Nodal	R	R	R
Suportabilidade dos Ramos	R	R	R
Capacidade das Subestações	R	R	R
Geografia	N	N	R

Legenda: OP – Objetivo Prioritário; O – Objetivo; N – Não se aplica; e R – Restrição.

algoritmos específicos, uma vez que demandam uma formulação particular. Tais objetivos não integram a função objetivo estabelecida neste trabalho, ainda que não haja qualquer restrição das técnicas aqui propostas a sua inclusão;

4) As restrições de Radialidade, Tensão Nodal, Suportabilidade e Capacidade são comuns a todos os enfoques, então são considerados em conjunto e invariavelmente;

5) A restrição geográfica se aplica unicamente ao Planejamento, de modo que é preferível tratá-la antes do processo de otimização restringindo as arestas candidatas do Planejamento àquelas que atendem a essa restrição;

6) Finalizando, para efeito do Planejamento, alocação de novas subestações e ampliação de subestações existentes são modelados como arestas candidatas fictícias.

O problema geral da Configuração de Redes pode agora ser formulado em termos de  $G_S$  e  $T_S$  como segue:

$$\text{Maximizar } f(T_S, D). \quad (3.11)$$

Por definição,  $T_S$  modela as restrições de conexão e radialidade. Adicionalmente, a equação (3.11) está sujeita a restrições de: igualdade impostas pelo fluxo de carga; corrente admissível, válida para todos os setores e arestas habilitadas; limites de tensão nos setores; e capacidade dos nós fonte. A função  $f$  é definida efetivamente como uma soma ponderada, em que cada termo representa matematicamente um dos objetivos do problema. Os termos são normalizados visando atribuir a cada um uma mesma resolução na função global:

$$f(T_S, D) = w_1 f_1 + w_2 f_2 + w_3 f_3 + w_4 f_4, \quad (3.12a)$$

em que:

$$f_1(T_S, D) = 1 - \frac{\Delta E_P}{(\Delta E_P + \Delta E_E)}, \quad (3.12b)$$

$$f_2(T_S, T_S^0) = 1 - \frac{n_{op}}{n_{tt}}, \quad (3.12c)$$

$$f_3(T_S) = \frac{\sum_{i \in X_L} c_i P_{nomi}}{\sum_{l \in X_S} c_l P_{noml}} e \quad (3.12d)$$

$$f_4(T_S) = 1 - \frac{\sum_{k \in \mathbf{A}} \$k}{\sum_{j \in \mathbf{B}} \$j}, \quad (3.12e)$$

são as funções que denotam as parcelas a serem maximizadas segundo a relação de prioridades estabelecida pelos valores  $w_1$ ,  $w_2$ ,  $w_3$  e  $w_4$ . Sendo:

$\Delta E_P$  a energia perdida no intervalo de tempo considerado -  $D$ ;

$\Delta E_E$  a energia entregue no intervalo de tempo considerado -  $D$ ;

$T_S^0$  a floresta correspondente a configuração original ou inicial;

$n_{op}$  o número de manobras realizadas para obter  $T_S$  de  $T_S^0$ , desconsiderando as chaves manobradas para isolar o defeito no caso de estudo de Restabelecimento;

$n_{tt}$  o número de chaves de  $G_S$  desconsiderando as chaves manobradas para isolar o defeito no caso de estudo de Restabelecimento;

$c_k$  o nível de prioridade no atendimento à carga  $k$ ;

$P_{nomk}$  a carga ativa nominal do nó de carga  $k$ ;

**A** o conjunto formado pelas arestas candidatas habilitadas para obter  $T_S$  no eventual estudo de Planejamento;

**B** o conjunto formado por todas as arestas candidatas no eventual estudo de Planejamento;

$\$k$  o custo associado à aresta candidata  $k$ .

De posse da função objetivo do problema e das restrições, é possível definir a função de aptidão. Para tanto, a idéia é considerar cada uma das restrições

diretamente como um fator multiplicativo sobre  $f(T_s, D)$ ; resultando em uma função que reúne, simultaneamente, todos os objetivos e restrições do problema. Tais fatores são obtidos, respectivamente, para as restrições de radialidade, tensão nodal, suportabilidade e capacidade da seguinte forma:

$$k_1 = e^{x_1 \frac{nv_{radial}}{n(X_L)}}, \quad (3.13a)$$

$$k_2 = e^{x_2 \frac{nv_V}{n(X_S)}}, \quad (3.13b)$$

$$k_3 = e^{x_3 \frac{nv_I}{n(A_{ST})+n(X_S)}} e \quad (3.13c)$$

$$k_4 = e^{x_4 \frac{nv_S}{n(X_F)}}. \quad (3.13d)$$

Em que:

$k_i$  é a penalização associada à função  $f_i$ ;

$x_i$  é o parâmetro negativo de controle da penalização  $i$ ;

$nv_{radial}$  é o resultado da soma dos nós desconectados e do número de malhas para a configuração atual do sistema;

$nv_V$  é o número de nós em que houve violação da restrição de tensão nodal;

$nv_I$  é o número de setores e arestas em que houve violação da restrição de suportabilidade de corrente;

$nv_S$  é o número de nós fonte em que houve violação da restrição de capacidade.

Observe que conforme a definição dos termos de (3.12) e (3.13), esses fatores são definidos como grandezas normalizadas. Logo, eles tendem a ser menos dependentes do sistema estudado, o que simplifica seu ajuste. A função de aptidão pode ser finalmente definida por:

$$f_{ap} = k_1 k_2 k_3 k_4 f(T_s, D). \quad (3.14)$$

Os parâmetros de controle  $w_1, w_2, w_3, w_4, x_1, x_2, x_3$  e  $x_4$  devem ser

selecionados em função do estudo escolhido e dos objetivos estabelecidos. Na subseção 3.3.3 é apresentado um critério para seleção destes parâmetros de forma automática e com garantia de que o atendimento às restrições prevalece sobre a otimização dos objetivos do problema. Um tipo de seletividade de  $f_{ap}$ .

### 3.3.3 Seleção dos Parâmetros

Da forma como foi definida na seção anterior,  $f_{ap}$  possui oito parâmetros: quatro deles são relativos aos objetivos do problema e os demais às restrições. Ocorre uma única interseção, entre o peso  $w_3$  e a restrição cujo parâmetro de controle é  $x_1$ , de modo que quando um destes valores é diferente de zero o outro é necessariamente nulo. Trata-se da restrição de conexão, que somente para o Restabelecimento deve ser considerada como um objetivo.

A idéia para simplificar o ajuste dos parâmetros é definir um dos pesos  $w_i$ , estabelecendo um objetivo prioritário em cada variante da Configuração, e calcular os demais valores tomando como referência suas contribuições na aptidão total em relação ao objetivo prioritário. Para exemplificar, considere um estudo de Reconfiguração cuja prioridade seja minimizar perdas e que tenha como objetivo secundário mitigar manobras. Os outros objetivos não se aplicariam, logo:  $w_3 = w_4 = 0$ .

Supondo que  $w_1$  foi estipulado, uma alteração de  $p_o\%$  em  $f_1$  implica em uma variação total em (3.12) de  $w_1 p_o / 100$ . De forma semelhante, um acréscimo ou redução de  $b$  manobras implica em uma variação total de  $w_2 b / n_{tt}$ . Assim, é possível calcular  $w_2$  com base em  $w_1$  com a garantia de que  $p_o$  equivale a  $b$  na perspectiva do AG, pela seguinte expressão:

$$w_2 = w_1 \frac{p_o n_{tt}}{100b} . \quad (3.15)$$

O valor de  $w_2$  resultante é geralmente pequeno, embora seja suficiente para garantir que, entre duas soluções com perdas muito próximas, a solução com menor número de manobras prevaleça. O critério também depende pouco do sistema sob estudo, como estabelecido previamente.

A mesma lógica de cálculo pode ser estendida para os outros enfoques da Configuração, bem como para o cálculo das restrições. Neste segundo caso, a equivalência deve ser definida entre  $p_r\%$  de variação em  $f_1$  e, por exemplo,  $nv_v$  violações de tensão nodal. Isto é:

$$w_1 \frac{p_r}{100} = 1 - e^{-\frac{x_2 nv_v}{n(X_S)}}.$$

Desenvolvendo, é possível obter uma expressão para o cálculo de  $x_2$  em termos de  $w_1$ :

$$x_2 = \frac{n(X_S)}{nv_v} \ln\left(1 - \frac{w_1 p_r}{100}\right). \quad (3.16)$$

O resultado de (3.16) é aproximado e pressupõe que a parcela associada a  $f_1$  no somatório dos objetivos é predominante. Na Tabela 3.8, apresenta-se um resumo com os ajustes obtidos generalizando o procedimento adotado acima para calcular os parâmetros nos três enfoques possíveis. No caso particular do Planejamento, a definição dos pesos requer um artifício: a escolha é feita de modo que as funções  $f_1$  e  $f_4$  meçam custos em unidades monetárias. A variável  $\$_{MWh}$  é o custo por MWh e as funções *max* e *min* definidas na Tabela 3.8 retornam, respectivamente, o maior e menor valores dentre um par de argumentos. Obviamente, para desabilitar um objetivo em qualquer situação, basta zerar seu peso ignorando o valor sugerido na tabela.

### 3.3.4 Memória Genética

Como discutido na Seção 3.1, os AGs tentam emular a evolução natural de modo que, no curso das gerações, as soluções mais aptas se perpetuem. Assim, durante a execução

**Tabela 3.8 – Resumo da metodologia proposta para ajuste automático dos parâmetros da Equação (3.14), considerando os enfoques: Reconfiguração, Restabelecimento e Planejamento.**

Enfoque do Problema		
Reconfiguração	Restabelecimento	Planejamento
$w_1$ estipulado	$w_1 = 0$	$w_1 = (\Delta E_P + \Delta E_E) \$_{MWh}$
$w_2 = w_1 \frac{P_o n_{II}}{100b}$	$w_2 = w_3 \frac{P_o n_{II}}{100b}$	$w_2 = 0$
$w_3 = 0$	$w_3$ estipulado	$w_3 = 0$
$w_4 = 0$	$w_4 = 0$	$w_4 = \sum_{j \in \mathbf{B}} \$_j$
$x_1 = \frac{n(X_L)}{nv_{radial}} \ln(1 - \frac{w_1 p_r}{100})$	$x_1 = 0$	$x_1 = \frac{n(X_L)}{nv_{radial}} \ln(1 - \frac{w_x p_r}{100})$
$x_2 = \frac{n(X_S)}{nv_v} \ln(1 - \frac{w_1 p_r}{100})$	$x_2 = \frac{n(X_S)}{nv_v} \ln(1 - \frac{w_3 p_r}{100})$	$x_2 = \frac{n(X_S)}{nv_v} \ln(1 - \frac{w_x p_r}{100})$
$x_3 = \frac{n_{aux}}{nv_I} \ln(1 - \frac{w_1 p_r}{100})$	$x_3 = \frac{n_{aux}}{nv_I} \ln(1 - \frac{w_3 p_r}{100})$	$x_3 = \frac{n_{aux}}{nv_I} \ln(1 - \frac{w_x p_r}{100})$
$x_4 = \frac{n(X_F)}{nv_S} \ln(1 - \frac{w_1 p_r}{100})$	$x_4 = \frac{n(X_F)}{nv_S} \ln(1 - \frac{w_3 p_r}{100})$	$x_4 = \frac{n(X_F)}{nv_S} \ln(1 - \frac{w_x p_r}{100})$

Legenda:  $p_o, p_r$  – percentuais para objetivos e restrições;  $n_{aux} = n(A_{ST}) + n(X_S)$ ;  $\$_{MWh}$  – Custo do MWh; e  $w_x = \max(w_1, w_4) / \min(w_1, w_4)$ .

do AG, pode haver várias cópias de alguns indivíduos na população e, enquanto não surgirem outros indivíduos mais aptos, essas soluções tendem a prosseguir se perpetuando geração após geração.

A idéia da memória genética (MG) é manter uma lista auxiliar com indivíduos, de modo que, durante a avaliação da população, apenas os indivíduos que não constem na lista precisem ter sua aptidão calculada. O objetivo é maximizar o número de buscas bem sucedidas à MG, enquanto minimiza-se o tempo com estas buscas. Nesse sentido, propõe-se usar um lista simplesmente ligada, com tamanho

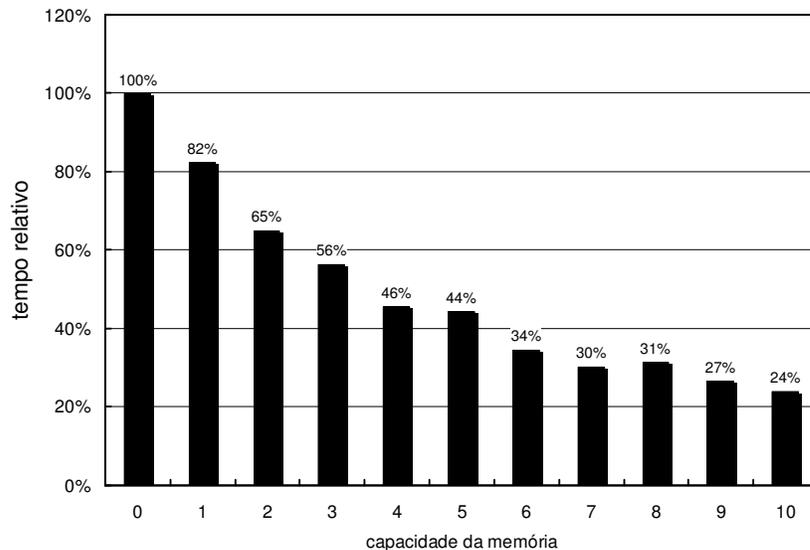
máximo fornecido como parâmetro de entrada e atualizada continuamente, de modo que o primeiro da lista seja o indivíduo mais recentemente utilizado pelo AG.

O procedimento de atualização da MG tem três possibilidades: i) se o indivíduo procurado não consta na MG – sua aptidão é calculada, ele assume o topo da lista e, caso o tamanho da MG extrapole o número de indivíduos permitido, o último elemento da lista é excluído; ii) se o indivíduo procurado consta na MG mas não está no topo – o indivíduo é removido de sua posição atual para assumir o topo da lista; e iii) se o indivíduo procurado consta na MG e já está no topo – nenhuma alteração precisa ser feita. As alterações descritas são realizadas em uma lista ligada apenas pela manipulação de apontadores, logo com complexidade reduzida.

Embora o algoritmo de atualização da MG seja relativamente simples, sua inclusão causa um aumento na complexidade do AG Básico. De modo geral, quanto mais complexa computacionalmente for a função de aptidão, maior tende a ser a economia no tempo de execução do AG ao adotar uma MG. No caso específico da Configuração, especificamente para o Planejamento de uma rede com 63 barras adotando uma população de 150 indivíduos, foram realizadas várias simulações variando o tamanho da MG e registrando a redução proporcionada no tempo total de execução do AG. Os valores obtidos estão plotados no gráfico da Figura 3.6 e indicam que o benefício obtido com a MG tende a saturar a partir de aproximadamente 8% do tamanho da população. A redução no tempo de execução para este caso é de cerca de 75% em relação ao AG sem MG (BRAZ, 2004).

### **3.4 Fluxograma do AG Proposto**

O AG Básico implementado é esquematizado no fluxograma da Figura 3.7. Neste algoritmo, a população inicial pode ser criada de forma aleatória ou com base em

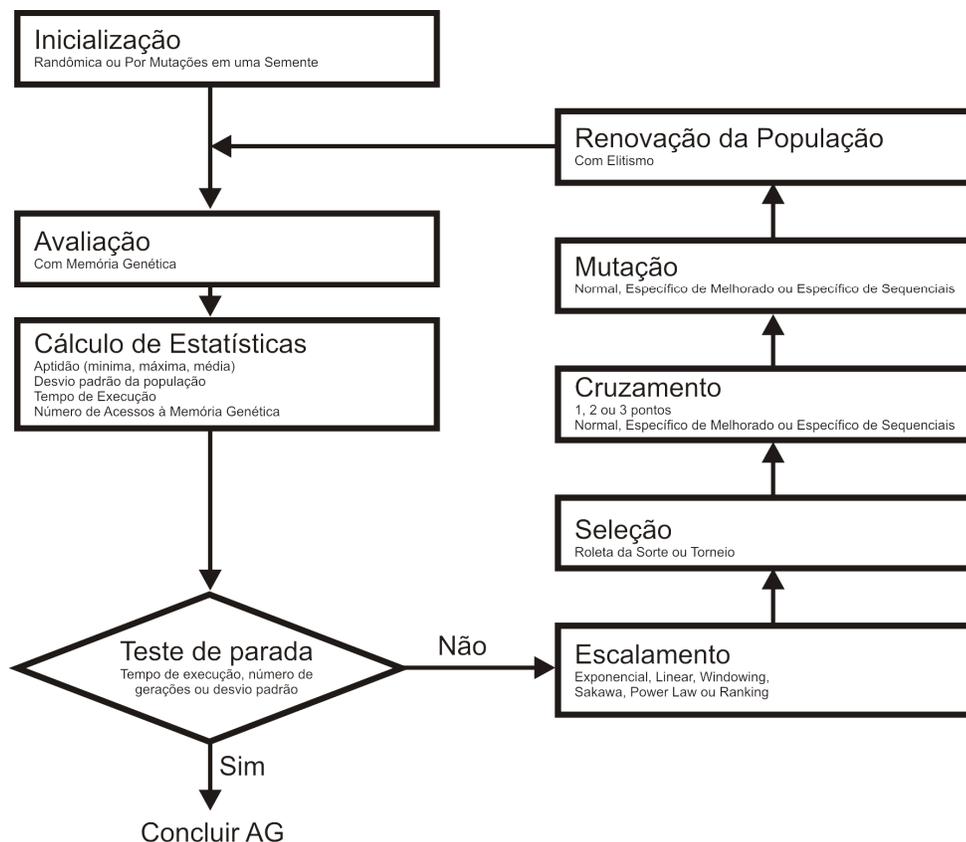


**Figura 3.6 – Redução no tempo total de um AG no Planejamento de uma rede fictícia de 63 barras para diferentes ajustes de tamanho da memória genética.**

sucessivas mutações em uma configuração base. Essa segunda alternativa é especialmente interessante para estudos de Reconfiguração de rede, em que a configuração existente geralmente representa uma boa configuração de partida para o AG.

Na etapa de avaliação da população utiliza-se uma memória genética como descrito na subseção 3.3.4 para, sempre que possível, evitar o cálculo da função de aptidão. Após essa avaliação, o algoritmo gera um relatório com estatísticas da geração atual, como melhor aptidão, média da população, desvio padrão e tempo de execução. Contudo, é possível ajustar o algoritmo para gerar este relatório em uma frequência menor.

Durante o processo de seleção é possível escolher entre o método da roleta ou por torneio. Já o cruzamento entre indivíduos pode ser realizado pelo operador de cruzamento simples com 1, 2 ou 3 pontos de corte, ou, quando aplicável, por operadores específicos das codificações por vetor característico melhorado e sequenciais, descritos nos Capítulos 4 e 5. Para mutação, as opções são semelhantes:



**Figura 3.7 – Fluxograma do AG implementado.**

emprega-se o operador de mutação simples, conforme probabilidade pré-estabelecida, ou operadores específicos das representações por vetor característico melhorado e sequenciais.

Adotam-se populações estacionárias, de modo que o tamanho das populações não variam no curso do AG nem ocorre concorrência entre pais e filhos. Em função dos parâmetros selecionados, um número ajustável de eleitos passa de uma geração para a outra sem enfrentar o processo de seleção. Por fim, o algoritmo pode utilizar como critérios de parada o número de gerações, tempo de execução ou o desvio padrão das aptidões, ou alguma combinação destes critérios.

Nos Capítulos 4 e 5, o fluxograma da Figura 3.7 é complementado pela descrição das duas representações sequenciais propostas neste trabalho: Subtrativa e Aditiva. No Capítulo 6, apresenta-se uma variação deste algoritmo chamada de AGS.

## Capítulo 4

# Codificação Sequencial Subtrativa

---

Este capítulo foi dividido em sete partes: na primeira, apresenta-se um algoritmo simples para obtenção de malhas fundamentais, conceito chave para Subtrativa; a segunda, denominada de Matemática de Malhas Fundamentais, descreve de que modo a alteração de uma malha pode afetar outras da rede e como controlar estas alterações de modo sistemático; as duas seções seguintes englobam, respectivamente, a codificação proposta e seu algoritmo de decodificação; a quinta seção aborda a questão dos operadores genéticos para a representação Subtrativa; na seção seguinte, são expostas algumas melhorias nos algoritmos e estruturas de dados utilizados na formulação de Subtrativa com intuito de reduzir sua complexidade; e por fim, na seção Métricas do Código, é feita uma análise sobre como o código Subtrativo deve se posicionar dentro das métricas habitualmente adotadas na avaliação comparativa de representações.

### 4.1 Algoritmo para Determinação de Malhas Fundamentais

A codificação Subtrativa se baseia no conceito de malhas fundamentais, assim é razoável que seja apresentado um modo sistemático de obtê-las para uma rede qualquer. O algoritmo sugerido parte de qualquer floresta  $T_S$  definida no grafo de setores  $G_S$  de uma rede elétrica e consiste em determinar uma malha fundamental para cada corda definida por  $T_S$  em  $G_S$ .

O procedimento para obter a malha fundamental associada à corda  $c_i$  é o seguinte: i) selecionar os nós  $n_1$  e  $n_2$  adjacentes a  $c_i$ ; ii) obter o caminho  $p_1$  em  $T_S$ , desde  $n_1$  até o nó fonte que o supre como um conjunto de arestas; iii) obter o caminho  $p_2$  em  $T_S$ , desde  $n_2$  até o nó fonte que o supre como um conjunto de arestas; iv) determinar a malha  $M_i = [(p_1 \cup p_2) - (p_1 \cap p_2)] \cup c_i$ .

Na Tabela 4.1 ilustra-se a obtenção das cinco malhas fundamentais para o exemplo da Figura 3.3 (Capítulo 3) e de acordo com as etapas detalhadas.

## 4.2 Matemática de Malhas Fundamentais

As técnicas de representação de redes baseadas em malhas fundamentais têm em comum o fato de sempre partirem da rede malhada obtida com o fechamento de todas as chaves e de manterem uma lista com malhas fundamentais cujo propósito é auxiliar na decodificação dos indivíduos. Em termos gerais, a idéia é abrir uma chave em cada malha da rede obtendo ao final uma rede radial conexa. Ou de forma equivalente,

**Tabela 4.1 – Etapas do algoritmo de obtenção de Malhas Fundamentais para o sistema da Figura 3.3 (Capítulo 3).**

It	Corda	Passo i	Passos ii, iii e iv	Ordenando
1	$S_3$	$n_1 = 6$ $n_2 = 7$	$p_1 = \{S_1, S_2\}$ $p_2 = \{S_1, S_{12}, S_{11}, S_{13}\}$ $M_0 = \{S_2, S_{12}, S_{11}, S_{13}, S_3\}$	$M_0 = \{S_2, S_3, S_{11}, S_{12}, S_{13}\}$
2	$S_6$	$n_1 = 5$ $n_2 = 9$	$p_1 = \{S_1, S_{12}, S_{11}, S_{10}, S_8, S_7\}$ $p_2 = \{S_1, S_{12}, S_{11}, S_{13}, S_4, S_5\}$ $M_1 = \{S_{10}, S_8, S_7, S_{13}, S_4, S_5, S_6\}$	$M_1 = \{S_4, S_5, S_6, S_7, S_8, S_{10}, S_{13}\}$
3	$S_9$	$n_1 = 4$ $n_2 = 5$	$p_1 = \{S_1, S_{12}, S_{11}, S_{10}\}$ $p_2 = \{S_1, S_{12}, S_{11}, S_{10}, S_8, S_7\}$ $M_2 = \{S_8, S_7, S_9\}$	$M_2 = \{S_7, S_8, S_9\}$
4	$S_{15}$	$n_1 = 4$ $n_2 = 10$	$p_1 = \{S_1, S_{12}, S_{11}, S_{10}\}$ $p_2 = \{S_1, S_{14}\}$ $M_3 = \{S_{12}, S_{11}, S_{10}, S_{14}, S_{15}\}$	$M_3 = \{S_{10}, S_{11}, S_{12}, S_{14}, S_{15}\}$
5	$S_{16}$	$n_1 = 10$ $n_2 = 11$	$p_1 = \{S_1, S_{14}\}$ $p_2 = \{S_1, S_{12}, S_{11}, S_{10}, S_8\}$ $M_4 = \{S_{14}, S_{12}, S_{11}, S_{10}, S_8, S_{16}\}$	$M_4 = \{S_8, S_{10}, S_{11}, S_{12}, S_{14}, S_{16}\}$

adicionar uma corda a cada malha de  $G_S$  obtendo ao final uma floresta  $T_S$ . Contudo, em todos os casos consultados no estado da arte, há situações em que este procedimento simples conduz a uma configuração de rede que viola tanto a restrição de radialidade quanto de conexão. Esta afirmação não contradiz a revisão do Capítulo 2, uma vez que, dentre as representações nomeadas como ótimas na Tabela 2.1, nenhuma é baseada em manter listas de malhas fundamentais.

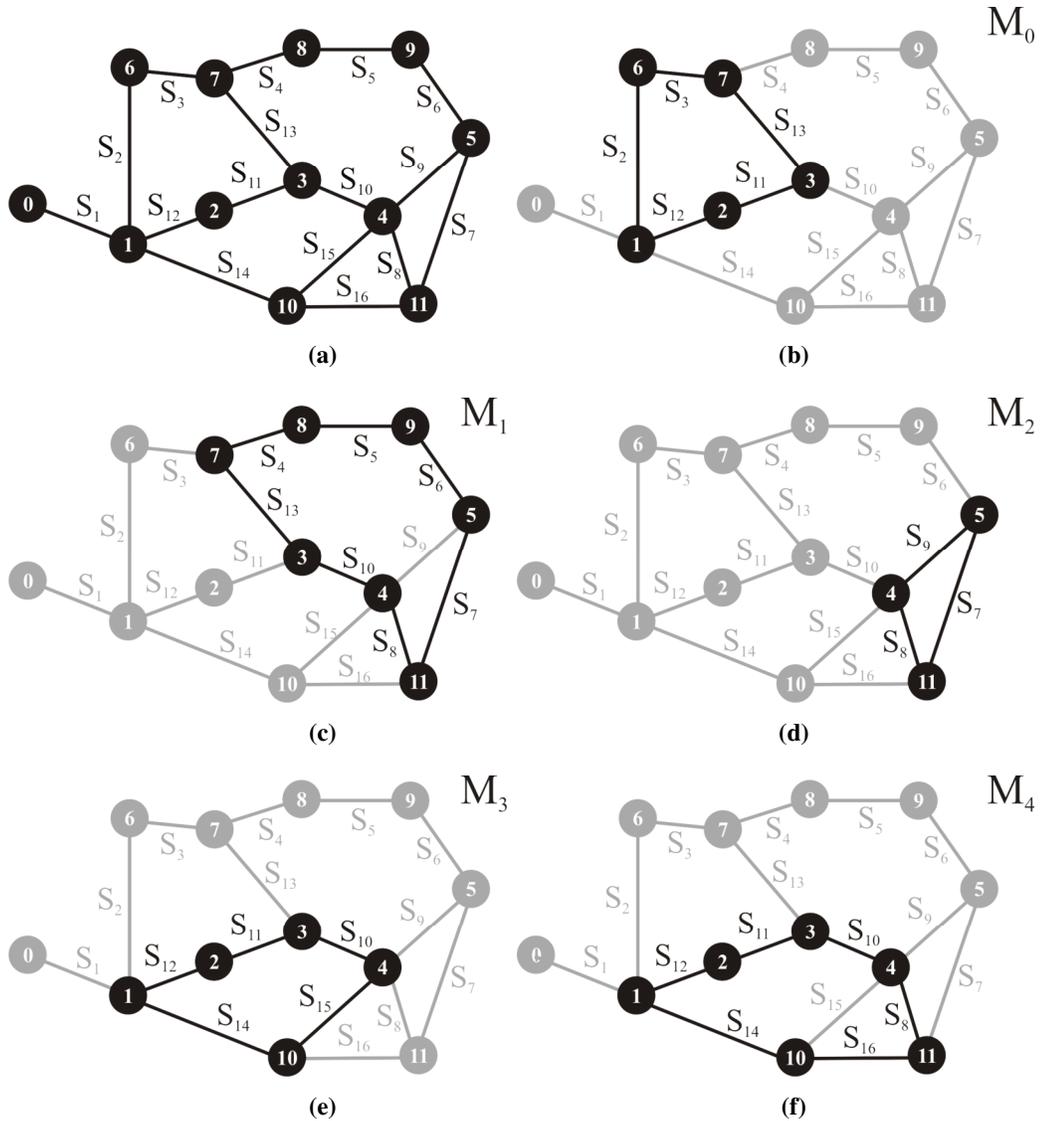
O problema das técnicas baseadas em malhas fundamentais propostas anteriormente é que elas não consideram apropriadamente as possíveis alterações que a eliminação de uma malha pode acarretar nas outras da rede. Para exemplificar, considere a Figura 4.1, onde ilustram-se graficamente a rede inicial considerada nos exemplos deste capítulo e suas malhas fundamentais obtidas na Tabela 4.1. Três exemplos são apresentados com a finalidade de expor a idéia de uma aritmética de malhas e em todos se supõe que a rede encontra-se na sua condição inicial, com todas as arestas habilitadas:

### **1) Eliminação da malha $M_0$ (Figura 4.1b) pela abertura da chave $S_3$**

Consultando a Figura 4.1, observa-se que a chave  $S_3$  não ocorre em nenhuma outra malha além de  $M_0$ , logo ela é uma chave *exclusiva*. Operando  $S_3$  não ocorre qualquer mudança nas outras malhas, como pode ser observado nas Figuras 4.1 de (c)-(f). De fato, como regra geral, manobras em chaves exclusivas não provocam alteração em outras malhas;

### **2) Eliminação da malha $M_0$ (Figura 4.1b) pela abertura da chave $S_{13}$**

A chave  $S_{13}$  é *compartilhada* pelas malhas  $M_0$  e  $M_1$  como ilustram as Figuras 4.1b e 4.1c. Ao operar esta chave, as malhas  $M_0$  e  $M_1$  deixam de existir, dando lugar a uma nova malha que engloba as duas. Isto é, podemos dizer que a abertura de  $S_{13}$  excluiu  $M_0$



**Figura 4.1 – Malhas fundamentais obtidas pelo algoritmo descrito na Seção 4.1:**  
 a) rede malhada inicial; b-f) malhas fundamentais de 0 até 4.

e *expandiu*  $M_1 = \{S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_{10}, S_{11}, S_{12}\}$ . Nenhuma das outras malhas sofre qualquer alteração devido a esta manobra.

### 3) Eliminação da malha $M_3$ (Figura 4.1e) pela abertura da chave $S_{11}$

A chave  $S_{11}$  é compartilhada pelas malhas  $M_0$ ,  $M_3$  e  $M_4$ , como ilustram as Figuras 4.1b, 4.1e e 4.1f. Assim, sua abertura além de eliminar  $M_3$ , implica na alteração das malhas  $M_0$  e  $M_4$ . Em  $M_0$  a condição é semelhante aquela descrita em 2, uma vez que  $M_0$  deve expandir sobre a área coberta por  $M_3$  em virtude da referida manobra. O resultado é  $M_0$

=  $\{S_2, S_3, S_{10}, S_{13}, S_{14}, S_{15}\}$ . O caso de  $M_4$  é diferente, porque em vez de implicar em expansão a manobra resulta na *redução* de  $M_4$ . Observe que abrir  $S_{11}$  na Figura 4.1e modifica 4.1f, resultando em  $M_4 = \{S_8, S_{15}, S_{16}\}$ . Como regra, chaves compartilhadas sempre implicam em expansões ou reduções das malhas envolvidas que permanecem ativas.

Os exemplos 1, 2 e 3 demonstram graficamente as duas operações possíveis quando malhas fundamentais são manipuladas: expansão e redução. Resta definir uma forma sistemática de implementar estas operações, de preferência com complexidade mínima. Para tanto, define-se a matriz  $L$  com  $n(A_{NA})$  linhas e  $n(A_S)$  colunas; ou seja, a linha  $i$  de  $L$  representa a malha fundamental  $M_i$  e há uma coluna para cada aresta de  $G_S$ . O valor  $L_{ij}$  é 1 se a chave associada a coluna  $j$  pertence a malha  $i$  e, em caso contrário, é 0. Do modo que, para a rede da Figura 4.1a:

	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$	$S_8$	$S_9$	$S_{10}$	$S_{11}$	$S_{12}$	$S_{13}$	$S_{14}$	$S_{15}$	$S_{16}$
$M_0$	0	1	1	0	0	0	0	0	0	0	1	1	1	0	0	0
$M_1$	0	0	0	1	1	1	1	1	0	1	0	0	1	0	0	0
$M_2$	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
$M_3$	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0
$M_4$	0	0	0	0	0	0	0	1	0	1	1	1	0	1	0	1

A matriz  $L$  pode ser vista como um *Mapa das Malhas* da rede, similar a matriz de ciclos definida para dígrafos em Netto (2006). A soma de uma linha  $i$  fornece quantas arestas formam  $M_i$ , enquanto a soma de uma coluna  $j$  indica o número de malhas que compartilham a aresta  $S_{j+1}$ .

A principal vantagem de  $L$  é a facilidade com que expansões e reduções podem ser executadas. Define-se um único operador, denominado de XOR, e que opera sobre duas linhas  $i$  e  $j$  de  $L$ . Sua sintaxe é  $XOR(L[i], L[j])$ , retornando uma nova linha com o mesmo comprimento e formato binário de  $L[i]$  ou  $L[j]$ .

O nome XOR faz alusão ao operador *ou-exclusivo*, por ser exatamente a

operação realizada tomando os elementos de cada linha termo a termo. O procedimento de eliminação da malha  $i$  pela abertura da chave associada à coluna  $j$  pode ser executado da seguinte maneira: para toda malha  $k$  que ainda não foi eliminada cujo elemento da coluna  $j$  é não nulo, execute o operador  $XOR(L[i],L[k])$  e coloque o resultado na linha  $k$ .

Reescrevendo o exemplo anterior adotando este procedimento:

### 1) Eliminação da malha $M_0$ (Figura 4.1b) pela abertura da chave $S_3$

Como a coluna referente à  $S_3$  em  $L$  não tem nenhum outro valor diferente de zero, nada precisa ser feito;

### 2) Eliminação da malha $M_0$ (Figura 4.1b) pela abertura da chave $S_{13}$

A coluna referente à  $S_{13}$  tem um valor não nulo na linha associada a  $M_1$ , logo a nova linha  $M_1$  é obtida:

	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$	$S_8$	$S_9$	$S_{10}$	$S_{11}$	$S_{12}$	$S_{13}$	$S_{14}$	$S_{15}$	$S_{16}$
$M_0$	0	1	1	0	0	0	0	0	0	0	1	1	1	0	0	0
$M_1$	0	0	0	1	1	1	1	1	0	1	0	0	1	0	0	0
Novo $M_1$	0	1	1	1	1	1	1	1	0	1	1	1	0	0	0	0

que implica em:  $M_1 = \{S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_{10}, S_{11}, S_{12}\}$ ;

### 3) Eliminação da malha $M_3$ (Figura 4.1e) pela abertura da chave $S_{11}$

Neste caso, há valores não nulos na coluna referente à  $S_{11}$  nas linhas de  $M_0$  e  $M_4$ . O procedimento é o mesmo anterior:

	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$	$S_8$	$S_9$	$S_{10}$	$S_{11}$	$S_{12}$	$S_{13}$	$S_{14}$	$S_{15}$	$S_{16}$
$M_3$	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0
$M_0$	0	1	1	0	0	0	0	0	0	0	1	1	1	0	0	0
Novo $M_0$	0	1	1	0	0	0	0	0	0	1	0	0	1	1	1	0

que implica em:  $M_0 = \{S_2, S_3, S_{10}, S_{13}, S_{14}, S_{15}\}$ . E para  $M_4$ :

	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$	$S_8$	$S_9$	$S_{10}$	$S_{11}$	$S_{12}$	$S_{13}$	$S_{14}$	$S_{15}$	$S_{16}$
$M_3$	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0
$M_4$	0	0	0	0	0	0	0	1	0	1	1	1	0	1	0	1
Novo $M_4$	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1

que implica em:  $M_4 = \{S_8, S_{15}, S_{16}\}$ . Isto significa que os mesmos resultados obtidos

antes foram reproduzidos por meio de um algoritmo simples baseado no Mapa das Malhas.

Uma aplicação extra para o mapa descrito é implementar o algoritmo descrito na Seção 4.1. Basta escrever  $p_1$ ,  $p_2$  e a própria corda no mesmo formato das linhas de  $L$ , e então obter a malha fundamental por duas operações XOR em qualquer ordem. Para demonstrar, considere o cálculo da malha  $M_0$ :

	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$	$S_8$	$S_9$	$S_{10}$	$S_{11}$	$S_{12}$	$S_{13}$	$S_{14}$	$S_{15}$	$S_{16}$
$p_1$	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$p_2$	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
$c_0$	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
$M_0$	0	1	1	0	0	0	0	0	0	0	1	1	1	0	0	0

O resultado é o mesmo obtido anteriormente:  $M_0 = \{S_2, S_3, S_{11}, S_{12}, S_{13}\}$ .

Tendo definido uma forma precisa de atualizar malhas fundamentais como resultado de diferentes manobras de abertura, é possível partir para a formulação efetiva da estratégia de codificação Subtrativa.

### 4.3 Codificação

A idéia é implementar o código de modo similar à codificação por Malhas Fundamentais Sem Corte apresentada nos Capítulos 2 e 3. A diferença desta técnica em relação às suas versões anteriores é que o Mapa das Malhas e o operador XOR, definidos na Seção 4.2, são usados em conjunto, de modo que uma abertura de chave seja codificada se, e somente se, tal chave efetivamente pertence a uma malha na rede atual.

O complicador é que, adotando a estratégia de atualização como descrita, o número de chaves nas malhas é variável; o que do ponto de vista dos AGs significa conferir um alfabeto dinâmico aos genes. A solução encontrada para viabilizar um AG com alfabeto dinâmico foi modificar os genes para codificarem um valor real definido

no intervalo  $[0,1[$ , em vez de um valor inteiro associado a uma chave específica obtida no pré-processamento.

Em síntese, propõe-se codificar uma rede com  $n(A_{NA})$  malhas fundamentais por meio de um cromossomo com  $n(A_{NA})$  genes reais com valor delimitado no intervalo  $[0,1[$ . Cada gene comanda necessariamente a abertura de uma malha, logo o espaço de busca, ainda que não possa ser calculado por uma equação direta, é formado exclusivamente por configurações radiais conexas e engloba todas as configurações com tais características. A codificação foi nomeada de Subtrativa porque começa com o grafo  $G_S$  com todas as arestas habilitadas e opera desabilitando – subtraindo – uma aresta por vez até definir uma floresta  $T_S$  em  $G_S$ .

Devido aos genes serem reais, há vários cromossomos que codificam uma mesma configuração. Na terminologia dos AGs, a Subtrativa gera vários genótipos para um único fenótipo, comportamento semelhante ao verificado na representação Network Random Keys (ROTHLAUF et al, 2002). Um exemplo numérico que ilustra melhor o código proposto é apresentado na Seção 4.4.

#### 4.4 Decodificação

O propósito da decodificação para Subtrativa é converter um cromossomo como descrito na Seção 4.3 em uma lista de cordas a serem inseridas em  $G_S$  para definir uma árvore  $T_S$ . Do ponto de vista elétrico, este processo consiste em obter uma lista de chaves a serem abertas. O procedimento de decodificação ocorre toda vez que se faz necessário avaliar um novo indivíduo, porém nas codificações com interpretação indireta, e em particular nas codificações sequenciais, o algoritmo de decodificação utiliza alguns dados que são obtidos uma única vez antes da execução do AG – etapa chamada de pré-processamento.

Para Subtrativa, a etapa de pré-processamento deve gerar além da matriz  $L$  apresentada na Seção 4.2, um vetor auxiliar inteiro  $\alpha$  com comprimento  $n(A_{NA})$ . O  $i$ -ésimo elemento de  $\alpha$  é carregado com o número de chaves da malha  $M_i$ , que é o resultado da soma dos elementos da linha  $i$  em  $L$ . Para o exemplo:  $\alpha = \{5, 7, 3, 5, 6\}$ .

De posse de  $L$  e  $\alpha$ , todas as arestas de  $G_S$  são habilitadas e o algoritmo de decodificação pode começar. A interpretação de um cromossomo ocorre da esquerda para direita e consiste em executar um mesmo procedimento  $n(A_{NA})$  vezes. O algoritmo em pseudo-código supondo  $g[i]$  como o valor do gene  $i$ ,  $A_s$  como um vetor de arestas e Abre como a rotina que implementa a abertura de uma aresta é o seguinte:

Algoritmo 4.1 - Decodificação Subtrativa

```

//linhas iniciadas com "//" são comentários
1 Para(inteiro i=0;i<n(ANA);i++)
    //ich_m: [0 = 1°chave], [1 = 2°chave], ...
2 inteiro ich_m=inteiro(g[n(ANA)-i-1]*α[i]);
3 inteiro col=-1;
4 inteiro soma=0;
    //col: marca a coluna em L selecionada por ich_m
5 Enquanto(soma<=ich_m) soma+=L[i][++col];
6 As[col].Abre();
7 Para(inteiro j=i+1;j<n(ANA);j++)
8     Se(L[j][col]) L[j]=XOR(L[i],L[j]);

```

Retomando o exemplo e a notação dos cromossomos do Capítulo 3, considere o cromossomo abaixo e a matriz  $L$  repetida por uma questão de conveniência:

$$\mathbf{v} = \begin{array}{|c|c|c|c|c|} \hline 4 & 3 & 2 & 1 & 0 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ \hline M_0 & M_1 & M_2 & M_3 & M_4 \\ \hline \end{array} \begin{array}{l} \leftarrow \text{genes} \\ \leftarrow \text{limite do alfabeto} \\ \leftarrow \text{cromossomo} \\ \leftarrow \text{segmentação} \end{array} \quad \mathbf{e}$$

$$\mathbf{L} = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 & S_8 & S_9 & S_{10} & S_{11} & S_{12} & S_{13} & S_{14} & S_{15} & S_{16} \\ \hline M_0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ \hline M_1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline M_2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline M_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ \hline M_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ \hline \end{array} .$$

A decodificação deste cromossomo é implementada em cinco iterações, como segue:

### Iteração 1 – Decodificação do gene 4

O primeiro passo é converter o gene para inteiro como na linha 2 do Algoritmo 4.1:

$ich\_m = \text{inteiro}(0.5 * 5) = \text{inteiro}(2.5) = 2$ ; 2 significa que a chave a ser aberta é a terceira da malha  $M_0$ , que resulta na aresta  $S_{11}$ . Tal aresta é compartilhada com  $M_3$  e  $M_4$ , de onde são gerados uma nova matriz  $L$  e um novo  $\alpha$ :

	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$	$S_8$	$S_9$	$S_{10}$	$S_{11}$	$S_{12}$	$S_{13}$	$S_{14}$	$S_{15}$	$S_{16}$
$M_0$	0	1	1	0	0	0	0	0	0	0	<b>1</b>	1	1	0	0	0
$M_1$	0	0	0	1	1	1	1	1	0	1	0	0	1	0	0	0
$M_2$	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
$M_3$	0	1	1	0	0	0	0	0	0	1	0	0	1	1	1	0
$M_4$	0	1	1	0	0	0	0	1	0	1	0	0	1	1	0	1

$$\alpha = \{5, 7, 3, 6, 7\}$$

### Iteração 2 – Decodificação do gene 3

Executando a linha 2 do Algoritmo 4.1:  $ich\_m = \text{inteiro}(0.5 * 7) = \text{inteiro}(3.5) = 3$ ; 3 significa que a aresta a ser desabilitada é a quarta da malha  $M_1$ , que resulta em  $S_7$ . Esta aresta é compartilhada apenas com  $M_2$ , logo são gerados uma nova matriz  $L$  e um novo  $\alpha$ :

	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$	$S_8$	$S_9$	$S_{10}$	$S_{11}$	$S_{12}$	$S_{13}$	$S_{14}$	$S_{15}$	$S_{16}$
$M_0$	0	1	1	0	0	0	0	0	0	0	<b>1</b>	1	1	0	0	0
$M_1$	0	0	0	1	1	1	<b>1</b>	1	0	1	0	0	1	0	0	0
$M_2$	0	0	0	1	1	1	0	0	1	1	0	0	1	0	0	0
$M_3$	0	1	1	0	0	0	0	0	0	1	0	0	1	1	1	0
$M_4$	0	1	1	0	0	0	0	1	0	1	0	0	1	1	0	1

$$\alpha = \{5, 7, 6, 6, 7\}$$

### Iteração 3 – Decodificação do gene 2

Executando a linha 2 do Algoritmo 4.1:  $ich\_m = \text{inteiro}(0.5 * 6) = \text{inteiro}(3) = 3$ ; 3 significa que a aresta a ser desabilitada é a quarta da malha  $M_2$ , resultando em  $S_9$ ; que é exclusiva e, conseqüentemente, não requer alteração em  $L$  ou  $\alpha$ :

	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>	S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>	S <sub>12</sub>	S <sub>13</sub>	S <sub>14</sub>	S <sub>15</sub>	S <sub>16</sub>
$M_0$	0	1	1	0	0	0	0	0	0	0	<b>1</b>	1	1	0	0	0
$M_1$	0	0	0	1	1	1	<b>1</b>	1	0	1	0	0	1	0	0	0
$M_2$	0	0	0	1	1	1	0	0	<b>1</b>	1	0	0	1	0	0	0
$M_3$	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>
$M_4$	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>

$$\alpha = \{5, 7, 6, 6, 7\}$$

#### Iteração 4 – Decodificação do gene 1

Executando a linha 2 do Algoritmo 4.1:  $ich\_m = \text{inteiro}(0.5 \cdot 6) = \text{inteiro}(3) = 3$ ; 3

significa que a aresta a ser desabilitada é a quarta da malha  $M_3$ , que resulta em  $S_{13}$ . Tal

aresta é compartilhada com  $M_3$ , resultando em novos  $L$  e  $\alpha$ :

	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>	S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>	S <sub>12</sub>	S <sub>13</sub>	S <sub>14</sub>	S <sub>15</sub>	S <sub>16</sub>
$M_0$	0	1	1	0	0	0	0	0	0	0	<b>1</b>	1	1	0	0	0
$M_1$	0	0	0	1	1	1	<b>1</b>	1	0	1	0	0	1	0	0	0
$M_2$	0	0	0	1	1	1	0	0	<b>1</b>	1	0	0	1	0	0	0
$M_3$	0	1	1	0	0	0	0	0	0	1	0	0	<b>1</b>	1	1	0
$M_4$	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>						

$$\alpha = \{5, 7, 6, 6, 3\}$$

#### Iteração 5 – Decodificação do gene 0

Executando a linha 2 do Algoritmo 4.1:  $ich\_m = \text{inteiro}(0.5 \cdot 3) = \text{inteiro}(1.5) = 1$ ; 1

significa que a aresta selecionada é a segunda da malha  $M_4$ , que resulta em  $S_{15}$ . Esta

aresta é necessariamente exclusiva da malha  $M_4$ , visto que se trata da última malha

ainda ativa na rede. A decodificação pode ser resumida pelo mapa final obtido:

	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>	S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>	S <sub>12</sub>	S <sub>13</sub>	S <sub>14</sub>	S <sub>15</sub>	S <sub>16</sub>
$M_0$	0	1	1	0	0	0	0	0	0	0	<b>1</b>	1	1	0	0	0
$M_1$	0	0	0	1	1	1	<b>1</b>	1	0	1	0	0	1	0	0	0
$M_2$	0	0	0	1	1	1	0	0	<b>1</b>	1	0	0	1	0	0	0
$M_3$	0	1	1	0	0	0	0	0	0	1	0	0	<b>1</b>	1	1	0
$M_4$	0	0	0	0	0	0	0	1	0	0	0	0	0	0	<b>1</b>	<b>1</b>

A configuração de rede obtida é apresentada na Figura 4.2 e conta com as seguintes

cordas:  $S_7$ ,  $S_9$ ,  $S_{11}$ ,  $S_{13}$  e  $S_{15}$ .

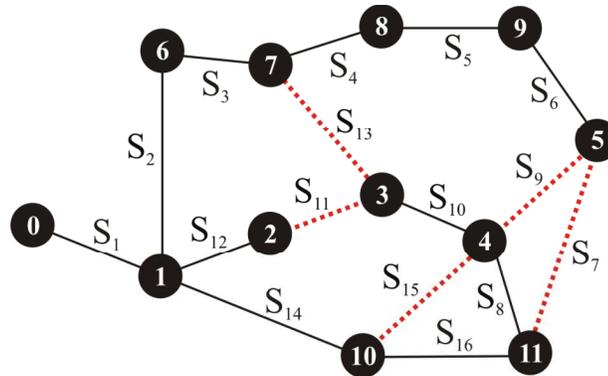


Figura 4.2 – Rede obtida pelo Algoritmo 4.1 decodificando o cromossomo exemplo de Subtrativa.

## 4.5 Operadores Genéticos

Do modo que Subtrativa foi apresentada nas seções anteriores, é possível utilizar os operadores genéticos convencionais com garantia de que nenhuma configuração que viole as restrições de radialidade e conexão seja gerada. Dentre as codificações descritas como ótimas no Capítulo 2, a única que compartilha essa propriedade com Subtrativa é Network Random Keys (ROTHLAUF et al, 2002). No Capítulo 5, apresenta-se a Codificação Sequencial Aditiva que também possui tal propriedade.

Obviamente, permitir usar operadores genéticos convencionais não significa que eles proporcionem desempenho ótimo. Propõe-se nesta seção uma variação dos operadores convencionais, que sugere um desempenho melhor que a abordagem convencional, porém sem caracterizar o uso de operadores específicos. Ambas as alternativas de operadores são alvo de pesquisa na Seção 4.7 sobre como o código Subtrativo se adéqua às métricas descritas na subseção 3.2.3.

Antes de partir para a formulação dos operadores propostos, considere o cromossomo do exemplo da Seção 4.4 quando submetido a duas simulações:

**Teste 1:** Avaliar a alteração provocada pela modificação do gene 0, tal que:



A configuração obtida difere da referência por quatro arestas:  $\{S_3, S_{12}\}$  em vez de  $\{S_{11}, S_{13}\}$ . No pior caso, a alteração do primeiro gene de um cromossomo poderia conduzir a um descendente com todas as cordas diferentes dos pais.

Com base nos comentários anteriores, é possível concluir que para Subtrativa uma alteração no final de um cromossomo tende a gerar descendentes com pequena diferença dos pais – *busca local*; de forma oposta, alterações no início do cromossomo podem resultar em modificações mais significativas – *busca global*. Com o intuito de favorecer o desempenho dos operadores em termos de localidade e hereditariedade, propõe-se usar um recurso apresentado originalmente por Zhang et al (2008) para associar a busca local aos melhores indivíduos no AG e a busca global aos piores.

A idéia é definir variáveis que controlam o alcance permitido dos operadores de cruzamento e mutação com base na aptidão dos indivíduos e na média da população, tal que:

$$N_c = \begin{cases} \left( c_1 \frac{f_{\max} - f_i'}{f_{\max} - \bar{f}} + c_2 \right) \cdot (l-1) & f_i' > \bar{f} \\ (l-1) & f_i' \leq \bar{f} \end{cases} \quad (4.1)$$

$$N_m = \begin{cases} \left( m_1 \frac{f_{\max} - f_i}{f_{\max} - \bar{f}} + m_2 \right) \cdot l & f_i > \bar{f} \\ l & f_i \leq \bar{f} \end{cases} \quad (4.2)$$

Sendo:

$c_1, c_2$  coeficientes associados ao alcance do cruzamento:  $c_1 + c_2 = 1$ ;

$m_1, m_2$  coeficientes associados ao alcance da mutação:  $m_1 + m_2 < 1$ ;

$f_{\max}$  maior aptidão da população;

$f_i'$  maior aptidão entre os dois cromossomos selecionados para cruzamento;

$f_i$  aptidão do cromossomo selecionado para mutação;

$\bar{f}$  aptidão média da população.

Os pontos onde ocorrem o cruzamento  $C_p$  e a mutação  $M_p$  são definidos de forma randômica como em uma operação convencional, entretanto, limitados ao alcance obtido anteriormente da seguinte forma:

$$\begin{cases} C_p = rand \cdot N_c \\ M_p = rand \cdot N_m \end{cases} \quad (4.3)$$

Em que: *rand* é um número obtido com distribuição normal no intervalo  $[0,1[$ . No sentido de simplificar eventuais referências, os operadores formulados em (4.1), (4.2) e (4.3) são doravante identificados como *cruzamento na calda* e *mutação na calda*.

## 4.6 Otimizando o algoritmo e as estruturas

Esta seção trata de algumas modificações simples que se adotadas podem melhorar o desempenho de Subtrativa. São duas subseções: na primeira, apresentam-se duas simplificações do Mapa das Malhas  $L$ , cujo propósito é minimizar o número de arestas nas malhas fundamentais da rede; e na segunda, propõe-se uma versão mais compacta de  $L$  que leva em consideração sua inerente esparsidade.

### 4.6.1 Reduzindo o Mapa das Malhas

A melhoria mais simples no algoritmo como foi apresentado é excluir de  $L$  ainda na fase de pré-processamento toda coluna que seja completamente nula, uma vez que tal condição ocorre quando uma aresta não integra nenhuma malha. Afinal, seja qual for a configuração ótima, essa aresta estará habilitada. Embora apenas  $S_1$  pudesse ser eliminada desta forma no exemplo apresentado, é provável que várias arestas possam ser eliminadas para redes reais.

Outra melhoria possível está diretamente relacionada com o número de arestas compartilhadas em  $L$ . Na verdade, considerando que durante o processo de decodificação estas arestas implicam em atualizações, que por sua vez impõem maior complexidade computacional, é possível afirmar que quanto menos arestas compartilhadas menor a complexidade da decodificação. Propõe-se minimizar o número total de atualizações pela redução do número de chaves compartilhadas em  $L$ .

Mais uma vez, o algoritmo é definido em termos do Mapa das Malhas e do operador XOR. A idéia é testar XOR em todos os pares formados entre malhas em busca de reduções no número de arestas até que nenhuma redução seja possível. Para o exemplo, a condição de parada é que XOR não permita redução de malha para os seguintes pares:  $M_0-M_1$ ,  $M_0-M_2$ ,  $M_0-M_3$ ,  $M_0-M_4$ ,  $M_1-M_2$ ,  $M_1-M_3$ ,  $M_1-M_4$ ,  $M_2-M_3$ ,  $M_2-M_4$  e  $M_3-M_4$ .

O algoritmo 4.2 implementa o procedimento descrito em pseudo-código com a ajuda de uma matriz quadrada  $MM$  cuja ordem é  $n(A_{NA})$ ; usando um vetor temporário  $TMP$ ; supondo que existe uma função  $S_m$  que retorna o somatório dos elementos de um vetor; e considerando que existe um comando `continue`, que leva imediatamente à próxima avaliação de um laço. No início do algoritmo  $MM$  é inteiramente nula, de modo que o algoritmo termina somente quando todos os elementos do triângulo superior desta matriz forem não nulos:

Algoritmo 4.2 - Cálculo de L reduzido

```

1   boolean terminou=falso;
2   Enquanto (terminou==falso)
3       terminou=verdadeiro;
4       Para(inteiro i=0;i<n(ANA)-1;i++)
5           Para(inteiro j=i+1;j<n(ANA);j++)
6               Se(MM[i][j]!=0) continue;
7               TMP=XOR(L[i],L[j]);
8               MM[i][j]=1;
```

```

9           Se (Sm(L[i])>Sm(L[j])) e (Sm(L[i])>Sm(TMP))
10              L[i]=TMP;
11           Para (inteiro k=0;k<i;k++) MM[k][i]=0;
12           Para (inteiro k=i+1;k<j;k++) MM[i][k]=0;
13           terminou=falso;
14           continue;
15           Se (Sm(L[j])>Sm(L[i])) e (Sm(L[j])>Sm(TMP))
16              L[j]=TMP;
17           Para (inteiro k=0;k<i;k++) MM[k][j]=0;
18           terminou=falso;
19           continue;

```

Essa versão de  $L$  com menos arestas compartilhadas também trás consigo a vantagem de reduzir a influência da configuração utilizada para obtê-la nos eventuais resultados obtidos. Contudo, o procedimento descrito não garante a obtenção do Mapa das Malhas Mínimo de  $G_S$  que seria a condição ótima para reduzir as atualizações em  $L$ .

Retomando a rede exemplo da Figura 4.1, ao ser executado o Algoritmo 4.2 permite reduzir as malhas  $M_1$  e  $M_4$ , respectivamente, de 7 para 6 arestas e de 6 para 3 arestas. O Mapa das Malhas reduzido é apresentado a seguir:

$$L = \begin{array}{c} \begin{array}{cccccccccccccccc} & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 & S_8 & S_9 & S_{10} & S_{11} & S_{12} & S_{13} & S_{14} & S_{15} & S_{16} \\ M_0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ M_1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ M_2 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ M_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ M_4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{array} \end{array} .$$

#### 4.6.2 Uma versão vetorial para $L$

A formulação de  $L$  como foi feita na Seção 4.2, mesmo com as reduções propostas na subseção 4.6.1, desconsidera sua esparsidade. Uma alternativa para economizar espaço ao representar sistemas maiores e, simultaneamente, reduzir a complexidade do Algoritmo 4.1 consiste em utilizar um vetor de vetores em vez de uma matriz. A idéia é definir um vetor para cada linha  $L[i]$ , o qual armazena uma lista com todas as posições em que  $L[i]$  tem valores não nulos. Adicionalmente, cada elemento no vetor possui um

par  $\{vetor, posição\}$  que identifica a próxima ocorrência da aresta em questão no Mapa das Malhas, com o objetivo de simplificar o processo de atualização.

Para exemplificar, considere o Mapa das Malhas da Seção 4.2 e sua nova versão  $L'$  utilizando vetores com indexação começando em 0:

	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$	$S_8$	$S_9$	$S_{10}$	$S_{11}$	$S_{12}$	$S_{13}$	$S_{14}$	$S_{15}$	$S_{16}$
$M_0$	0	1	1	0	0	0	0	0	0	0	1	1	1	0	0	0
$M_1$	0	0	0	1	1	1	1	1	0	1	0	0	1	0	0	0
$M_2$	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
$M_3$	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0
$M_4$	0	0	0	0	0	0	0	1	0	1	1	1	0	1	0	1

e

$M_0$	1	2	10	11	12												
	0	0	0	0	3	1	3	2	1	6							
$M_1$	3	4	5	6	7	9	12										
	0	0	0	0	2	0	2	1	3	0	0	0					
$M_2$	6	7	8														
	0	0	4	0	0	0											
$M_3$	9	10	11	13	14												
	4	1	4	2	4	3	4	4	0	0							
$M_4$	7	9	10	11	13	15											
	0	0	0	0	0	0	0	0	0	0	0						

Considerou-se a aresta  $S_{i+1}$  associada a coluna  $i$  e uma aresta sem ocorrência posterior foi modelada fazendo  $vetor = 0$ . Por exemplo, a aresta associada a coluna 1 ( $S_2$ ) está presente na primeira posição de  $M_0$  e não ocorre em nenhuma outra malha, pois o valor armazenado em  $vetor$  é 0. Já a aresta associada a coluna 10 ( $S_{11}$ ) ocorre em  $M_0$  e novamente no vetor 3 ( $M_3$ ) na posição 1 (2º elemento do vetor), conforme indica o par  $\{vetor, posição\} = \{3, 1\}$ .

O arranjo de  $L$  usando vetores simplifica o operador XOR, que em vez de realizar a operação *ou-exclusivo*  $n(A_S)$  vezes, passa a simplesmente mesclar duas listas, evitando elementos repetidos – notação XOR'. Esse procedimento é semelhante ao implementado pelo método clássico de ordenação *MERGE-SORT* (CORMEN et al, 2002).

De posse de  $L'$  e de  $XOR'$ , é possível reescrever o algoritmo de decodificação de Subtrativa. As mesmas considerações feitas para o Algoritmo 4.1 se aplicam, mas considere adicionalmente que o acesso ao  $i$ -ésimo elemento do  $j$ -ésimo vetor é feito pela construção  $L'[i][j]$ , como era feito antes para uma matriz. O índice da coluna a que se refere o elemento é obtido de  $L'[i][j].col$  e o Par {vetor, posição} da próxima ocorrência da referida aresta é  $L'[i][j].vetpos$ :

Algoritmo 4.3 - Decodificação Subtrativa -  $L'$

```

1   Para (inteiro i=0; i<n(ANA); i++)
2       inteiro ich_m=inteiro(g[n(ANA)-i-1]* $\alpha$ [i]);
3       AS[L'[i][ich_m].col].Abre();
4       Par [vet,pos]=L'[i][ich_m].vetpos;
5       Enquanto (vet>0)
6           Par [vet_prox,pos_prox]=L'[vet][pos].vetpos;
7           L'[vet]=XOR'(L'[i],L'[vet]);
8           [vet,pos]=[vet_prox,pos_prox];

```

## 4.7 Métricas do Código

Como discutido no Capítulo 3, foram propostas no estado da arte uma série de métricas com o intuito de avaliar mais precisamente diferentes representações. O objetivo desta seção é apresentar Subtrativa tomando como base tais métricas.

### 4.7.1 Espaço

Utilizando notação assintótica para o espaço como fizeram Raidl & Julstrom (2003), o código Subtrativo ocupa  $O(n(A_{NA}))$ . Esse valor pode ser obtido a partir dos totais de vértices e arestas descontando os nós fonte, como:  $n(A_{NA}) = n(A_S) - n(X_L)$ .

### 4.7.2 Tempo

A métrica tempo engloba a complexidade dos algoritmos de mutação, cruzamento e decodificação. Em relação aos operadores genéticos, a complexidade é  $O(n(A_{NA}))$ . Tal

qual a métrica espaço, uma vez que se propõe utilizar operadores convencionais. Este valor é válido mesmo no caso das variantes de operadores propostas na Seção 4.5, pois os dados necessários podem ser obtidos sem complexidade adicional na etapa de avaliação dos indivíduos. A situação é diferente para a decodificação, já que a complexidade deve ser calculada com base no Algoritmo 4.3.

O trecho do Algoritmo 4.3 que predomina na análise de complexidade é a chamada do operador XOR' para atualização de  $L'$  (Linha 7). São três níveis de laços de repetição: no nível mais interno, XOR' é executado com complexidade  $O(n_1+n_2)$ , em que:  $n_1$  e  $n_2$  são o número de arestas em cada uma das malhas recebidas como argumento; o número de vezes que XOR' é chamado é igual ao número  $n_3$  de malhas que compartilham uma determinada aresta; e, externamente, essas operações são repetidas  $n(A_{NA})$  vezes. Os valores  $n_1$ ,  $n_2$  e  $n_3$  variam em função de  $G_S$  e do indivíduo que está sendo decodificado, de modo que, assumindo valores médios, o algoritmo como um todo seria  $O(n(A_{NA})n_3(n_1+n_2))$ .

Para analisar o pior caso, podemos assumir que  $n_1$ ,  $n_2$  e  $n_3$  são funções do número de arestas na rede, o que conduz a uma complexidade cúbica. Esse resultado é considerado negativo para o código Subtrativo, se comparado às complexidades obtidas pelas representações nomeadas como ótimas no Capítulo 2. Entretanto, considerando que as redes de distribuição geralmente são planares e assumindo que o procedimento de construção e redução de  $L$  ou  $L'$  se aproxima das malhas mínimas da rede:  $n_3 \approx 2$ . Também não parece razoável considerar que o comprimento médio das malhas cresça como função do número de arestas. Se fosse possível supor que  $n_1$  e  $n_2$  assumem algum valor constante, a complexidade do código subtrativo seria  $O(n(A_{NA}))$ . Nesse caso, o pessimismo de antes poderia estar dando lugar ao excesso de otimismo na definição das premissas.

De fato, esse tema demandaria um estudo mais detalhado, porém julga-se estar além do escopo do presente trabalho. Considera-se como uma estimativa razoável, sem maior profundidade, complexidade quadrática.

#### **4.7.3 Factibilidade**

A idéia da métrica factibilidade é maximizar o número de restrições inerentes ao problema que se deseja resolver que são incorporadas a um determinado código e seus operadores genéticos. Nesse sentido, Subtrativa é semelhante as melhores técnicas apresentadas no Capítulo 2, porque é capaz de representar somente configurações radiais conexas. Nenhuma das técnicas consultadas ou propostas se propõe a embutir no código outras restrições elétricas, cuja avaliação exigiria a execução de um algoritmo de fluxo de carga.

#### **4.7.4 Cobertura**

Por definição, o código Subtrativo representa apenas soluções radiais conexas e compreende todas as configurações deste tipo. Os operadores genéticos, por serem convencionais ou uma variação relacionada somente ao seu alcance, permitem que operações genéticas sucessivas gerem qualquer solução no espaço de busca. Portanto, a cobertura do código é plena.

#### **4.7.5 Tendência**

O conceito de tendência é amplo e se aplica ao código propriamente dito e aos seus respectivos operadores genéticos. Um código tendencioso no contexto da Configuração significa que ele tende a gerar um tipo particular de rede, o que pode ser uma vantagem, se a tendência é na direção da solução ótima global, ou uma desvantagem, caso contrário. Seja qual for a situação, a tendência do código tem efeito na etapa de

inicialização, reduzindo a diversidade da população inicial. Enquanto operadores genéticos tendenciosos têm impacto no processo de otimização propriamente dito, podendo gerar uma busca não uniforme através do espaço das soluções.

O procedimento adotado no estado da arte para avaliar tendência de uma estratégia de codificação particular tem sido comparar indivíduos gerados aleatoriamente por meio deste código com indivíduos gerados aleatoriamente por um código de referência, reconhecidamente sem tendência.

Rothlauf (2006) realizou várias simulações com grafos completos de diferentes tamanhos usando como referência a representação Número Prufer. Suas análises, em geral, utilizam o conceito de distância (subseção 3.2.3) entre as soluções geradas aleatoriamente e grafos em formato de estrela e árvores geradoras mínimas, com objetivo de pesquisar se o código tende a uma ou outra topologia.

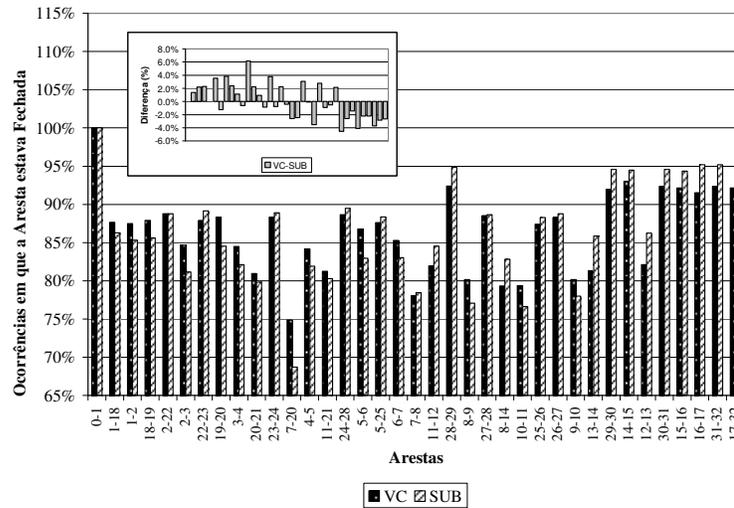
Raidl & Julstrom (2003) também utilizam grafos completos, porém o comparativo é com topologias dos tipos estrela e caminho, sendo adotada como métrica o diâmetro da árvore. Em que, *diâmetro* de uma árvore é o número de arestas do seu caminho mais longo.

O objetivo dessa seção é avaliar Subtrativa e seus operadores com relação à métrica tendência. Um procedimento semelhante ao que foi utilizado nas referências citadas é adotado, utilizando como referência a representação Vetor Característico – VC. Não faz sentido utilizar grafos completos, uma vez que redes de distribuição são geralmente fracamente malhadas. Utiliza-se o sistema teste apresentado na Figura 1.2 do Capítulo 1, porém considerando que todos os seus ramos possuem chave (BARAN & WU, 1989). Trata-se de uma rede com uma única subestação, 32 barras e 37 ramos com chave.

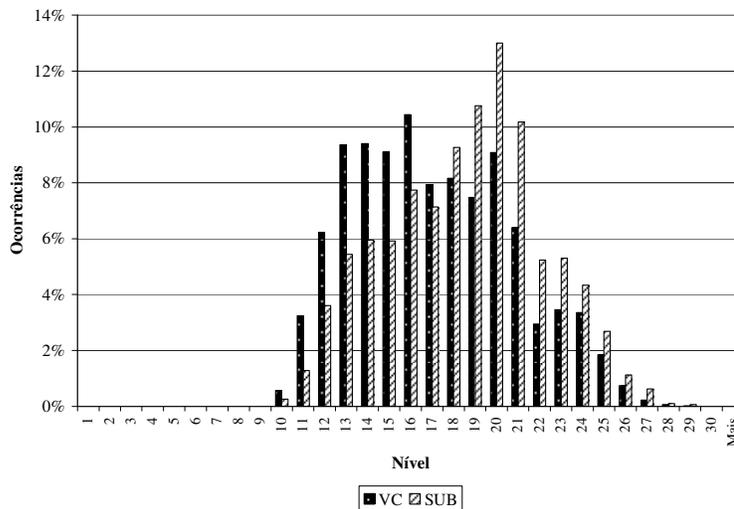
A idéia para levar a efeito a análise do código é utilizar conjuntos de 5000 indivíduos radiais conexos gerados de forma aleatória e avaliar três métricas de comparação: o percentual com que cada aresta comparece fechada, a profundidade máxima das árvores geradas e a distribuição das aptidões. Tais resultados são apresentados nas Figuras 4.3, 4.4 e 4.5.

Na Figura 4.3 plotou-se a frequência com que cada aresta aparece fechada na amostra, sendo apresentado no canto superior esquerdo as diferenças observadas fazendo VC menos Subtrativa. As arestas são posicionadas no eixo das abscissas de modo que as mais próximas da subestação fiquem à esquerda e as mais distantes à direita. Observam-se diferenças pouco significativas entre os códigos, sendo a diferença nula para a aresta 0-1, pois 100% das soluções radiais conexas exigem esta aresta habilitada; máxima para 7-20, quando Subtrativa mantém esta aresta fechada em uma taxa 6.2% menor que VC; e apresenta um valor médio de 2.3%, tomando todas as diferenças percentuais em módulo. Contudo, apesar da amplitude das diferenças ser pequena, há uma tendência consistente de Subtrativa em fechar com maior frequência as arestas plotadas no extremo direito do gráfico. Isto é, as arestas mais distantes da subestação.

Partindo para a análise com base na maior profundidade observada nas soluções (Figura 4.4), é possível verificar que a tendência de Subtrativa por fechar chaves mais distantes da subestação implica igualmente em uma tendência por codificar árvores com maior profundidade. A vantagem neste caso é que as diferenças observadas entre VC e Subtrativa são mais significativas, o que facilita a análise. Esse resultado sugere que a comparação em termos de profundidades é mais efetiva que o percentual de soluções em que as arestas aparecem fechadas, ainda que para esse caso, ambas as métricas resultem na mesma conclusão.

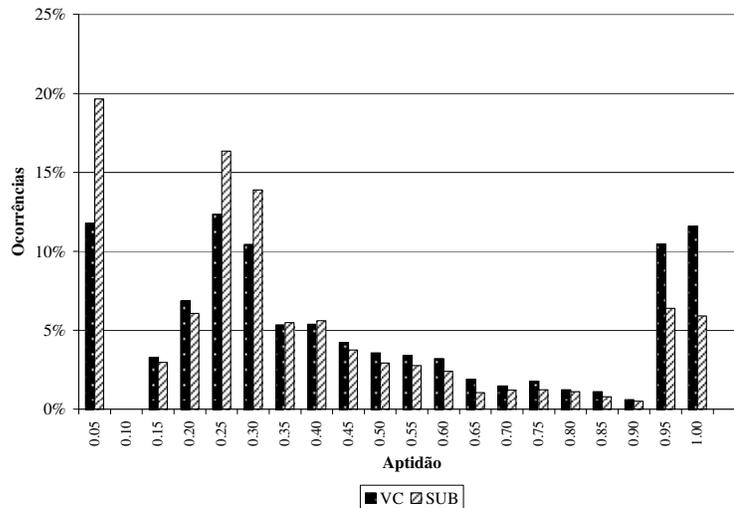


**Figura 4.3 – Comparação entre 5000 indivíduos gerados de forma aleatória utilizando as representações Vetor Característico – VC e Subtrativa – SUB: percentual em que cada aresta comparece com estado fechado nos indivíduos gerados.**



**Figura 4.4 – Comparação entre 5000 indivíduos gerados de forma aleatória utilizando as representações Vetor Característico – VC e Subtrativa – SUB: percentual de soluções por profundidade máxima.**

A análise com base nas aptidões (Figura 4.5), difere das anteriores porque é específica da Configuração de Redes. Sua dependência do problema ocorre através da função de aptidão, que neste caso foi formulada considerando exclusivamente as perdas ativas das soluções. Verifica-se no comparativo que Subtrativa tende a sobre-representar configurações com pouca aptidão e sub-representar soluções candidatas com alta aptidão. Para exemplificar, considere as duas últimas faixas do gráfico: 10.5% das



**Figura 4.5 – Comparação entre 5000 indivíduos gerados de forma aleatória utilizando as representações Vetor Característico – VC e Subtrativa – SUB: percentual de soluções por faixa de aptidão.**

soluções tem aptidão no intervalo  $[0.90 - 0.95[$  para VC contra 6.4% de soluções correspondentes usando Subtrativa; na faixa  $[0.95 - 1.00[$  os percentuais são, respectivamente, de 11.6% e 5.9%.

Em síntese, os testes realizados apontam que Subtrativa tende a representar árvores com maior profundidade, ou seja, tende ao tipo de rede caminho. Considerando que, em média, redes de distribuição com perdas mínimas possuem profundidade reduzida; teoricamente, a tendência se apresenta para o código Subtrativo aplicado à Configuração como uma característica negativa.

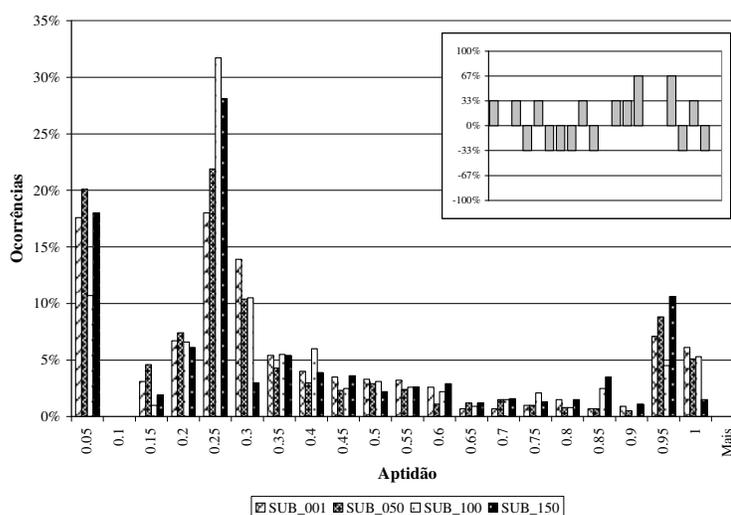
Partindo para a análise de tendência do operador de cruzamento, avaliou-se o comportamento de uma população com 1000 indivíduos no curso de 150 gerações sem pressão seletiva e adotando apenas operações de cruzamento. Em uma primeira etapa, o operador convencional é utilizado e, em seguida, adota-se o operador de cruzamento na calda definido na Seção 4.5.

A referência nesse caso é a própria população gerada de forma aleatória no início da simulação, sendo analisadas além das características da população inicial

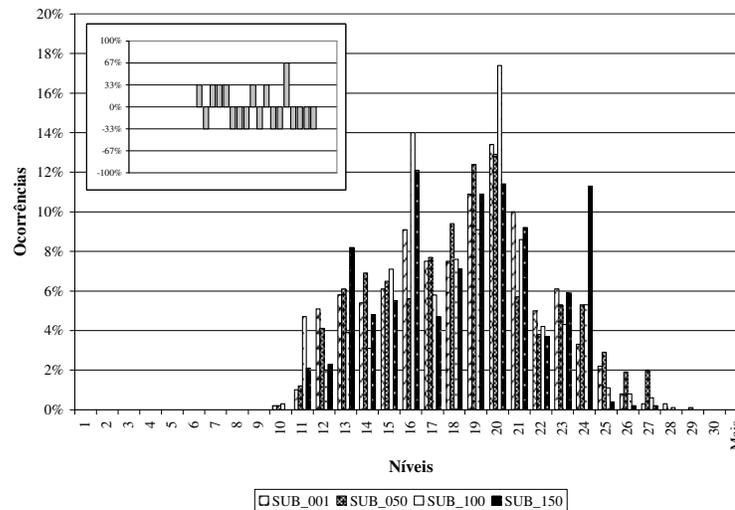
(geração 1), os resultados observados ao término das gerações 50, 100 e 150. Na Figura 4.6 apresenta-se a distribuição das aptidões quando se adota o cruzamento convencional, estando plotado no canto superior direito um *índice de tendência* calculado com base nos quatro valores disponíveis para cada intervalo de aptidão. O índice de tendência é uma percentagem no intervalo [-100,+100] e consiste em uma medida de como um conjunto de valores evoluiu durante a simulação. Por exemplo: um valor +100% significa que houve aumento em todas as transições, isto é: 1-50, 50-100 e 100-150; enquanto -100% representa redução em todas as transições.

Embora haja variações significativas em algumas faixas de aptidão da Figura 4.6 nas 150 gerações simuladas, aparentemente não há qualquer tendência a favor de um conjunto específico de soluções, sugerindo oscilações aleatórias. O índice de tendência reforça esse resultado, pois em nenhuma faixa de aptidão ele foi extremo, tampouco houve aglomeração consistente de valores negativos ou positivos. As aptidões médias observadas por geração avaliada foram: 0.36, 0.34, 0.36 e 0.37.

Apresentam-se na Figura 4.7 resultados da mesma simulação, porém



**Figura 4.6 – Evolução por faixa de aptidão para 1000 indivíduos com codificação Subtrativa implementando apenas cruzamento convencional, após as gerações: 1 (SUB\_001), 50 (SUB\_050), 100 (SUB\_100) e 150 (SUB\_150). Índices de Tendência no canto superior direito.**

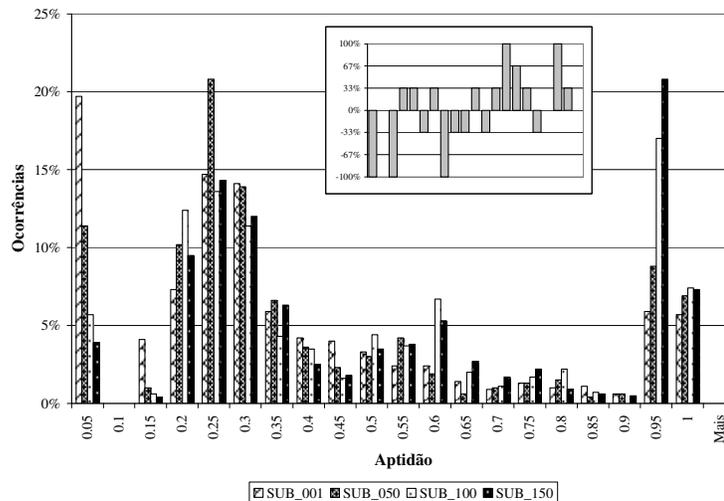


**Figura 4.7 – Evolução por profundidade máxima para 1000 indivíduos com codificação Subtrativa adotando cruzamento convencional, após as gerações: 1 (SUB\_001), 50 (SUB\_050), 100 (SUB\_100) e 150 (SUB\_150). Índices de Tendência no canto superior esquerdo.**

relativos à métrica profundidade máxima das árvores geradas. Dessa vez o gráfico com o índice de tendência foi posicionado no canto superior esquerdo. Novamente, as variações observadas no curso das gerações parecem aleatórias, não sendo observado nenhum caso em que o índice de tendência atinja um de seus extremos. As profundidades médias observadas nos quatro momentos pesquisados, corroboram essa análise: 18.32, 18.52, 18.26 e 18.46.

Nas Figuras 4.8 e 4.9 são apresentados os resultados obtidos ao repetir as simulações adotando a variação proposta na Seção 4.5: Cruzamento na Calda. Aparentemente, esta variação impõe uma tendência positiva ao código Subtrativo, porém moderada.

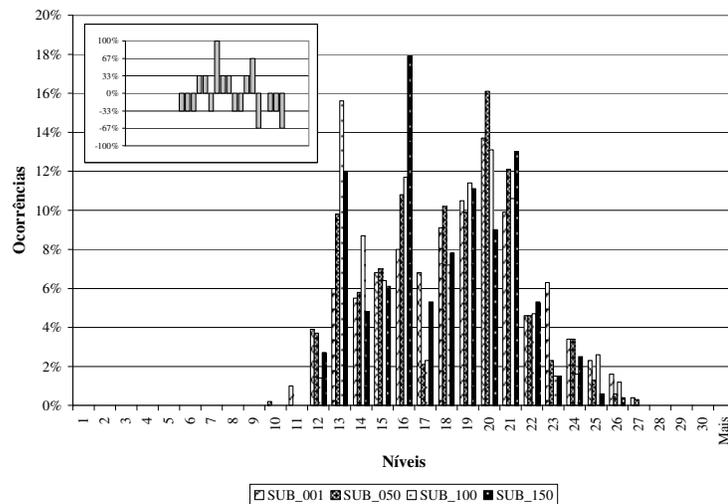
Considerando a Figura 4.8 é possível verificar que soluções com aptidão menor que 0.2 sofreram redução consistente em sua participação na população, enquanto soluções com aptidões maiores que 0.9 tiveram acréscimo. Este resultado fica mais explícito tomando como base o índice de tendência, em que observa-se uma concentração de valores negativos à esquerda e de valores positivos à direita. Os dados



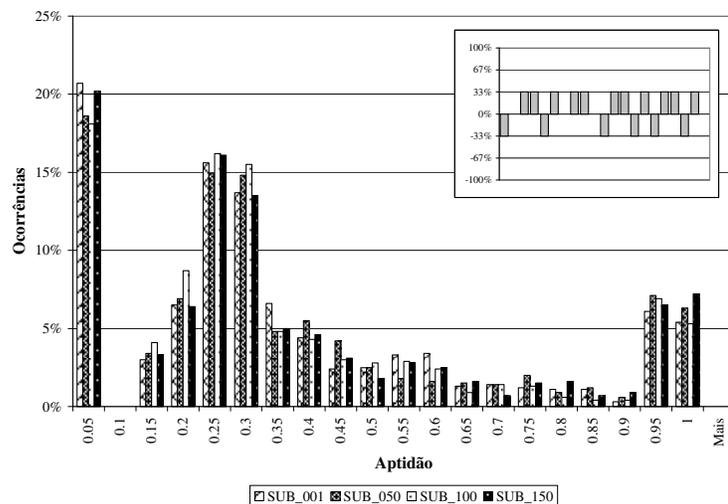
**Figura 4.8 – Evolução por faixa de aptidão para 1000 indivíduos com codificação Subtrativa implementando apenas cruzamento na calda, após as gerações: 1 (SUB\_001), 50 (SUB\_050), 100 (SUB\_100) e 150 (SUB\_150). Índices de Tendência no centro da figura.**

apresentados na Figura 4.9 permitem visualizar como ocorre tal melhoria: há redução na participação das soluções com profundidades menores que 13 e, igualmente, nas soluções com profundidades maiores que 22; sendo observado acréscimo nas soluções com profundidade entre 13 e 22. O acréscimo mais significativo ocorre nas soluções com profundidade máxima 16. Os seguintes valores médios foram obtidos: 0.34, 0.38, 0.48 e 0.51 para aptidões e 18.43, 18.06, 17.67 e 17.64 para profundidades, confirmando as inferências feitas.

Para finalizar as simulações de tendência de Subtrativa, resta testar o operador de mutação. De forma objetiva, deseja-se responder se o operador convencional ou a mutação na calda proposta na Seção 4.5, quando aplicados ao código Subtrativo, impõem algum tipo de tendência. Para tanto, foram levadas a efeito simulações similares àquelas realizadas para o cruzamento: avaliou-se o comportamento de uma população com 1000 indivíduos no curso de 150 gerações sem pressão seletiva e adotando apenas operações de mutação; primeiro com o operador convencional e, em seguida, com o operador de mutação na calda.



**Figura 4.9 – Evolução por profundidade máxima para 1000 indivíduos com codificação Subtrativa adotando apenas cruzamento na calda, após as gerações: 1 (SUB\_001), 50 (SUB\_050), 100 (SUB\_100) e 150 (SUB\_150). Índices de Tendência no canto superior esquerdo.**



**Figura 4.10 – Evolução por faixa de aptidão para 1000 indivíduos com codificação Subtrativa implementando apenas mutação convencional, após as gerações: 1 (SUB\_001), 50 (SUB\_050), 100 (SUB\_100) e 150 (SUB\_150). Índices de Tendência no canto superior direito.**

Na Figura 4.10 apresenta-se a distribuição das aptidões quando se adota a mutação convencional, estando plotado no canto superior direito os índices de tendência. Não há indicação de tendência a favor de um conjunto específico de soluções, sendo observadas variações pouco significativas nas aptidões e índices positivos e negativos distribuídos de modo aparentemente randômico. As aptidões médias observadas por geração foram: 0.34, 0.36, 0.34 e 0.36.

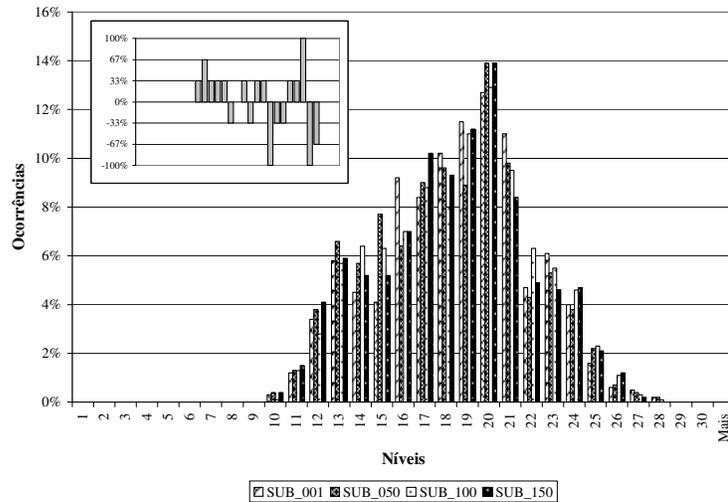
Apresentam-se na Figura 4.11 resultados da mesma simulação da Figura 4.10, entretanto relativos à métrica profundidade máxima das árvores geradas. O gráfico com o índice de tendência foi posicionado no canto superior esquerdo. Nesse caso, houve uma concentração de índices de tendência positivos no intervalo de 10 a 15, porém essa aparente tendência não se confirma quando se leva em consideração as pequenas amplitudes das variações observadas. As profundidades médias nos quatro momentos pesquisados foram as seguintes: 18.48, 18.23, 18.52 e 18.33.

Nas Figuras 4.12 e 4.13 são apresentados os resultados obtidos ao repetir as simulações adotando a mutação na calda. Diferente da tendência positiva observada na variante do cruzamento, os resultados obtidos para a mutação na calda não sugerem qualquer tendência. Isto é, foi mantida a operação uniforme verificada no operador de mutação convencional. Os seguintes valores médios foram obtidos: 0.36, 0.35, 0.37 e 0.35 para aptidões e 18.51, 18.49, 18.54 e 18.58 para profundidades, confirmando as inferências feitas.

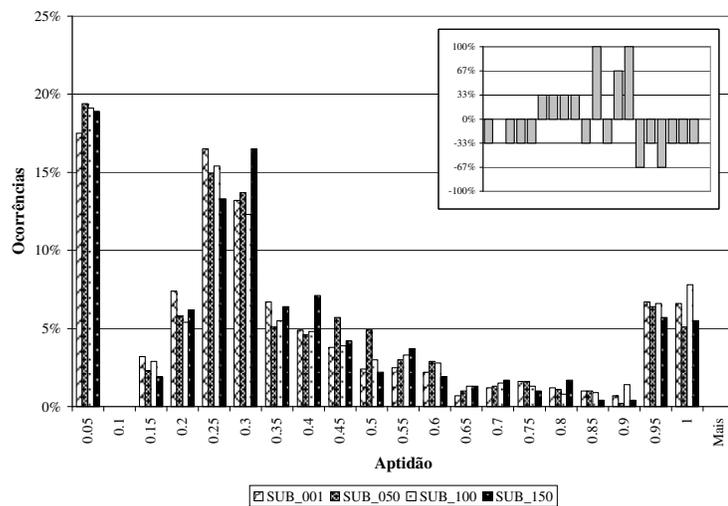
#### **4.7.6 Localidade**

Como formulado na subseção 3.2.3, a localidade é uma medida de quão bem genótipos vizinhos correspondem a fenótipos vizinhos. Vizinhos no espaço dos genótipos denota separados por uma única operação de mutação, enquanto o termo vizinhos no espaço dos fenótipos significa separados pela menor diferença possível entre soluções. No caso da Configuração, a diferença mínima entre duas topologias é de um par de arestas, uma vez que o número total de arestas habilitadas é constante para toda rede radial conexa.

O objetivo dessa seção é avaliar a localidade de Subtrativa quando são utilizados operadores de mutação convencional e na calda (Seção 4.5). De modo semelhante à subseção 4.7.5, utiliza-se o sistema teste apresentado na Figura 1.2



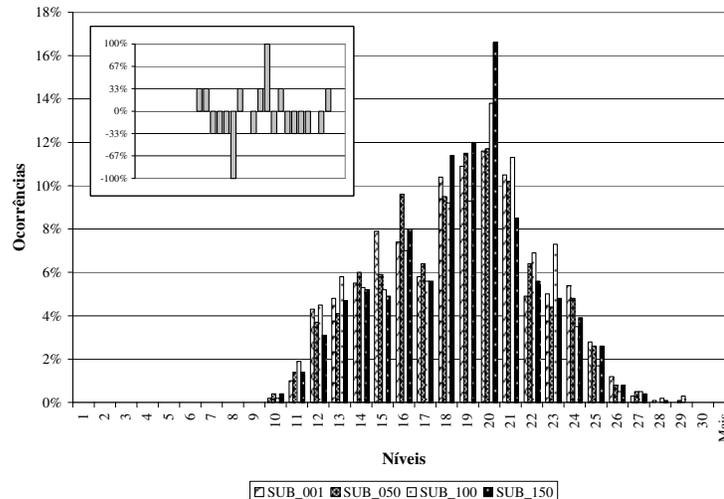
**Figura 4.11 – Evolução por profundidade máxima para 1000 indivíduos com codificação Subtrativa adotando mutação convencional, após as gerações: 1 (SUB\_001), 50 (SUB\_050), 100 (SUB\_100) e 150 (SUB\_150). Índices de Tendência no canto superior esquerdo.**



**Figura 4.12 – Evolução por faixa de aptidão para 1000 indivíduos com codificação Subtrativa implementando apenas mutação na calda, após as gerações: 1 (SUB\_001), 50 (SUB\_050), 100 (SUB\_100) e 150 (SUB\_150). Índices de Tendência no canto superior direito.**

considerando que todos os seus ramos possuem chave (BARAN & WU, 1989). A simulação consiste em aplicar uma única vez o operador de mutação em 5000 indivíduos codificados usando Subtrativa e registrar, para cada um dos pares pai-filho, o respectivo número de arestas diferentes (d1f). Observe que para uma localidade ideal, esta diferença seria sempre igual a dois:  $d1f = 2$ .

Os resultados obtidos com o operador convencional e com a mutação na



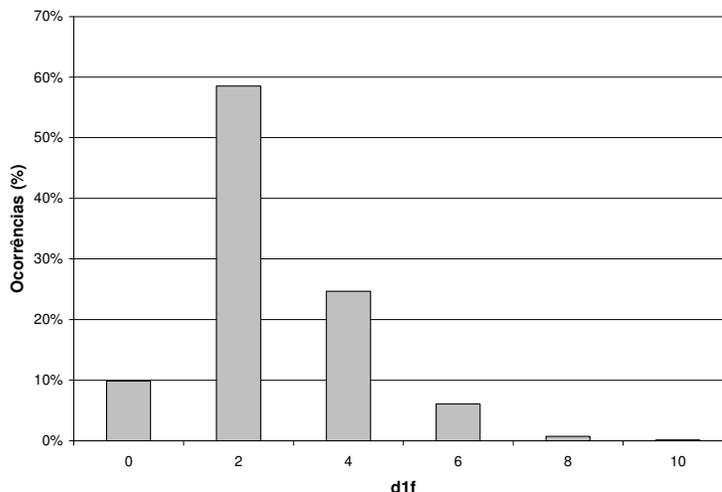
**Figura 4.13 – Evolução por profundidade máxima para 1000 indivíduos com codificação Subtrativa adotando mutação na calda, após as gerações: 1 (SUB\_001), 50 (SUB\_050), 100 (SUB\_100) e 150 (SUB\_150). Índices de Tendência no canto superior esquerdo.**

calda, nesta ordem, são apresentados nas Figuras 4.14 e 4.15. Não há diferença significativa de um operador para o outro, sendo obtida para ambos a condição ideal ( $d1f = 2$ ) em, aproximadamente, 60% dos casos. Considerando valores médios há uma diferença mínima: 2.60 para mutação convencional e 2.53 para mutação na calda. Embora retratem uma condição diferente da ideal, os valores obtidos asseguram uma localidade adequada ao código Subtrativo.

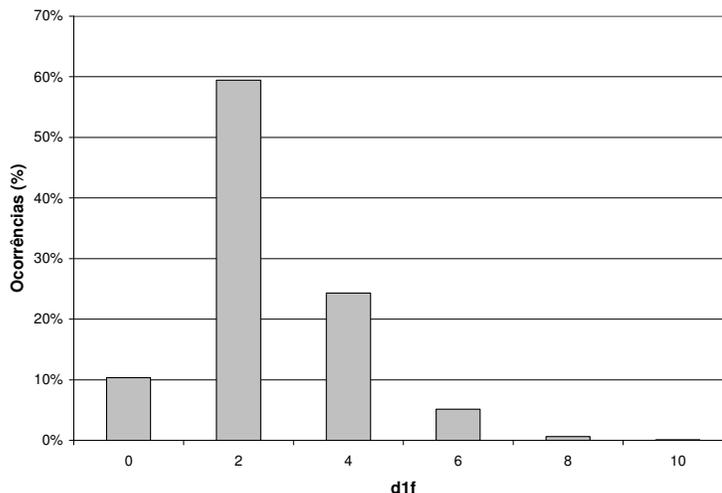
Uma observação interessante é o fato de em, aproximadamente, 10% das amostras a mutação não provocar nenhuma alteração nos fenótipos. Ela sugere que deve ser adotada para Subtrativa uma taxa de mutação um pouco maior que o usual, de modo que o número efetivo de mutações seja o desejado. Recalculando os percentuais das Figuras 4.14 e 4.15, desconsiderando as ocorrências em que  $d1f=0$ , resulta no seguinte percentual aproximado para a condição ideal ( $d1f = 2$ ): 66.67%.

#### 4.7.7 Hereditariedade

O conceito de hereditariedade estabelece que um indivíduo gerado por cruzamento deve representar uma solução que combina majoritariamente partes das soluções associadas



**Figura 4.14 – Taxas de ocorrência observadas para distâncias entre pai e filho (d1f) após simulação com 5000 mutações convencionais com código Subtrativo.**



**Figura 4.15 – Taxas de ocorrência observadas para distâncias entre pai e filho (d1f) após simulação com 5000 mutações na calda com código Subtrativo.**

aos seus pais (subseção 3.2.3). Aplicado à Configuração, implica que se dois indivíduos 1 e 2 gerarem um filho  $f$  por cruzamento, as distâncias entre 1 e  $f$  ( $d1f$ ) e entre 2 e  $f$  ( $d2f$ ) devem ser menores ou iguais à distância observada entre 1 e 2 ( $d12$ ). Novamente, distância entre soluções é definida como número de chaves em que o estado (aberta – fechada) é diferente.

O objetivo dessa seção é avaliar a hereditariedade do código Subtrativo quando são utilizados operadores de cruzamento convencional e na calda (Seção 4.5). O

sistema teste da Figura 1.2 é utilizado mais uma vez, supondo que todos os seus ramos têm chave. A idéia é gerar 5000 filhos por meio de operações de cruzamento em pais criados de forma aleatória usando Subtrativa, registrando em cada um desses casos as distâncias  $d_{12}$ ,  $d_{1f}$  e  $d_{2f}$ .

Para uma hereditariedade ideal,  $d_{12} \geq d_{1f}$  e  $d_{12} \geq d_{2f}$  deve sempre ocorrer. Logo, ilustram-se nas Tabelas 4.2 e 4.3 as taxas de ocorrência verificadas nas simulações comparando-se  $d_{12}$  com o maior valor entre  $d_{1f}$  e  $d_{2f}$ . Primeiro, para o cruzamento convencional e, em seguida, o cruzamento na calda. Os valores sombreados em ambas as tabelas destacam os casos em que observou-se distância do filho para um dos pais maior que a distância entre os pais, violando a condição ideal almejada. Observe que o número de violações é reduzido, totalizando 3.8% e 3.5% quando são utilizados, respectivamente, os operadores convencional e na calda.

Embora retratem uma condição diferente da ideal, os valores obtidos asseguram uma hereditariedade adequada ao código Subtrativo, sendo observada uma vantagem pouco significativa quando o operador de cruzamento na calda é escolhido.

#### 4.7.8 Restrição

A métrica restrição se confunde com factibilidade, estando ambas relacionadas com a inclusão no código e operadores genéticos do maior número de restrições inerentes ao

**Tabela 4.2 – Taxas de ocorrência obtidas no estudo de hereditariedade: distância entre pais ( $d_{12}$ ) versus maior distância observada entre pais e filhos ( $d_{1f}$  ou  $d_{2f}$ ) após operação de cruzamento convencional em indivíduos com codificação Subtrativa.**

	Maior entre $d_{1f}$ e $d_{2f}$ (%)							Total geral
	0	2	4	6	8	10		
d12	0	0.04	0.00	0.00	0.00	0.00	0.00	0.04
	2	0.00	0.10	0.04	0.02	0.00	0.00	0.16
	4	0.00	0.24	1.84	0.62	0.08	0.00	2.78
	6	0.00	0.00	4.50	9.66	1.74	0.06	15.96
	8	0.00	0.00	2.88	17.18	21.56	1.24	42.86
	10	0.00	0.00	0.00	9.56	15.34	13.30	38.20
Total geral	0.04	0.34	9.26	37.04	38.72	14.60	100.00	

**Tabela 4.3 – Taxas de ocorrência obtidas no estudo de hereditariedade: distância entre pais (d12) versus maior distância observada entre pais e filhos (d1f ou d2f) após operação de cruzamento na calda em indivíduos com codificação Subtrativa.**

		Maior entre d1f e d2f (%)						Total geral
		0	2	4	6	8	10	
d12	0	0.04	0.00	0.00	0.00	0.00	0.00	0.04
	2	0.00	0.20	0.02	0.00	0.00	0.00	0.22
	4	0.00	0.26	1.74	0.28	0.06	0.00	2.34
	6	0.00	0.00	3.88	10.70	1.50	0.08	16.16
	8	0.00	0.00	1.96	15.00	23.14	1.54	41.64
	10	0.00	0.00	0.00	5.96	14.42	19.22	39.60
Total geral		0.04	0.46	7.60	31.94	39.12	20.84	100.00

problema que se deseja resolver. Nesse sentido, os argumentos apresentados na subseção 4.7.3 são válidos aqui: o código Subtrativo é semelhante às melhores técnicas apresentadas no Capítulo 2, sendo capaz de representar somente configurações de rede radiais conexas.

#### 4.7.9 Hibridização

Raidl & Julstrom (2003) definem a métrica hibridização como a habilidade de embutir heurísticas do problema que se deseja resolver diretamente nos operadores genéticos. Este trabalho não inclui qualquer tentativa nesse sentido, podendo ser um dos desdobramentos para pesquisas futuras. Contudo, propõe-se uma modalidade diferente no Capítulo 6: *hibridização no código*.

## Capítulo 5

# Codificação Sequencial Aditiva

---

Este capítulo foi organizado em seis seções: na primeira, denominada de Matemática de Setores, apresenta-se uma forma sistemática de operar com a idéia de fronteira em um grafo  $G_S$ ; as duas seções seguintes descrevem, respectivamente, a codificação Aditiva e seu algoritmo de decodificação; a quarta seção é dedicada aos operadores genéticos; na seção posterior, algumas melhorias nos algoritmos e estruturas de dados até então utilizados são expostas, visando reduzir a complexidade da representação; e, por fim, na seção Métricas do Código, é feita uma análise sobre como o código Aditivo se posiciona em relação as métricas habitualmente adotadas na avaliação comparativa de representações.

### 5.1 Matemática de Setores

A codificação Aditiva parte do grafo  $G_S$  com todas as arestas desabilitadas e procede a conexão dos nós de carga um por vez, sequencialmente. Antes de conectar o primeiro nó de carga, o conjunto de arestas candidatas ou de *fronteira* é formado apenas pelas arestas incidentes em nós fonte. Ao habilitar a primeira aresta, um nó de carga é conectado à rede e o conjunto de fronteira precisa ser atualizado. Esse processo de atualização é contínuo, de modo que criar e atualizar a fronteira constituem um conceito chave para o código Aditivo.

Nesta seção propõe-se uma estrutura semelhante ao Mapa das Malhas apresentado na Seção 4.2, denominada de Mapa dos Setores, cuja finalidade é viabilizar o controle e atualização do conjunto de arestas de fronteira com complexidade mínima. O Mapa dos Setores  $R$  resulta da alteração de uma estratégia clássica de representação de grafos: matriz de incidências. Ela consiste em uma matriz inteira com  $n(X_S)$  linhas e  $n(A_S)$  colunas. O número de elementos não-negativos da linha  $i$  é igual ao número de arestas incidentes no nó  $i$ , enquanto o número de elementos não-negativos em qualquer coluna é igual a dois, uma vez que toda aresta de  $A_S$  é necessariamente incidente em dois setores.

A diferença do mapa proposto em relação à representação clássica é que, para tirar vantagem do fato das colunas terem sempre apenas dois elementos não-negativos, define-se da seguinte maneira um elemento  $(i, j)$  de  $R$ : o valor é negativo se a aresta  $j$  não incidir no setor  $i$ ; e, em caso contrário, tal valor identifica a linha de  $R$  associada ao outro setor adjacente a  $j$ . Essa idéia fica mais clara no exemplo numérico adiante.

Além de  $R$ , define-se um novo operador de ou-exclusivo com operação sobre sequências de valores inteiros, denominado de IXOR. Este operador utiliza o seguinte quadro funcional quando aplicado termo a termo – sendo: N qualquer inteiro negativo e PZ qualquer inteiro não-negativo:

Argumentos		IXOR
N	N	N
N	PZ	PZ*
PZ	N	PZ*
PZ	PZ	N

\*o resultado é o próprio argumento PZ.

Retomando o caso exemplo apresentado na Figura 3.3 (Capítulo 3),  $R$  é definido por 12 linhas e 16 colunas da seguinte maneira (valores negativos omitidos):

	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>	S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>	S <sub>12</sub>	S <sub>13</sub>	S <sub>14</sub>	S <sub>15</sub>	S <sub>16</sub>
0	<b>1</b>															
1	<b>0</b>	<b>6</b>										<b>2</b>		<b>10</b>		
2											<b>3</b>	<b>1</b>				
3										<b>4</b>	<b>2</b>		<b>7</b>			
4								<b>11</b>	<b>5</b>	<b>3</b>					<b>10</b>	
5						<b>9</b>	<b>11</b>		<b>4</b>							
6		<b>1</b>	<b>7</b>													
7			<b>6</b>	<b>8</b>									<b>3</b>			
8				<b>7</b>	<b>9</b>											
9					<b>8</b>	<b>5</b>										
10														<b>1</b>	<b>4</b>	<b>11</b>
11							<b>5</b>	<b>4</b>								<b>10</b>

Observe que há necessariamente dois valores não-negativos por coluna, nas linhas associadas aos setores adjacentes a cada uma das chaves. Para exemplificar, considere a chave S<sub>6</sub>: o valor 9 na posição (5, S<sub>6</sub>) significa que ela é incidente no setor 5 e sua outra extremidade incide em 9. A utilidade dessa propriedade se comprova adiante, na definição do Algoritmo 5.1. O conjunto de arestas de fronteira para a condição inicial é obtido aplicando IXOR às linhas de R associadas aos nós fontes; porém, como o único nó fonte é 0, sua linha é usada diretamente obtendo-se:

	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>	S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>	S <sub>12</sub>	S <sub>13</sub>	S <sub>14</sub>	S <sub>15</sub>	S <sub>16</sub>
atual =	<b>1</b>															

Partindo dessa configuração inicial, o conjunto de arestas de fronteira é formado exclusivamente por S<sub>1</sub>. Habilitar esta chave significa executar IXOR entre a configuração atual e a linha 1 de R, uma vez que é este o valor armazenado na posição selecionada:

	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>	S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>	S <sub>12</sub>	S <sub>13</sub>	S <sub>14</sub>	S <sub>15</sub>	S <sub>16</sub>
IXOR: atual	<b>1</b>															
R[1]	<b>0</b>	<b>6</b>										<b>2</b>		<b>10</b>		
atual		<b>6</b>										<b>2</b>		<b>10</b>		

O novo conjunto de arestas de fronteira é obtido diretamente do novo atual: S<sub>2</sub>, S<sub>12</sub> e S<sub>14</sub>.

Supondo que a aresta  $S_{14}$  seja a próxima selecionada, este procedimento é implementado novamente por uma operação de IXOR, agora entre a configuração *atual* e a linha 10 de  $R$ :

		$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$	$S_8$	$S_9$	$S_{10}$	$S_{11}$	$S_{12}$	$S_{13}$	$S_{14}$	$S_{15}$	$S_{16}$
IXOR:	atual		<b>6</b>										<b>2</b>		<b>10</b>		
	$R[10]$														<b>1</b>	<b>4</b>	<b>11</b>
	atual		<b>6</b>											<b>2</b>		<b>4</b>	<b>11</b>

O novo conjunto de arestas de fronteira é  $S_2$ ,  $S_{12}$ ,  $S_{15}$  e  $S_{16}$ . Na verdade, observe que a informação sobre as arestas de fronteira fica encapsulada no vetor *atual*, sendo possível guiar todo o processo com base exclusivamente nele.

Tendo definido uma forma precisa de obter a lista de arestas de fronteira para diferentes condições, é possível partir para a formulação efetiva da estratégia de codificação Aditiva.

## 5.2 Codificação

A idéia é que cada gene informe uma aresta da fronteira a ser habilitada, sendo o conjunto de fronteira atualizado continuamente utilizando  $R$  e IXOR, conforme descrito na Seção 5.1. O complicador de tal representação é o fato da fronteira ter tamanho variável, conduzindo a um alfabeto dinâmico como ocorreu com Subtrativa (Capítulo 4). Naturalmente, o artifício para lidar com esta dinâmica é o mesmo descrito anteriormente: define-se um cromossomo para o código Aditivo como sendo formado por genes reais com valor restrito ao intervalo  $[0,1]$ . O número de genes necessários é igual ao de nós de carga  $n(X_L) = n(A_{ST})$ , visto que cada gene conecta necessariamente um nó de carga à rede.

A codificação foi nomeada de Aditiva porque começa com o grafo  $G_S$  com todas as arestas desabilitadas e opera habilitando – adicionando – uma aresta por

vez até definir uma floresta  $T_S$  em  $G_S$ . Em virtude dos genes serem reais, mais uma vez há vários cromossomos que codificam uma mesma solução e não é possível formular uma equação direta para o cálculo do espaço de busca. O fato é que, como todas as possíveis conexões são consideradas no processo de avaliação dos genes, o espaço de busca resultante contém todas as configurações radiais conexas e somente elas. Um exemplo numérico que ilustra melhor o código proposto é apresentado na Seção 5.3.

### 5.3 Decodificação

O objetivo da decodificação para Aditiva é converter um cromossomo como descrito na Seção 5.2 em uma lista de ramos que juntos estabelecem uma árvore  $T_S$  em  $G_S$ . Do ponto de vista elétrico, o processo consiste em obter uma lista de chaves a serem fechadas. Para levar a efeito tal procedimento é necessário criar previamente  $R$  como descrito na Seção 5.1, iniciar apropriadamente o vetor `atual` e carregar a variável `acc` com o número de elementos não-negativos de `atual`. De posse destas variáveis auxiliares, todas as arestas de  $G_S$  são desabilitadas e o algoritmo de decodificação pode começar.

A interpretação de um cromossomo consiste em executar  $n(A_{ST})$  vezes um mesmo procedimento como descrito no Algoritmo 5.1. Considera-se  $g[i]$  como o gene  $i$ ,  $A_S$  como o vetor de arestas, `Fecha` como a rotina que implementa o fechamento de uma aresta e `IXOR` retornando versões atualizadas de `atual` e `acc`.

#### Algoritmo 5.1 - Decodificação Aditiva

```

//linhas iniciadas com "//" são comentários
1 Para(inteiro i=0;i<n(AST);i++)
    //ich_f: [0 = 1°chave], [1 = 2°chave], ...
2 inteiro ich_f=inteiro(g[n(AST)-i-1]*acc);
3 inteiro col=-1;
4 inteiro soma=0;

```

```

//col: marca a coluna em R selecionada por ich_f
5 Enquanto(soma<=ich_f)
6     Se(atual[++col]>=0) soma++;
//implementa o fechamento da aresta
7 As[col].Fecha();
//atualiza R conforme apresentado na Seção 5.1
8 [atual,acc]=IXOR(atual,R[atual[col]]);

```

Retomando o exemplo da Figura 3.3, propõe-se decodificar o seguinte cromossomo para ilustrar o Algoritmo 5.1 em ação:

$$v = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|} \hline 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ \hline 1^{\text{a}} & 2^{\text{a}} & 3^{\text{a}} & 4^{\text{a}} & 5^{\text{a}} & 6^{\text{a}} & 7^{\text{a}} & 8^{\text{a}} & 9^{\text{a}} & 10^{\text{a}} & 11^{\text{a}} \\ \hline \end{array}$$

Por uma questão de conveniência,  $R$  é repetido a seguir. Além dele, são fornecidos o vetor `atual` e a variável `acc` iniciais:

	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>	S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>	S <sub>12</sub>	S <sub>13</sub>	S <sub>14</sub>	S <sub>15</sub>	S <sub>16</sub>
0	<b>1</b>															
1	<b>0</b>	<b>6</b>										<b>2</b>		<b>10</b>		
2											<b>3</b>	<b>1</b>				
3										<b>4</b>	<b>2</b>		<b>7</b>			
4								<b>11</b>	<b>5</b>	<b>3</b>					<b>10</b>	
5						<b>9</b>	<b>11</b>		<b>4</b>							
6		<b>1</b>	<b>7</b>													
7			<b>6</b>	<b>8</b>									<b>3</b>			
8				<b>7</b>	<b>9</b>											
9					<b>8</b>	<b>5</b>										
10														<b>1</b>	<b>4</b>	<b>11</b>
11							<b>5</b>	<b>4</b>								<b>10</b>

	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>	S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>	S <sub>12</sub>	S <sub>13</sub>	S <sub>14</sub>	S <sub>15</sub>	S <sub>16</sub>	acc
atual	<b>1</b>																<b>1</b>

A decodificação deste cromossomo é implementada em 11 iterações:

### Iteração 1 – Decodificação do gene 10

Nesta primeira iteração `acc=1` e, seja qual for o alelo do primeiro gene, implica `valor=0`; resultando na seleção de  $S_1$  e em aplicar  $IXOR(atual, R[1])$ . O resultado é um novo vetor `atual` válido para a interpretação do próximo gene na iteração 2:

	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>	S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>	S <sub>12</sub>	S <sub>13</sub>	S <sub>14</sub>	S <sub>15</sub>	S <sub>16</sub>	acc	
atual	<b>1</b>																	
R[1]	<b>0</b>	<b>6</b>										<b>2</b>		<b>10</b>				
novo atual		<b>6</b>										<b>2</b>		<b>10</b>				<b>3</b>

### Iteração 2 – Decodificação do gene 9

Executando a linha 2 do Algoritmo 5.1:  $\text{valor}=\text{inteiro}(0.5*3)=\text{inteiro}(1.5)=1$ ; o que implica selecionar o segundo valor candidato em atual: S<sub>12</sub>. Como o valor desta posição em atual é 2, a atualização é realizada da seguinte maneira:

	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>	S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>	S <sub>12</sub>	S <sub>13</sub>	S <sub>14</sub>	S <sub>15</sub>	S <sub>16</sub>	acc	
atual		<b>6</b>										<b>2</b>		<b>10</b>				
R[2]											<b>3</b>	<b>1</b>						
novo atual		<b>6</b>									<b>3</b>			<b>10</b>				<b>3</b>

### Iteração 3 – Decodificação do gene 8

Executando a linha 2 do Algoritmo 5.1:  $\text{valor}=\text{inteiro}(0.5*3)=\text{inteiro}(1.5)=1$ ; o que implica selecionar o segundo valor candidato em atual: S<sub>11</sub>. O valor em R é 3, então:

	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>	S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>	S <sub>12</sub>	S <sub>13</sub>	S <sub>14</sub>	S <sub>15</sub>	S <sub>16</sub>	acc	
atual		<b>6</b>									<b>3</b>			<b>10</b>				
R[3]										<b>4</b>	<b>2</b>		<b>7</b>					
novo atual		<b>6</b>								<b>4</b>			<b>7</b>	<b>10</b>				<b>4</b>

### Iteração 4 – Decodificação do gene 7

Executando a linha 2 do Algoritmo 5.1:  $\text{valor}=\text{inteiro}(0.5*4)=\text{inteiro}(2)=2$ ; o que implica selecionar o terceiro valor candidato em atual: S<sub>13</sub>. O valor em R é 7, assim:

	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>	S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>	S <sub>12</sub>	S <sub>13</sub>	S <sub>14</sub>	S <sub>15</sub>	S <sub>16</sub>	acc	
atual		<b>6</b>								<b>4</b>			<b>7</b>	<b>10</b>				
R[7]			<b>6</b>	<b>8</b>									<b>3</b>					
novo atual		<b>6</b>	<b>6</b>	<b>8</b>						<b>4</b>				<b>10</b>				<b>5</b>

### Iteração 5 – Decodificação do gene 6

Executando a linha 2 do Algoritmo 5.1:  $\text{valor}=\text{inteiro}(0.5*5)=\text{inteiro}(2.5)=2$ ; o que implica selecionar o terceiro valor candidato em atual: S<sub>4</sub>. O valor em R é 8, logo:

	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>	S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>	S <sub>12</sub>	S <sub>13</sub>	S <sub>14</sub>	S <sub>15</sub>	S <sub>16</sub>	acc
atual		6	6	8						4				10			
R[8]				7	9												
novo atual		6	6		9					4				10			5

### Iteração 6 – Decodificação do gene 5

Executando a linha 2 do Algoritmo 5.1:  $\text{valor}=\text{inteiro}(0.5*5)=\text{inteiro}(2.5)=2$ ; o que implica selecionar o terceiro valor candidato em atual: S<sub>5</sub>. O valor em R é 9, logo:

	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>	S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>	S <sub>12</sub>	S <sub>13</sub>	S <sub>14</sub>	S <sub>15</sub>	S <sub>16</sub>	acc
atual		6	6		9					4				10			
R[9]					8	5											
novo atual		6	6			5				4				10			5

### Iteração 7 – Decodificação do gene 4

Executando a linha 2 do Algoritmo 5.1:  $\text{valor}=\text{inteiro}(0.5*5)=\text{inteiro}(2.5)=2$ ; o que implica selecionar o terceiro valor candidato em atual: S<sub>6</sub>. O valor em R é 5, então:

	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>	S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>	S <sub>12</sub>	S <sub>13</sub>	S <sub>14</sub>	S <sub>15</sub>	S <sub>16</sub>	acc
atual		6	6			5				4				10			
R[5]						9	11		4								
novo atual		6	6				11		4	4				10			6

### Iteração 8 – Decodificação do gene 3

Executando a linha 2 do Algoritmo 5.1:  $\text{valor}=\text{inteiro}(0.5*6)=\text{inteiro}(3)=3$ ; o que implica selecionar o quarto valor candidato em atual: S<sub>9</sub>. O valor em R é 4, desta forma:

	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>	S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>	S <sub>12</sub>	S <sub>13</sub>	S <sub>14</sub>	S <sub>15</sub>	S <sub>16</sub>	acc
atual		6	6				11		4	4				10			
R[4]								11	5	3					10		
novo atual		6	6				11	11						10	10		6

### Iteração 9 – Decodificação do gene 2

Executando a linha 2 do Algoritmo 5.1:  $\text{valor}=\text{inteiro}(0.5*6)=\text{inteiro}(3)=3$ ; o que implica selecionar o quarto valor candidato em atual: S<sub>8</sub>. O valor em R é 11, assim:

	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>	S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>	S <sub>12</sub>	S <sub>13</sub>	S <sub>14</sub>	S <sub>15</sub>	S <sub>16</sub>	acc
atual		6	6				11	11						10	10		
R[11]							5	4								10	
novo atual		6	6											10	10	10	
																	5

### Iteração 10 – Decodificação do gene 1

Executando a linha 2 do Algoritmo 5.1:  $\text{valor} = \text{inteiro}(0.5 * 5) = \text{inteiro}(2.5) = 2$ ; o que implica selecionar o terceiro valor candidato em *atual*: S<sub>14</sub>. O valor em *R* é 10, então:

	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>	S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>	S <sub>12</sub>	S <sub>13</sub>	S <sub>14</sub>	S <sub>15</sub>	S <sub>16</sub>	acc
atual		6	6											10	10	10	
R[10]														1	4	11	
novo atual		6	6														
																	2

### Iteração 11 – Decodificação do gene 0

Executando a linha 2 do Algoritmo 4.3:  $\text{valor} = \text{inteiro}(0.5 * 2) = \text{inteiro}(1) = 1$ ; o que implica selecionar o segundo valor candidato em *atual*: S<sub>3</sub>. O valor em *R* é 6, logo:

	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>	S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>	S <sub>12</sub>	S <sub>13</sub>	S <sub>14</sub>	S <sub>15</sub>	S <sub>16</sub>	acc
atual		6	6														
R[6]		1	7														
novo atual																	
																	0

Esta última atualização é desnecessária, visto que sempre leva a um vetor inteiramente negativo:  $\text{acc} = 0$ . Ela foi incluída para demonstrar que após habilitar 11 arestas, seguramente não resta nenhum setor desconectado e *atual* está vazio. A configuração de rede obtida neste exemplo é apresentada na Figura 5.1.

## 5.4 Operadores Genéticos

Do mesmo modo que Subtrativa, Aditiva admite que sejam utilizados operadores genéticos convencionais com garantia de que nenhuma configuração que viole as restrições de radialidade e conexão seja gerada. É igualmente verdade que essa

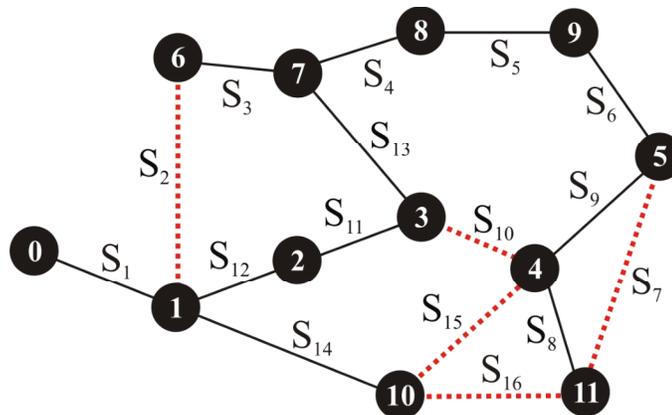


Figura 5.1 – Rede obtida pelo Algoritmo 5.1 decodificando o cromossomo exemplo de Aditiva.

característica não garante que os operadores convencionais proporcionem desempenho ótimo. Assim, propõe-se utilizar com o código Aditivo os mesmos operadores de cruzamento e mutação na calda descritos na Seção 4.5.

## 5.5 Otimizando o algoritmo e as estruturas

Esta seção trata de algumas modificações simples que se adotadas podem melhorar o desempenho de Aditiva. São duas subseções: na primeira, duas simplificações do Mapa dos Setores  $R$  são descritas, cujo propósito é minimizar o número de setores e arestas a serem considerados no processo de otimização; e na segunda, propõe-se uma versão mais compacta de  $R$  que leva em consideração sua esparsidade.

### 5.5.1 Reduzindo o Mapa dos Setores

A idéia é incluir no final da etapa de pré-processamento uma checagem para verificar potenciais simplificações em  $R$ . A primeira simplificação consiste em excluir de  $R$  setores que sejam eletricamente inatingíveis; tal condição pode vir a acontecer principalmente em estudos de Restabelecimento, nos quais um ou mais setores defeituosos precisam ser isolados do sistema antes de iniciar o estudo.

O Algoritmo 5.2 implementa a seleção dos setores inatingíveis em uma rede, fornecendo a referida lista a partir dos itens que restarem no vetor auxiliar OFF após sua execução. Três rotinas auxiliares são utilizadas: Localiza, que procura por um valor (1° argumento) em um vetor (2° argumento) retornando a posição do elemento, quando bem sucedido, ou um valor negativo, caso contrário; Adiciona, que insere um elemento (1° argumento) no final de um vetor (2° argumento); e Remove, que exclui o elemento de uma determinada posição (1° argumento) no vetor (2° argumento). Considera-se que os vetores começam no índice 0, que seu acesso usa colchetes [], que  $A_S$  é o vetor com as arestas,  $X_F$  é o vetor com os nós fonte e que  $X_L$  é o vetor com os nós de carga:

Algoritmo 5.2 - Obtenção de setores inatingíveis

```

1   Vetor ON =  $X_F$ ;
2   Vetor OFF =  $X_L$ ;
3   inteiro ich = 0;
4   inteiro pos = -1;
5   Enquanto(ich < n(ON))
6       Para(inteiro i=0; i < n( $A_S$ ); i++)
7           pos=Localiza(R[ON[ich]][i], OFF);
8           Se(pos >= 0)
9               Adiciona(OFF[pos], ON);
10              Remove(pos, OFF);
11          ich++;

```

O segundo caso de otimização de  $R$  é bem mais comum e ocorre quando há um único não-negativo em uma linha de  $R$ , indicando que há uma única aresta  $S_i$  conectando o setor ao sistema. Neste caso, seja qual for a configuração ótima, necessariamente  $S_i$  estará habilitada. Então, propõe-se realizar uma operação de IXOR para criar um super-nó pela união dos nós adjacentes a  $S_i$ . Eventualmente, pode haver reduções desta natureza em cascata, de modo que o algoritmo de redução de setores só



e

0	0	1						
1	0	0	1	6	11	2	13	10
2	10	3	11	1				
3	9	4	10	2	12	7		
4	7	11	8	5	9	3	14	10
5	5	9	6	11	8	4		
6	1	1	2	7				
7	2	6	3	8	12	3		
8	3	7	4	9				
9	4	8	5	5				
10	13	1	14	4	15	11		
11	6	5	7	4	15	10		

Novamente, considerou-se a aresta  $S_{i+1}$  é associada a coluna  $i$ . Desse modo, o par  $\{\text{aresta}, \text{nó remoto}\} = \{10,3\}$  na 1ª posição do vetor de  $R'[2]$  significa que a aresta associada a coluna 10 ( $S_{11}$ ) tem uma extremidade incidente no setor 2 e outra no setor 3.

Como ocorreu com  $L'$  na subseção 4.6.2, o arranjo de  $R$  usando vetores simplifica o operador IXOR, que em vez de realizar a operação *ou-exclusivo*  $n(A_S)$  vezes, passa a simplesmente mesclar duas listas, evitando elementos *aresta* repetidos – notação IXOR'. Esse procedimento é semelhante ao implementado pelo método clássico de ordenação *MERGE-SORT* (CORMEN et al, 2002).

De posse de  $R'$  e de IXOR', é possível reescrever o algoritmo de decodificação de Aditiva (Algoritmo 5.3). As mesmas considerações feitas para o Algoritmo 5.1 se aplicam, mas considere adicionalmente que `atual` tem o formato dos vetores de  $R'$  e que a sintaxe para acessar a *aresta* e o *nó remoto* do  $i$ -ésimo elemento do  $j$ -ésimo vetor em  $R'$  é a seguinte:  $R'[i][j].col$  e  $R'[i][j].remoto$ , nesta ordem:

Algoritmo 5.3 - Decodificação Aditiva -  $R'$

```

1   Para (inteiro i=0; i<n(AST); i++)
2       inteiro ich_f=inteiro(g[n(AST)-i-1]*acc);
3       As[atual[ich_f].col].Fecha();
4       [atual, acc]=IXOR'(atual, R[atual[ich_f].remoto]);

```

## 5.6 Métricas do Código

Essa seção tem como objetivo descrever Aditiva com base nas métricas apresentadas no Capítulo 3.

### 5.6.1 Espaço

Utilizando notação assintótica para o espaço como fizeram Raidl & Julstrom (2003), o código Aditivo ocupa  $O(n(A_{ST}))$ . Em que, por definição:  $n(A_{ST}) = n(X_L)$ .

### 5.6.2 Tempo

A métrica tempo engloba operadores genéticos e o código propriamente dito. A complexidade dos operadores genéticos é  $O(n(A_{ST}))$ , tal qual a métrica espaço, uma vez que se propõe utilizar operadores convencionais. Este valor é válido mesmo no caso das variantes que operam na calda, pois os dados necessários podem ser obtidos sem complexidade adicional na etapa de avaliação dos indivíduos. Para a decodificação, a complexidade deve ser calculada com base no Algoritmo 5.3.

O trecho do Algoritmo 5.3 que predomina na análise de complexidade é a chamada do operador IXOR' para atualização de `atual` (Linha 4). São dois níveis de laços de repetição: no mais interno, IXOR' é executado com complexidade  $O(g_1+g_2)$ , em que:  $g_1$  e  $g_2$  são, respectivamente, os números atuais de arestas de fronteira e incidentes no nó de carga que está sendo conectado à rede; e, externamente, essas operações são repetidas  $n(A_{ST})$  vezes. Os valores  $g_1$  e  $g_2$  variam em função de  $G_S$  e do

indivíduo que está sendo decodificado, de modo que, assumindo valores médios, o algoritmo como um todo seria  $O(n(A_{ST})(g_1+g_2))$ .

O valor médio de  $g_2$  pode ser obtido de modo determinístico com base no número de arestas e nós da rede:  $2n(A_S)/n(X_S)$ . Assumindo o pior caso, em que  $g_1$  é em média igual ao número total de arestas  $n(A_S)$ , é possível reescrever expressão da complexidade, substituindo  $n(A_{ST})$  por  $n(X_L)$  – subseção 5.6.1:

$$n(X_L) \left( n(A_S) + 2 \frac{n(A_S)}{n(X_S)} \right) = n(X_L)n(A_S) \left( 1 + \frac{2}{n(X_S)} \right) \cong n(X_L)n(A_S). \quad (5.1)$$

A simplificação utilizada presume que o número de nós é grande ao ponto do erro cometido ser mínimo, assim a complexidade do código Aditivo de modo conservativo seria quadrática.

De fato, esse tema demandaria um estudo mais detalhado, principalmente para determinar com a devida propriedade como varia a fronteira no curso do Algoritmo 5.3. Contudo, julga-se que esta investigação está além do escopo do presente trabalho, ficando como uma sugestão para um trabalho futuro.

### 5.6.3 Factibilidade

A métrica factibilidade está relacionada com incorporar restrições a um código e seus operadores de modo a minimizar a representação de soluções inviáveis. Nesse sentido, o código Aditivo é similar ao código Subtrativo, sendo válidos os comentários da subseção 4.7.3.

### 5.6.4 Cobertura

A codificação Aditiva representa apenas soluções radiais conexas e compreende todas as configurações deste tipo. Os operadores genéticos, por serem convencionais ou uma variação relacionada somente ao seu alcance, permitem que operações genéticas

sucedivas gerem qualquer solução no espaço de busca. A cobertura de Aditiva é plena.

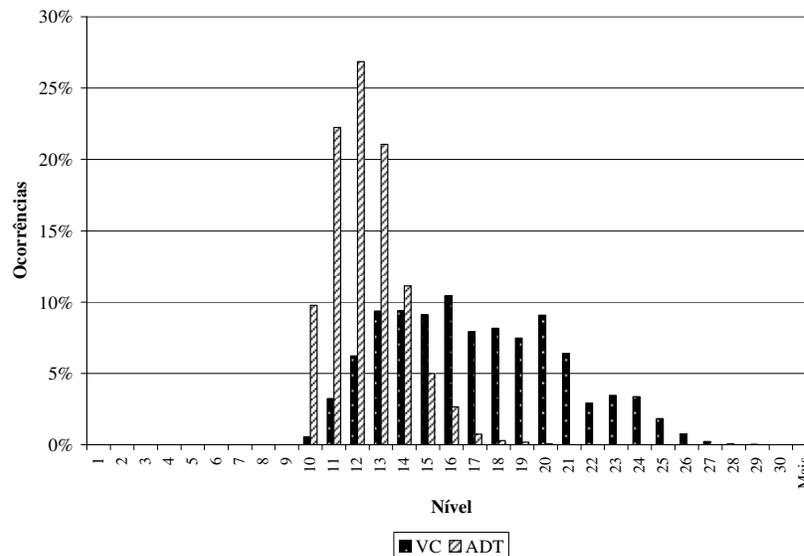
### 5.6.5 Tendência

O conceito de tendência engloba o código e seus operadores genéticos. Tendência no código implica gerar com maior frequência um tipo de rede, sendo vantajosa se, e somente se, as topologias favorecidas abarcam o ótimo global. Já tendência nos operadores têm impacto no processo de otimização propriamente dito, podendo gerar uma busca não uniforme através do espaço de busca.

O procedimento adotado nesta seção para avaliar tendências no código Aditivo e seus operadores é similar ao que foi utilizado na subseção 4.7.5. A representação Vetor Característico – VC foi utilizada como referência, bem como o sistema teste da Figura 1.2 (Capítulo 1) considerando todos os seus ramos com chave (BARAN & WU, 1989). As análises no código são realizadas primeiro e, em seguida, os operadores genéticos são avaliados: cruzamento convencional, cruzamento na calda, mutação convencional e mutação na calda.

Nos testes de tendência do código foram utilizados conjuntos de 5000 indivíduos radiais conexos gerados de forma aleatória, sendo plotadas a profundidade das árvores geradas (Figura 5.2) e a distribuição das aptidões (Figura 5.3). A frequência com que cada aresta aparece fechada na amostra foi omitida por ter se mostrado a menos efetiva dentre as métricas utilizadas na subseção 4.7.5.

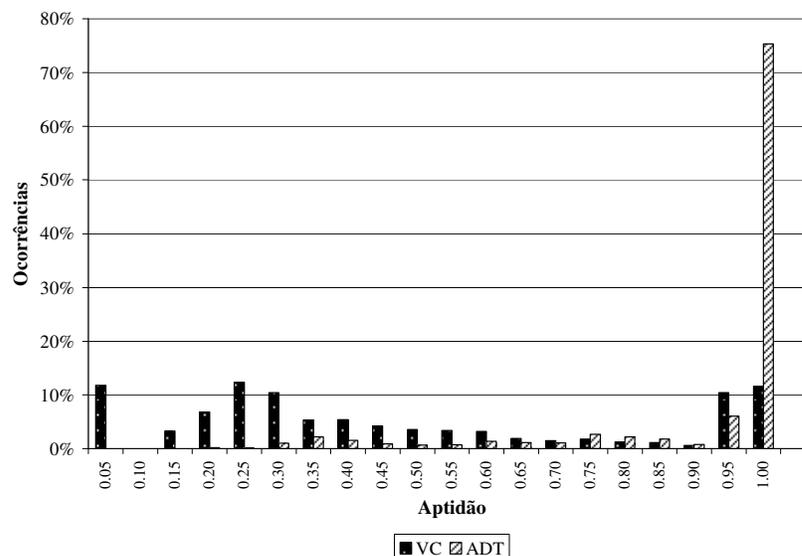
A análise da Figura 5.2 indica uma tendência significativa do código Aditivo em representar topologias com baixas profundidades. Selecionando os níveis com ocorrência superior a 5%, a codificação Aditiva se restringe ao intervalo 10-14 contra 12-21 do código Subtrativo. Isto é, Aditiva não só concentra suas soluções em uma faixa com metade da largura de Subtrativa como privilegia as topologias com no



**Figura 5.2 – Comparação entre 5000 indivíduos gerados de forma aleatória utilizando as representações Vetor Característico – VC e Aditiva – ADT: percentual de soluções por profundidade máxima.**

máximo 10, 11 e 12 níveis, as quais praticamente não ocorrem em Subtrativa. O efeito dessa tendência em termos de perdas elétricas das soluções pode ser observado na Figura 5.3: Aditiva apresenta uma forte tendência por codificar soluções com alta aptidão, obtendo uma taxa de representação na maior faixa de aptidões (0.95-1.00) de 75% contra 12% de VC. Obviamente, esses resultados são válidos apenas para o sistema testado, devendo ser tratados como indicativos de boa performance para o código Aditivo sempre que as melhores soluções apresentarem baixas profundidades.

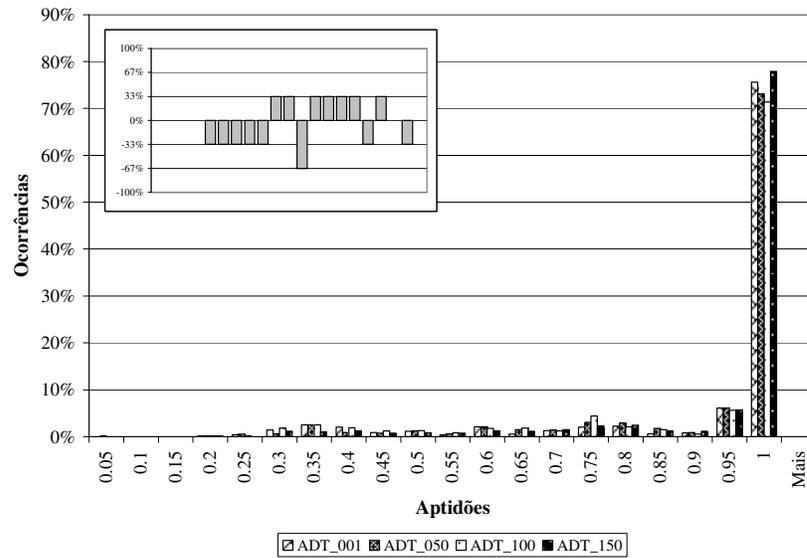
Partindo para a análise de tendência do operador de cruzamento convencional aplicado ao código Aditivo, avaliou-se o comportamento de uma população com 1000 indivíduos no curso de 150 gerações sem pressão seletiva e adotando apenas o cruzamento convencional. A evolução observada por faixas de aptidão ao término das gerações 1, 50, 100 e 150 é apresentada na Figura 5.4. No canto superior esquerdo plotou-se o índice de tendência calculado com base nos quatro valores disponíveis.



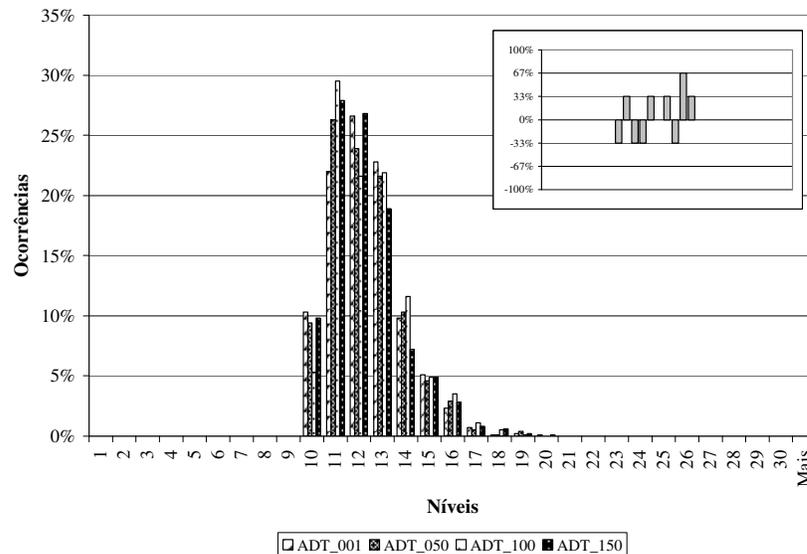
**Figura 5.3 – Comparação entre 5000 indivíduos gerados de forma aleatória utilizando as representações Vetor Característico – VC e Aditiva – ADT: percentual de soluções por faixa de aptidão.**

Embora haja cinco índices de tendência negativos consecutivos nos intervalos de aptidão menores, os dados no gráfico principal não sugerem tendência. De fato, os índices de tendência se mostram pouco efetivos quando as taxas de ocorrência são muito baixas. Essa também é a conclusão analisando a evolução nas maiores profundidades (Figura 5.5), cuja variação é aparentemente randômica. Os seguintes valores médios foram obtidos: 0.88, 0.88, 0.87 e 0.90 para aptidões e 12.31, 12.28, 12.43 e 12.22 para profundidades, confirmando as inferências feitas.

Nas Figuras 5.6 e 5.7 são apresentados os resultados obtidos ao repetir as simulações realizadas com o operador de cruzamento convencional, mas adotando cruzamento na calda. Devido a forte tendência do código Aditivo, os dados de aptidão ficam concentrados no intervalo 0.95-1.00 o que dificulta a visualização das mudanças que ocorrem nas demais faixas no curso da simulação. Tendo em vista a pouca efetividade dos índices de tendência para taxas de ocorrência baixas, as análises do cruzamento na calda são levadas a efeito comparando-se os resultados obtidos com



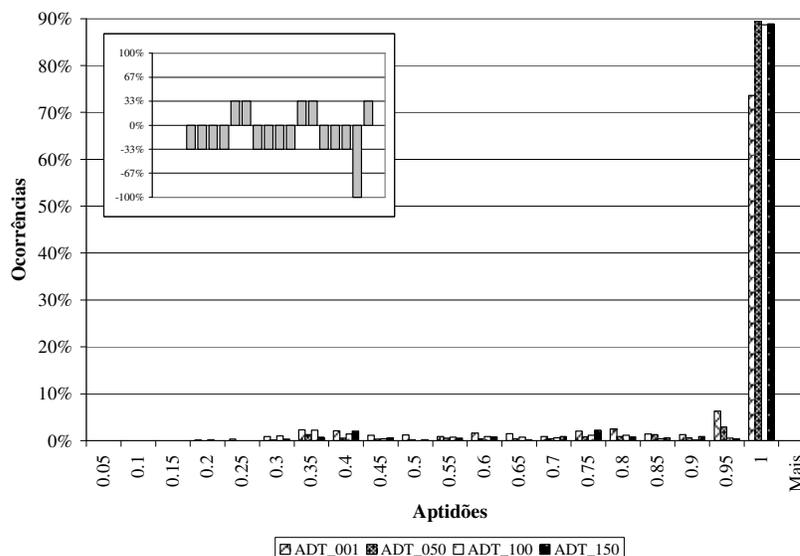
**Figura 5.4 – Evolução por faixa de aptidão para 1000 indivíduos com codificação Aditiva implementando apenas cruzamento convencional, após as gerações: 1 (ADT\_001), 50 (ADT\_050), 100 (ADT\_100) e 150 (ADT\_150). Índices de Tendência no canto superior direito.**



**Figura 5.5 – Evolução por profundidade máxima para 1000 indivíduos com codificação Aditiva adotando cruzamento convencional, após as gerações: 1 (ADT\_001), 50 (ADT\_050), 100 (ADT\_100) e 150 (ADT\_150). Índices de Tendência no canto superior esquerdo.**

aqueles apresentados nas Figuras 5.4 e 5.5, referentes aos testes com o cruzamento convencional.

Comparando as Figuras 5.6 com 5.4, observa-se redução em praticamente todas as faixas de aptidão, exceto 0.95-1.00. Sendo a evolução mais

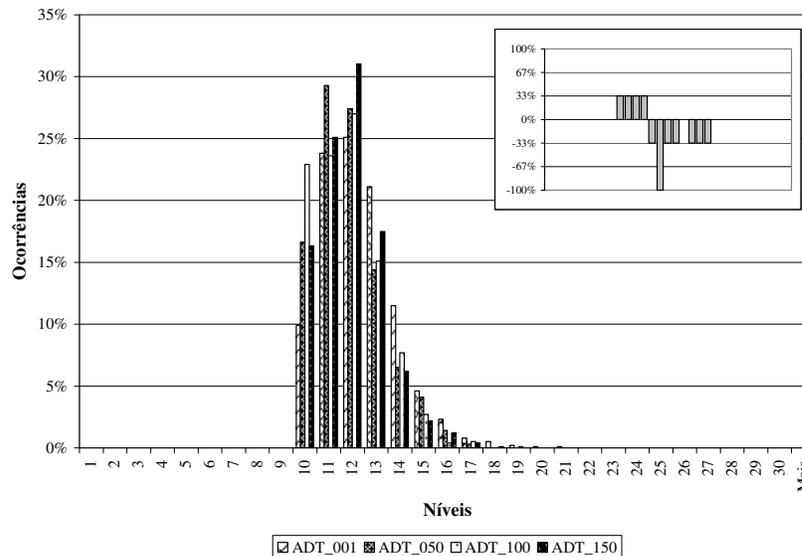


**Figura 5.6 – Evolução por faixa de aptidão para 1000 indivíduos com codificação Aditiva implementando apenas cruzamento na calda, após as gerações: 1 (ADT\_001), 50 (ADT\_050), 100 (ADT\_100) e 150 (ADT\_150). Índices de Tendência no canto superior esquerdo.**

significativa no intervalo entre 0.90-0.95, quando o índice de tendência assume seu extremo negativo: -100%. Além disso, embora ocorra um acréscimo significativo no intervalo 0.95-1.00 entre as gerações 1-50, ele não se efetiva nas transições seguintes. Não se sabe se ocasionado por algum tipo de saturação, haja visto que a taxa de ocorrência nessa faixa de aptidão atinge 89% a partir da geração 50, contra um valor máximo de 78% nos testes com o cruzamento convencional.

Os dados apresentados nas Figuras 5.7 e 5.5 permitem uma melhor visualização. Verifica-se um decréscimo consistente na participação dos níveis maiores que 13 entre as gerações 1 e 50 quando o operador na calda é utilizado. Porém, a exemplo do que ocorreu com as aptidões, esse processo não se confirma nas transições posteriores. Esse comportamento fica aparente nos dados médios observados: 0.88, 0.93, 0.91 e 0.92 para aptidões e 12.34, 11.84, 11.74 e 11.86 para profundidades.

Como conclusão, os resultados obtidos associando cruzamento na calda ao código Aditivo indicam uma tendência positiva moderada e sugerem a existência de algum tipo de saturação no processo. A forte tendência no código aparece como um

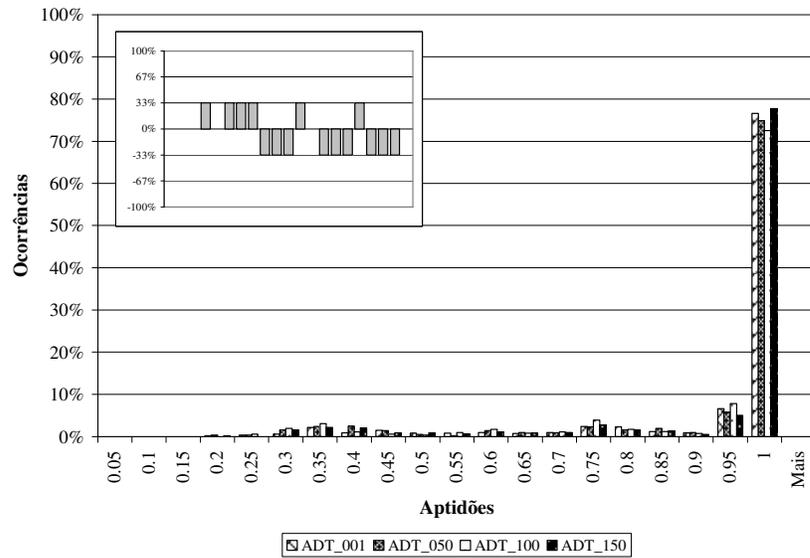


**Figura 5.7 – Evolução por profundidade máxima para 1000 indivíduos com codificação Aditiva adotando apenas cruzamento na calda, após as gerações: 1 (ADT\_001), 50 (ADT\_050), 100 (ADT\_100) e 150 (ADT\_150). Índices de Tendência no canto superior esquerdo.**

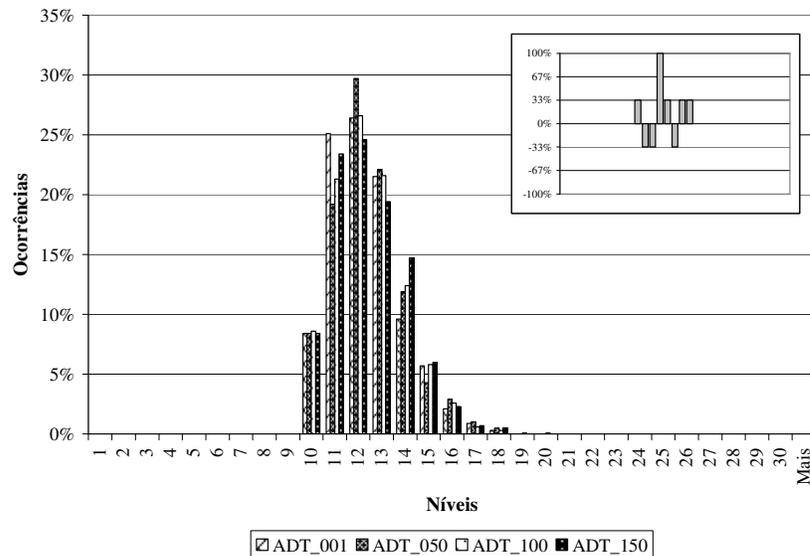
complicador para as análises, pois implica taxas de ocorrência baixas nos histogramas e, conseqüentemente, ineficácia dos índices de tendência.

Iniciando os testes com os operadores de mutação, o objetivo é estabelecer se o operador convencional ou a mutação na calda impõem algum tipo de tendência quando são aplicados ao código Aditivo. Para tanto, foram levadas a efeito simulações similares àquelas realizadas para o cruzamento: avaliou-se o comportamento de uma população com 1000 indivíduos no curso de 150 gerações sem pressão seletiva e adotando apenas operações de mutação; primeiro com o operador convencional e, em seguida, com o operador de mutação na calda.

Na Figura 5.8 apresenta-se a distribuição das aptidões quando se adota a mutação convencional, estando plotado no canto superior esquerdo seus índices de tendência. Embora fique evidente mais uma vez a forte tendência do código propriamente dito, não há indicação de tendência a favor de um conjunto específico de soluções em virtude das operações de mutação. Mesmo considerando os índices de tendência, foram verificados valores com amplitude baixa e distribuídos sem



**Figura 5.8 – Evolução por faixa de aptidão para 1000 indivíduos com codificação Aditiva implementando apenas mutação convencional, após as gerações: 1 (ADT\_001), 50 (ADT\_050), 100 (ADT\_100) e 150 (ADT\_150). Índices de Tendência no canto superior direito.**



**Figura 5.9 – Evolução por profundidade máxima para 1000 indivíduos com codificação Aditiva adotando mutação convencional, após as gerações: 1 (ADT\_001), 50 (ADT\_050), 100 (ADT\_100) e 150 (ADT\_150). Índices de Tendência no canto superior esquerdo.**

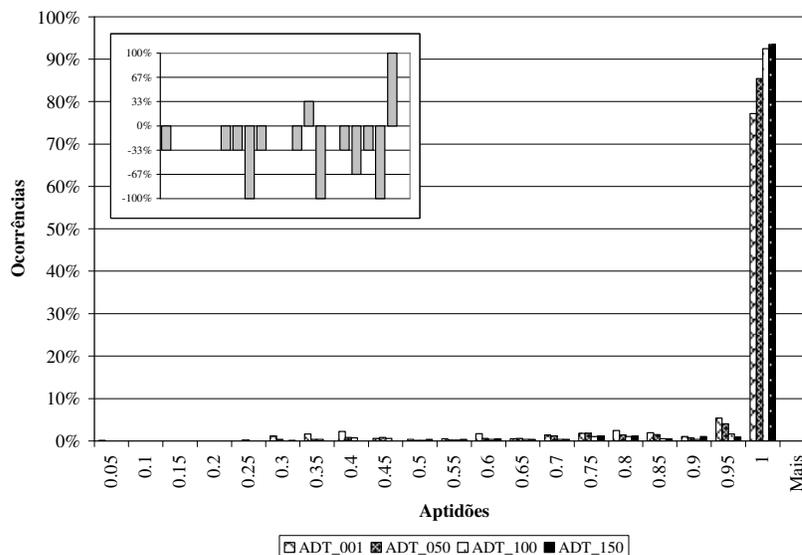
demonstrar um padrão. As aptidões médias observadas por geração foram: 0.89, 0.88, 0.89 e 0.89.

Ainda tratando dos testes com a mutação convencional, a análise da Figura 5.9 confirma as indicações do gráfico de distribuição das aptidões. A

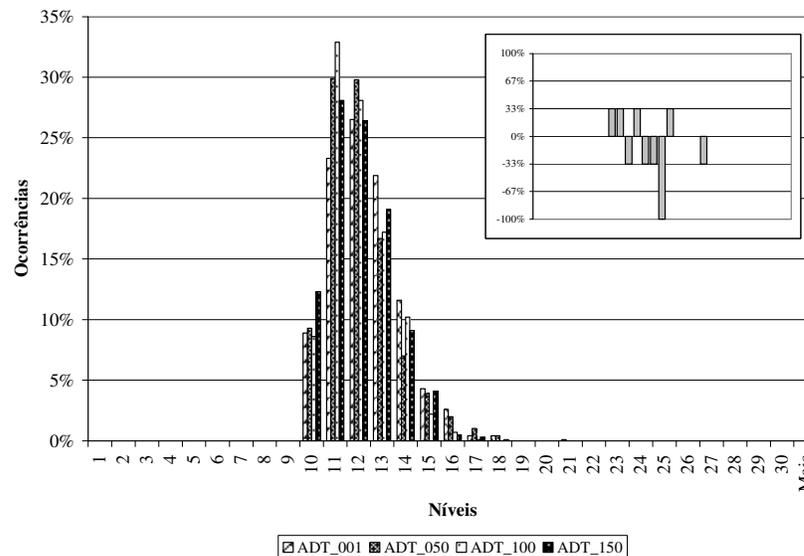
profundidade máxima das árvores codificadas parece variar de forma aleatória durante a simulação; exceção para o nível 14, que apresenta sem motivo aparente um aumento consistente: de 9.6% na amostragem inicial para 14.7% no final da simulação. As profundidades médias observadas nos quatro momentos pesquisados foram as seguintes: 12.31, 12.42, 12.42 e 12.42, apoiando a ausência de tendência.

Nas Figuras 5.10 e 5.11 são apresentados os resultados obtidos ao repetir as simulações adotando mutação na calda. Comparando as Figuras 5.10 e 5.8, fica claro um deslocamento das faixas com aptidões menores para o maior intervalo possível (0.95-1.00), que sai de um valor máximo de 77.7% para 93.5%. Também se verifica uma tendência na evolução da máxima profundidade das soluções, ocorrendo aumento das ocorrências nos níveis 10-12 e redução no intervalo 13-18.

Conclui-se que a mutação na calda impõe tendência ao código Aditivo, sendo tal efeito positivo para o caso testado. Os seguintes valores médios foram obtidos: 0.89, 0.92, 0.94 e 0.95 para aptidões e 12.33, 12.09, 11.97 e 12.01 para profundidades, confirmando as inferências feitas.



**Figura 5.10 – Evolução por faixa de aptidão para 1000 indivíduos com codificação Aditiva implementando apenas mutação na calda, após as gerações: 1 (ADT\_001), 50 (ADT\_050), 100 (ADT\_100) e 150 (ADT\_150). Índices de Tendência no canto superior direito.**

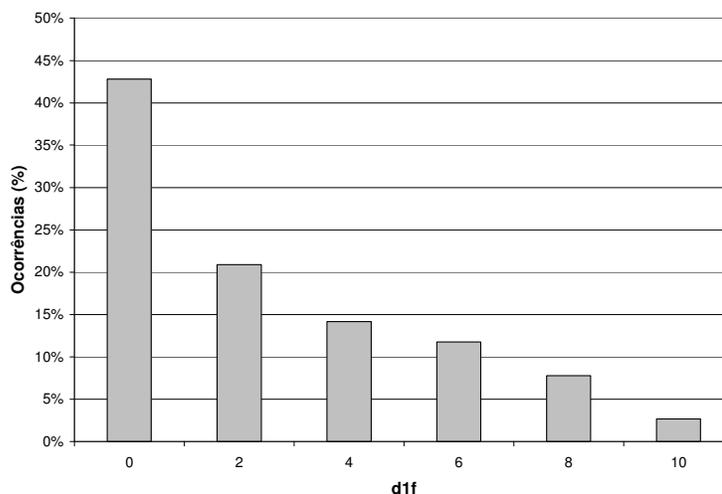


**Figura 5.11 – Evolução por profundidade máxima para 1000 indivíduos com codificação Aditiva adotando mutação na calda, após as gerações: 1 (ADT\_001), 50 (ADT\_050), 100 (ADT\_100) e 150 (ADT\_150). Índices de Tendência no canto superior esquerdo.**

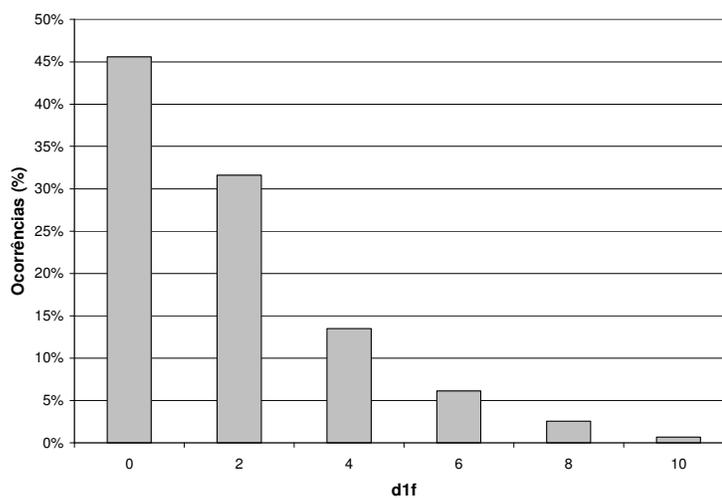
### 5.6.6 Localidade

O objetivo dessa seção é avaliar a localidade de Aditiva quando são utilizados operadores de mutação convencional e de mutação na calda, utilizando para tanto a mesma metodologia da subseção 4.7.6: implementam-se 5000 operações de mutação em indivíduos gerados de forma aleatória, sendo armazenada em cada uma dessas operações a distância entre pai e filho ( $d_{1f}$ ). Define-se *distância* entre duas configurações como o número de chaves cujo estado é diferente, sendo  $d_{1f} = 2$  a distância mínima para soluções radiais conexas distintas.

Os resultados obtidos com o operador convencional e com a mutação na calda, nesta ordem, são apresentados nas Figuras 5.12 e 5.13. Nos testes com mutação convencional (Figura 5.12), apenas 20.9% das ocorrências apresentam a localidade ideal ( $d_{1f} = 2$ ). Mesmo tolerando  $d_{1f} = 4$ , as ocorrências adequadas não chegariam a 36%. Com base neste teste, há uma forte indicação de baixa localidade para o código Aditivo quando são empregados operadores de mutação convencional. Partindo para a mutação



**Figura 5.12 – Taxas de ocorrência observadas para distâncias entre pai e filho (d1f) após simulação com 5000 mutações convencionais com código Aditivo.**



**Figura 5.13 – Taxas de ocorrência observadas para distâncias entre pai e filho (d1f) após simulação com 5000 mutações na calda com código Aditivo.**

na calda (Figura 5.13), os resultados são um pouco melhores. Aproximadamente, 45% de ocorrências aceitáveis, contudo a conclusão é a mesma: baixa localidade.

Embora os testes realizados indiquem baixa localidade, há um atenuante: a maioria das ocorrências resultou em  $d1f = 0$ , o que equivale a ausência de mutação. Se todas as ocorrências desse tipo fossem excluídas, os percentuais de adequação ( $d1f = 2$ ) para mutação convencional e na calda seriam de, aproximadamente, 36.5% e 58.1%, nesta ordem. Tais valores denotam uma condição diferente da ideal, mas tolerável desde

que, no sentido de obter uma localidade adequada para o código Aditivo, taxas de mutação significativamente maiores que o usual sejam utilizadas.

### 5.6.7 Hereditariedade

Essa seção é dedicada ao levantamento da hereditariedade do código Aditivo quando são utilizados cruzamento convencional e na calda. Algo semelhante ao que foi implementado para Subtrativa na subseção 4.7.7. Recapitulando, hereditariedade adequada implica que se dois indivíduos 1 e 2 gerarem um filho  $f$  por cruzamento, as distâncias entre 1 e  $f$  ( $d_{1f}$ ) e entre 2 e  $f$  ( $d_{2f}$ ) devem ser menores ou iguais à distância observada entre 1 e 2 ( $d_{12}$ ). Novamente, distância entre soluções é definida como número de chaves em que o estado (aberta – fechada) é diferente.

O procedimento adotado é semelhante ao da subseção 4.7.7: utiliza-se o sistema teste da Figura 1.2 supondo que todos os seus ramos têm chave; são gerados 5000 filhos por meio de operações de cruzamento em pais criados de forma aleatória usando Aditiva; e, para cada filho criado, são armazenadas as distâncias  $d_{12}$ ,  $d_{1f}$  e  $d_{2f}$ . Além disso, visando simplificar a análise, considera-se hereditariedade adequada sempre que  $d_{12}$  é maior ou igual ao maior valor entre  $d_{1f}$  e  $d_{2f}$ .

Ilustram-se nas Tabelas 5.1 e 5.2 as taxas de ocorrência verificadas nas simulações comparando-se  $d_{12}$  com o maior valor entre  $d_{1f}$  e  $d_{2f}$ . Primeiro, para o cruzamento convencional e, em seguida, para o cruzamento na calda. Os valores sombreados em ambas as tabelas destacam os casos em que observou-se distância do filho para um dos pais maior que a distância entre os pais, violando a condição ideal almejada. O total de violações na Tabela 5.1 é alto: 28.5%. Esse valor significa que a cada dez operações de cruzamento, aproximadamente três violam a regra. Resultado melhor foi obtido com o operador de cruzamento na calda, reduzindo o total de

**Tabela 5.1 – Taxas de ocorrência obtidas no estudo de hereditariedade: distância entre pais (d12) versus maior distância observada entre pais e filhos (d1f ou d2f) após operação de cruzamento convencional em indivíduos com codificação Aditiva.**

		Maior entre d1f e d2f (%)					Total geral	
		0	2	4	6	8		10
d12	0	0.02	0.00	0.02	0.00	0.00	0.00	0.04
	2	0.00	0.16	0.12	0.20	0.04	0.06	0.58
	4	0.00	0.16	1.24	1.64	1.24	0.74	5.02
	6	0.00	0.00	1.52	8.24	8.42	4.06	22.24
	8	0.00	0.00	0.60	7.62	21.56	11.94	41.72
	10	0.00	0.00	0.00	1.78	11.52	17.10	30.40
Total geral		0.02	0.32	3.50	19.48	42.78	33.90	100.00

**Tabela 5.2 – Taxas de ocorrência obtidas no estudo de hereditariedade: distância entre pais (d12) versus maior distância observada entre pais e filhos (d1f ou d2f) após operação de cruzamento na calda em indivíduos com codificação Aditiva.**

		Maior entre d1f e d2f (%)					Total geral	
		0	2	4	6	8		10
d12	0	0.04	0.00	0.00	0.00	0.00	0.02	0.06
	2	0.00	0.24	0.26	0.26	0.02	0.00	0.78
	4	0.00	0.28	2.70	2.20	0.90	0.12	6.20
	6	0.00	0.00	2.06	10.96	5.98	1.40	20.40
	8	0.00	0.00	0.44	7.22	25.60	7.28	40.54
	10	0.00	0.00	0.00	1.32	9.46	21.24	32.02
Total geral		0.04	0.52	5.46	21.96	41.96	30.06	100.00

violações para 18.4%. Porém, esse valor ainda é considerado alto.

Os resultados dos testes desta seção sugerem hereditariedade baixa ou, no melhor dos cenários, aceitável para Aditiva. A formulação de um operador de recombinação que proporcione hereditariedade alta ao código Aditivo se mostra como um melhoramento necessário para sua melhor performance, porém optou-se por sugerilo como uma pesquisa futura.

### 5.6.8 Restrição

A exemplo do que ocorreu na subseção 4.7.8 para Subtrativa, os argumentos apresentados na subseção 5.6.3 são válidos aqui: o código Aditivo é semelhante às melhores técnicas descritas no Capítulo 2, sendo capaz de representar somente configurações de rede radiais conexas.

### **5.6.9 Híbridização**

Este trabalho não inclui qualquer tentativa no sentido de embutir nos operadores genéticos heurísticas associadas à Configuração, podendo esse ser um dos desdobramentos para pesquisas futuras. Contudo, propõe-se uma modalidade diferente no Capítulo 6: *hibridização no código*.

## Capítulo 6

# Algoritmo Genético Sequencial

---

Este capítulo foi dividido em três seções: na primeira, apresentam-se motivação e fundamentos que suportam o Algoritmo Genético Sequencial (AGS); em seguida, o algoritmo é formulado, discutindo suas repercussões em termos de código, alternativas de controle do espaço de busca e os critérios heurísticos propostos; e, na terceira seção, são realizadas algumas simulações com objetivo de evidenciar as vantagens do AGS em relação à métrica tendência – um tipo de hibridização no código.

### 6.1 Motivação e fundamentos

As representações Subtrativa e Aditiva descritas nos Capítulos 4 e 5 possuem algumas semelhanças e, mesmo nas diferenças, possuem uma relação de dualidade, como evidenciado na Tabela 6.1. Considerando o processo de decodificação de um indivíduo, os códigos têm em comum que o gene é sempre utilizado para escolher um elemento em um conjunto de candidatos. A escolha é por uma chave para ser aberta no código Subtrativo e por uma chave para ser fechada no código Aditivo, mas a constante é que existem alternativas: 1º, 2º, 3º, e assim por diante.

Considerando que, durante a interpretação dos genes nos Algoritmos 4.3 (subseção 4.6.2) e 5.3 (subseção 5.5.2), há em média  $n$  alternativas, a restrição do conjunto de opções às primeiras  $n/10$  implica reduzir o espaço de busca das soluções a

Tabela 6.1 – Comparativo das Codificações Sequenciais Subtrativa e Aditiva.

Módulo	Aspecto	Codificação	
		Subtrativa	Aditiva
Pré-processamento	Condição da rede	Arestas habilitadas	Arestas desabilitadas
	Variáveis	Mapa das Malhas e $\alpha$	Mapa dos Setores e acc
Codificação	Tipo	Codificação por Arestas	Codificação por Arestas
	Alfabeto	[0, 1[	[0, 1[
	Número de genes	$n(A_{NA})$	$n(A_{ST})$
	Espaço de Busca	Radiais conexas	Radiais conexas
	Tipo	Sequencial	Sequencial
Decodificação	Operador auxiliar	XOR	IXOR
	Rotina requerida	Abre	Fecha
Operadores Genéticos	Tipo	Convencionais e Calda	Convencionais e Calda
Melhoramentos	Variáveis	Mapa das Malhas Vetorial	Mapa dos Setores Vetorial
	Operador auxiliar	XOR'	IXOR'

10% do tamanho original. Isto é, 10% das soluções radiais conexas existentes. Certamente, essa redução torna a busca realizada pelo AG mais fácil. Entretanto, caso a solução ótima global não esteja nos 10% representados, ela seria inócua. Para que a redução no espaço de busca se constitua em uma vantagem, é preciso aumentar a probabilidade da solução ótima global ser mantida. Essa é a oportunidade que serve como fundamento para o algoritmo proposto nesse capítulo.

O AGS combina redução no espaço de busca das soluções a uma heurística que ordena as alternativas de modo que as mais desejáveis ocupem as primeiras posições e as menos desejáveis as últimas. O efeito da heurística é aumentar a chance da solução ótima global ser representada, mesmo quando se utiliza uma fração do espaço de busca original. O problema é que não se conhece uma heurística ideal, tampouco outra quase ideal que, para um espaço de busca com  $N\%$  das configurações

radiais conexas, garanta a representação do ótimo global para qualquer rede de distribuição. A solução adotada para conviver com as imperfeições dos critérios heurísticos foi tornar o parâmetro  $N$  variável, sendo de responsabilidade do AGS o seu controle. Define-se  $N$  como *Fator de Busca* do AGS.

## 6.2 Formulação do Algoritmo

O fluxograma do algoritmo proposto é apresentado na Figura 6.1, sendo destacadas em cinza as diferenças em relação ao AG Básico da Seção 3.4. A alteração mais significativa se refere a adição do módulo Controle do Fator de Busca, o qual é descrito na subseção 6.2.1. Outra mudança ocorre no módulo de Avaliação, que passa a enviar aos algoritmos de decodificação o  $N$  atual. O modo como os algoritmos de

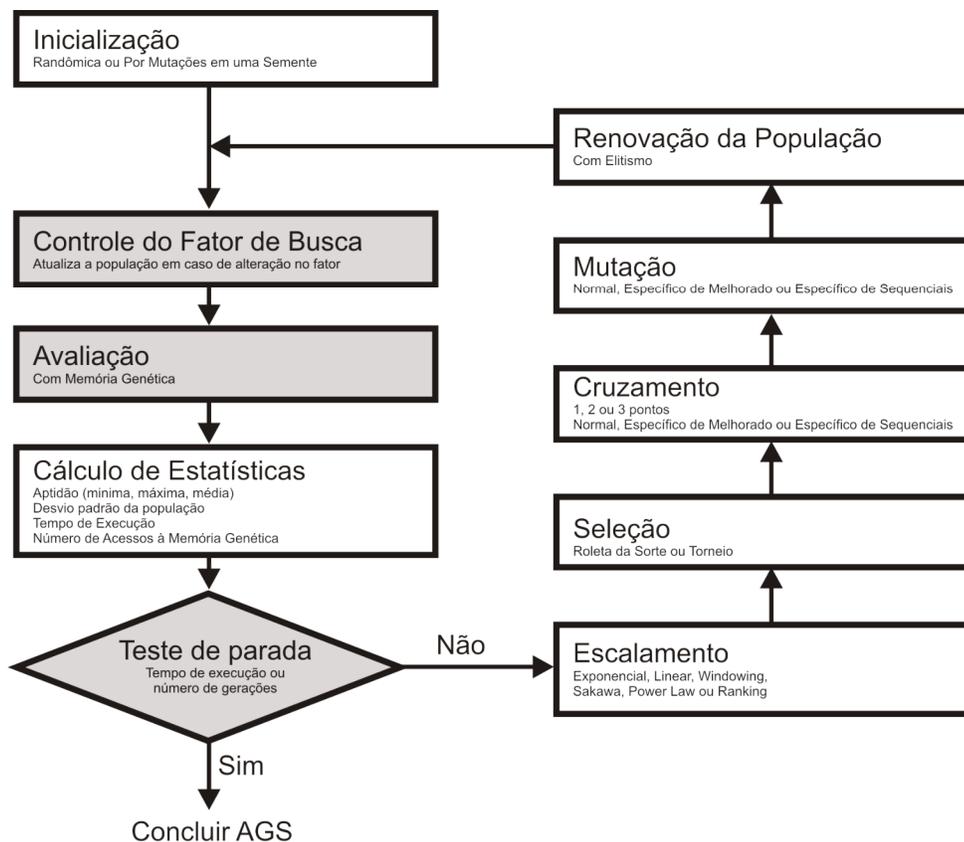


Figura 6.1 – Fluxograma do AGS implementado.

decodificação de Subtrativa e Aditiva utilizam este fator é o assunto da subseção 6.2.2. Por fim, a modificação implementada no Teste de Parada consiste apenas em retirar da lista de critérios a alternativa desvio padrão, uma vez que sua natureza estocástica é incompatível com as lógicas de controle de  $N$  apresentadas na subseção 6.2.1. Complementando a formulação do AGS, são descritas na subseção 6.2.3 as heurísticas selecionadas para os testes com o AGS no Capítulo 7.

### 6.2.1 Controle do Fator de Busca $N$

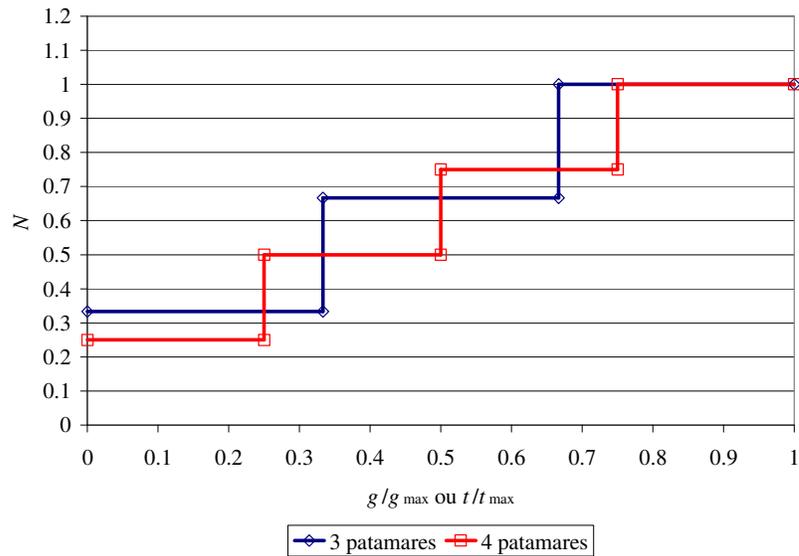
A idéia é controlar  $N$  de modo que ele assuma durante a execução do AGS  $n_{pat}$  patamares crescentes, começando em  $N_0$  e terminando em  $N_{final}$ . Se o critério de parada é o número de gerações  $g_{max}$ , cada patamar se aplica durante um certo número de gerações. Isto é, a duração do patamar é medida como uma fração de  $g_{max}$ . A situação é semelhante no caso de parada por um tempo máximo de execução  $t_{max}$ , sendo os patamares válidos por frações de  $t_{max}$ . Desse modo, qualquer estratégia de controle pode ser representada de forma genérica por um gráfico de  $N$  versus uma duração em pu (por unidade). Sendo  $g_{max}$  ou  $t_{max}$  a referência dependendo do critério de parada adotado. A única restrição é que tal referência seja conhecida no início da execução do algoritmo, motivo pelo qual o critério de parada pelo desvio padrão não pode ser utilizado.

Propõe-se duas estratégias de controle de  $N$ : *Larguras Iguais* e *Larguras Proporcionais*. No controle por larguras iguais, um patamar  $i$  ( $i = 1, 2, \dots, n_{pat}$ ) tem duração  $1/n_{pat}$ , enquanto no controle por larguras proporcionais sua duração é obtida a partir da expressão:  $2i/(n_{pat}(n_{pat}+1))$ . O fator de busca efetivamente adotado no patamar  $i$  é calculado nos dois casos pela mesma expressão:

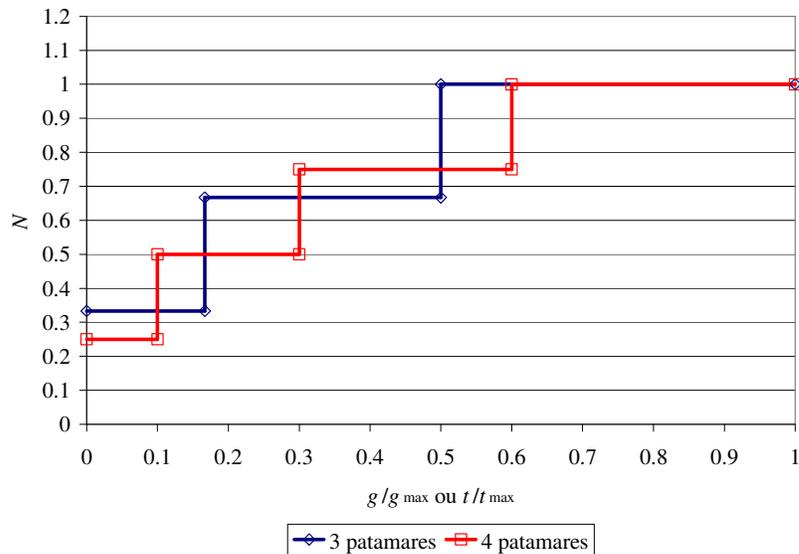
$$N = N_0 + (i - 1) \left( \frac{N_{final} - N_0}{n_{pat} - 1} \right). \quad (6.1)$$

Para exemplificar, considere o exemplo da Figura 6.2. Nele são apresentadas as curvas geradas pelos controles por larguras iguais (Figura 6.2a) e larguras proporcionais (Figura 6.2b) quando  $n_{\text{pat}} = 3$  e  $n_{\text{pat}} = 4$ . Nas duas condições adotou-se  $N_{\text{final}} = 1.00$ , sendo  $N_0 = 0.33$  para  $n_{\text{pat}} = 3$  e  $N_0 = 0.25$  para  $n_{\text{pat}} = 4$ .

Como tudo tem um custo, sempre que o módulo de controle de  $N$  comanda uma mudança de patamar, se faz necessário atualizar a população. Essa



(a)



(b)

**Figura 6.2 – Exemplo das estratégias de controle de  $N$  para 3 e 4 patamares: a) Larguras Iguais e b) Larguras Proporcionais.**

operação é levada a efeito pela multiplicação de todos os genes pela razão entre o fator praticado até aquele momento pelo novo. Essa correção nos genes é detalhada na subseção 6.2.2, juntamente com os códigos de decodificação propriamente ditos. Por hora, observe apenas que o custo computacional dessa atualização restringe os valores viáveis de  $n_{pat}$ . Recomenda-se mantê-lo no intervalo de 2 até 5. Essa discussão é retomada nos testes do Capítulo 7.

### 6.2.2 Repercussão nos algoritmos de decodificação

Uma única alteração precisa ser feita em cada um dos algoritmos de decodificação, precisamente na linha 2 de ambos. Considere que  $N$  é o fator de busca em percentagem fornecido pelo AGS na chamada dessas rotinas (as alterações aparecem sombreadas):

#### Algoritmo 4.3 - Decodificação Subtrativa - $L'$

```

1   Para (inteiro  $i=0; i < n(A_{NA}); i++$ )
2   inteiro  $ich_m = \text{inteiro}((N/100) * g[n(A_{NA}) - i - 1] * \alpha[i]);$ 
3    $A_s[L'[i][ich_m].col].Abre();$ 
4   Par  $[vet, pos] = L'[i][ich_m].vetpos;$ 
5   Enquanto ( $vet > 0$ )
6       Par  $[vet\_prox, pos\_prox] = L'[vet][pos].vetpos;$ 
7        $L'[vet] = XOR'(L'[i], L'[vet]);$ 
8        $[vet, pos] = [vet\_prox, pos\_prox];$ 

```

#### Algoritmo 5.3 - Decodificação Aditiva - $R'$

```

1   Para (inteiro  $i=0; i < n(A_{ST}); i++$ )
2   inteiro  $ich_f = \text{inteiro}((N/100) * g[n(A_{ST}) - i - 1] * acc);$ 
3    $A_s[atual[ich_f].col].Fecha();$ 
4    $[atual, acc] = IXOR'(atual, R[atual[ich_f].remoto]);$ 

```

Com essa alteração, o intervalo possível para  $ich_m$  e  $ich_f$  passa a ser controlado por  $N$ : se  $N=0$ , uma única configuração pode ser representada, pois  $ich_m$  e  $ich_f$  são sempre nulos; se  $N=100$ , os algoritmos voltam a sua formulação anterior, em que todas as configurações radiais conexas são codificadas; e, para qualquer valor neste

intervalo, o AGS estará implementando uma busca em uma fração das configurações radiais conexas.

Retomando a descrição iniciada na subseção 6.2.1, sobre a atualização da população que deve ocorrer no AGS toda vez que  $N$  é atualizado, considere a decodificação de um gene qualquer cujo alelo é  $g$  quando o fator de busca é  $N$ . Tomando a linha 2 do Algoritmo 4.3, o valor de  $i_{ch\_m}$  é obtido pelo truncamento da seguinte expressão:  $(N/100) * g * \alpha[i]$ . Quando o controle do fator de busca comanda a atualização de  $N$  para um novo valor  $N'$ , é importante que o resultado desta expressão seja mantido, garantindo que o fenótipo anteriormente codificado não seja modificado. Assim, a seguinte igualdade deve ser satisfeita:

$$(N/100) * g * \alpha[i] = (N'/100) * g' * \alpha[i].$$

Em que:  $g'$  é o novo alelo e  $\alpha[i]$  é o mesmo anterior, uma vez que depende das malhas da rede e nenhuma delas foi alterada. Manipulando a igualdade já sem  $\alpha[i]$ :

$$g' = g \frac{N}{N'}. \quad (6.2)$$

A equação (6.2) também é válida para Aditiva, pois o equacionamento é similar. Resumindo, o procedimento de atualização consiste em alterar os genes para garantir que seus respectivos fenótipos sejam mantidos. Observe que pela estratégia de controle formulada em (6.1),  $N$  varia sempre de modo crescente. Assim, (6.2) age sempre no sentido de diminuir o alelo e, conseqüentemente, não há risco de violação do limite dos genes.

### 6.2.3 Heurísticas

Como descrito na Seção 6.1, o AGS requer uma heurística que aumente a chance de manter a solução ótima global quando um espaço de busca reduzido é utilizado. Essa melhoria ocorre por meio da ordenação das arestas, tal que a melhor alternativa seja

associada ao alelo 0, a segunda melhor ao alelo 1, e assim por diante. Nesse sentido, há duas formulações possíveis: heurísticas de pré-processamento e heurísticas de execução.

As heurísticas de pré-processamento são as mais simples e com menor custo computacional, uma vez que implementam uma única ordenação das arestas antes do AGS ser executado. As heurísticas de execução por sua vez tendem a ser mais complexas computacionalmente e também mais efetivas, já que implementam ao menos uma ordenação durante a execução do algoritmo; no caso limite, podem demandar uma ordenação por gene interpretado.

Neste trabalho, optou-se por adotar apenas heurísticas de pré-processamento, sendo testadas quatro ordenações: por níveis crescente (N+), por níveis decrescente (N-), por corrente crescente (I+) e por corrente decrescente (I-). Em que, níveis e correntes são determinados considerando a rede malhada obtida fechando todas as chaves. O nível de uma aresta é calculado como a média dos níveis de seus nós adjacentes, sendo o nível de um nó definido por uma busca em largura na rede malhada obtida fechando todas as chaves. O procedimento é ilustrado na Figura 6.3 para o sistema exemplo da Figura 3.3, sendo os níveis dos nós escritos entre colchetes [] e das arestas entre chaves {}. As ordenações baseadas em correntes se referem às correntes calculadas para cada aresta por um algoritmo de fluxo de carga matricial (CEPEL, 2010), sendo considerados apenas seus respectivos módulos.

As heurísticas N+ e N- representam os extremos possíveis da idéia de que as redes de distribuição com menores perdas tendem a ter profundidades baixas. Trata-se de uma formulação empírica, que pode ser vista como uma recomendação para fechar as chaves com níveis baixos e abrir as demais. Nesse sentido, espera-se que N+ apresente melhor desempenho quando combinada ao código Aditivo, pois sua tendência é por expor chaves com os menores níveis ao fechamento. De modo oposto, espera-se

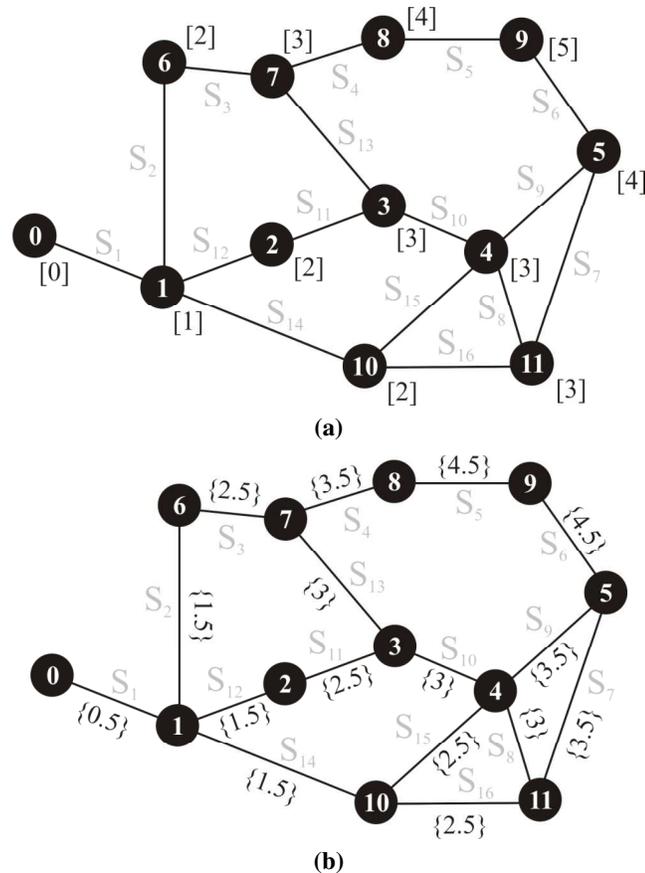


Figura 6.3 – Cálculo dos níveis para rede da Figura 3.3 considerando todas as chaves fechadas: (a) níveis dos nós e (b) níveis das arestas.

desempenho superior de Subtrativa quando  $N^-$  é utilizada. Nesse caso, a ordenação opera expondo mais as chaves com níveis maiores, justamente aquelas que a heurística recomenda abrir.

As ordenações  $I^+$  e  $I^-$ , por sua vez, são inspiradas nos numerosos algoritmos da literatura técnica que se apóiam na abertura sequencial de chaves (MERLIN & BACK, 1975). O conceito adotado é que nas configurações com perdas mínimas as chaves abertas são geralmente aquelas em que, caso fosse calculado o fluxo de carga com a rede malhada, a corrente observada seria mínima. Nesse caso, a leitura é que Aditiva leva vantagem se puder fechar chaves por onde passam correntes altas. Logo, deve apresentar melhor desempenho com  $I^-$ . Em oposição, Subtrativa deve ter melhor desempenho com  $I^+$ .

Embora sejam propostas apenas quatro ordenações, baseadas essencialmente em dois critérios heurísticos, não há qualquer restrição do AGS ao uso de outras técnicas. O objetivo é demonstrar que as ordenações combinadas com  $N$  impactam nos espaços de busca das soluções e, portanto, podem influenciar positivamente na performance do AGS. A formulação, validação e testes de outras heurísticas de ordenação parece indicar um campo promissor para pesquisas futuras. A mesma análise pode ser feita em relação às estratégias de controle de  $N$  descritas na subseção 6.2.1.

### 6.3 Métricas do Algoritmo

Como discutido nas Seções 6.1 e 6.2, a adoção do fator de busca  $N$  com intuito de reduzir o espaço de busca das soluções configura uma vantagem para o AGS, se e somente se, o critério heurístico selecionado para ordenação das arestas é capaz de provocar uma tendência no código, no sentido de representar a solução ótima global quando o espaço de busca está reduzido ( $N < 100\%$ ). Sendo assim, o objetivo dessa seção é avaliar as heurísticas propostas na subseção 6.2.3 quanto a sua habilidade de provocar tal tendência positiva nos códigos Subtrativo e Aditivo.

A idéia é implementar testes semelhantes aos que foram conduzidos nas Seções 4.7.5 e 5.6.5 visando analisar tendências nos códigos, mas adotando as quatro heurísticas apresentadas na subseção 6.2.3 mais uma ordem aleatória de referência RD. Três valores para  $N$  são utilizados: 33.3%, 66.6% e 100%. Em cada teste são gerados 5000 indivíduos de forma aleatória, sendo avaliadas as distribuições de aptidões e profundidades máximas. Utiliza-se o sistema teste da Figura 1.2, supondo que todos os seus ramos têm chave.

As distribuições observadas para as aptidões quando o código Subtrativo é utilizado e considerando  $N$  igual a 33.3%, 66.6% e 100%, nesta ordem, são apresentadas na Tabela 6.2. Em que, o intervalo de aptidão com maior taxa de ocorrências por coluna consta sombreado, destacando a tendência verificada no código. O mesmo formato é utilizado na Tabela 6.3 para apresentar os resultados obtidos com o código Aditivo.

Antes de analisar as heurísticas propriamente ditas, considere os resultados obtidos quando  $N=100%$ , nas Tabelas 6.2 e 6.3. Observe que, seja qual for a técnica de ordenação adotada, as taxas de ocorrência em todas as faixas de aptidão são semelhantes entre si e em relação aos perfis levantados nas Figuras 4.5 e 5.3, respectivamente, para os códigos Subtrativo e Aditivo. Esses resultados demonstram que as técnicas de ordenação de arestas sozinhas ( $N=100%$ ) são incapazes de modificar as características de tendência inerentes aos códigos sequenciais.

Iniciando as análises das heurísticas quando combinadas ao código Subtrativo (Tabela 6.2), observe pelos valores sombreados que RD, N+ e I- concentram

**Tabela 6.2 – Taxas de ocorrência por faixa de aptidão para um conjunto de 5000 indivíduos gerados aleatoriamente com código Subtrativo, fatores de busca  $N$  de 33.3%, 66.6% e 100%, adotando as seguintes heurísticas de ordenação das arestas: Randômica (RD), Níveis crescente (N+), Níveis decrescente (N-), Correntes crescente (I+) e Correntes decrescente (I-).**

	$N=33.3\%$					$N=66.6\%$					$N=100\%$					
	RD	N+	N-	I+	I-	RD	N+	N-	I+	I-	RD	N+	N-	I+	I-	
	0.05	0.17	0.42	0.00	0.00	0.56	0.17	0.25	0.00	0.00	0.25	0.18	0.18	0.19	0.17	0.19
	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	0.15	0.02	0.11	0.00	0.00	0.06	0.03	0.06	0.00	0.00	0.06	0.03	0.04	0.03	0.03	0.04
	0.20	0.17	0.14	0.00	0.00	0.18	0.07	0.09	0.00	0.00	0.12	0.06	0.07	0.07	0.07	0.07
	0.25	0.36	0.22	0.00	0.00	0.18	0.16	0.26	0.02	0.03	0.28	0.17	0.17	0.16	0.16	0.17
	0.30	0.05	0.07	0.00	0.00	0.01	0.11	0.17	0.11	0.09	0.13	0.14	0.13	0.13	0.14	0.13
	0.35	0.06	0.04	0.00	0.00	0.00	0.07	0.05	0.08	0.05	0.03	0.05	0.05	0.06	0.06	0.06
	0.40	0.02	0.00	0.00	0.02	0.00	0.07	0.04	0.07	0.06	0.03	0.05	0.05	0.05	0.06	0.05
	0.45	0.02	0.00	0.00	0.00	0.00	0.05	0.03	0.05	0.06	0.02	0.04	0.04	0.04	0.04	0.04
	0.50	0.01	0.00	0.00	0.00	0.00	0.04	0.02	0.04	0.07	0.02	0.03	0.03	0.03	0.03	0.03
Aptidão	0.55	0.02	0.00	0.00	0.01	0.00	0.03	0.01	0.05	0.05	0.01	0.02	0.03	0.02	0.03	0.03
	0.60	0.01	0.00	0.00	0.01	0.00	0.03	0.01	0.04	0.05	0.01	0.02	0.02	0.02	0.02	0.02
	0.65	0.00	0.00	0.00	0.01	0.00	0.02	0.00	0.02	0.03	0.01	0.01	0.01	0.01	0.01	0.01
	0.70	0.01	0.00	0.00	0.01	0.00	0.01	0.00	0.03	0.03	0.00	0.01	0.01	0.01	0.01	0.01
	0.75	0.00	0.00	0.02	0.01	0.00	0.01	0.00	0.03	0.04	0.00	0.01	0.01	0.01	0.01	0.01
	0.80	0.00	0.00	0.02	0.01	0.00	0.01	0.00	0.03	0.03	0.00	0.01	0.01	0.01	0.01	0.01
	0.85	0.01	0.00	0.00	0.01	0.00	0.01	0.00	0.02	0.02	0.00	0.01	0.01	0.01	0.01	0.01
	0.90	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.01	0.00	0.01	0.00	0.00	0.01	0.01
	0.95	0.02	0.00	0.01	0.03	0.00	0.08	0.00	0.11	0.13	0.01	0.08	0.07	0.07	0.07	0.07
	1.00	0.04	0.00	0.95	0.86	0.00	0.04	0.00	0.27	0.24	0.00	0.07	0.06	0.07	0.06	0.06
Mais	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Máximo	0.36	0.42	0.95	0.86	0.56	0.17	0.26	0.27	0.24	0.28	0.18	0.18	0.19	0.17	0.19	

**Tabela 6.3 – Taxas de ocorrência por faixa de aptidão para um conjunto de 5000 indivíduos gerados aleatoriamente com código Aditivo, fatores de busca  $N$  de 33.3%, 66.6% e 100%, adotando as seguintes heurísticas de ordenação das arestas: Randômica (RD), Níveis crescente (N+), Níveis decrescente (N-), Correntes crescente (I+) e Correntes decrescente (I-).**

	$N=33.3\%$					$N=66.6\%$					$N=100\%$				
	RD	N+	N-	I+	I-	RD	N+	N-	I+	I-	RD	N+	N-	I+	I-
0.05	0.00	0.00	1.00	0.63	0.00	0.00	0.00	0.16	0.17	0.00	0.00	0.00	0.00	0.00	0.00
0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.15	0.01	0.00	0.00	0.02	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00
0.20	0.00	0.00	0.00	0.35	0.00	0.01	0.00	0.27	0.30	0.00	0.00	0.00	0.00	0.00	0.00
0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.13	0.15	0.00	0.00	0.00	0.00	0.00	0.00
0.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.10	0.00	0.01	0.01	0.01	0.01	0.01
0.35	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.02	0.03	0.00	0.03	0.02	0.02	0.02	0.02
0.40	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.02	0.02	0.00	0.02	0.02	0.02	0.02	0.02
0.45	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.02	0.04	0.00	0.01	0.01	0.01	0.01	0.01
0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.02	0.00	0.01	0.01	0.01	0.01	0.01
0.55	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.03	0.00	0.01	0.01	0.01	0.01	0.01
0.60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.02	0.00	0.02	0.02	0.02	0.01	0.02
0.65	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.03	0.02	0.00	0.01	0.01	0.01	0.01	0.01
0.70	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.01	0.00	0.01	0.01	0.01	0.01	0.01
0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.02	0.00	0.02	0.02	0.03	0.03	0.02
0.80	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.02	0.01	0.00	0.02	0.02	0.02	0.02	0.03
0.85	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.02	0.01	0.00	0.02	0.02	0.01	0.02	0.01
0.90	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.01	0.01
0.95	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.14	0.04	0.01	0.05	0.06	0.06	0.06	0.06
1.00	0.99	1.00	0.00	0.00	1.00	0.87	0.99	0.00	0.00	0.98	0.75	0.76	0.75	0.74	0.75
Mais	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Máximo	0.99	1.00	1.00	0.63	1.00	0.87	0.99	0.27	0.30	0.98	0.75	0.76	0.75	0.74	0.75

suas ocorrências nos intervalos com aptidões baixas, seja qual for o  $N$  utilizado. Esse resultado indica que estas ordenações não são efetivas quando o código Subtrativo é utilizado. O desempenho de N- e I+ é diferente. Elas demonstram uma forte concentração no maior intervalo de aptidão possível quando  $N=33.3\%$ , demonstrando efetividade quando associadas à Subtrativa. Porém, esse efeito aparece atenuado para  $N=66.6\%$ , começando a prevalecer a tendência inerente ao código Subtrativo.

Restringindo a análise às heurísticas que demonstraram efetividade, N- aparenta ser melhor que I+ para o sistema testado, pois obteve concentrações maiores na maior faixa de aptidão (0.95-1.00): primeiro 0.95 contra 0.86 e, em seguida, 0.27 contra 0.24. Contudo, somente testes com outros sistemas permitiriam generalizar tais conclusões.

Os resultados do código Aditivo (Tabela 6.3) confirmam a relação de dualidade com a representação Subtrativa. Verifica-se um desempenho adequado quando são utilizadas as ordenações RD, N+ e I-, enquanto N- e I+ se mostram inadequadas. Embora RD demonstre uma tendência positiva, seu desempenho é inferior

ao obtido adotando os critérios heurísticos N+ e I-. Esse comportamento de RD sugere que ordenações aleatórias implicam sempre resultados intermediários, entre N+ e N- ou entre I+ e I-. Em termos de concentração das soluções nas melhores faixas de aptidão, a técnica que obteve os melhores resultados quando associada com Aditiva foi N+.

Levando em consideração que as análises com base em aptidões dependem da função utilizada, que nesse caso são as perdas ativas, apresentam-se nas Tabelas 6.4 e 6.5 os mesmos resultados segundo a métrica profundidade máxima. As tabelas se referem aos códigos Subtrativo e Aditivo quando adotam-se  $N=33.3%$ ,  $N=66.6%$  e  $N=100%$ , nesta ordem.

Assumindo a distribuição quando  $N=100%$  na Tabela 6.4 como inerente ao código Subtrativo, observe que  $N$  opera como uma perturbação em torno da profundidade 20. Quando o fator é menor ( $N=33.3%$ ), as ordenações N- e I+ concentram suas representações na profundidade 12, N+ em 21 e as demais em 20; para  $N=66.6%$ , N- e I+ têm seus máximos de agrupamento, respectivamente, nas

**Tabela 6.4 – Taxas de ocorrência por profundidade máxima para um conjunto de 5000 indivíduos gerados aleatoriamente com código Subtrativo, fatores de busca N de 33.3%, 66.6% e 100%, adotando as seguintes heurísticas de ordenação das arestas: Randômica (RD), Níveis crescente (N+), Níveis decrescente (N-), Correntes crescente (I+) e Correntes decrescente (I-).**

	N=33.3%					N=66.6%					N=100%				
	RD	N+	N-	I+	I-	RD	N+	N-	I+	I-	RD	N+	N-	I+	I-
10	0.00	0.00	0.11	0.07	0.00	0.00	0.00	0.02	0.01	0.00	0.00	0.00	0.00	0.00	0.00
11	0.03	0.00	0.26	0.16	0.00	0.01	0.00	0.06	0.04	0.00	0.01	0.02	0.02	0.02	0.02
12	0.02	0.00	0.30	0.29	0.00	0.03	0.00	0.10	0.09	0.00	0.04	0.04	0.04	0.04	0.04
13	0.00	0.00	0.24	0.12	0.00	0.04	0.01	0.13	0.12	0.02	0.06	0.05	0.06	0.06	0.05
14	0.10	0.02	0.06	0.12	0.02	0.10	0.01	0.11	0.10	0.02	0.06	0.06	0.06	0.06	0.06
15	0.00	0.01	0.02	0.08	0.01	0.07	0.02	0.09	0.11	0.03	0.06	0.06	0.07	0.06	0.06
16	0.05	0.03	0.01	0.07	0.01	0.09	0.05	0.12	0.12	0.06	0.08	0.08	0.08	0.08	0.08
17	0.05	0.00	0.00	0.02	0.03	0.05	0.04	0.09	0.10	0.06	0.07	0.07	0.07	0.07	0.08
18	0.11	0.03	0.00	0.02	0.11	0.09	0.09	0.09	0.08	0.10	0.09	0.10	0.09	0.09	0.08
19	0.16	0.09	0.00	0.02	0.12	0.12	0.14	0.08	0.08	0.12	0.11	0.11	0.10	0.10	0.11
20	0.24	0.13	0.00	0.01	0.16	0.16	0.16	0.05	0.07	0.19	0.13	0.12	0.13	0.14	0.13
21	0.16	0.17	0.00	0.01	0.15	0.08	0.14	0.03	0.04	0.11	0.10	0.11	0.11	0.10	0.11
22	0.05	0.08	0.00	0.00	0.12	0.05	0.08	0.01	0.02	0.05	0.05	0.05	0.05	0.06	0.05
23	0.00	0.13	0.00	0.00	0.08	0.04	0.08	0.01	0.01	0.07	0.05	0.05	0.05	0.04	0.05
24	0.03	0.15	0.00	0.00	0.06	0.03	0.09	0.00	0.01	0.07	0.04	0.05	0.04	0.04	0.04
25	0.00	0.11	0.00	0.00	0.09	0.03	0.05	0.00	0.00	0.07	0.02	0.02	0.02	0.02	0.02
26	0.00	0.05	0.00	0.00	0.04	0.01	0.02	0.00	0.00	0.03	0.01	0.01	0.01	0.01	0.01
27	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.01
28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Mais	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Máximo	0.24	0.17	0.30	0.29	0.16	0.16	0.16	0.13	0.12	0.19	0.13	0.12	0.13	0.14	0.13

**Tabela 6.5 – Taxas de ocorrência por profundidade máxima para um conjunto de 5000 indivíduos gerados aleatoriamente com código Aditivo, fatores de busca N de 33.3%, 66.6% e 100%, adotando as seguintes heurísticas de ordenação das arestas: Randômica (RD), Níveis crescente (N+), Níveis decrescente (N-), Correntes crescente (I+) e Correntes decrescente (I-).**

	N=33.3%					N=66.6%					N=100%				
	RD	N+	N-	I+	I-	RD	N+	N-	I+	I-	RD	N+	N-	I+	I-
10	0.97	0.99	0.00	0.00	0.31	0.34	0.56	0.00	0.00	0.16	0.09	0.09	0.09	0.08	0.09
11	0.00	0.00	0.00	0.00	0.59	0.18	0.25	0.01	0.00	0.33	0.23	0.23	0.22	0.22	0.23
12	0.00	0.01	0.00	0.00	0.06	0.23	0.13	0.04	0.03	0.25	0.26	0.27	0.26	0.26	0.26
13	0.02	0.00	0.00	0.00	0.03	0.17	0.05	0.15	0.12	0.16	0.21	0.21	0.21	0.22	0.22
14	0.00	0.00	0.00	0.34	0.01	0.02	0.01	0.33	0.38	0.05	0.11	0.11	0.12	0.12	0.11
15	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.16	0.15	0.03	0.06	0.06	0.06	0.05	0.05
16	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.09	0.11	0.01	0.03	0.03	0.03	0.03	0.02
17	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.06	0.06	0.00	0.01	0.01	0.01	0.01	0.01
18	0.01	0.00	0.00	0.59	0.00	0.00	0.00	0.08	0.07	0.00	0.00	0.00	0.00	0.00	0.00
19	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.04	0.04	0.00	0.00	0.00	0.00	0.00	0.00
20	0.00	0.00	0.64	0.01	0.00	0.00	0.00	0.03	0.04	0.00	0.00	0.00	0.00	0.00	0.00
21	0.00	0.00	0.06	0.00	0.00	0.00	0.00	0.02	0.01	0.00	0.00	0.00	0.00	0.00	0.00
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
23	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
24	0.00	0.00	0.27	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
27	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Mais	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Máximo	0.97	0.99	0.64	0.59	0.59	0.34	0.56	0.33	0.38	0.33	0.26	0.27	0.26	0.26	0.26

profundidades 13 e 16, já tendendo à profundidade 20. Análise similar pode ser feita para o código Aditivo, mas com tendência no sentido da profundidade 12.

É prudente ressaltar que as indicações dessa seção não podem ser generalizadas, por se referirem a um único sistema de teste. Todavia, os resultados obtidos são significativos e estão aderentes às expectativas discutidas na subseção 4.2.3. Ficou demonstrado para esse sistema de pequenas proporções que há heurísticas adequadas para cada uma das codificações sequenciais. Consequentemente, o AGS com código e heurística adequados deve apresentar um desempenho superior ao AG Básico. Testes com reconfigurações de diferentes sistemas teste e vários ajustes do AG e do AGS são o objeto de estudo do Capítulo 7.

## Capítulo 7

# Resultados

---

Este capítulo foi organizado em quatro seções: na primeira, os cinco sistemas de distribuição selecionados para os testes de validação e efetividade são descritos; na segunda seção, a metodologia usada para avaliar a efetividade das técnicas propostas é apresentada; nas duas seções posteriores, implementam-se os testes propriamente ditos das codificações sequenciais aplicadas ao AG Básico em uma primeira etapa e, em seguida, do AGS como proposto no Capítulo 6. No sentido de consolidar os resultados do capítulo, as seções com os testes do AG e AGS terminam com uma síntese dos resultados obtidos.

### 7.1 Descrição dos Sistemas Teste

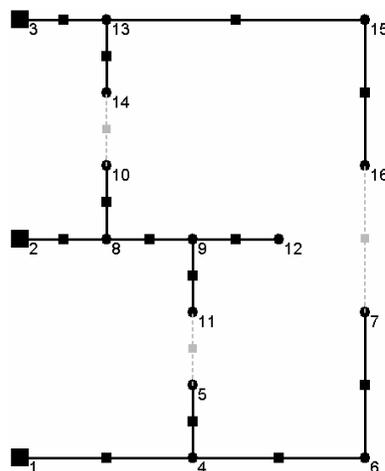
Para realizar os estudos de caso foram utilizados cinco sistemas de distribuição disponíveis na literatura técnica sobre Configuração de Redes. Os dois primeiros são sistemas clássicos identificados nesta seção como sistemas de *16-nós* e *33-nós*. Eles foram apresentados, respectivamente, por Cinvanlar et al (1988) e Baran & Wu (1989), sendo utilizados principalmente para validar as rotinas desenvolvidas. Para levar a efeito os testes de efetividade das codificações propostas e do AGS foram usados os seguintes sistemas: *70-nós* (DAS, 2006); *94-nós* (SU & LEE, 2003); e *137-nós* (MANTOVANI et al, 2000).

Apresenta-se na Tabela 7.1 um resumo das características mais relevantes dos sistemas testados, sendo suas configurações originais visualizadas nas Figuras 7.1 até 7.5. Seus dados completos são fornecidos no Apêndice A.

Uma característica comum aos sistemas de teste é a suposição de que todo ramo possui chave habilitada. De fato, supondo que os nós representam cargas normais, esta seria a condição mais complexa para um estudo de Reconfiguração,

**Tabela 7.1 – Dados referentes às configurações iniciais dos três sistemas usados nos testes.**

Dados	Sistema				
	16-nós	33-nós	70-nós	94-nós	137-nós
Subestações	3	1	2	11	2
Barras	13	32	68	83	135
Ramos	16	37	79	96	156
Cordas	3	5	11	13	21
Carga Ativa Total (MW)	28.700	3.715	4.468	28.350	18.314
Carga Reativa Total (Mvar)	5.900	2.300	3.059	20.700	7.933
Tensão Nominal (kV)	23.00	12.66	11.00	11.40	13.80
Carga Ativa Média (MW)	2.208	0.116	0.066	0.342	0.136
Desvio padrão Ativo	1.469	0.093	0.046	0.465	0.187
Carga Reativa Média (MW)	0.454	0.072	0.045	0.249	0.059
Desvio padrão Reativo	1.278	0.107	0.034	0.354	0.080
Perdas Ativas (MW)	0.5114	0.2027	0.2275	0.5320	0.3204
Perdas Ativas (%)	1.782	5.456	5.093	1.877	1.749
Tensão Mínima (kV)	22.29	11.56	9.95	10.59	12.84
Tensão Mínima (pu)	0.969	0.913	0.905	0.929	0.931



**Figura 7.1 – Diagrama do sistema 16-nós (CINVANLAR et al, 1988).**

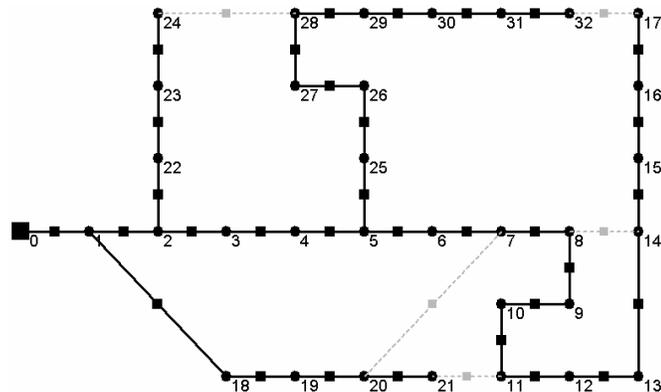


Figura 7.2 – Diagrama do sistema 33-nós (BARAN & WU, 1989).

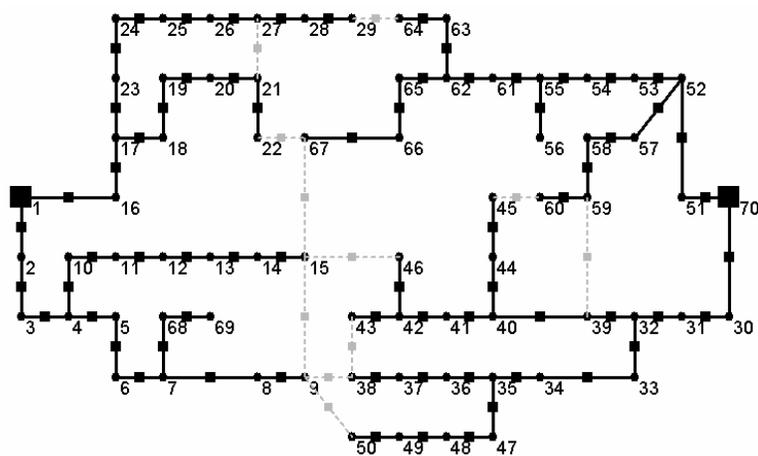


Figura 7.3 – Diagrama do sistema 70-nós (DAS, 2006).

devido ao grande número de chaves. Outra abordagem, até mais factível, seria considerar que os nós fornecidos para estes sistemas já se referem a setores e que se tratam de sistemas consideravelmente maiores que 16, 33, 70, 94 e 137 nós.

## 7.2 Metodologia do Estudo

Embora a formulação apresentada na Seção 3.3 seja geral, aplicando-se aos três enfoques da Configuração, optou-se por restringir os testes deste capítulo ao enfoque Reconfiguração. Não há impedimento em usar os códigos sequenciais ou mesmo o AGS nos outros enfoques, contudo é válido reconhecer que uma aplicação realista para Restabelecimento ou Planejamento requer ajustes adicionais nos algoritmos das Figuras

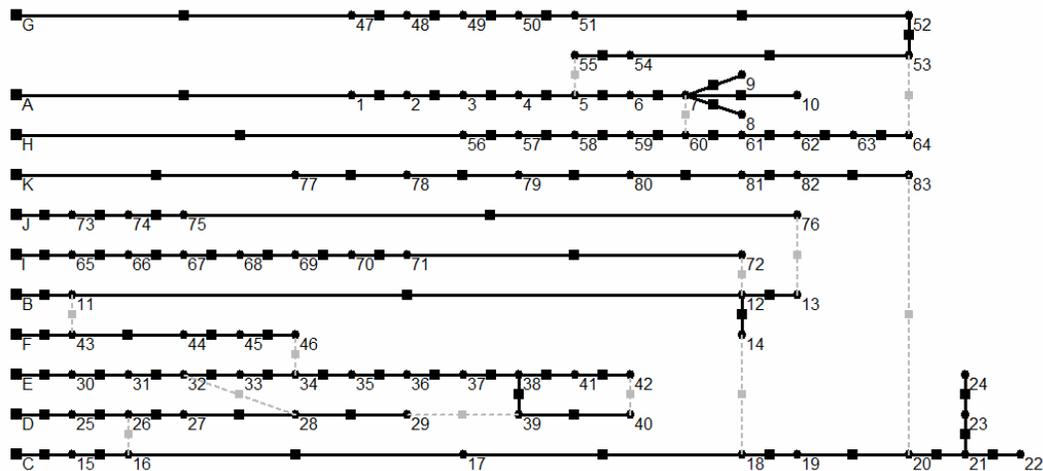


Figura 7.4 – Diagrama do sistema 94-nós (SU & LEE, 2003).

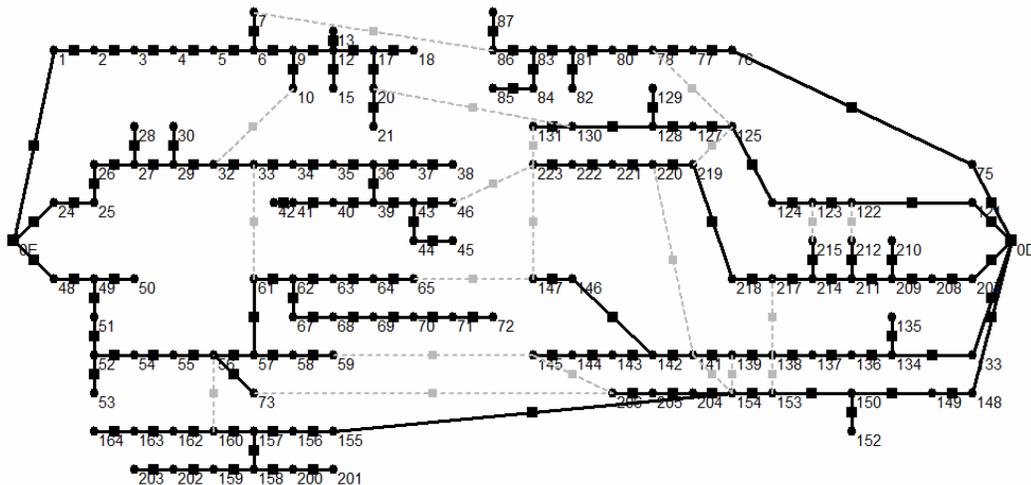


Figura 7.5 – Diagrama do sistema 137-nós (MANTOVANI et al, 2000).

3.7 e 6.1. No Restabelecimento, a prioridade seria reduzir o tempo de processamento e no Planejamento, além de incluir na solução eventuais reforços e expansões, a solução deveria contemplar alguma redundância no sentido de maximizar a confiabilidade da rede. Tais modificações estão além do objetivo principal do capítulo, que é validar os códigos sequencias associados ao AG Básico e ao próprio AGS.

Observe que a existência de uma heurística suportando Restabelecimento ou Planejamento suscita uma análise sobre a viabilidade de usar uma variante do AGS. Contudo, optou-se por sugerir essa pesquisa como um desdobramento deste trabalho.

Os testes foram levados a efeito primeiramente no AG Básico, tendo sido utilizadas as codificações Subtrativa, Aditiva e outras duas representações de referência na Reconfiguração dos cinco sistemas teste descritos na Seção 7.1. As codificações selecionadas como referência foram vetor característico (VC) e vetor característico melhorado (*Melhorado*).

Além da comparação direta com VC e Melhorado, os testes foram realizados visando permitir o cruzamento com resultados de vários trabalhos clássicos sobre o tema; isto é, foram escolhidos os mesmos sistemas teste e um ajuste dos parâmetros da função de aptidão (Equação 3.14) que implica em considerar os mesmos objetivos e restrições destas referências técnicas. Adicionalmente, as simulações foram conduzidas quando Subtrativa e Aditiva são associadas aos operadores genéticos convencionais e na calda, totalizando seis representações: VC, Melhorado, Subtrativa com operador convencional (SUB-CONV), Subtrativa com operador na calda (SUB-CALDA), Aditiva com operador convencional (ADT-CONV) e Aditiva com operador na calda (ADT-CALDA).

O intuito dos testes com o AG Básico é posicionar os códigos sequenciais em termos de desempenho em relação às suas concorrentes no estado da arte e, adicionalmente, estabelecer qual operador genético é o mais indicado para cada código sequencial.

Com relação aos testes com o AGS, a idéia inicial era testá-lo na Reconfiguração dos sistemas de 70, 94 e 137 nós, com Subtrativa e Aditiva associadas aos operadores convencional e na calda, adotando as quatro heurísticas formuladas na subseção 6.2.3, as duas estratégias de controle de  $N$  da subseção 6.2.1 e variando o número de patamares no intervalo de 2 a 5. Essa lista de alternativas resultaria em  $3*2*4*2*4 = 192$  combinações, tendo sido imediatamente abandonada.

Ao invés de explorar os diferentes ajustes simultaneamente, propõe-se separar os testes em três etapas: na primeira, são testadas as heurísticas estipulando o controle de  $N$  por larguras iguais e  $n_{\text{pat}} = 3$ ; em seguida, de posse das heurísticas indicadas para cada código e ainda com  $n_{\text{pat}} = 3$ , implementa-se o comparativo dos controles de  $N$  por larguras iguais e proporcionais; e, por fim, são levados a efeito os testes variando somente  $n_{\text{pat}}$ . Em todos os casos, o AGS é usado na Reconfiguração dos três sistemas teste maiores, sempre adotando Subtrativa e Aditiva associadas aos respectivos operadores recomendados pelos testes iniciais com o AG.

Nos testes, optou-se por realizar dez execuções para cada uma das configurações, de modo a permitir uma avaliação mais precisa em termos de robustez dos algoritmos. São apresentados a aptidão, geração em que foi obtida e tempo total de processamento da melhor execução verificada e, posteriormente, esse mesmo conjunto de dados, mas referente à média das dez execuções. A diferença entre os dados da melhor execução e a média indica com que frequência cada ajuste obtém as melhores soluções. Além disso, são apresentadas a melhor aptidão e a média das aptidões da população inicial com o objetivo de analisar tendências nos códigos.

Complementando os resultados, são apresentadas estimativas do tempo médio de obtenção da solução final ( $T_C$ ) e do tempo médio de avaliação por indivíduo ( $T_A$ ).  $T_C$  é considerado uma medida efetiva sobretudo para aplicações de maior porte, porque nestes casos é comum estabelecer como critério de parada a permanência de um mesmo indivíduo por um número fixo de gerações consecutivas. Nesta condição, o tempo total de execução será ligeiramente maior que  $T_C$ . A expressão utilizada para calculá-lo é:

$$T_C = \frac{g'}{g_{\max}} T_M. \quad (7.1)$$

Sendo,  $g'$  o número médio de gerações para obter a solução final;  $g_{max}$  o número total de gerações; e  $T_M$  o tempo total médio de execução do algoritmo. Aproveitando parte das grandezas definidas para (7.1),  $T_A$  é calculado pela seguinte expressão:

$$T_A = \frac{T_M}{g_{max} n_{POP}}. \quad (7.2)$$

Em que,  $n_{POP}$  é o tamanho da população adotado.

Considerando que todos os testes sejam levados a efeito em uma mesma máquina e com carregamento semelhante,  $T_A$  pode ser visto como uma estimativa da complexidade dos algoritmos de cada uma das representações testadas.

Observe que restringir os testes à Reconfiguração implica relaxar a restrição de tempo de processamento, de modo que somente quando duas técnicas obtiverem uma mesma solução é que  $T_M$ ,  $T_C$  ou  $T_A$  se tornam preponderantes.

Por uma questão de unicidade das informações, nos testes os valores de perdas ativas são apresentados em MW. Além disso, todas as execuções foram realizadas em um computador pessoal com processador Intel® Core 2 Duo de 2.4GHz e 3GB de memória. Como os testes visam subsidiar recomendações de ajustes para o AG e AGS, mesmo pequenas diferenças entre aptidões obtidas são consideradas significativas. O limite estabelecido foi de seis casas decimais.

### 7.3 Testes com o AG

Essa seção foi organizada em cinco subseções, uma dedicada a cada um dos sistemas teste. Em todos os casos, o enfoque é a Reconfiguração visando reduzir exclusivamente perdas. Utiliza-se fator de demanda unitário e a carga é modelada como de potência constante. No AG, optou-se pela seleção por roleta e inicialização randômica, contudo, incluindo na população inicial a configuração em operação.

### 7.3.1 Sistema de 16-nós

Para este estudo, as restrições de capacidade das subestações, suportabilidade dos condutores e nível de tensão foram desabilitadas e os ajustes do AG foram os mesmos de Enacheanu et al (2008) para codificação VC: população de 30 indivíduos, critério de parada ajustado para 40 gerações, e probabilidades de mutação e cruzamento fixas de 0.1 e 0.4, nesta ordem. A diferença foi que se utilizou escalamento linear com valor máximo de 100, mínimo de 10 e decremento de 6, elitismo do melhor indivíduo da população e uma probabilidade de mutação de 0.15 somente para Aditiva (subseção 5.6.6). Como será mostrado adiante, os ajustes visam permitir um comparativo com várias referências.

Apresentam-se na Tabela 7.2 os resultados médios obtidos para a população inicial ao utilizar os seis ajustes implementados. O maior e o menor valores em cada coluna são destacados, respectivamente, por um fundo cinza e por negrito. Esta formatação é adotada sempre que possível nas tabelas posteriores.

Claramente, VC obtém os piores resultados em termos de população inicial, sendo Aditiva a melhor seja qual for o operador. Além disso, observa-se que mesmo com performance inferior à Aditiva, o código Subtrativo parte com vantagem sobre Melhorado e VC. Essa diferença é mais contundente comparando as médias obtidas, reforçando o melhor desempenho de Subtrativa.

Em relação à busca e otimização propriamente ditas, são apresentadas na Tabela 7.3 as respostas obtidas com cada uma das seis configurações testadas. Observe que VC é a única incapaz de obter o valor ótimo, cuja aptidão é 0.984190 para o ajuste escolhido (0.4661 MW de perdas ativas). Melhorado, Subtrativa e Aditiva, independente do operador utilizado, não só obtiveram tal resposta como este foi o resultado nas 10 execuções, uma vez que a aptidão média é igual a da melhor solução.

**Tabela 7.2 – Aptidões médias das populações iniciais do AG, para as 10 execuções realizadas no sistema de 16-nós.**

Codificação	Aptidões médias da população inicial em dez execuções	
	Melhor	Média
VC	<b>0.982758</b>	<b>0.587570</b>
Melhorado	0.982929	0.861540
SUB-CONV	0.983587	0.956880
SUB-CALDA	0.983056	0.953000
ADT-CONV	0.983810	<b>0.977970</b>
ADT-CALDA	<b>0.983855</b>	0.977480

**Tabela 7.3 – Resultados obtidos para o sistema de 16-nós em 10 execuções do AG: melhor execução e dados médios.**

Codificação	Melhor solução			Média em 10 execuções			$T_C$ (s)	$T_A$ (s)
	Aptidão	Geração	Tempo (s)	Aptidão	Geração	Tempo (s)		
VC	<b>0.983605</b>	<b>10.0</b>	<b>0.0</b>	<b>0.983121</b>	4.7	0.6	0.071	<b>0.001</b>
Melhorado	0.984190	<b>0.0</b>	<b>0.0</b>	0.984190	17.0	0.8	<b>0.340</b>	0.001
SUB-CONV	0.984190	<b>0.0</b>	1.0	0.984190	2.7	1.0	<b>0.068</b>	0.001
SUB-CALDA	0.984190	1.0	1.0	0.984190	7.5	1.0	0.188	0.001
ADT-CONV	0.984190	<b>0.0</b>	<b>2.0</b>	0.984190	2.6	<b>2.1</b>	0.137	<b>0.002</b>
ADT-CALDA	0.984190	1.0	1.0	0.984190	<b>2.4</b>	2.0	0.120	0.002

As cordas da melhor configuração obtida, identificadas pelos nós incidentes, são 7-16, 8-10 e 9-11 (Figura 7.6). A redução nas perdas ativas verificada foi de 8.9 %.

Devido ao empate nas aptidões para cinco configurações, a escolha do melhor desempenho para este sistema deve ser baseada em  $T_C$  e  $T_A$ . Avaliando  $T_C$ , SUB-CONV é a mais rápida para convergir, com um tempo médio de 68 ms. Em seguida, aparece ADT-CALDA com 120 ms. Verifica-se também que os códigos sequenciais, seja qual for o operador, levam vantagem em relação à representação Melhorado. A menor diferença é para SUB-CALDA: 188 ms contra 340 ms (81%). Além disso, o operador na calda aparentemente melhora o desempenho de Aditiva e piora Subtrativa.

Tomando os tempos de avaliação por indivíduo ( $T_A$ ), Aditiva indica a maior complexidade dentre as configurações testadas. Entretanto, os tempos são tão

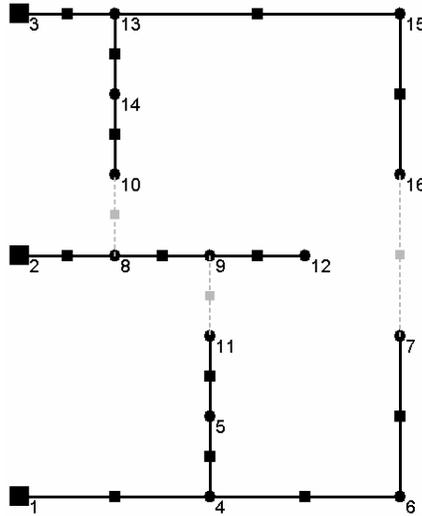


Figura 7.6 – Melhor configuração obtida pelo AG para o sistema 16-nós.

pequenos que essa análise é considerada pouco efetiva para o sistema 16-nós.

Comparando com o estado da arte, a melhor solução obtida reproduz os resultados de referências clássicas, como: Mantovani et al (2000), Zhu (2002), Hong & Ho (2005) e Enacheanu et al (2008). A única diferença é com relação a Mendoza et al (2006a) cuja melhor solução tem as seguintes cordas: 8-10, 5-11 e 7-16. Esta solução implica em perdas ativas de 0.4839MW, logo 3.8% maiores que o melhor valor obtido nesta seção. Tratando de complexidade, o trabalho que obtém a solução ótima com o menor tempo de execução é Enacheanu et al (2008), com 2.1 segundos para executar um total de 150 avaliações:  $T_A = 14$  ms. A diferença também é significativa considerando  $T_C$ , pois Enacheanu et al (2008) obtiveram a solução ótima na geração 9 de um total de 15:  $T_C = 2.1*(9/15) = 1.26$  segundos.

Os resultados apresentados nessa seção são promissores para os códigos sequenciais, no entanto, há ressalvas: esse sistema é extremamente pequeno e, conseqüentemente, seu espaço de busca é limitado; AGs com alguns parâmetros diferentes estão sendo comparados; e a análise de complexidade baseada no tempo de execução para testes realizados em máquinas diferentes tem pouco efeito. Nesse

sentido, as comparações entre os códigos sequenciais e as representações de referência se mostram mais efetivas. A subseção 7.3.2 descreve testes com um sistema um pouco maior.

### 7.3.2 Sistema de 33-nós

As restrições de capacidade das subestações e suportabilidade dos condutores foram desabilitadas neste estudo, entretanto, a restrição de tensão nodal foi considerada com tolerância de 90% - 105% (ANEEL, 2010). Os ajustes do AG foram os mesmos de Enacheanu et al (2008) para codificação VC: população de 40 indivíduos, critério de parada ajustado para 50 gerações, e probabilidades de mutação e cruzamento fixas de 0.1 e 0.4, nesta ordem. A diferença foi que se adotou escalamento linear com valor máximo de 100, mínimo de 10 e decremento de 3, elitismo do melhor indivíduo da população e probabilidade de mutação de 0.15 somente para Aditiva.

Apresentam-se na Tabela 7.4 os resultados médios obtidos para a população inicial nos seis ajustes implementados. Como antes, VC apresenta o pior desempenho em termos de população inicial e Aditiva é a melhor, seja qual for o operador utilizado. Subtrativa aparece novamente entre Aditiva e Melhorado quando a análise é com base no melhor indivíduo, porém Melhorado supera Subtrativa quando a média é considerada.

Tomando os resultados da melhor execução (Tabela 7.5), novamente apenas VC foi incapaz de chegar ao valor ótimo, cuja aptidão é 0.965165 para o ajuste escolhido (0.1395 MW de perdas ativas). Embora todas as configurações exceto VC tenham ao menos uma vez obtido a melhor solução, a vantagem de ADT-CALDA neste teste é significativa, uma vez que foi o único ajuste capaz de obter a solução ótima em todas as execuções. O segundo melhor resultado foi obtido por ADT-CONV, com uma

**Tabela 7.4 – Aptidões médias das populações iniciais do AG, para as 10 execuções realizadas no sistema de 33-nós.**

Codificação	Aptidões médias da população inicial em dez execuções	
	Melhor	Média
VC	0.951336	0.138290
Melhorado	0.952167	0.528110
SUB-CONV	0.958553	0.446650
SUB-CALDA	0.958660	0.439810
ADT-CONV	0.964355	0.898130
ADT-CALDA	0.964309	0.900290

**Tabela 7.5 – Resultados obtidos para o sistema de 33-nós em 10 execuções do AG: melhor execução e dados médios.**

Codificação	Melhor solução			Média em 10 execuções			$T_C$ (s)	$T_A$ (s)
	Aptidão	Geração	Tempo (s)	Aptidão	Geração	Tempo (s)		
VC	0.962035	7.0	2.0	0.953188	6.3	1.9	0.239	0.001
Melhorado	0.965165	32.0	5.0	0.964337	31.9	4.1	2.616	0.002
SUB-CONV	0.965165	45.0	4.0	0.964837	29.7	3.8	2.257	0.002
SUB-CALDA	0.965165	20.0	3.0	0.964745	30.7	3.7	2.272	0.002
ADT-CONV	0.965165	11.0	6.0	0.965130	10.9	8.3	1.809	0.004
ADT-CALDA	0.965165	15.0	7.0	0.965165	16.3	8.4	2.738	0.004

média de 0.965130. Isto é, mesmo sem considerar  $T_C$  ou  $T_A$ , Aditiva apresentou o melhor desempenho dentre todos os ajustes, seja qual for o operador utilizado.

Considerando tempos de processamento e restringindo a análise às representações que chegaram ao ótimo pelo menos uma vez, ADT-CALDA é a mais lenta com relação ao tempo para convergir e também na avaliação por indivíduo. Todavia, como dito na Seção 7.2, a análise com base nos tempos só é preponderante quando ocorre um empate em termos de aptidões, o que não vale para este teste. Comparando com base nos operadores genéticos, o resultado foi diferente do que havia sido observado para 16-nós: o operador na calda tem efeito aumentando  $T_C$  em Aditiva e Subtrativa. A melhor configuração obtida é apresentada na Figura 7.7 e possui as seguintes cordas: 6-7, 8-9, 13-14, 24-28 e 31-32.

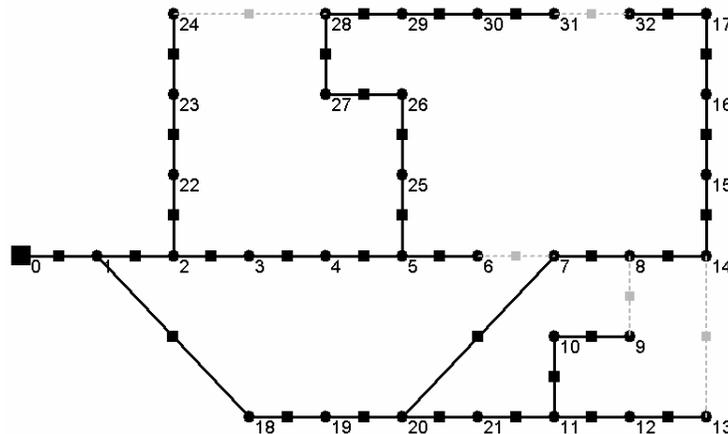


Figura 7.7 – Melhor configuração obtida pelo AG para o sistema 33-nós.

No conjunto de referências consultadas no estado da arte, seis trabalhos obtiveram a mesma solução ótima apresentada nesta seção: Mantovani et al (2000), Lin et al (2000), Zhu (2002), Mendoza et al (2006a), Wang & Cheng (2008) e Enacheanu et al (2008). Computacionalmente, a solução mais rápida é a de Mendoza et al (2006a), em que  $T_A = 12$  ms. Logo, Subtrativa e Aditiva apresentam tempos de processamento melhores que o estado da arte: de 2 ms até 4 ms. Porém, ressaltando que tal análise é imprecisa por adotar resultados obtidos com equipamentos e condições diferentes. A redução observada nas perdas ativas foi de 31.2 %.

### 7.3.3 Sistema de 70-nós

Esse é o primeiro dos sistemas selecionados para os testes de efetividade dos códigos, uma vez que suas dimensões permitem uma análise mais realista. Novamente, são usados os parâmetros do AG fornecidos em Enacheanu et al (2008) para VC: população de 70 indivíduos, critério de parada de 100 gerações, e probabilidades de mutação e cruzamento fixas, respectivamente, de 0.1 e 0.4. Adicionalmente, foram utilizados escalamento linear com valor máximo de 100, mínimo de 10 e decremento 1, elitismo do melhor indivíduo da população e uma probabilidade de mutação 0.15 somente para Aditiva.

As suportabilidades dos cabos são consideradas, bem como uma tolerância na variação de tensão das barras de 90% - 105%. Permanece desabilitada apenas a restrição de capacidade das subestações. Apresentam-se na Tabela 7.6 os resultados médios obtidos para a população inicial ao utilizar as seis configurações testadas e, na Tabela 7.7, os resultados propriamente ditos.

Pela análise da Tabela 7.6, fica clara uma diferença deste teste em relação ao que foi observado antes com 16-nós e 33-nós: a melhor solução codificada inicialmente por VC, Melhorado, SUB-CONV e SUB-CALDA possui uma mesma aptidão. De fato, tal solução corresponde a configuração inicial da rede, que é incluída sempre nas populações iniciais. Isto é, com o aumento do espaço de busca das soluções, apenas Aditiva foi capaz de gerar aleatoriamente soluções melhores que a configuração em operação, confirmando sua forte tendência positiva. Além disso, a exemplo do que

**Tabela 7.6 – Aptidões médias das populações iniciais do AG, para as 10 execuções realizadas no sistema de 70-nós.**

Codificação	Aptidões médias da população inicial em dez execuções	
	Melhor	Média
VC	0.953895	0.051710
Melhorado	0.953895	0.085250
SUB-CONV	0.953895	0.059760
SUB-CALDA	0.953895	0.059600
ADT-CONV	0.955213	0.442920
ADT-CALDA	0.954879	0.430950

**Tabela 7.7 – Resultados obtidos para o sistema de 70-nós em 10 execuções do AG: melhor execução e dados médios.**

Codificação	Melhor solução			Média em 10 execuções			$T_c$ (s)	$T_A$ (s)
	Aptidão	Geração	Tempo (s)	Aptidão	Geração	Tempo (s)		
VC	0.957083	59.0	20.0	0.954230	9.3	19.1	1.776	0.003
Melhorado	0.957967	82.0	32.0	0.957521	69.3	34.1	23.631	0.005
SUB-CONV	0.958260	26.0	32.0	0.957196	38.3	29.8	11.413	0.004
SUB-CALDA	0.958282	83.0	26.0	0.957992	50.7	28.5	14.450	0.004
ADT-CONV	0.958260	33.0	65.0	0.957985	38.5	65.4	25.179	0.009
ADT-CALDA	0.958282	77.0	61.0	0.958064	42.8	70.4	30.131	0.010

havia sido observado com 33-nós, Melhorado supera Subtrativa na média das aptidões.

Partindo para a Tabela 7.7, a melhor solução encontrada tem uma aptidão de 0.958282 para o ajuste escolhido (0.20309 MW de perdas ativas). Tal solução foi obtida somente por SUB-CALDA e ADT-CALDA, indicando uma vantagem do operador na calda neste teste. Os desempenhos de SUB-CONV e ADT-CONV foram inferiores aos pares com operador na calda, mas ainda melhor que o apresentado por VC e Melhorado avaliando o melhor resultado obtido. Contudo, Melhorado se recupera quando são comparados resultados médios, ficando à frente de SUB-CONV, com 0.957521. A melhor média observada foi de ADT-CALDA: 0.958064.

Computacionalmente, ADT-CALDA foi a configuração mais lenta tanto para convergir como na avaliação por indivíduo, repetindo o resultado obtido com 33-nós. Todavia, como não houve empate em termos de aptidões, essa análise pode ser considerada secundária. As cordas relativas a melhor configuração obtida são as seguintes: 9-15, 9-38, 13-14, 15-67, 21-27, 28-29, 38-43, 39-59, 40-44, 49-50 e 62-65 (Figura 7.8).

Nas Figuras 7.9 e 7.10 é possível observar a evolução das codificações Subtrativa e Aditiva por geração em relação aos códigos de referência. O comparativo é

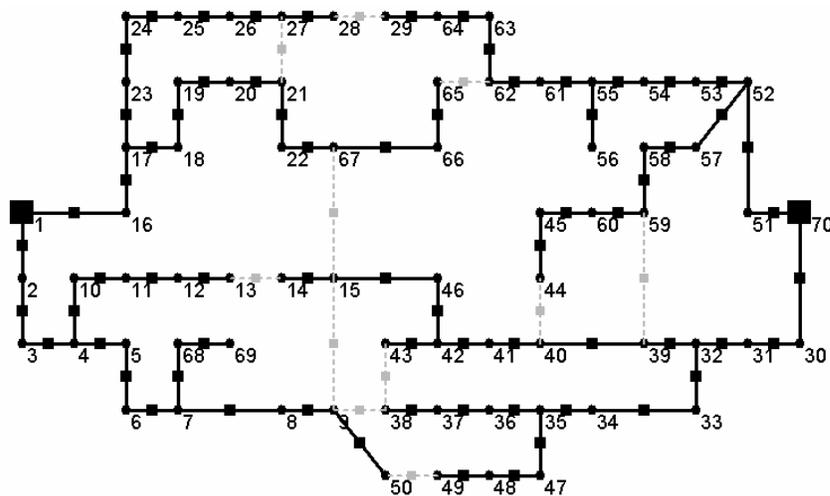
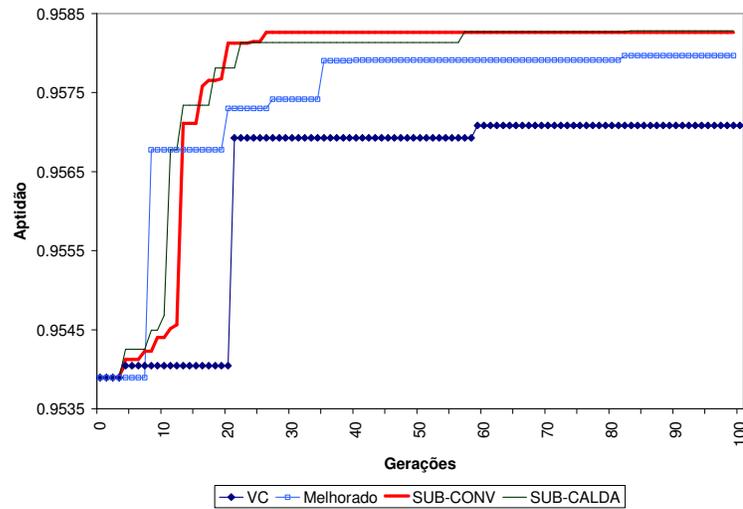
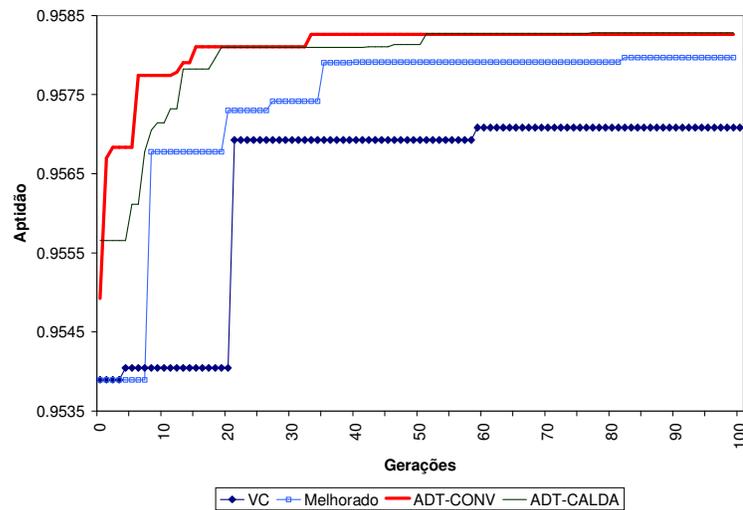


Figura 7.8 – Melhor configuração obtida pelo AG para o sistema 70-nós.



**Figura 7.9 – Comparativo das melhores execuções no sistema 70-nós por geração: Subtrativa contra VC e Melhorado.**



**Figura 7.10 – Comparativo das melhores execuções no sistema 70-nós por geração: Aditiva contra VC e Melhorado.**

feito tomando o melhor resultado de cada configuração, escolhido com base na aptidão, tempo de execução e geração em que a solução final foi obtida, nesta ordem. Esse procedimento é adotado em todos os comparativos posteriores deste capítulo.

Observe que além de terminar com uma solução melhor que VC e Melhorado na Figura 7.9, Subtrativa supera as referências durante a maior parte do tempo: desde as gerações 13 e 12, respectivamente, para operadores convencionais e na

calda. Essa vantagem é ainda mais destacada com Aditiva (Figura 7.10), visto que VC e Melhorado são superadas durante toda a execução.

Se o comparativo é levado a efeito com base na evolução dos códigos em relação ao tempo de execução, as conclusões são semelhantes. Apresentam-se nas Figuras 7.11 e 7.12 a evolução de Subtrativa e Aditiva com relação aos códigos de referência para essa condição. Nesses gráficos fica claro também, com base nos eixos das abscissas, que Aditiva apresenta maior complexidade computacional que Subtrativa e que a adoção dos operadores na calda diminui o tempo total de execução em ambos os casos: Subtrativa e Aditiva.

Considerando o estado da arte, a configuração obtida é melhor que àquela fornecida por Das (2006), cujas perdas ativas são de 0.2055 MW, e se equivale à configuração encontrada pelo AG convencional de Enacheanu et al (2008)<sup>2</sup>. A redução total nas perdas ativas foi de, aproximadamente, 10.7 %.

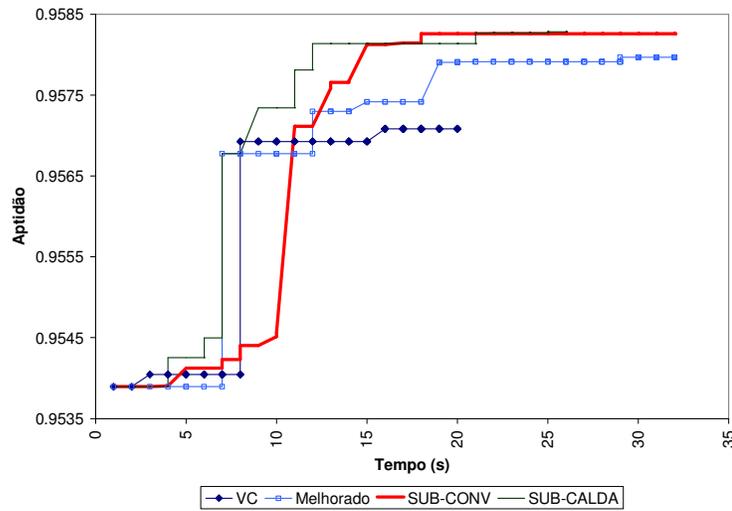
#### **7.3.4 Sistema de 94-nós**

Os seguintes parâmetros são usados neste estudo: população de 100 indivíduos, critério de parada de 200 gerações, e probabilidades de mutação e cruzamento fixas, respectivamente, de 0.1 e 0.4. Uma exceção foi feita para Aditiva, em que a probabilidade de mutação foi ajustada para 0.15, conforme sugestão apresentada na subseção 5.6.6.

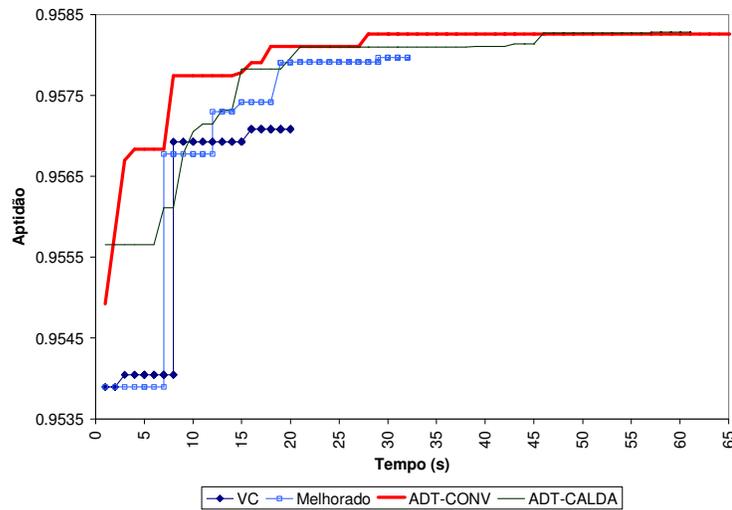
Além dos parâmetros descritos, foram utilizados escalamento linear com valor máximo de 200, mínimo de 20 e decremento 2 e elitismo do melhor indivíduo da população. A única restrição considerada foi o nível de tensão nas barras, limitado ao

---

<sup>2</sup> Ao testar sua codificação baseada em Matroids, Enacheanu et al (2008) obtiveram uma configuração supostamente melhor que a solução apresentada aqui. Aparentemente, a chave  $S_{71}$  ligando os nós 47 e 68 em Enacheanu et al (2008) conecta de fato os nós 47 e 16 no sistema original. Fazendo esta correção, a configuração apresentada implica em perdas maiores, da ordem de 0.2048MW.



**Figura 7.11 – Comparativo das melhores execuções no sistema 70-nós por tempo de execução: Subtrativa contra VC e Melhorado.**



**Figura 7.12 – Comparativo das melhores execuções no sistema 70-nós por tempo de execução: Aditiva contra VC e Melhorado.**

intervalo 90% - 105%. Apresentam-se na Tabela 7.8 os resultados médios obtidos para a população inicial ao utilizar as seis configurações testadas e, na Tabela 7.9, os respectivos resultados.

Os dados da Tabela 7.8 confirmam a tendência observada para 70-nós, em que apenas o código Aditivo foi capaz de gerar na inicialização uma solução melhor que a própria configuração inicial da rede. Porém, essa vantagem parece ter decrescido, sugerindo que mesmo Aditiva partirá da configuração em operação para sistemas ainda

**Tabela 7.8 – Aptidões médias das populações iniciais do AG, para as 10 execuções realizadas no sistema de 94-nós.**

Codificação	Aptidões médias da população inicial em dez execuções	
	Melhor	Média
VC	0.982124	0.036960
Melhorado	0.982124	0.045360
SUB-CONV	0.982124	0.148730
SUB-CALDA	0.982124	0.146480
ADT-CONV	0.982235	0.609600
ADT-CALDA	0.982141	0.619730

**Tabela 7.9 – Resultados obtidos para o sistema de 94-nós em 10 execuções do AG: melhor execução e dados médios.**

Codificação	Melhor solução			Média em 10 execuções			$T_c$ (s)	$T_A$ (s)
	Aptidão	Geração	Tempo (s)	Aptidão	Geração	Tempo (s)		
VC	0.982936	43.0	86.0	0.982553	47.8	88.0	21.032	0.004
Melhorado	0.984056	180.0	215.0	0.983783	157.6	184.2	145.150	0.009
SUB-CONV	0.984065	127.0	175.0	0.983844	97.9	190.5	93.250	0.010
SUB-CALDA	0.984065	85.0	178.0	0.983897	119.8	203.8	122.076	0.010
ADT-CONV	0.984065	54.0	410.0	0.983082	73.3	440.1	161.297	0.022
ADT-CALDA	0.984065	87.0	342.0	0.983241	85.0	406.3	172.678	0.020

maiores. Ocorre o esperado com Aditiva em relação a média, posicionando-se novamente como a melhor. Entretanto, Subtrativa aparece à frente de Melhorado, o que contraria os resultados de 33-nós e 70-nós.

Na execução propriamente dita (Tabela 7.9), VC e Melhorado foram as únicas que não chegaram a melhor solução, cuja aptidão é 0.984065 para o ajuste escolhido (0.4699 MW de perdas ativas). Subtrativa e Aditiva, independente do operador, chegaram a melhor solução ao menos uma vez. Todavia, tomando como base os valores médios, o desempenho de Aditiva foi significativamente inferior ao de Subtrativa. Na verdade, pela primeira vez, Aditiva foi superada também por Melhorado tomando valores médios: 0.983241 contra 0.983783.

Ainda com relação a Tabela 7.9, a exemplo do que havia ocorrido com 70-nós, as versões de Subtrativa e Aditiva que usam operadores na calda obtiveram os

melhores resultados. Sendo de SUB-CALDA o melhor resultado dentre as seis configurações testadas, com uma aptidão média de 0.983897. A convergência mais rápida dentre as configurações que obtiveram a solução ótima ao menos uma vez foi SUB-CONV:  $T_C = 93.250$  segundos; seguida por SUB-CALDA, ADT-CONV e ADT-CALDA. A indicação é que o operador na calda contribuiu aumentando o tempo para convergência e que tal aumento repercutiu na melhoria das aptidões alcançadas.

Tratando de complexidade, não houve imprevistos: Aditiva requer em média o dobro de tempo que Subtrativa para avaliar um indivíduo. A melhor configuração obtida é apresentada na Figura 7.13 e possui as seguintes cordas: 6-7, 11-43, 12-13, 14-18, 16-26, 28-32, 33-34, 38-39, 41-42, 54-55, 61-62, 71-72 e 82-83.

Para evidenciar a evolução do AG Básico quando as codificações sequenciais são adotadas em comparação com VC e Melhorado, nas Figuras 7.14 e 7.15 são apresentados os gráficos de aptidão no curso das gerações. Mais uma vez, o comparativo é implementado tomando o melhor resultado de cada configuração. Veja que na maior parte das gerações a solução encontrada pelos códigos Subtrativo e Aditivo é melhor que a de VC e Melhorado. Se o comparativo é implementado com base nas aptidões em relação ao tempo de execução, as conclusões são semelhantes

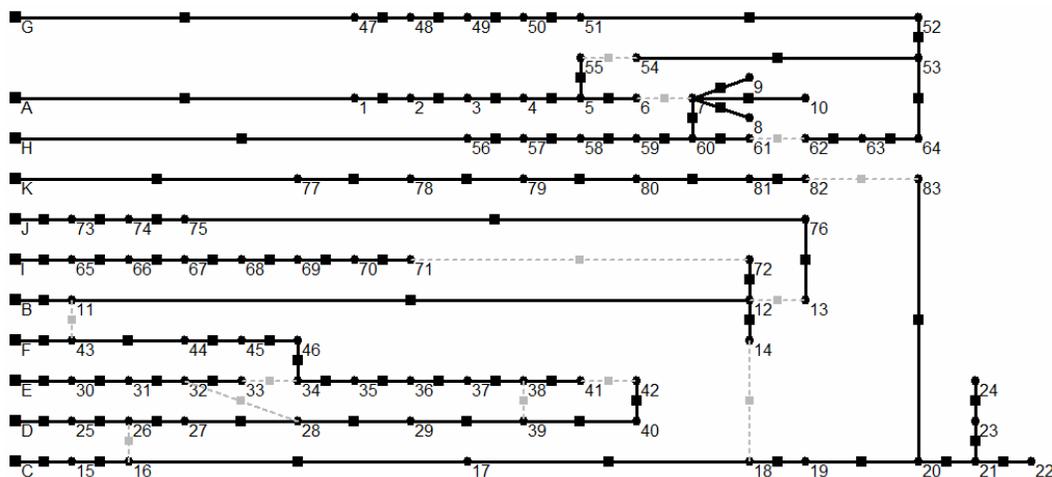


Figura 7.13 – Melhor configuração obtida pelo AG para o sistema 94-nós.

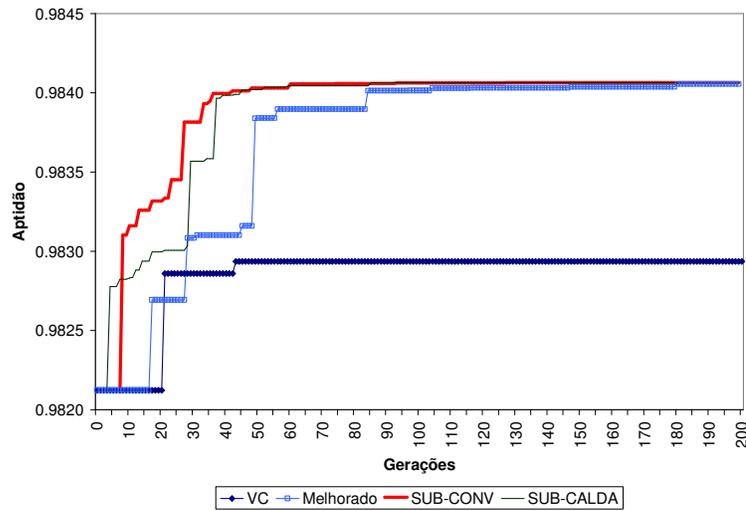


Figura 7.14 – Comparativo das melhores execuções no sistema 94-nós por geração: Subtrativa contra VC e Melhorado.

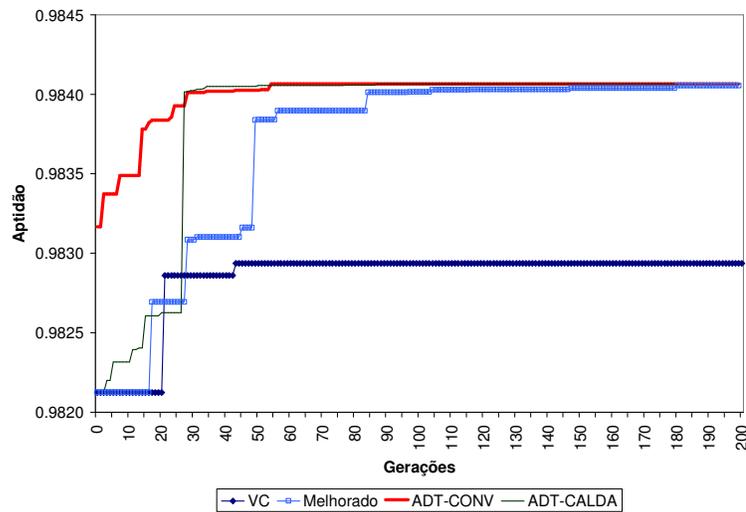


Figura 7.15 – Comparativo das melhores execuções no sistema 94-nós por geração: Aditiva contra VC e Melhorado.

(Figuras 7.16 e 7.17).

Considerando o estado da arte, a solução ótima encontrada reproduz os melhores resultados das seguintes referências: Su & Lee (2003), Su et al (2005), Chiou et al (2005), Wang & Cheng (2008), Chang (2008), Raju & Bijwe (2008) e Jazebi et al (2008). Nenhum dos trabalhos consultados obteve uma topologia com perdas menores, sendo o algoritmo apresentado por Su & Lee (2003) o mais rápido dentre todos:  $T_C = 36.15$  segundos. A redução nas perdas ativas foi de, aproximadamente, 11.7 %.

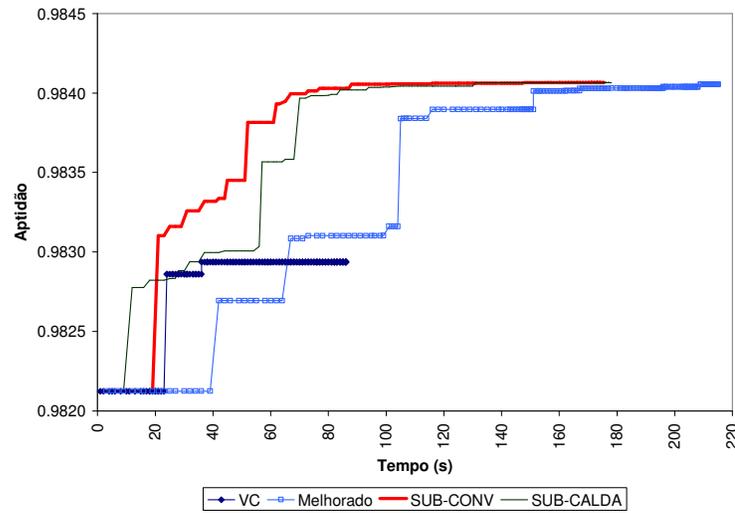


Figura 7.16 – Comparativo das melhores execuções no sistema 94-nós por tempo de execução: Subtrativa contra VC e Melhorado.

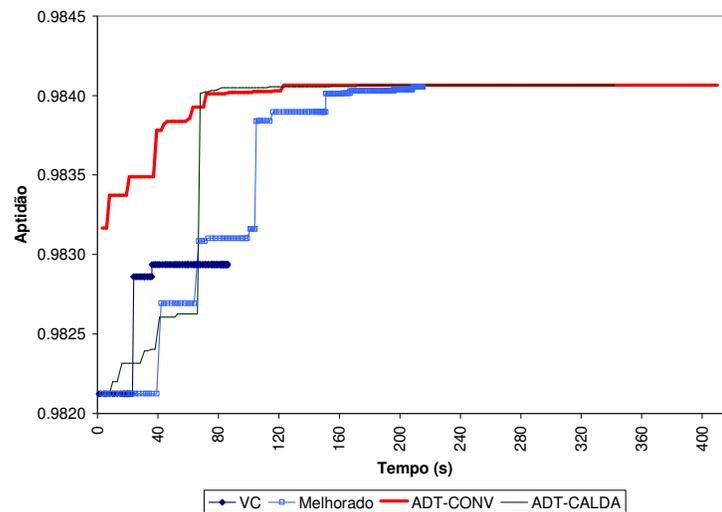


Figura 7.17 – Comparativo das melhores execuções no sistema 94-nós por tempo de execução: Aditiva contra VC e Melhorado.

### 7.3.5 Sistema de 137-nós

O espaço de busca para este sistema é o maior dentre os sistemas teste, tornando-o o mais realista. São usados os seguintes parâmetros para o AG: população de 150 indivíduos, critério de parada de 300 gerações e probabilidades de mutação e cruzamento fixas, respectivamente, de 0.1 e 0.4. Novamente a exceção foi Aditiva, em que se adotou uma probabilidade de mutação de 0.15. Adicionalmente, foram utilizados

escalamento linear com valor máximo de 200, mínimo de 20 e decremento 1, e elitismo do melhor indivíduo da população. A única restrição considerada foi a tensão nas barras, limitada ao intervalo 90% - 105%.

Apresentam-se na Tabela 7.10 os resultados obtidos para a população inicial nas seis configurações testadas. Como vinha sendo indicado nos testes anteriores, devido ao aumento no espaço de busca, nem mesmo Aditiva obteve na inicialização uma solução melhor que a própria configuração inicial da rede, cuja aptidão é 0.983285. Contudo, Aditiva permanece como a melhor considerando a aptidão média da população inicial, com uma vantagem significativa em relação às demais. Subtrativa aparece novamente atrás de Melhorado, de certo modo, indicando um padrão.

Partindo para os resultados da Tabela 7.11, fica clara mais uma vez a desvantagem de VC e Melhorado em relação a qualquer das configurações que usam

**Tabela 7.10 – Aptidões médias das populações iniciais do AG, para as 10 execuções realizadas no sistema de 137-nós.**

Codificação	Aptidões médias da população inicial em dez execuções	
	Melhor	Média
VC	0.983285	<b>0.008190</b>
Melhorado	0.983285	0.017310
SUB-CONV	0.983285	0.009530
SUB-CALDA	0.983285	0.009920
ADT-CONV	0.983285	0.339890
ADT-CALDA	0.983285	0.356390

**Tabela 7.11 – Resultados obtidos para o sistema de 137-nós em 10 execuções do AG: melhor execução e dados médios.**

Codificação	Melhor solução			Média em 10 execuções			$T_c$ (s)	$T_A$ (s)
	Aptidão	Geração	Tempo (s)	Aptidão	Geração	Tempo (s)		
VC	<b>0.984886</b>	<b>29.0</b>	<b>750.0</b>	0.983645	<b>72.9</b>	<b>743.0</b>	<b>180.549</b>	<b>0.017</b>
Melhorado	0.985010	299.0	2533.0	0.984353	238.1	2662.6	2113.217	0.059
SUB-CONV	0.985107	288.0	1184.0	0.985025	134.5	1122.9	503.434	0.025
SUB-CALDA	0.985129	211.0	1095.0	0.985072	161.0	1081.3	580.298	0.024
ADT-CONV	0.985057	173.0	2620.0	0.984984	82.6	2825.9	778.064	0.063
ADT-CALDA	0.985057	64.0	2829.0	0.984989	90.7	2746.3	830.298	0.061

código sequencial. Tais representações de referência obtiveram no melhor caso uma solução com aptidão 0.985010, contra uma melhor solução geral de 0.985129 (0.2809 MW de perdas ativas) encontrada exclusivamente por SUB-CALDA.

De acordo com a média das dez execuções, o melhor resultado é igualmente de SUB-CALDA, com média 0.985072. Em seguida, aparecem SUB-CONV, ADT-CALDA e ADT-CONV, indicando não só uma melhor performance de Subtrativa em relação à Aditiva, como uma vantagem consistente dos operadores na calda em comparação com os convencionais. As cordas relativas a melhor configuração obtida são as seguintes: 7-86, 9-10, 20-130, 39-43, 57-61, 59-145, 65-147, 78-125, 125-219, 131-223, 137-138, 138-217, 139-154, 141-154, 141-220, 144-145, 154-204, 155-156, 211-212, 214-215 e 222-223 (Figura 7.18).

Embora não seja possível generalizar os resultados da Tabela 7.11, eles reforçam o argumento que códigos mais elaborados como Subtrativa e Aditiva podem melhorar o desempenho do AG principalmente nos sistemas de grande porte, em que a explosão combinatória no número de soluções infactíveis pode comprometer a busca.

Considerando a convergência, a representação mais rápida dentre os códigos sequenciais foi SUB-CONV:  $T_C = 503.434$  segundos. Seguida por SUB-

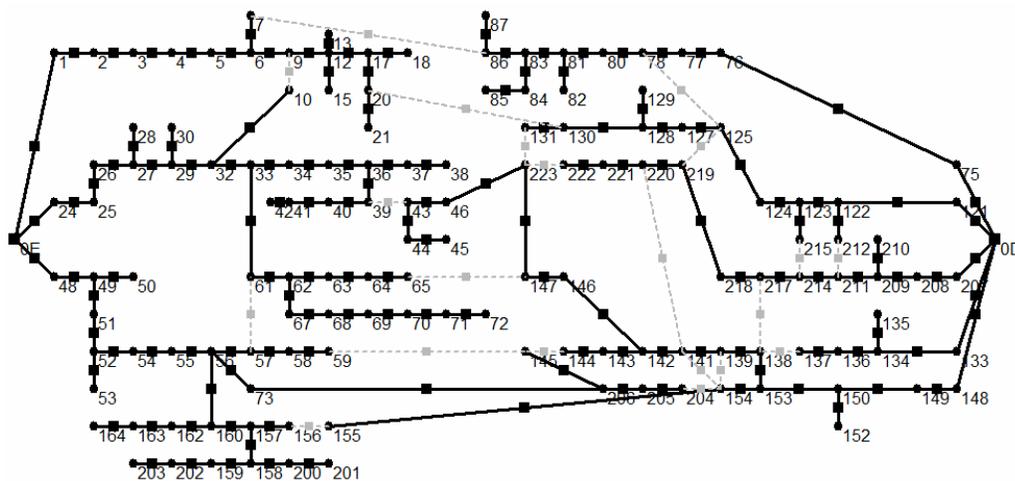
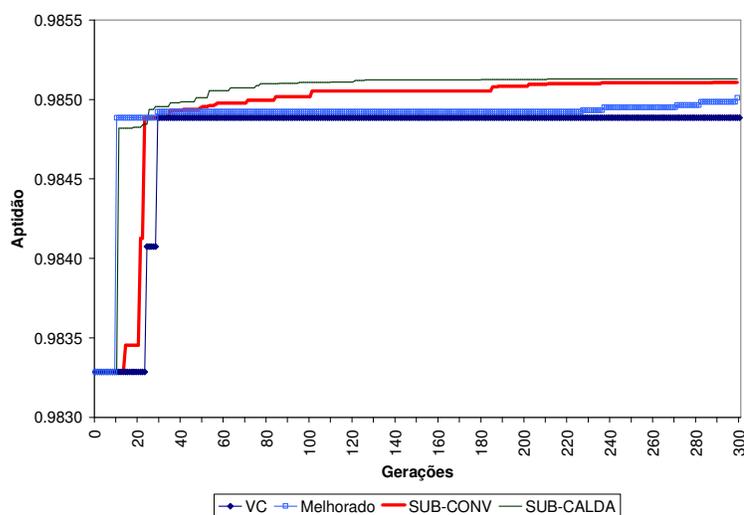


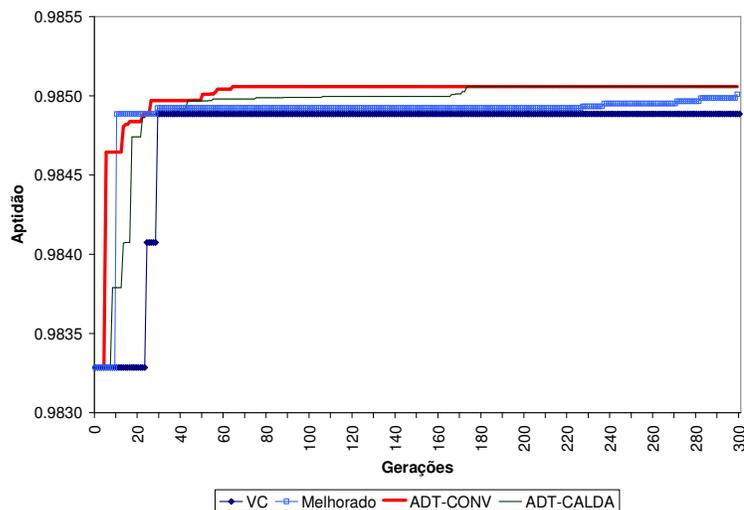
Figura 7.18 – Melhor configuração obtida pelo AG para o sistema 137-nós.

CALDA, ADT-CONV e ADT-CALDA, nesta ordem. Os resultados apontam mais uma vez que os operadores na calda tendem a aumentar  $T_C$ .

Nas Figuras 7.19 e 7.20 plotou-se a evolução do melhor indivíduo por geração, respectivamente, de Subtrativa e Aditiva em comparação com VC e Melhorado. Novamente, fica evidente o melhor desempenho dos códigos sequenciais não só pelo final da execução, mas no curso de praticamente todas as gerações. Se o comparativo é levado a efeito com base na evolução de cada configuração em relação ao



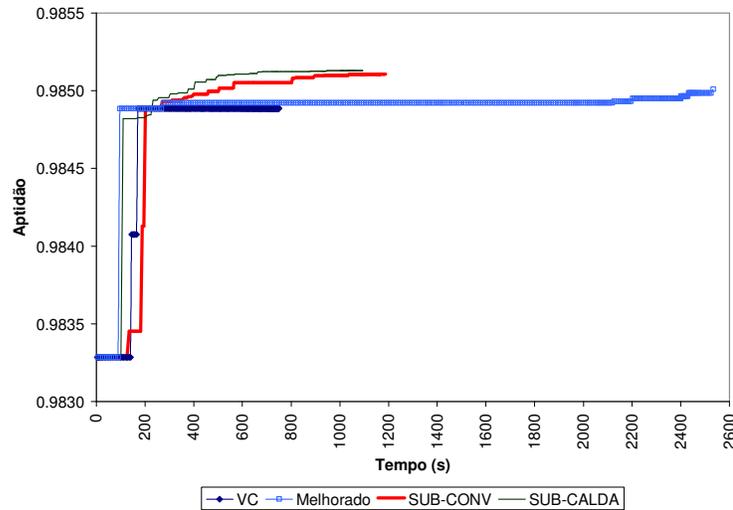
**Figura 7.19 – Comparativo das melhores execuções no sistema 137-nós por geração: Subtrativa contra VC e Melhorado.**



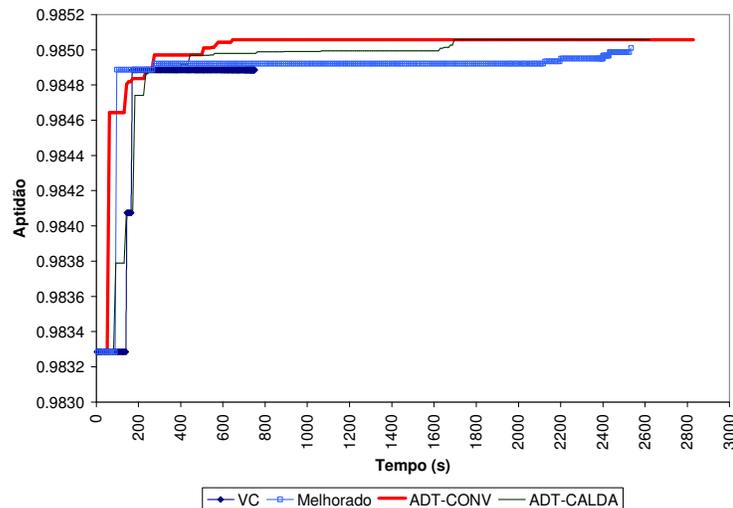
**Figura 7.20 – Comparativo das melhores execuções no sistema 137-nós por geração: Aditiva contra VC e Melhorado.**

tempo de execução, as conclusões são semelhantes (Figuras 7.21 e 7.22). Isto é, o fato de  $T_A$  ser maior nos códigos sequenciais é compensado com sobras pela melhoria proporcionada no processo de busca.

Os gráficos das Figuras 7.21 e 7.22 também evidenciam a maior complexidade de Aditiva em relação a Subtrativa e um resultado interessante com relação aos tempos de processamento quando se utilizam operadores convencionais e na calda. Embora os operadores na calda tendam a aumentar  $T_C$ , os gráficos indicam uma



**Figura 7.21 – Comparativo das melhores execuções no sistema 137-nós por tempo de execução: Subtrativa contra VC e Melhorado.**



**Figura 7.22 – Comparativo das melhores execuções no sistema 137-nós por tempo de execução: Aditiva contra VC e Melhorado.**

redução no tempo total de execução quando eles são utilizados. Essa redução fica evidente avaliando  $T_A$  e se deve ao aumento na taxa de utilização da memória genética (subseção 3.3.4). O argumento é que os operadores na calda provocam menos perturbações nas populações, o que favorece a memória.

Para exemplificar a melhoria de desempenho da memória genética com operadores na calda, considere a execução retratada na Figura 7.21. As taxas de utilização da memória para SUB-CONV e SUB-CALDA foram, respectivamente, de 60.74% e 63.28%, e proporcionaram uma redução no tempo total de execução em segundos de 1184 para 1095. Repetindo o procedimento para ADT-CONV e ADT-CALDA (Figura 7.22), as taxas foram, respectivamente, de 12.19% e 18.12% e proporcionaram uma redução no tempo total de execução de 2829 para 2620 segundos.

Considerando o estado da arte, a configuração obtida supera a melhor configuração fornecida em Mantovani et al (2000), cuja perda é 0.28550 MW, mas é ligeiramente inferior ao melhor resultado de Carreno et al (2008): 0.28019 MW. As reduções nas perdas obtidas, respectivamente, por Mantovani et al (2000), Carreno et al (2008) e nos testes desta seção foram as seguintes: 10.89 %, 12.55 % e 12.30 %.

### **7.3.6 Síntese dos resultados - AG**

Nas Seções 7.3.1 até 7.3.5 foram apresentados os resultados obtidos para cada um dos sistemas teste em separado. O objetivo dessa seção é avaliar esses resultados do ponto de vista global, enfatizando três assuntos específicos: população inicial, capacidade de encontrar as melhores soluções e complexidade computacional.

Ao discutir o tema população inicial deve ser entendido aptidão dos indivíduos da população inicial, que é uma característica diretamente relacionada com a tendência de cada código. De forma direta, uma representação que possui uma tendência

positiva deve gerar uma população inicial com aptidão média maior que a de uma representação sem tendência; e esta, por sua vez, deve originar uma população com aptidão maior que a de uma representação com uma tendência negativa.

A Tabela 7.12 resume as aptidões médias observadas para as seis configurações testadas nos cinco sistemas teste utilizados. Claramente, os melhores resultados são obtidos por Aditiva, confirmando a forte tendência positiva prevista no Capítulo 5. Além disso, os piores resultados são sem dúvida de VC, apresentando a menor média para todos os sistemas testados. O fraco desempenho de VC parece contrariar os resultados do Capítulo 5, quando ela aparece com um desempenho melhor que Subtrativa em relação a tendência do código. Entretanto, veja que naqueles testes foram computados apenas os indivíduos radiais conexos. Assim, é factível associar o fraco desempenho ao grande número de inactíveis representados.

Ignorando Aditiva, a expectativa era que a segunda posição fosse ocupada por Melhorado, pois os testes com Subtrativa no Capítulo 4 indicaram uma leve tendência negativa. Contudo, Subtrativa aparece na segunda posição para os sistemas 16-nós e 94-nós. Para 16-nós, com um espaço de busca reduzido, esse resultado não chega a ser inesperado; afinal, seu espaço de busca compreende somente soluções radiais conexas, o que não se verifica em Melhorado. Entretanto, para 94-nós

**Tabela 7.12 – Resumo das aptidões médias das populações iniciais dos cinco sistemas teste: simulações com o AG Básico.**

Codificação	Aptidões médias da população inicial em dez execuções				
	16-nós	33-nós	70-nós	94-nós	137-nós
VC	0.587570	0.138290	0.051710	0.036960	0.008190
Melhorado	0.861540	0.528110	0.085250	0.045360	0.017310
SUB-CONV	0.956880	0.446650	0.059760	0.148730	0.009530
SUB-CALDA	0.953000	0.439810	0.059600	0.146480	0.009920
ADT-CONV	0.977970	0.898130	0.442920	0.609600	0.339890
ADT-CALDA	0.977480	0.900290	0.430950	0.619730	0.356390

esse efeito não deveria prevalecer.

Analisando as características dos sistemas testados (Tabela 7.1), 94-nós difere de 70-nós e 137-nós, principalmente, pelo maior número de subestações e por apresentar uma distribuição das cargas menos homogênea. A hipótese é que por se tratar do sistema com distribuição de cargas menos homogênea, 94-nós seja o menos aderente à regra que confere melhores aptidões às redes com menores profundidades. Com melhores soluções tendendo a profundidades maiores, Subtrativa seria beneficiada e, em oposição, Aditiva apresentaria perda de desempenho. Esse resultado de fato ocorre, sendo 94-nós o único caso em que Aditiva é superada por Melhorado tomando dados médios.

Considerando o melhor indivíduo da população inicial, os resultados observados nas Tabelas 7.2, 7.4, 7.6, 7.8 e 7.10 indicam que para sistemas de maior porte a melhor solução será sempre a própria configuração em operação, em virtude da explosão combinatória no espaço de busca.

Partindo para os resultados propriamente ditos, considere as Tabelas 7.13 e 7.14, nas quais são ilustradas as melhores soluções obtidas nos testes com o AG Básico em termos absolutos e médios, nesta ordem. O fundo cinza destacando os maiores valores de cada coluna e negrito para os menores indica nas duas tabelas o melhor desempenho dos códigos sequenciais em relação a VC e Melhorado. Essa vantagem é mais destacada sobretudo avaliando os melhores resultados, pois Aditiva apresenta em termos médios uma queda de performance no sistema 94-nós, como comentado nas análises da população inicial.

No comparativo direto entre Subtrativa e Aditiva os resultados confirmam as análises dos Capítulos 4 e 5. Na Tabela 7.14, Aditiva apresenta uma vantagem consistente para 16-nós, 33-nós e 70-nós, sendo superada nos testes com 94-

**Tabela 7.13 – Resumo das melhores aptidões obtidas nos testes do AG Básico para os cinco sistemas testados.**

Codificação	Melhor solução				
	16-nós	33-nós	70-nós	94-nós	137-nós
VC	0.983605	0.962035	0.957083	0.982936	0.984886
Melhorado	0.984190	0.965165	0.957967	0.984056	0.985010
SUB-CONV	0.984190	0.965165	0.958260	0.984065	0.985107
SUB-CALDA	0.984190	0.965165	0.958282	0.984065	0.985129
ADT-CONV	0.984190	0.965165	0.958260	0.984065	0.985057
ADT-CALDA	0.984190	0.965165	0.958282	0.984065	0.985057

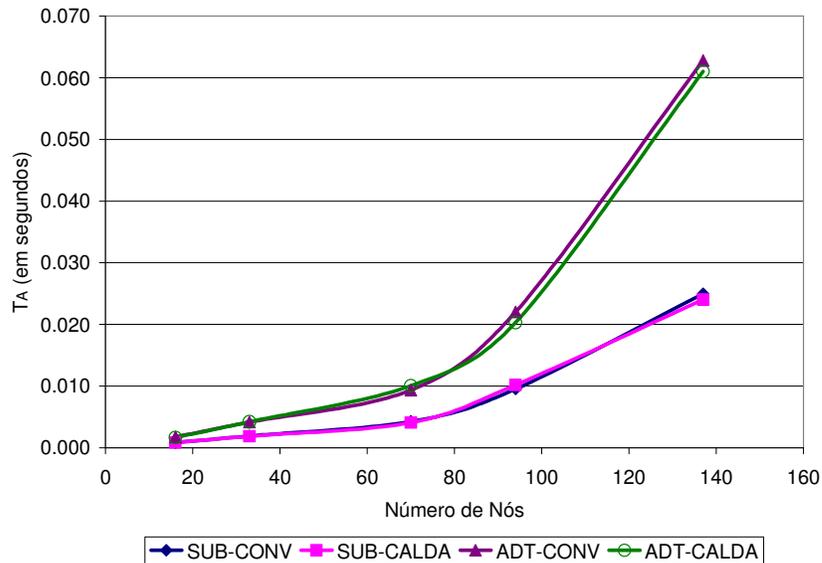
**Tabela 7.14 – Resumo das aptidões médias obtidas nos testes do AG Básico para os cinco sistemas testados.**

Codificação	Média em 10 execuções				
	16-nós	33-nós	70-nós	94-nós	137-nós
VC	0.983121	0.953188	0.954230	0.982553	0.983645
Melhorado	0.984190	0.964337	0.957521	0.983783	0.984353
SUB-CONV	0.984190	0.964837	0.957196	0.983844	0.985025
SUB-CALDA	0.984190	0.964745	0.957992	0.983897	0.985072
ADT-CONV	0.984190	0.965130	0.957985	0.983082	0.984984
ADT-CALDA	0.984190	0.965165	0.958064	0.983241	0.984989

nós e 137-nós. Isto é, a forte tendência positiva de Aditiva prevalece nos sistemas menores, uma vez que eles possuem igualmente menores espaços de busca das soluções. Todavia, a medida que os sistemas se tornam maiores, esse efeito é mitigado e passam a prevalecer os problemas detectados no Capítulo 5 relativos a localidade e hereditariedade de Aditiva.

Em relação aos operadores genéticos, observa-se uma vantagem consistente do operador na calda, válida tanto para Subtrativa como para Aditiva. Porém, em geral, provocando diferenças pouco significativas.

Para finalizar, apresenta-se no gráfico da Figura 7.23 a evolução observada nos tempos de avaliação por indivíduo ( $T_A$ ) para os cinco sistemas testados. Esse valor pode ser utilizado como uma medida aproximada da complexidade dos códigos sequenciais e evidencia a maior complexidade de Aditiva.



**Figura 7.23 – Evolução observada no tempo de avaliação por indivíduo ( $T_A$ ) em função do número de nós do sistema testado.**

## 7.4 Testes com o AGS

Como discutido na Seção 7.2, os testes com o AGS foram divididos em três conjuntos. Em uma primeira etapa, as melhores heurísticas para cada código são determinadas; em seguida, é implementado um comparativo entre as estratégias de controle de  $N$  por larguras iguais e proporcionais; e, por fim, algumas simulações são levadas a efeito no sentido de determinar qual o melhor ajuste do número de patamares de  $N$ .

Considerando que o AGS é destinado principalmente aos sistemas de maior porte, os testes desta seção se restringem aos sistemas 70-nós, 94-nós e 137-nós. Novamente, o enfoque é na Reconfiguração, com objetivo de minimizar perdas, utiliza-se fator de demanda unitário e carga modelada por potência constante. No AGS, adotou-se seleção de indivíduos por roleta; inicialização da população randômica, entretanto, incluindo a configuração em operação; e somente operadores genéticos na calda. Os demais parâmetros foram os mesmos usados nos testes das Seções 7.3.3, 7.3.4 e 7.3.5.

#### 7.4.1 Heurísticas: N+, N-, I+ e I-

Nesta seção, são simuladas combinações de SUB-CALDA e ADT-CALDA com as quatro heurísticas formuladas no Capítulo 6, resultando em oito ajustes:

- SUB-CALDA com N+ (SUB-CALDA/N+),
- SUB-CALDA com N- (SUB-CALDA/N-),
- SUB-CALDA com I+ (SUB-CALDA/I+),
- SUB-CALDA com I- (SUB-CALDA/I-),
- ADT-CALDA com N+ (ADT-CALDA/N+),
- ADT-CALDA com N- (ADT-CALDA/N-),
- ADT-CALDA com I+ (ADT-CALDA/I+) e
- ADT-CALDA com I- (ADT-CALDA/I-).

Adotam-se como referência os resultados obtidos anteriormente com o AG Básico: SUB-CALDA e ADT-CALDA.

Em todos os testes, adota-se a mesma metodologia da Seção 7.3. Na qual, cada configuração é executada 10 vezes, sendo apresentados os melhores resultados e a média obtida. Além disso, estipula-se controle de  $N$  por larguras iguais, com  $N_0 = 0.33$ ,  $N_{\text{final}} = 1.00$  e  $n_{\text{pat}} = 3$ . Os resultados obtidos aplicando o AGS na Reconfiguração de 70-nós, 94-nós e 137-nós são expostos, respectivamente, nas Tabelas 7.15, 7.16 e 7.17.

Restringindo a análise à Subtrativa, se observa que quando ela é combinada a N- ou I+ o desempenho do AGS tende a ser melhor que o do AG Básico. O efeito é mais perceptível nos valores médios, o que indica que o AGS tende a ser mais robusto que o AG Básico.

No comparativo direto entre as heurísticas consideradas adequadas ao código Subtrativo, N- obteve melhores resultados quando aplicada na Reconfiguração

**Tabela 7.15 – Resultados obtidos com o AGS combinando SUB-CALDA e ADT-CALDA às heurísticas N+, N-, I+ e I- para o sistema de 70-nós:  $N$  com larguras iguais e  $n_{pat} = 3$ .**

Codificação	Melhor solução			Média em 10 execuções			$T_C$ (s)	$T_A$ (s)
	Aptidão	Geração	Tempo (s)	Aptidão	Geração	Tempo (s)		
SUB-CALDA	0.958282	83.0	26.0	0.957992	50.7	28.5	14.450	0.004
SUB-CALDA/N+	0.957990	97.0	39.0	0.957080	93.1	38.2	35.564	0.005
SUB-CALDA/N-	0.958288	30.0	32.0	0.958010	38.5	34.2	13.167	0.005
SUB-CALDA/I+	0.958145	18.0	37.0	0.957953	34.1	37.3	12.719	0.005
SUB-CALDA/I-	0.957549	91.0	35.0	0.955956	93.5	36.6	34.221	0.005
ADT-CALDA	0.958282	77.0	61.0	0.958064	42.8	70.4	30.131	0.010
ADT-CALDA/N+	0.958102	12.0	112.0	0.958073	48.6	109.0	52.974	0.016
ADT-CALDA/N-	0.956095	99.0	62.0	0.955509	91.8	69.1	63.434	0.010
ADT-CALDA/I+	0.957962	98.0	68.0	0.957638	91.1	74.1	67.505	0.011
ADT-CALDA/I-	0.958102	67.0	89.0	0.958057	35.9	106.3	38.162	0.015

**Tabela 7.16 – Resultados obtidos com o AGS combinando SUB-CALDA e ADT-CALDA às heurísticas N+, N-, I+ e I- para o sistema de 94-nós:  $N$  com larguras iguais e  $n_{pat} = 3$ .**

Codificação	Melhor solução			Média em 10 execuções			$T_C$ (s)	$T_A$ (s)
	Aptidão	Geração	Tempo (s)	Aptidão	Geração	Tempo (s)		
SUB-CALDA	0.984065	85.0	178.0	0.983897	119.8	203.8	122.076	0.010
SUB-CALDA/N+	0.984039	195.0	240.0	0.983592	190.0	222.5	211.375	0.011
SUB-CALDA/N-	0.984065	144.0	250.0	0.984065	89.3	314.1	140.246	0.016
SUB-CALDA/I+	0.984065	22.0	293.0	0.984065	44.3	352.9	78.167	0.018
SUB-CALDA/I-	0.984056	197.0	235.0	0.983709	193.8	236.1	228.781	0.012
ADT-CALDA	0.984065	87.0	342.0	0.983241	85.0	406.3	172.678	0.020
ADT-CALDA/N+	0.984040	132.0	469.0	0.983791	114.9	481.6	276.679	0.024
ADT-CALDA/N-	0.983270	178.0	387.0	0.982370	177.3	350.5	310.718	0.018
ADT-CALDA/I+	0.983431	183.0	315.0	0.982533	159.6	343.5	274.113	0.017
ADT-CALDA/I-	0.984065	179.0	420.0	0.984062	157.7	460.1	362.789	0.023

de 70-nós e 94-nós, sendo superada por I+ no teste com 137-nós. Houve melhoria da solução ótima em relação aos resultados da Seção 7.3 nos sistemas 70-nós e 137-nós: de 0.958282 para 0.958288 e de 0.985131 para 0.985235, nesta ordem. Essas novas soluções ótimas implicam, respectivamente, 0.20306 MW e 0.28019 MW de perdas ativas.

Tratando de convergência, os resultados não foram homogêneos. Para 70-nós, houve redução de  $T_C$  tanto em SUB-CALDA/N- como SUB-CALDA/I+

**Tabela 7.17 – Resultados obtidos com o AGS combinando SUB-CALDA e ADT-CALDA às heurísticas N+, N-, I+ e I- para o sistema de 137-nós:  $N$  com larguras iguais e  $n_{pat} = 3$ .**

Codificação	Melhor solução			Média em 10 execuções			$T_C$ (s)	$T_A$ (s)
	Aptidão	Geração	Tempo (s)	Aptidão	Geração	Tempo (s)		
SUB-CALDA	0.985131	<b>115.0</b>	<b>780.0</b>	0.985059	<b>122.8</b>	<b>833.0</b>	<b>255.731</b>	<b>0.014</b>
SUB-CALDA/N+	<b>0.985122</b>	347.0	966.0	0.985045	375.8	1148.0	1078.546	0.019
SUB-CALDA/N-	<b>0.985122</b>	159.0	1208.0	<b>0.984885</b>	164.3	1368.7	562.194	0.023
SUB-CALDA/I+	0.985235	148.0	1684.0	0.985127	158.0	1547.0	611.065	0.026
SUB-CALDA/I-	0.985131	397.0	1477.0	0.985024	368.0	1235.6	1136.752	0.021
ADT-CALDA	0.984990	<b>119.0</b>	2325.0	0.984986	<b>73.9</b>	2676.7	<b>494.520</b>	0.045
ADT-CALDA/N+	0.985012	253.0	2823.0	0.984783	214.7	2866.0	1538.326	0.048
ADT-CALDA/N-	0.983471	356.0	<b>1734.0</b>	0.983407	331.1	<b>1974.1</b>	1634.061	<b>0.033</b>
ADT-CALDA/I+	<b>0.983335</b>	366.0	2065.0	<b>0.983315</b>	329.9	2016.8	1663.356	0.034
ADT-CALDA/I-	0.985235	356.0	2553.0	0.985214	305.6	2658.0	2030.712	0.044

quando comparado com SUB-CALDA: de 14.450 segundos para 13.167 e 12.719 segundos. Em 94-nós, apenas SUB-CALDA/I+ apresentou melhoria, saindo de 122.076 para 78.167 segundos. E, no sistema 137-nós, o efeito nas duas heurísticas foi aumentar  $T_C$ , com uma variação superior a 100%: de 255.731 para 562.194 segundos, no melhor caso. Esse resultado não se verifica em  $T_A$ , onde as variações foram sempre inferiores a 50%.

Considere agora os resultados obtidos por Aditiva quando combinada a cada uma das heurísticas descritas. Reforçando a relação de dualidade com Subtrativa, Aditiva obteve seus melhores resultados quando combinada a N+ e I-. Isto é, o oposto de Subtrativa nos três sistemas testados. Mais uma vez, o efeito se mostra mais significativo nos valores médios que nas melhores execuções, indicando melhoria na robustez do algoritmo. Embora tenha ocorrido melhoria na solução ótima da Seção 7.3 para o sistema 137-nós: de 0.984990 para 0.985235.

Comparando especificamente N+ com I-, a vantagem aparente é de I- visto que obteve resultados melhores em 94-nós e 137-nós. Na verdade, tanto N- combinada ao código Subtrativo, como N+ combinada a Aditiva tiveram resultados

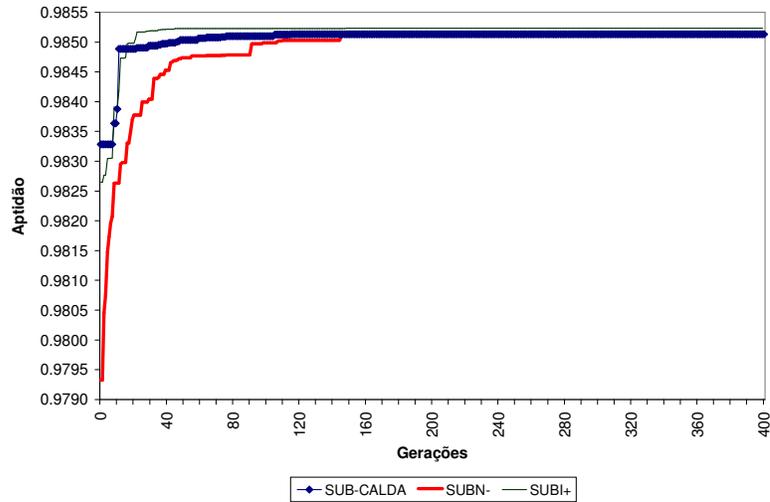
inferiores nos testes com 137-nós, sugerindo que as heurísticas baseadas em correntes são mais adequadas para sistemas maiores.

Avaliando a convergência do AGS com Aditiva, os resultados foram consistentes ao indicar aumento de  $T_C$ . O pior caso ocorreu para 137-nós, sendo a variação da ordem de 300%. Novamente, esse resultado não se verifica em  $T_A$ , onde a maior variação foi de 60%.

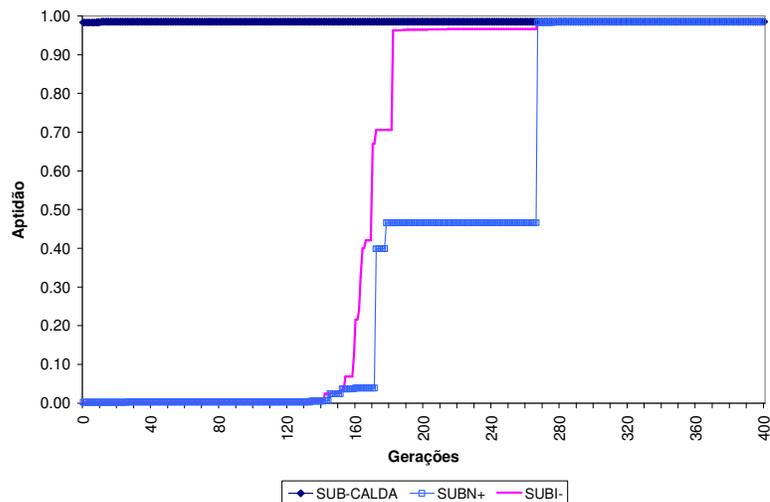
Tratando-se do desempenho do melhor ajuste de Subtrativa em relação ao melhor ajuste de Aditiva, os resultados são muito próximos. Para 70-nós, SUB-CALDA/N- vence na melhor solução (0.958288), mas é de ADT-CALDA/N+ o melhor desempenho na média: 0.958073. Com 94-nós ocorre um empate em termos de melhor solução, entretanto, SUB-CALDA/N- é a melhor na média, obtendo a solução ótima em 100% das execuções: 0.984065. Por fim, para 137-nós, ocorre novo empate na melhor solução; dessa vez entre SUB-CALDA/I+ e ADT-CALDA/I-, que obtiveram uma solução com aptidão 0.985235. Contudo, na média a vantagem é de ADT-CALDA/I- com 0.985214. Para não deixar de escolher um melhor ajuste, optou-se por privilegiar os resultados de 137-nós escolhendo para os testes da subseção 7.4.2 os seguintes ajustes: SUB-CALDA/I+ e ADT-CALDA/I-.

Antes de partir para a subseção 7.4.2, considere a evolução do AGS na Reconfiguração do sistema 137-nós. As Figuras 7.24 e 7.25 são relativas ao código Subtrativo, enquanto as Figuras 7.26 e 7.27 são relativas ao código Aditivo. Essas figuras evidenciam, principalmente, como a escolha da heurística errada para determinada representação pode comprometer o desempenho do AGS. Esse é o caso ilustrado nas Figuras 7.25 e 7.27.

Adicionalmente, observe que o ganho de desempenho com o AGS se verifica na maior parte das gerações com SUB-CALDA/I+ e em todas as gerações com



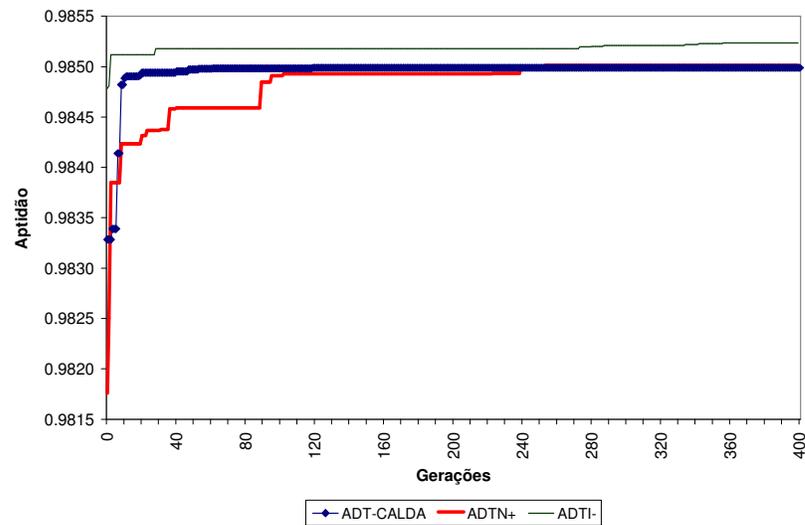
**Figura 7.24 – Comparativo das melhores execuções no sistema 137-nós por geração: SUB-CALDA com AG (Seção 7.3) contra AGS e heurísticas indicadas para SUB (N- e I+).**



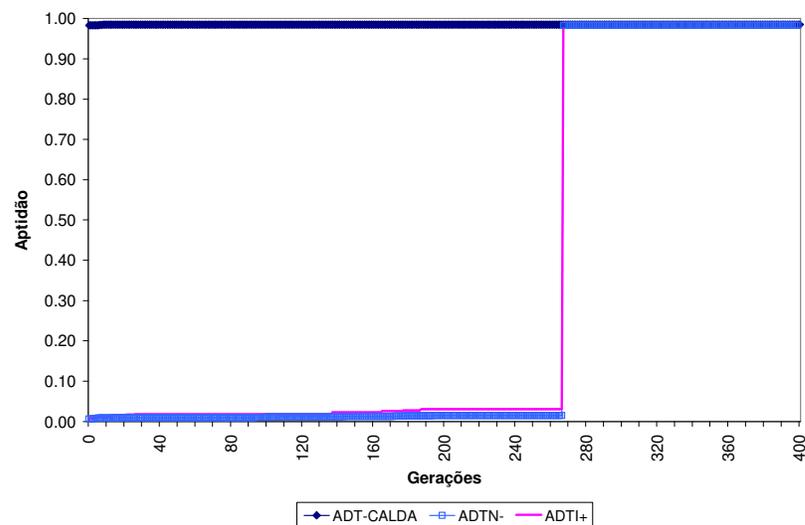
**Figura 7.25 – Comparativo das melhores execuções no sistema 137-nós por geração: SUB-CALDA com AG (Seção 7.3) contra AGS e heurísticas não indicadas para SUB (N+ e I-).**

ADT-CALDA/I-. Isto é, além das diferenças de desempenho avaliadas no final da execução do algoritmo (Tabelas 7.15 até 7.17), o AGS indica capacidade de obter configurações de rede ótimas, ou sub-ótimas, em uma velocidade maior que o AG Básico. Tal característica pode representar uma vantagem na Reconfiguração de sistemas de grande porte, sendo este um resultado aderente aos objetivos formulados no Capítulo 1.

#### 7.4.2 Controle do Fator de Busca $N$ : Larguras Iguais e Proporcionais



**Figura 7.26 – Comparativo das melhores execuções no sistema 137-nós por geração: ADT-CALDA com AG (Seção 7.3) contra AGS e heurísticas indicadas para ADT (N+ e I-).**



**Figura 7.27 – Comparativo das melhores execuções no sistema 137-nós por geração: ADT-CALDA com AG (Seção 7.3) contra AGS e heurísticas não indicadas para ADT (N- e I+).**

Até a presente seção, os resultados obtidos pelo AGS com SUB-CALDA/I+ e ADT-CALDA/I- e controle de  $N$  por larguras iguais foram superiores aos apresentados pelo AG Básico com SUB-CALDA e ADT-CALDA e estes, por sua vez, foram significativamente melhores que os obtidos pelo AG Básico com as referências VC e Melhorado.

O objetivo dessa seção é determinar se o controle de  $N$  por larguras

proporcionais (PROP) é capaz de melhorar ainda mais a busca em relação ao controle por larguras iguais (IGUAL). Para garantir igualdade de condições no comparativo, os dois ajustes utilizam os mesmos parâmetros para o módulo de controle de  $N$ :  $N_0 = 0.33$ ,  $N_{\text{final}} = 1.00$  e  $n_{\text{pat}} = 3$ .

Considere as Tabelas 7.18, 7.19 e 7.20 associadas, respectivamente, aos resultados obtidos na Reconfiguração dos sistemas 70-nós, 94-nós e 137-nós. Em relação ao código Subtrativo, observe que os resultados são muito próximos. Avaliando a melhor solução encontrada, ocorre um empate para 70-nós e 94-nós, enquanto IGUAL apresenta pequena vantagem em 137-nós: 0.985235 contra 0.985230. Tomando como base as aptidões médias, IGUAL apresenta melhor desempenho em 70-nós, mas é

**Tabela 7.18 – Resultados obtidos com o AGS combinando SUB-CALDA/I+ e ADT-CALDA/I- às lógicas de controle de  $N$  por larguras iguais e proporcionais: sistema de 70-nós e  $n_{\text{pat}} = 3$ .**

Codificação	Melhor solução			Média em 10 execuções			$T_C$ (s)	$T_A$ (s)
	Aptidão	Geração	Tempo (s)	Aptidão	Geração	Tempo (s)		
IGUAL/SUB-CALDA/I+	0.958145	18.0	37.0	0.957953	34.1	37.3	12.719	0.005
PROP/SUB-CALDA/I+	0.958145	20.0	32.0	0.957946	37.0	34.6	12.802	0.005
IGUAL/ADT-CALDA/I-	0.958102	67.0	89.0	0.958057	35.9	106.3	38.162	0.015
PROP/ADT-CALDA/I-	0.958102	69.0	106.0	0.958051	37.6	106.3	39.969	0.015

**Tabela 7.19 – Resultados obtidos com o AGS combinando SUB-CALDA/I+ e ADT-CALDA/I- às lógicas de controle de  $N$  por larguras iguais e proporcionais: sistema de 94-nós e  $n_{\text{pat}} = 3$ .**

Codificação	Melhor solução			Média em 10 execuções			$T_C$ (s)	$T_A$ (s)
	Aptidão	Geração	Tempo (s)	Aptidão	Geração	Tempo (s)		
IGUAL/SUB-CALDA/I+	0.984065	22.0	293.0	0.984065	44.3	352.9	78.167	0.018
PROP/SUB-CALDA/I+	0.984065	43.0	228.0	0.984065	62.3	292.7	91.176	0.015
IGUAL/ADT-CALDA/I-	0.984065	179.0	420.0	0.984062	157.7	460.1	362.789	0.023
PROP/ADT-CALDA/I-	0.984065	138.0	393.0	0.984062	123.7	452.9	280.119	0.023

**Tabela 7.20 – Resultados obtidos com o AGS combinando SUB-CALDA/I+ e ADT-CALDA/I- às lógicas de controle de  $N$  por larguras iguais e proporcionais: sistema de 137-nós e  $n_{\text{pat}} = 3$ .**

Codificação	Melhor solução			Média em 10 execuções			$T_C$ (s)	$T_A$ (s)
	Aptidão	Geração	Tempo (s)	Aptidão	Geração	Tempo (s)		
IGUAL/SUB-CALDA/I+	0.985235	148.0	1684.0	0.985127	158.0	1547.0	611.065	0.026
PROP/SUB-CALDA/I+	0.985230	122.0	1162.0	0.985138	129.2	1291.2	417.058	0.022
IGUAL/ADT-CALDA/I-	0.985235	356.0	2553.0	0.985214	305.6	2658.0	2030.712	0.044
PROP/ADT-CALDA/I-	0.985235	274.0	2060.0	0.985206	229.8	2550.3	1465.147	0.043

superado por PROP em 94-nós e 137-nós. Porém, as diferenças são sempre pequenas; ao ponto de estarem além da sexta casa decimal para 94-nós (Tabela 7.19).

Analisando os tempos de execução de Subtrativa, a escolha de IGUAL ou PROP indica um efeito mais significativo nos resultados. Ao optar por PROP nos três sistemas testados, houve redução no tempo total de execução e, conseqüentemente, em  $T_A$ . Em 137-nós, essa redução foi da ordem de 32%.

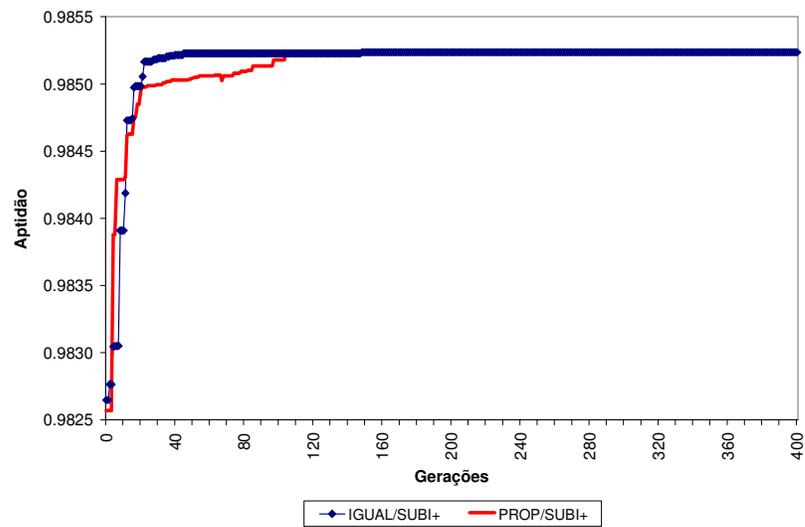
Com relação ao código Aditivo, os resultados também são muito parecidos. Verifica-se um empate triplo em termos de melhor solução obtida. Enquanto, considerando dados médios, a vantagem é de IGUAL em 70-nós e 137-nós, empatando em 94-nós. A exemplo do que ocorreu com Subtrativa, o efeito mais significativo foi no tempo de avaliação por indivíduo, PROP permitindo uma redução da ordem de 28% nos testes com 137-nós.

Para ilustrar a evolução por geração de cada uma das estratégias de controle, considere nas Figuras 7.28 e 7.29 os resultados obtidos para 137-nós, respectivamente, para Subtrativa e Aditiva. Como discutido, não se observa uma vantagem consistente em Subtrativa tampouco em Aditiva. Para não deixar de apontar o melhor ajuste dessa seção, utiliza-se na subseção 7.4.3 o controle PROP.

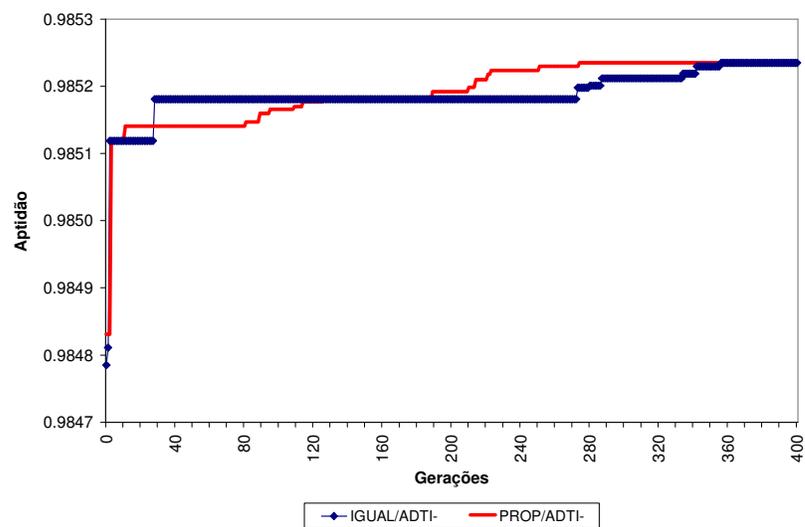
### 7.4.3 Número de Patamares de $N$ : 2, 3, 4 e 5

Para finalizar os testes com o AGS, implementam-se nesta seção algumas simulações variando o número de patamares do fator de busca  $N$ , de 2 até 5. Os testes são levados a efeito com o código Subtrativo combinado aos operadores na calda, heurística I+ e controle de  $N$  por larguras proporcionais. A configuração de Aditiva é a mesma, diferindo apenas na heurística escolhida: I-.

Adota-se  $N_{\text{final}} = 1.00$  em todos os ajustes, alterando  $N_0$  em função do



**Figura 7.28 – Comparativo por geração das melhores execuções de SUB-CALDA/I+ (SUBI+) no sistema 137-nós: larguras iguais (IGUAL) contra larguras proporcionais (PROP).**



**Figura 7.29 – Comparativo por geração das melhores execuções de ADT-CALDA/I- (ADTI-) no sistema 137-nós: larguras iguais (IGUAL) contra larguras proporcionais (PROP).**

número de patamares. Para  $n_{pat}$  no intervalo de 2 até 5, são usados os seguintes valores de  $N_0$ : 0.50, 0.33, 0.25 e 0.20, nesta ordem. Novamente, são utilizados como sistemas de teste 70-nós, 94-nós e 137-nós, sendo os resultados apresentados, respectivamente, nas Tabelas 7.21, 7.22 e 7.23. O procedimento usado é o mesmo de antes: são realizadas 10 execuções para cada um dos ajustes e apresentam-se nas tabelas os dados da melhor execução e da média.

**Tabela 7.21 – Resultados obtidos com o AGS combinando SUB-CALDA/I+ e ADT-CALDA/I-, controle de  $N$  por larguras proporcionais e variando o número de patamares 2-5: sistema 70-nós.**

Codificação	Melhor solução			Média em 10 execuções			$T_c$ (s)	$T_A$ (s)
	Aptidão	Geração	Tempo (s)	Aptidão	Geração	Tempo (s)		
SUBI+2PAT	<b>0.958145</b>	92.0	29.0	0.957769	44.7	31.0	13.857	<b>0.004</b>
SUBI+3PAT	<b>0.958145</b>	20.0	32.0	0.957946	37.0	34.6	12.802	0.005
SUBI+4PAT	0.958288	18.0	33.0	0.958166	32.4	35.4	11.470	0.005
SUBI+5PAT	0.958288	<b>15.0</b>	30.0	0.958232	<b>28.1</b>	35.4	<b>9.947</b>	0.005
ADTI-2PAT	0.958102	37.0	<b>74.0</b>	0.958080	42.3	<b>86.6</b>	36.632	<b>0.012</b>
ADTI-3PAT	0.958102	69.0	106.0	<b>0.958051</b>	37.6	106.3	39.969	0.015
ADTI-4PAT	0.958102	89.0	104.0	0.958065	53.4	105.8	56.497	0.015
ADTI-5PAT	0.958102	69.0	98.0	<b>0.958051</b>	<b>34.2</b>	105.5	<b>36.081</b>	0.015

**Tabela 7.22 – Resultados obtidos com o AGS combinando SUB-CALDA/I+ e ADT-CALDA/I-, controle de  $N$  por larguras proporcionais e variando o número de patamares 2-5: sistema 94-nós.**

Codificação	Melhor solução			Média em 10 execuções			$T_c$ (s)	$T_A$ (s)
	Aptidão	Geração	Tempo (s)	Aptidão	Geração	Tempo (s)		
SUBI+2PAT	0.984065	89.0	<b>222.0</b>	0.983912	65.9	255.2	84.088	<b>0.013</b>
SUBI+3PAT	0.984065	43.0	228.0	0.984065	62.3	292.7	91.176	0.015
SUBI+4PAT	0.984065	23.0	278.0	0.984065	13.7	343.8	23.550	0.017
SUBI+5PAT	0.984065	<b>15.0</b>	319.0	0.984065	<b>9.9</b>	358.7	<b>17.756</b>	<b>0.018</b>
ADTI-2PAT	0.984065	<b>75.0</b>	<b>327.0</b>	0.984064	<b>88.7</b>	<b>434.8</b>	<b>192.834</b>	<b>0.022</b>
ADTI-3PAT	0.984065	138.0	393.0	0.984062	123.7	452.9	280.119	0.023
ADTI-4PAT	0.984065	134.0	405.0	<b>0.984060</b>	138.4	467.6	323.579	0.023
ADTI-5PAT	0.984065	151.0	449.0	0.984062	135.9	477.5	324.461	0.024

Os resultados obtidos com Subtrativa indicam uma melhoria consistente no desempenho do AGS quando o número de patamares considerados cresce. Em termos de melhor execução, as soluções ótimas foram obtidas com 4 ou 5 patamares para 70-nós, com todos os ajustes para 94-nós e com 4 ou 5 patamares para 137-nós. Tomando resultados médios, a melhor performance ocorre sempre quando  $n_{pat} = 5$ , o que confirma a hipótese levantada.

Considerando a complexidade, observa-se um aumento de  $T_A$  a medida que cresce o número de patamares. No entanto, esse aumento no tempo total de execução é acompanhado por um aumento na velocidade de convergência, de modo que

**Tabela 7.23 – Resultados obtidos com o AGS combinando SUB-CALDA/I+ e ADT-CALDA/I–, controle de  $N$  por larguras proporcionais e variando o número de patamares 2-5: sistema 134-nós.**

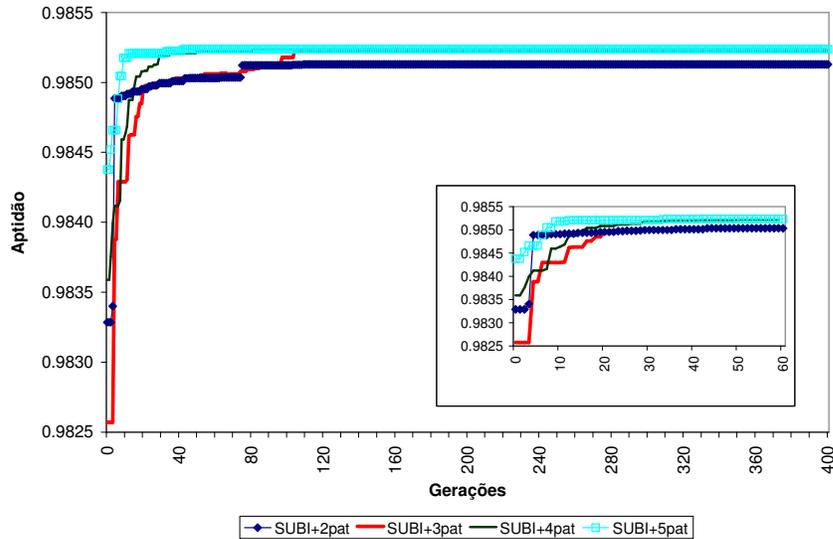
Codificação	Melhor solução			Média em 10 execuções			$T_C$ (s)	$T_A$ (s)
	Aptidão	Geração	Tempo (s)	Aptidão	Geração	Tempo (s)		
SUBI+2PAT	<b>0.985129</b>	109.0	<b>1024.0</b>	<b>0.985078</b>	101.7	<b>1134.4</b>	288.421	<b>0.019</b>
SUBI+3PAT	0.985230	122.0	1162.0	0.985138	129.2	1291.2	417.058	0.022
SUBI+4PAT	0.985235	81.0	1556.0	0.985150	145.6	1380.4	502.466	0.023
SUBI+5PAT	0.985235	<b>42.0</b>	<b>1797.0</b>	0.985206	<b>36.7</b>	<b>1468.7</b>	<b>134.753</b>	<b>0.024</b>
ADTI-2PAT	0.985235	<b>200.0</b>	2318.0	0.985198	<b>176.6</b>	2661.9	<b>1175.229</b>	0.044
ADTI-3PAT	0.985235	274.0	<b>2060.0</b>	0.985206	229.8	<b>2550.3</b>	1465.147	<b>0.043</b>
ADTI-4PAT	0.985235	261.0	2325.0	0.985199	246.1	2653.0	1632.258	0.044
ADTI-5PAT	0.985235	272.0	2542.0	<b>0.985197</b>	220.2	2750.5	1514.150	0.046

nos três sistemas houve redução de  $T_C$  quando  $n_{pat} = 5$ .

Ilustram-se na Figura 7.30 as curvas com as evoluções de Subtrativa para as melhores execuções em 137-nós quando  $n_{pat}$  varia de 2 até 5. Fica evidente o ganho na velocidade com que o algoritmo obtém as melhores soluções quando o ajuste de  $n_{pat}$  é maior. Principalmente, na miniatura posicionada no canto inferior direito, na qual se destaca a evolução nas primeiras gerações do AGS. Obviamente, deve haver um limite para que o aumento do ajuste de  $n_{pat}$  deixe de ser vantajoso para o desempenho. Essa é mais uma investigação sugerida como um possível desdobramento do presente trabalho.

Partindo para as análises com Aditiva, os resultados são diferentes do que foi observado com Subtrativa. Olhando os melhores resultados, Aditiva se mostra imune ao ajuste do número de patamares, obtendo uma mesma resposta seja qual for o ajuste dos patamares e seja qual for o sistema testado.

Nos valores médios, a maior aptidão ocorre com 2 patamares para 70-nós e 94-nós, e com 3 patamares em 137-nós. Logo, o oposto do que foi observado com Subtrativa, em que os melhores resultados foram obtidos com  $n_{pat} = 5$ . Contudo, o resultado mais surpreendente é o intervalo entre a pior e a melhor média encontrada, que é significativamente mais estreito para Aditiva. Por exemplo, multiplicando as



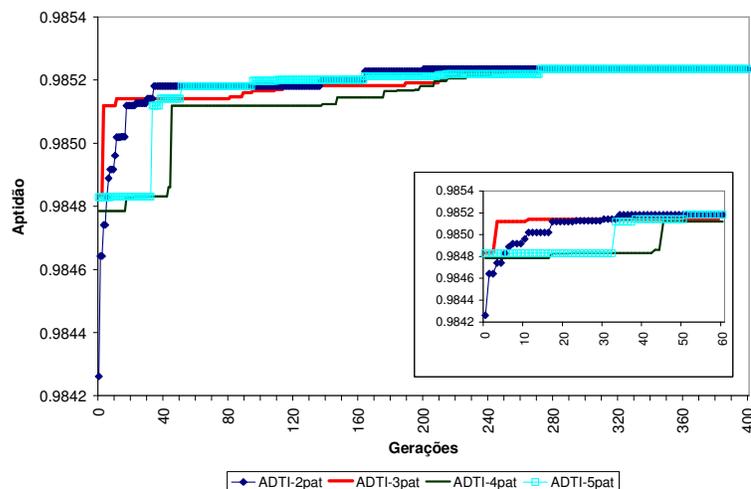
**Figura 7.30 – Evolução por geração das melhores execuções de SUB-CALDA/I+ com largura proporcional para o sistema 137-nós: 2, 3, 4 e 5 patamares do fator de busca  $N$ .**

aptidões obtidas por  $10^6$ , as diferenças são as seguintes: 29 contra 463 para 70-nós, 4 contra 153 para 94-nós e 9 contra 463 para 137-nós.

A indicação é que a forte tendência do código Aditivo atenua o efeito de  $n_{pat}$ . Porém, como indicaram as Tabelas 7.15 até 7.17, esse resultado não se estende ao algoritmo propriamente dito. Resumindo, o AGS com Aditiva se mostrou extremamente robusto e pouco dependente do parâmetro  $n_{pat}$ .

Analisando a complexidade do código Aditivo, os resultados reforçam a aparente imunidade em relação a  $n_{pat}$  observada nas aptidões. Verifica-se um acréscimo em  $T_A$ , mas sem um ganho associado na velocidade de convergência do algoritmo.

Ilustram-se na Figura 7.31 as curvas com as evoluções de Aditiva nas melhores execuções em 137-nós com  $n_{pat}$  variando de 2 até 5. Fica evidente a diferença em relação a Subtrativa, pois são os ajustes com 2 e 3 patamares que chegam com maior velocidade nas melhores soluções. A miniatura no canto inferior direito destaca esse resultado, enfatizando as primeiras gerações do AGS.



**Figura 7.31 – Evolução por geração das melhores execuções de ADT-CALDA/I– com largura proporcional para o sistema 137-nós: 2, 3, 4 e 5 patamares do fator de busca  $N$ .**

#### 7.4.4 Síntese dos resultados - AGS

Diferente da Seção 7.3, as análises dos resultados do AGS já foram realizadas considerando os sistemas teste conjuntamente. Desse modo, objetiva-se resumir as subseções 7.4.1 até 7.4.3, em que foram avaliadas, respectivamente, as heurísticas  $N+$ ,  $N-$ ,  $I+$  e  $I-$ , quanto a sua adequação aos códigos Subtrativo e Aditivo; os controles de  $N$  por larguras iguais e proporcionais; e os diferentes ajustes de  $n_{pat}$ .

Os resultados dos testes da subseção 7.4.1 são resumidos na Tabela 7.24 e confirmam as previsões do Capítulo 6:  $N-$  e  $I+$  se mostraram adequadas ao código Subtrativo, enquanto  $N+$  e  $I-$  se mostraram adequadas à Aditiva. Avaliando as melhores execuções, houve melhoria na solução ótima em relação aos testes da Seção 7.3 no sistema 70-nós, utilizando o AGS combinado ao código Subtrativo; e no sistema 137-nós, com o AGS combinado tanto à Subtrativa quanto à Aditiva. Resultados ainda melhores foram obtidos quando a análise é feita tomando como base os dados médios: todos os ajustes em que heurística e representação adequados foram utilizados superaram os resultados do AG.

**Tabela 7.24 – Resumo dos resultados obtidos pelo AGS nos testes da subseção 7.4.1 para os sistemas 70-nós, 94-nós e 137-nós: melhor aptidão e média em 10 execuções.**

Codificação	Melhor solução			Média em 10 execuções		
	70-nós	94-nós	137-nós	70-nós	94-nós	137-nós
SUB-CALDA	0.958282	0.984065	0.985131	0.957992	0.983897	0.985059
SUB-CALDA/N+	0.957990	<b>0.984039</b>	<b>0.985122</b>	0.957080	<b>0.983592</b>	0.985045
SUB-CALDA/N-	0.958288	0.984065	<b>0.985122</b>	0.958010	0.984065	<b>0.984885</b>
SUB-CALDA/I+	0.958145	0.984065	0.985235	0.957953	0.984065	0.985127
SUB-CALDA/I-	<b>0.957549</b>	0.984056	0.985131	<b>0.955956</b>	0.983709	0.985024
ADT-CALDA	0.958282	0.984065	0.984990	0.958064	0.983241	0.984986
ADT-CALDA/N+	0.958102	0.984040	0.985012	0.958073	0.983791	0.984783
ADT-CALDA/N-	<b>0.956095</b>	<b>0.983270</b>	0.983471	<b>0.955509</b>	<b>0.982370</b>	0.983407
ADT-CALDA/I+	0.957962	0.983431	<b>0.983335</b>	0.957638	0.982533	<b>0.983315</b>
ADT-CALDA/I-	0.958102	0.984065	0.985235	0.958057	0.984062	0.985214

Com base nos resultados da subseção 7.4.1, optou-se por restringir as simulações da subseção 7.4.2 aos ajustes SUB-CALDA/I+ e ADT-CALDA/I-, variando apenas o controle de  $N$  entre larguras iguais e proporcionais. Os resultados apontaram desempenhos similares em termos de aptidões obtidas, tanto na melhor execução quanto na média. Porém, o controle proporcional permitiu redução nos tempos de execução, sendo escolhido como ajuste sugerido para a subseção 7.4.3.

Nas simulações da subseção 7.4.3, foram pesquisados diferentes ajustes do parâmetro  $n_{pat}$ , sendo obtidos resultados significativamente diferentes para Subtrativa e Aditiva. Em síntese, são indicados ajustes maiores de  $n_{pat}$  para Subtrativa, com expectativa de melhoria de performance tanto em aptidões como no tempo de convergência; e ajustes menores de  $n_{pat}$  para Aditiva, apenas com o intuito de evitar o acréscimo de complexidade que ocorreria sem um ganho de performance associado.

Como conclusões dos testes realizados, o AGS se mostrou factível à Configuração de sistemas de distribuição, desde que usado com ajustes adequados dos parâmetros. Recomenda-se o seguinte ajuste para o código Subtrativo: operadores na calda, heurísticas N- ou I+, controle de  $N$  por larguras proporcionais e  $n_{pat} = 5$  ou maior. E, para Aditiva: operadores na calda, heurísticas N+ ou I-, controle de  $N$  por larguras proporcionais e  $n_{pat} = 2$  ou 3.

## Capítulo 8

# Conclusões e Sugestões

---

Neste trabalho, o problema da Configuração de Redes de Distribuição foi formulado para ser resolvido empregando-se AGs. A formulação do problema é geral, englobando simultaneamente os três enfoques clássicos do problema: Reconfiguração de rede para redução de perdas ativas, Restabelecimento de serviço após faltas permanentes e Planejamento da expansão.

Apresenta-se uma função de aptidão que engloba os vários objetivos e restrições do problema por meio de termos normalizados, permitindo mitigar sua dependência em relação às redes estudadas. Também é proposto um procedimento para ajuste automático dos parâmetros dessa função, garantindo uma seletividade entre restrições e objetivos, nesta ordem; bem como, uma estrutura de dados que minimiza o número de chamadas à função de aptidão, chamada de Memória Genética. Contudo, apesar dessa formulação generalizada, o enfoque principal do trabalho é no desenvolvimento de novas técnicas que permitam aos AGs configurar redes de grande porte em um tempo factível. Nesse sentido, os testes implementados se restringem ao enfoque Reconfiguração.

Tendo em mente que a melhoria no modo como as redes são codificadas incide diretamente no desempenho dos AGs, duas técnicas novas de representação das soluções são apresentadas em uma primeira etapa do trabalho: Subtrativa e Aditiva . Elas têm em comum a vantagem de codificarem exclusivamente configurações radiais

conexas; o fato de utilizarem uma lógica de codificação das soluções cuja interpretação é sequencial; e não se restringirem às redes planares. Além disso, tal qual a representação Network Random Keys (ROTHLAUF et al, 2002), elas não requerem operadores genéticos específicos. Ainda assim, variações chamadas de cruzamento na calda e mutação na calda foram formuladas e testadas, obtendo desempenho superior aos operadores convencionais.

Destacando as diferenças entre Subtrativa e Aditiva, sua relação é de dualidade; como se uma fosse o oposto da outra. Isto fica evidente nos testes implementados nos Capítulos 4 e 5 para posicioná-las com base nas métricas espaço, tempo, factibilidade, cobertura, tendência, hereditariedade, localidade, restrição e hibridização. A principal desvantagem comum aos códigos Subtrativo e Aditivo reside na maior complexidade de seus algoritmos de decodificação, porém estruturas de dados e operadores matemáticos que simplificam a decodificação foram propostos: Mapa das Malhas  $L$  e  $L'$ , operadores  $XOR$  e  $XOR'$ , Mapa dos Setores  $R$  e  $R'$  e operadores  $IXOR$  e  $IXOR'$ .

Com base nos resultados obtidos com o AG Básico no Capítulo 7, observa-se que as referidas codificações são efetivas na melhoria do desempenho. Os testes foram levados a efeito em cinco sistemas disponíveis na literatura específica sobre Configuração de Redes e usaram como referência duas codificações tradicionais: Vetor Característico e Vetor Característico Melhorado. Tanto no comparativo direto com estas codificações de referência, como em relação aos resultados apresentados no estado da arte, os testes demonstraram que as codificações sequenciais são promissoras. Adicionalmente, houve uma forte convergência entre as previsões realizadas com base nas métricas clássicas de análise de representações e os resultados efetivamente obtidos, demonstrando sua efetividade.

Embora Subtrativa e Aditiva caracterizem uma contribuição ao estado da arte, considera-se como principal inovação deste trabalho de tese a formulação, implementação e validação do AG Sequencial (AGS). Esse algoritmo tem como pré-requisito a utilização de um código sequencial e se diferencia do AG Básico por combinar um espaço de busca com tamanho controlado a um critério de otimização heurístico.

Os testes com o AGS englobaram a utilização dos códigos Subtrativo e Aditivo combinados a quatro heurísticas de ordenação das arestas, duas filosofias de controle do espaço de busca (fator de busca  $N$ ) e quatro ajustes do número de patamares de  $N$ . Foram utilizados os três sistemas maiores dentre os cinco usados nos testes com o AG Básico, tendo sido observados ganhos consistentes de desempenho mesmo tomando como referência os resultados obtidos anteriormente por Subtrativa e Aditiva. A melhoria de performance foi observada considerando as melhores soluções obtidas, a média de várias execuções e a própria evolução por gerações, sendo mais evidentes nos resultados médios.

Novamente, os resultados dos testes foram aderentes às previsões feitas com base nas métricas de análise de representações; isto é, os testes com o AGS no Capítulo 7 reproduzem as previsões do Capítulo 6. Uma vantagem adicional com relação ao AGS diz respeito a idéia de melhorar a performance de um AG pela manipulação da tendência do código utilizado. Teoricamente, esse princípio pode ser generalizado para qualquer problema de busca de árvores de cobertura em grafos, o que expande significativamente suas possíveis aplicações.

Várias sugestões de pesquisas futuras estão indicadas no texto, sendo as consideradas mais relevantes e algumas adicionais enumeradas a seguir:

- otimizar o cálculo da aptidão dos indivíduos no AG, melhorando o modelo e implementação do algoritmo de fluxo de carga;
- propor um novo operador de mutação para o código Subtrativo no sentido de aprimorar sua localidade, proporcionando uma distância ótima ( $d_{lf} = 2$ ) em uma taxa maior que os 66.67% atuais;
- propor um novo operador de mutação para o código Aditivo no sentido de aprimorar sua localidade, proporcionando uma distância ótima ( $d_{lf} = 2$ ) em uma taxa maior que os 36.5% e 58.1% verificados atualmente com os operadores convencionais e na calda, nesta ordem;
- propor um novo operador de cruzamento para o código Aditivo visando aprimorar sua hereditariedade, proporcionando uma taxa de violações inferior aos 28.5% e 18.4% observados atualmente, respectivamente, para os operadores convencionais e na calda;
- testar o AGS com outras heurísticas além das quatro testadas nesse texto, sobretudo heurísticas de execução, em que não há nenhuma indicação se o aumento na complexidade será compensado pela melhoria na performance;
- testar o AGS com outros controles do fator de busca e com um intervalo maior de patamares, visando aprimorar os ajustes recomendados neste trabalho;
- implementar testes com o AG e AGS com sistemas ainda maiores que os utilizados na tese e, principalmente, testar diferentes distribuições de carga, perfis de demanda e restrições elétricas em uma mesma rede; e
- implementar os devidos ajustes nos algoritmos formulados para testar a formulação generalizada também com os enfoques Restabelecimento e Planejamento.

## Referências Bibliográficas

---

- ANEEL, “Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional – PRODIST”. Revisão 1, 2010.
- AOKI, K, NARA, N., ITOH, M., SATOH, T., KUWABARA, H., “A new algorithm for service restoration in distribution systems”. IEEE Transactions on Power Delivery, Vol. 4, no. 3, pp. 1832-1839, 1989.
- BARAN, M. E., WU, F. F., “Network Reconfiguration in Distribution Systems for Loss Reduction and Load Balancing”. IEEE Transactions on Power Delivery, Vol. 4, No. 2, pp. 1401-1407, 1989.
- BEAN, J. C., “Genetics and Random Keys for Sequencing and Optimization”. Technical Report 92-43, Revised December 1993.
- BOROZAN, V., RAJIEIĆ, D., ACKOVSKI, R. “Minimum Loss Reconfiguration of Unbalanced Distribution Networks”. IEEE Transactions on Power Delivery, Vol. 12, no. 1, pp. 435-441, 1997a.
- BOROZAN, V., RAJAKOVIE, N., “Application Assessments of Distribution Network Minimum Loss Reconfiguration”. IEEE Transactions on Power Delivery, Vol. 12, No. 4, October, pp. 1786-1792, 1997b.
- BOULAXIS, N. G. PAPADOPOULOS, M. P., “Optimal Feeder Routing in Distribution System Planning Using Dynamic Programming Technique and GIS Facilities”, IEEE Transactions on Power Delivery, Vol. 17, pp. 242-247, 2002.

- BRAZ, H. D. M. “Algoritmos Genéticos para Configuração Ótima de Redes de Distribuição de Energia Elétrica”. Dissertação de Mestrado. Universidade Federal de Campina Grande, Campina Grande, Paraíba, Brasil, 2003.
- BRAZ, H. D. M., SOUZA, B. A., ALVES, H. N., “Evolutionary Computation for Planning of Distribution Network Expansion”, Proceedings of the IEEE/PES T&D, São Paulo, Brazil, 2004.
- BRETAS, N. G., DELBEM, A. C. B., CARVALHO, A., “Optimal energy restoration for general distribution systems by genetic algorithms”. Proceedings of International Conference on Power System Technology - POWERCON '98, Vol. 1, pp. 43-47, 1998.
- CARRANO, E. G., SOARES, L. A. E., TAKAHASHI, R. H. C., SALDANHA, R. R., NETO, O. M., “Electric Distribution Network Multiobjective Design Using a Problem-Specific Genetic Algorithm”. IEEE Transactions on Power Delivery, Vol. 21, No. 2, April, pp. 995-1005, 2006.
- CARRENO, E. M., ROMERO, R., PADILHA-FELTRIN, A., “An Efficient Codification to Solve Distribution Network Reconfiguration for Loss Reduction Problem”. IEEE Transactions on Power Systems, Vol. 23, No. 4, pp. 1542-1551, 2008.
- CARVALHO, P.M.S., FERREIRA, L.A.F.M., BARRUNCHO, L.M.F., “Optimization approach to dynamic restoration of distribution systems”. Electrical Power & Energy Systems, no. 29, pp. 222-229, 2007.
- CESPEDES R.G., “New method for the Analysis of Distribution Networks”. IEEE Transactions on Power Delivery, Vol. 5, No. 1, pp. 391-396, 1990.
- CEPEL, “Programa de análise de redes - ANAREDE”, versão 09.06.00, Departamento de Redes Elétricas, Fevereiro, 2010.

- CHANG, C.-F., “Reconfiguration and Capacitor Placement for Loss Reduction of Distribution Systems by Ant Colony Search Algorithm”. IEEE Transactions on Power Systems, Vol. 23, No. 4, pp. 1747-1755, November, 2002.
- CHEN, C.-S., LIN, C.-H., TSAI, H.-Y., “A rule-based expert system with colored petri net models for distribution system service restoration”. IEEE Transactions on Power Systems, Vol. 17, No. 4, pp. 1073-1080, 2002.
- CHEN, W.-H., TSAI, M.-S., KUO, H.-L., “Distribution System Restoration Using the Hybrid Fuzzy-Grey Method”. IEEE Transactions on Power Systems, Vol. 20, No. 1, February, pp. 199-205, 2005.
- CHIOU, J.-P., CHANG, C.-F., SU, C.-T., “Variable Scaling Hybrid Differential Evolution for Solving Network Reconfiguration of Distribution Systems”. IEEE Transactions on Power Systems, Vol. 20, No. 2, May, pp. 668-674, 2005.
- CHU, P. C., BEASLEY, J. E., “A Genetic Algorithm for the Generalized Assignment Problem”. Comput. Oper. Res., Vol. 24, No. 1, pp. 17-23, 1997.
- CIVANLAR, S., GRAINGER, J. J., YIN, H. E LEE, S. S. H. “Distribution Feeder Reconfiguration for Loss Reduction”. IEEE Transactions on Power Delivery, Vol. 3, No. 3, pp. 1217-1223, 1988.
- CORMEN T.H., LEISERSON C. E., RIVEST R. L., STEIN C., Algoritmos: teoria e prática. 2º edição, Rio de Janeiro: Editora Campus, 2002.
- DARWIN, C., “On the Origin of Species”. Down, Bromley, Kent, October 1st, 1859.
- DAS, D., “A Fuzzy Multiobjective Approach for Network Reconfiguration of Distribution Systems”. IEEE Transactions on Power Delivery, Vol. 21, No. 1, January, pp. 202-209, 2006.
- DELBEM A. C. B., CARVALHO, A. C. P. L. F., BRETAS, N. G., “Main chain representation for evolutionary algorithms applied to distribution system

- reconfiguration”. IEEE Transactions on Power Systems, Vol. 20, No. 1, pp. 425-436, 2005.
- ENACHEANU, B., RAISON, B., CAIRE, R., DEVAUX, O., BIENIA, W., HADJSAID, N., “Radial Network Reconfiguration Using Genetic Algorithm Based on the Matroid Theory”. IEEE Transactions on Power Systems, Vol. 23, No. 1, February, pp. 186-195, 2008.
- FERREIRA, L. A. F. M., CARVALHO, P. M. S., JORGE, L. A., GRAVE, S. N. C., BARRUNCHO, L. M. F., “Optimal Distribution Planning by Evolutionary Computation - How to Make it Work,” Transmission and Distribution Conference and Exposition, 2001 IEEE/PES, Vol. 1, pp. 469 – 475, 2001.
- FONTAN, D. M. S., “Reconfiguração de Sistemas de Distribuição usando um Algoritmo Evolutivo”. Dissertação de Mestrado. Universidade Federal de Campina Grande, Campina Grande, Paraíba, Brasil, 2008.
- FUDOU, H., GENJI, T., FUKUYAMA, Y., NAKANISHI, Y., “A Genetic Algorithm for Network Reconfiguration using Three Phase unbalanced Load Flow,” ISAP Intelligent Systems Applications to Power Systems, Seoul, Korea, 1997.
- FUKUYAMA, Y., CHIANG, H.-D., “A parallel genetic algorithm for service restoration in electric power distribution systems”. Proceedings of IEEE International Conference on Fuzzy Systems, Vol. 1, pp. 275 – 282, 1995.
- GOLDBERG, D. E., “Genetic Algorithms in Search, Optimization and Machine Learning”, Addison-Wesley, 1989.
- GOSWAMI, S. K., BASU, S. K., “A new algorithm for reconfiguration of distribution feeders for loss minimization”. IEEE Transactions on Power Delivery, PWRD-7, (3), pp. 1484-1490, 1992.

- HONG, Y.-Y., HO, S.-Y., "Determination of Network Configuration Considering Multiobjective in Distribution Systems Using Genetic Algorithms". IEEE Transactions on Power Systems, Vol. 20, No. 2, May, pp. 1062-1069, 2005.
- HOLLAND, J. H., "Adaptation in Natural and Artificial Systems", The University of Michigan Press, First Ed., 1975.
- HSU, Y.-Y., HUANG, H. M., KUO, H. C., PENG, S. K., CHANG, C. W., CHANG, K. J., YU, H. S., CHOW, C. E., KUO, R. T., "Distribution system service restoration using a heuristic search approach". IEEE Transactions on Power Delivery, Vol. 7, No. 2, pp. 734-740, 1992.
- HUANG, Y.-C., "Enhanced genetic algorithm-based fuzzy multi-objective approach to distribution network reconfiguration". IEE Proc.-Gener. Trans. Distrib., Vol. 149, No. 5, September, pp. 615-620, 2002.
- JAZEBI, S., HOSSEINIAN, S. H., POOYAN, M., VAHIDI, B., "Performance comparison of GA and DEA in Solving Distribution System Reconfiguration Problem". OPTIM2008, pp. 185-190, May, 2008.
- JONNAVITHULA, S., BILLINTON, ROY, "Minimum Cost Analysis of Feeder Routing in Distribution System Planning". IEEE Transactions on Power Delivery, Vol. 11, No. 4, October 1996, pp. 1935-1940, 1996.
- KASHEM, M. A., GANAPATHY, V., JASMON, G. B., "Network reconfiguration for load balancing in distribution networks". Proc.-Gener. Transm. Distrib., Vol. 146, No. 6, November 1999.
- KING, R. T. F. A., RADHA, B., RUGHOPUTH, H. C. S., "A Real-Parameter Genetic Algorithm for Optimal Network Reconfiguration". IEEE International Conference on Industrial Technology, 10-12 December, Vol. 1, pp. 54-59, 2003.

- \_\_\_\_\_, "A Fuzzy Logic Controlled Genetic Algorithm for Optimal Electrical Distribution Network Reconfiguration". Proceedings of the 2004 IEEE International Conference on Networking, Sensing & Control, Taipei, Taiwan, March 21-23, pp. 577-582, 2004.
- KRUSKAL, J. B., "On the shortest spanning subtree of a graph and the traveling salesman problem". Proc. Amer. Math. Soc., Vol. 7, pp. 48-50, 1956.
- LIN, W. M., CHANG, F. S., TSAY, M. T. "Distribution Feeder Reconfiguration with Refined Genetic Algorithm". IEE Proceedings Generation, Transmission and Distribution, 147(6), pp. 349-354, 2000.
- LIN, W. M., SU, Y. S., TSAY, M. T., "Genetic Algorithm for Optimal Distribution System Planning", POWERCON '98, Vol. 1, pp. 241 – 245, 1998.
- LIU, C-C, LEE, S. J., VENKATA, S. S., "An Expert System Operational Aid for Restoration and Loss Reduction of Distribution Systems", IEEE Transactions on Power Systems, Vol. 3, No. 2, pp. 619-626, 1988.
- LIU, C-C, LEE, S. J., VU, K., "Loss Minimization of Distribution Feeders: Optimality and Algorithms", IEEE Transactions on Power Delivery, Vol. 4, No. 2, April 1989, pp. 1281-1289, 1989.
- LIU, Y., GU, X., "Skeleton-Network Reconfiguration Based on Topological Characteristics of Scale-Free Networks and Discrete Particle Swarm Optimization". IEEE Transactions on Power Systems, Vol. 22, No. 3, August, pp. 1267-1274, 2007.
- MANJUNATH, K., MOHAN, M. R., "A New Hybrid Multi-Objective Quick Service Restoration technique for Electric Power Distribution Systems". Electrical Power and Energy Systems, 29, pp. 51-64, 2007.

- MANTOVANI, J. R. S., CASARI, F. E ROMERO, R. A. “Reconfiguração de Sistemas de Distribuição Radiais utilizando o critério da queda de tensão”. SBA Controle e Automação, Vol. 11, No. 3, 2000.
- MATOS M. A., MELO P. “Multiobjective Reconfiguration for Loss Reduction and Service Restoration Using Simulated Annealing”. IEEE Power Tech Conference, Budapest, Hungary, August, 1999.
- MENDOZA, J., LÓPEZ, R., MORALES, D., LÓPEZ, E., DESSANTE, P., MORAGA, R., “Minimal Loss Reconfiguration Using Genetic Algorithms With Restricted Population and Addressed Operators: Real Application”. IEEE Transactions on Power Systems, Vol. 21, No. 2, May, pp. 948-954, 2006a.
- MENDOZA, F., BERNAL-AGUSTÍN, J. L., DOMÍNGUEZ-NAVARRO, J. A., “NSGA and SPEA Applied to Multiobjective Design of Power Distribution Systems”. IEEE Transactions on Power Systems, Vol. 21, No. 4, pp. 1938-1945, 2006b.
- MERLIN, A., BACK, H., “Search for a Minimum -Loss Operating Spanning Tree Configuration for an Urban Power Distribution System”, 5th Power System Computation Conference, Cambridge, England, 1975.
- MICHALEWICZ, Z., “Genetic Algorithms + Data Structures = Evolution Programs”, Springer Verlag, NY, 1994.
- MIRANDA, V., RANITO, J. V., PROENÇA, L. M., “Genetic Algorithm in Optimal multistage distribution network planning,” IEEE Transactions on Power Systems, Vol. 9, No. 4, pp. 1927-1933, 1994.
- MIU, K. N., CHIANG, H.-D., YUAN, B., DARLING, G., “Fast service restoration for large-scale distribution systems with priority customers and constraints”. IEEE Transactions on Power Systems, Vol. 13, No. 3, pp. 789-795, 1998.

- MORELATO, A. L., MONTICELLI, A., “Heuristic Search Approach to Distribution Systems Restoration”. IEEE Transactions on Power Delivery, Vol. 4, No. 4, pp. 2235-2241, 1989.
- MORTON, A. B. E MAREELS, I. M. Y. “An efficient brute-force solution to the network reconfiguration problem”. IEEE Transactions on Power Delivery, Vol. 15, No. 3, 2000.
- NAGATA, T., IBA, N., TAO, Y., SASAKI, H., “A New Power System Restoration Framework by Means of Multi-Agent Approach”. IEEE/PES Transmission and Distribution Conference and Exhibition, Vol. 3, 6-10 Oct., pp. 1695-1700, 2002.
- NARA, K., SHIOSE, A., KITAGAWA, M., ISHIHARA, T., “Implementation of Genetic Algorithm for Distribution Systems Loss Minimum Re-Configuration”, IEEE Transactions on Power Systems, Vol. 7, No. 3, August, pp. 1044-1051, 1992.
- NARA, K., “Genetic Algorithm for Power Systems Planning”. Proceedings of the 4<sup>th</sup> International Conference on Advances in Power System Control, Operation and Management - APSCOM-97, Hong Kong, November, 1997.
- NARA, K., SONG, Y. H., “Modern Heuristics Application to Distribution System Optimization”. IEEE Power Engineering Society Winter Meeting, Vol. 2, pp. 826-832, 2002.
- NETTO, P. O. Boaventura, “Grafos: teoria, modelos, algoritmos”. 4. ed. Edgard Blucher, 2006.
- PEPONIS, G. J., PAPADOPOULOS, M. P., HATZIARGYRIOU, N. D. “Distribution Network Reconfiguration to Minimize Resistive Line Losses”. IEEE Transactions on Power Delivery, Vol. 10, No. 3, 1995.

- PEPONIS, G. J., PAPADOPOULOS, M. P., “New dynamic, branch exchange method for optimal distribution system planning,” IEE Proc-Gener. Transm. Distrib., Vol. 144, No. 3, 1997.
- PRASAD, K., RANJAN, R., SAHOO, N. C., CHATURVEDI, A., “Optimal Reconfiguration of Radial Distribution Systems Using a Fuzzy Mutated Genetic Algorithm”. IEEE Transactions on Power Delivery, Vol. 20, No. 2, April, pp. 1211-1213, 2005.
- QUEIROZ, L. M. O., LYRA, C., “Adaptive Hybrid Genetic Algorithm for Technical Loss Reduction in Distribution Networks under Variable Demands”. IEEE Transactions on Power Systems, Vol. 24, No. 1, February, pp. 445-453, 2009.
- RADCLIFFE, N. J., “Non-linear genetic representations”. See Männer and Manderick (1992), pp. 259–268, 1992.
- RAIDL, G. R., JULSTROM, B. A., “Edge-sets: An Effective Evolutionary Coding of Spanning Trees”. IEEE Transactions on Evolutionary Computation, Vol. 7, No. 3, June, pp. 225-239, 2003.
- RAJU, G. K. V., BIJWE, P. R., “An Efficient Algorithm for Minimum Loss Reconfiguration of Distribution System Based on Sensitivity and Heuristics”. IEEE Transactions on Power Systems, Vol. 23, No. 3, August, pp. 1280-1287, 2008.
- RAMOS, E. R., EXPÓSITO, A. G., SANTOS, J. R., IBORRA, F. L., “Path-Based Distribution Network Modeling: Application to Reconfiguration for Loss Reduction”. IEEE Transactions on Power Systems, Vol. 20, No. 2, May, pp. 556-564, 2005.

- RAMOS, E. R., MARTÍNEZ-RAMOS, J. L., EXPÓSITO, A. G., SALADO, A. J. U.,  
“Optimal Reconfiguration of Distribution Networks for Power Loss Reduction”.  
IEEE Porto Power Tech Conference, September 2001.
- RIVAS-DÁVALOS, F., MORENO-GOYTIA, E., GUTIÉRREZ-ALACARAZ, G.,  
TOVAR-HERNANDEZ, J., “Evolutionary Multi-Objective Optimization in  
Power Systems: State-of-the-Art”. Power Tech 2007, 1-5 July, Lausanne,  
Switzerland, 2007.
- ROTHLAUF, F., “Representations for Genetic and Evolutionary Algorithms”. Springer,  
2<sup>nd</sup> Edition, Heidelberg, Germany, 2006.
- ROTHLAUF, F., GOLDBERG, D. E., HEINZL, A., “Network random keys – A tree  
network representation scheme for genetic and evolutionary algorithms”.  
Evolutionary Computation 10 (1), pp. 75–97, 2002.
- RUDNICK, H., HARNISCH, I. E SANHUEZA, R. “Reconfiguration of Electric  
Distribution Systems”. Revista Facultad de Ingeniería, UTA (Chile), Vol. 4, 1997.
- SAKAWA, M. “Genetic Algorithms and Fuzzy Multiobjective Optimization”. Kluwer  
Academic Publishers, 1st edition, January, 2002.
- SANTOS, A. C., DELBEM, A. C. B., BRETAS, N. G., “A Multiobjective Evolutionary  
Algorithm with Node-depth Encoding for Energy Restoration”. Fourth  
International Conference on Natural Computation, IEEE Computer Society, pp.  
417-422, 2008.
- SARFI, R. F., SALAMA, M. M. A., CHIKHANI, A. Y., “Distribution System  
Reconfiguration for Loss Reduction: A new algorithm based on a set of Quantified  
Heuristic Rules”. Proceedings of the 1994 Canadian Conference on Electrical and  
Computer Engineering, pp. 125-130, New York: Institute of Electrical and  
Electronics Engineers, 1994a.

\_\_\_\_\_, "A survey of state of the art in Distribution Systems Reconfiguration for System Loss Reduction". Electric Power System Research, Vol. 31, No. 1, pp. 61-70, 1994b.

\_\_\_\_\_, "Distribution System Reconfiguration for Loss Reduction: An algorithm based on network partitioning theory". IEEE Transactions on Power Systems, Vol. 11, No. 1, 1995.

SARMA, N. D. R., PRASAD, V. C., PRAKASA RAO, K. S., SANKAR, V., "A new network reconfiguration technique for service restoration in distribution networks". IEEE Transactions on Power Delivery, Vol. 9, No. 4, pp. 1936-1942, 1994.

SHIN, J.-R., KIM, B.-S., PARK, J.-B., LEE, K. Y., "A New Optimal Routing Algorithm for Loss Minimization and Voltage Stability Improvement in Radial Power Systems". IEEE Transactions on Power Systems, Vol. 22, No. 2, May, pp. 648-657, 2007.

SHIRMOHAMMADI, D., "Service restoration in distribution networks via network reconfiguration". IEEE Transactions on Power Delivery, Vol. 7, No. 2, pp. 952-958, 1992.

SHIRMOHAMMADI, D., HONG, H. W., SEMLYEN, A. E LUO, G. X., "Reconfiguration of Electric Distribution Networks for Resistive Line Losses Reduction". IEEE Transactions on Power Delivery, Vol. 4, No. 2, pp. 1492-1498, 1989.

SIQING, S., YOUJIANG, S., YAN, L., WENQIN, Z. YIHAN, Y., "Integrating genetic algorithm with expert system for service restoration in distribution system". Proceedings of International Conference on Power System Technology - POWERCON '98, Vol. 1, pp. 265-269, 1998.

- SOUZA, B. A. Distribuição de Energia Elétrica. Campina Grande: CCT/UFPB-REENGE/PAPE, 1997.
- SOUZA, B. A., BRAZ, H. D. M., ALVES, H. N., “A Genetic Algorithm for Optimal Configuration of Distribution Feeders,” Proceedings of the Brazilian Symposium of Intelligent Automation – SBAI, Bauru, Brazil, 2003.
- SU, C.-T., CHANG, C.-F., CHIOU, J.-P., “Distribution Network Reconfiguration for Loss Reduction by Ant Colony Search Algorithm”. Electric Power Research 75 (2005), pp. 190-199, 2005.
- SU, C.-T., LEE, C.-S., “Network Reconfiguration of Distribution Systems Using Improved Mixed-Integer Hybrid Differential Evolution”. IEEE Transactions on Power Delivery, Vol. 18, No. 3, pp. 1022-1027, 2003.
- TANG, Y., “Power Distribution System Planning with Reliability Modeling and Optimization,” IEEE Transactions on Power Systems, Vol. 11, No. 1, pp. 181-189, 1996.
- TOUNE, S., FUDO, H., GENJI, T., FUKUYAMA, Y., NAKANISHI, Y., “Comparative study of modern heuristic algorithms to service restoration in distribution systems”. IEEE Transactions on Power Delivery, Vol. 17, No. 1, pp. 173-181, 2002.
- WANG, C., CHENG, H. Z., “Optimization of Network Configuration in Large Distribution Systems Using Plant Growth Simulation Algorithm”. IEEE Transactions on Power Systems, Vol. 23, No. 1, February, pp. 119-126, 2008.
- WATANABE, I., NODU, M., “A genetic algorithm for optimizing switching sequence of service restoration in distribution systems”. Congress on Evolutionary Computation – CEC2004, Vol. 2, pp. 1683 – 1690, 2004.

- WATANABE, I., KURIHARA, I., NAKACHI, Y., M., “A Hybrid Genetic Algorithm for Service Restoration Problems in Power Distribution Systems”. IEEE Congress on Evolutionary Computation CEC 2006, Vol. 2, pp. 1683 – 1690, 2006.
- ZHANG, D., FU, Z., ZHANG, L., “Joint Optimization for Power Loss Reduction in Distribution Systems”. IEEE Transactions on Power Systems, Vol. 23, No. 1, February, pp. 161-169, 2008.
- ZHOU,Q., SHIRMOHAMMADI, D., LIU, W.-H. E., “Distribution feeder reconfiguration for service restoration and load balancing”. IEEE Transactions on Power Systems, Vol. 12, No. 2, pp. 724-729, 1997.
- ZHU, J. Z., “Optimal reconfiguration of distribution network using the refined genetic algorithm”. ELSEVIER Electric Power Systems Research 62, pp. 37-42, 2002.

## Apêndice A

Neste apêndice são apresentados os dados completos dos sistemas utilizados nos testes do Capítulo 7. Como o aplicativo desenvolvido utiliza como entrada as coordenadas das barras e os dados dos condutores/estruturas dos ramos para calcular as impedâncias da rede, foi necessário definir coordenadas e condutores/estruturas fictícios para reproduzir os dados dos sistemas sob estudo. Formulou-se uma única tabela para cada sistema, na qual uma primeira parte estabelece identificadores e localização das subestações; uma segunda fornece os dados relativos às barras e suas respectivas origens; e um terceiro bloco define as cordas do sistema (chaves NA).

### A.1 Sistema de Cinvanlar et al (1988)

Este sistema possui 16-nós, sendo 3 deles subestações de 23 kV (Tabela A.1). A capacidade das subestações e a suportabilidade dos ramos são ajustadas para valores altos, o que equivale a desconsiderar tais restrições.

**Tabela A.1 – Dados de entrada referentes ao sistema 16-nós.**

Linha	De	Para	Cabo	R ( $\Omega$ /km)	XL ( $\Omega$ /km)	Imax (A)	X (km)	Y (km)	P(MW)	Q(Mvar)
1		1					0.00	3.00		
2		2					0.00	0.00		
3		3					0.00	-3.00		
4	1	4	C01	0.198375	0.264500	1000.00	2.00	-3.00	2.00	1.60
5	4	5	C02	0.423200	0.581900	1000.00	2.00	-2.00	3.00	0.40
6	4	6	C03	0.238050	0.476100	1000.00	4.00	-3.00	2.00	-0.40

Linha	De	Para	Cabo	R ( $\Omega$ /km)	XL ( $\Omega$ /km)	Imax (A)	X (km)	Y (km)	P(MW)	Q(Mvar)
7	6	7	C04	0.105800	0.105800	1000.00	4.00	-1.00	1.50	1.20
8	2	8	C05	0.581900	0.581900	1000.00	1.00	0.00	4.00	2.70
9	8	9	C06	0.423200	0.581900	1000.00	2.00	0.00	5.00	1.80
10	8	10	C07	0.581900	0.581900	1000.00	1.00	1.00	1.00	0.90
11	9	11	C08	0.581900	0.581900	1000.00	2.00	-1.00	0.60	-0.50
12	9	12	C09	0.423200	0.581900	1000.00	3.00	0.00	4.50	-1.70
13	3	13	C10	0.581900	0.581900	1000.00	1.00	3.00	1.00	0.90
14	13	14	C11	0.476100	0.634800	1000.00	1.00	2.00	1.00	-1.10
15	13	15	C12	0.141067	0.193967	1000.00	4.00	3.00	1.00	0.90
16	15	16	C13	0.105800	0.105800	1000.00	4.00	1.00	2.10	-0.80
17	5	11	C14	0.211600	0.211600	1000.00				
18	10	14	C15	0.211600	0.211600	1000.00				
19	7	16	C16	0.238050	0.317400	1000.00				

## A.2 Sistema de Baran & Wu (1989)

Este sistema é composto por 33-nós, sendo 1 deles relativo a uma subestação de 12.66 kV. Na Tabela A.2, seus dados de entrada são apresentados.

**Tabela A.2 – Dados de entrada referentes ao sistema 33-nós.**

Linha	De	Para	Cabo	R ( $\Omega$ /km)	XL ( $\Omega$ /km)	Imax (A)	X (km)	Y (km)	P(MW)	Q(Mvar)
1		0					0.00	0.00		
2	0	1	C01	0.092200	0.047000	1000.00	1.00	0.00	0.10	0.06
3	1	2	C02	0.493000	0.251100	1000.00	2.00	0.00	0.09	0.04
4	2	3	C03	0.366000	0.186400	1000.00	3.00	0.00	0.12	0.08
5	3	4	C04	0.381100	0.194100	1000.00	4.00	0.00	0.06	0.03
6	4	5	C05	0.819000	0.707000	1000.00	5.00	0.00	0.06	0.02
7	5	6	C06	0.187200	0.618800	1000.00	6.00	0.00	0.20	0.10
8	6	7	C07	0.711400	0.235100	1000.00	7.00	0.00	0.20	0.10
9	7	8	C08	1.030000	0.740000	1000.00	8.00	0.00	0.06	0.02
10	8	9	C09	1.044000	0.740000	1000.00	8.00	-1.00	0.06	0.02
11	9	10	C10	0.196600	0.065000	1000.00	7.00	-1.00	0.05	0.03
12	10	11	C11	0.374400	0.123800	1000.00	7.00	-2.00	0.06	0.04
13	11	12	C12	1.468000	1.155000	1000.00	8.00	-2.00	0.06	0.04
14	12	13	C13	0.541600	0.712900	1000.00	9.00	-2.00	0.12	0.08
15	13	14	C14	0.295500	0.263000	1000.00	9.00	0.00	0.06	0.01
16	14	15	C15	0.746300	0.545000	1000.00	9.00	1.00	0.06	0.02

Linha	De	Para	Cabo	R ( $\Omega$ /km)	XL ( $\Omega$ /km)	I <sub>max</sub> (A)	X (km)	Y (km)	P(MW)	Q(Mvar)
17	15	16	C16	1.289000	1.721000	1000.00	9.00	2.00	0.06	0.02
18	16	17	C17	0.732000	0.574000	1000.00	9.00	3.00	0.09	0.04
19	1	18	C18	0.058000	0.055300	1000.00	3.00	-2.00	0.09	0.04
20	18	19	C19	1.504200	1.355400	1000.00	4.00	-2.00	0.09	0.04
21	19	20	C20	0.409500	0.478400	1000.00	5.00	-2.00	0.09	0.04
22	20	21	C21	0.708900	0.937300	1000.00	6.00	-2.00	0.09	0.04
23	2	22	C22	0.451200	0.308300	1000.00	2.00	1.00	0.09	0.05
24	22	23	C23	0.898000	0.709100	1000.00	2.00	2.00	0.42	0.20
25	23	24	C24	0.896000	0.701100	1000.00	2.00	3.00	0.42	0.20
26	5	25	C25	0.203000	0.103400	1000.00	5.00	1.00	0.06	0.03
27	25	26	C26	0.284200	0.144700	1000.00	5.00	2.00	0.06	0.03
28	26	27	C27	1.059000	0.933700	1000.00	4.00	2.00	0.06	0.02
29	27	28	C28	0.804200	0.700600	1000.00	4.00	3.00	0.12	0.07
30	28	29	C29	0.507500	0.258500	1000.00	5.00	3.00	0.20	0.60
31	29	30	C30	0.974400	0.963000	1000.00	6.00	3.00	0.15	0.07
32	30	31	C31	0.310500	0.361900	1000.00	7.00	3.00	0.21	0.10
33	31	32	C32	0.341000	0.530200	1000.00	8.00	3.00	0.06	0.04
34	7	20	C33	0.707100	0.707100	1000.00				
35	8	14	C34	2.000000	2.000000	1000.00				
36	11	21	C35	2.000000	2.000000	1000.00				
37	17	32	C36	0.500000	0.500000	1000.00				
38	24	28	C37	0.250000	0.250000	1000.00				

### A.3 Sistema de Das (2006)

Este sistema é composto por 70-nós, sendo 2 deles subestações com tensão nominal 11 kV. Apresentam-se na Tabela A.3 seus dados de entrada.

**Tabela A.3 – Dados de entrada referentes ao sistema 70-nós.**

Linha	De	Para	Cabo	R ( $\Omega$ /km)	XL ( $\Omega$ /km)	I <sub>max</sub> (A)	X (km)	Y (km)	P(MW)	Q(Mvar)
1		1					1.00	5.00		
2		70					16.00	5.00		
3	1	2	C1	1.097000	1.074000	270.00	1.00	4.00	0.10	0.09
4	2	3	C2	1.463000	1.432000	270.00	1.00	3.00	0.06	0.04
5	3	4	C3	0.731000	0.716000	270.00	2.00	3.00	0.15	0.13
6	4	5	C4	0.366000	0.358000	270.00	3.00	3.00	0.08	0.05
7	5	6	C5	1.828000	1.790000	270.00	3.00	2.00	0.02	0.01

Linha	De	Para	Cabo	R ( $\Omega$ /km)	XL ( $\Omega$ /km)	I <sub>max</sub> (A)	X (km)	Y (km)	P(MW)	Q(Mvar)
8	6	7	C6	1.097000	1.074000	270.00	4.00	2.00	0.02	0.01
9	7	8	C7	0.366000	0.358000	270.00	6.00	2.00	0.01	0.01
10	8	9	C8	0.731000	0.716000	270.00	7.00	2.00	0.02	0.01
11	4	10	C9	1.080000	0.734000	208.00	2.00	4.00	0.02	0.01
12	10	11	C10	1.620000	1.101000	208.00	3.00	4.00	0.02	0.01
13	11	12	C11	1.080000	0.734000	208.00	4.00	4.00	0.05	0.04
14	12	13	C12	1.350000	0.917000	208.00	5.00	4.00	0.11	0.09
15	13	14	C13	0.810000	0.550000	208.00	6.00	4.00	0.03	0.02
16	14	15	C14	1.944000	1.321000	208.00	7.00	4.00	0.04	0.03
17	7	68	C15	1.080000	0.734000	208.00	3.00	5.00	0.06	0.03
18	68	69	C16	1.620000	1.101000	208.00	3.00	6.00	0.04	0.03
19	1	16	C17	0.548500	0.537000	270.00	4.00	6.00	0.02	0.01
20	16	17	C18	0.366000	0.358000	270.00	4.00	7.00	0.01	0.01
21	17	18	C19	1.463000	1.432000	270.00	5.00	7.00	0.03	0.02
22	18	19	C20	0.914000	0.895000	270.00	6.00	7.00	0.09	0.05
23	19	20	C21	0.804000	0.787000	270.00	6.00	6.00	0.05	0.03
24	20	21	C22	1.133000	1.110000	270.00	3.00	7.00	0.06	0.04
25	21	22	C23	0.475000	0.465000	270.00	3.00	8.00	0.10	0.08
26	17	23	C24	2.214000	1.505000	208.00	4.00	8.00	0.08	0.07
27	23	24	C25	1.620000	1.110000	208.00	5.00	8.00	0.10	0.06
28	24	25	C26	1.080000	0.734000	208.00	6.00	8.00	0.10	0.06
29	25	26	C27	0.540000	0.367000	208.00	7.00	8.00	0.12	0.07
30	26	27	C28	0.540000	0.367000	208.00	8.00	8.00	0.11	0.07
31	27	28	C29	1.080000	0.734000	208.00	16.00	3.00	0.08	0.05
32	28	29	C30	1.080000	0.734000	208.00	15.00	3.00	0.06	0.04
33	70	30	C31	0.183000	0.179000	270.00	14.00	3.00	0.01	0.01
34	30	31	C32	0.731000	0.716000	270.00	14.00	2.00	0.02	0.01
35	31	32	C33	0.731000	0.716000	270.00	12.00	2.00	0.05	0.03
36	32	33	C34	0.804000	0.787000	270.00	11.00	2.00	0.04	0.03
37	33	34	C35	0.585000	0.572500	270.00	10.00	2.00	0.06	0.04
38	34	35	C36	0.768000	0.752000	270.00	9.00	2.00	0.04	0.03
39	35	36	C37	0.731000	0.716000	270.00	8.00	2.00	0.03	0.03
40	36	37	C38	1.097000	1.074000	270.00	13.00	3.00	0.15	0.10
41	37	38	C39	1.463000	1.432000	270.00	11.00	3.00	0.06	0.04
42	32	39	C40	1.080000	0.734000	208.00	10.00	3.00	0.12	0.07
43	39	40	C41	0.270000	0.183500	208.00	9.00	3.00	0.09	0.06
44	40	41	C42	1.080000	0.734000	208.00	8.00	3.00	0.02	0.01
45	41	42	C43	1.836000	1.248000	208.00	11.00	4.00	0.02	0.01
46	42	43	C44	1.296000	0.881000	208.00	11.00	5.00	0.10	0.05

Linha	De	Para	Cabo	R ( $\Omega$ /km)	XL ( $\Omega$ /km)	I <sub>max</sub> (A)	X (km)	Y (km)	P(MW)	Q(Mvar)
47	40	44	C45	1.188000	0.807000	208.00	9.00	4.00	0.06	0.04
48	44	45	C46	0.540000	0.367000	208.00	11.00	1.00	0.09	0.07
49	42	46	C47	1.080000	0.734000	208.00	10.00	1.00	0.09	0.06
50	35	47	C48	0.540000	0.367000	208.00	9.00	1.00	0.10	0.07
51	47	48	C49	1.080000	0.734000	208.00	8.00	1.00	0.14	0.09
52	48	49	C50	1.080000	0.734000	208.00	15.00	5.00	0.06	0.04
53	49	50	C51	1.080000	0.734000	208.00	15.00	7.00	0.02	0.01
54	70	51	C52	0.366000	0.358000	270.00	14.00	7.00	0.04	0.03
55	51	52	C53	0.731500	0.716000	270.00	13.00	7.00	0.04	0.02
56	52	53	C54	1.463000	1.432000	270.00	12.00	7.00	0.03	0.02
57	53	54	C55	0.914000	0.895000	270.00	12.00	6.00	0.04	0.03
58	54	55	C56	1.097000	1.074000	270.00	14.00	6.00	0.08	0.05
59	55	56	C57	1.097000	1.074000	270.00	13.00	6.00	0.24	0.12
60	52	57	C58	0.190900	0.129400	208.00	13.00	5.00	0.13	0.11
61	57	58	C59	0.270000	0.183000	208.00	12.00	5.00	0.03	0.01
62	58	59	C60	0.810000	0.550000	208.00	11.00	7.00	0.01	0.01
63	59	60	C61	1.296000	0.881000	208.00	10.00	7.00	0.15	0.13
64	55	61	C62	1.188000	0.807000	208.00	10.00	8.00	0.05	0.03
65	61	62	C63	1.188000	0.807000	208.00	9.00	8.00	0.03	0.02
66	62	63	C64	0.810000	0.550000	208.00	9.00	7.00	0.13	0.12
67	63	64	C65	1.620000	1.101000	208.00	9.00	6.00	0.15	0.13
68	62	65	C66	1.080000	0.734000	208.00	7.00	6.00	0.03	0.02
69	65	66	C67	0.540000	0.367000	208.00	4.00	3.00	0.10	0.06
70	66	67	C68	1.080000	0.734000	208.00	5.00	3.00	0.04	0.03
71	9	50	C69	0.642053	0.513360	234.00				
72	9	38	C70	0.381000	0.244000	234.00				
73	15	46	C71	0.340500	0.272000	234.00				
74	22	67	C72	0.254000	0.203000	234.00				
75	29	64	C73	0.254000	0.203000	234.00				
76	45	60	C74	0.254000	0.203000	234.00				
77	43	38	C75	0.454000	0.363000	234.00				
78	39	59	C76	0.227000	0.181500	234.00				
79	21	27	C77	0.454000	0.363000	234.00				
80	15	9	C78	0.340500	0.272000	234.00				
81	67	15	C79	0.227000	0.181500	234.00				

#### A.4 Sistema de Su & Lee (2003)

Este sistema é composto por 94-nós, sendo 11 subestações com tensão nominal 11.4 kV. Apresentam-se na Tabela A.4 seus dados de entrada.

**Tabela A.4 – Dados de entrada referentes ao sistema 94-nós.**

Linha	De	Para	Cabo	R ( $\Omega$ /km)	XL ( $\Omega$ /km)	I <sub>max</sub> (A)	X (km)	Y (km)	P(MW)	Q(Mvar)
1		A					0.00	10.00		
2		B					0.00	5.00		
3		C					0.00	1.00		
4		D					0.00	2.00		
5		E					0.00	3.00		
6		F					0.00	4.00		
7		G					0.00	12.00		
8		H					0.00	9.00		
9		I					0.00	6.00		
10		J					0.00	7.00		
11		K					0.00	8.00		
12	A	1	C1	0.032400	0.110400	10000	6.00	10.00	0.00	0.00
13	1	2	C2	0.209600	0.430400	10000	7.00	10.00	0.10	0.05
14	2	3	C3	0.235800	0.484200	10000	8.00	10.00	0.30	0.20
15	3	4	C4	0.091700	0.188200	10000	9.00	10.00	0.35	0.25
16	4	5	C5	0.209600	0.430400	10000	10.00	10.00	0.22	0.10
17	5	6	C6	0.039300	0.080700	10000	11.00	10.00	1.10	0.80
18	6	7	C7	0.040500	0.138000	10000	12.00	10.00	0.40	0.32
19	7	8	C8	0.093736	0.192481	10000	13.00	9.50	0.30	0.20
20	7	9	C9	0.210906	0.433082	10000	13.00	10.50	0.30	0.23
21	7	10	C10	0.052400	0.107600	10000	14.00	10.00	0.30	0.26
22	B	11	C11	0.078600	0.161400	10000	1.00	5.00	0.00	0.00
23	11	12	C12	0.028383	0.057867	10000	13.00	5.00	1.20	0.80
24	12	13	C13	0.026200	0.053800	10000	14.00	5.00	0.80	0.60
25	12	14	C14	0.078600	0.161400	10000	13.00	4.00	0.70	0.50
26	C	15	C15	0.113400	0.386400	10000	1.00	1.00	0.00	0.00
27	15	16	C16	0.052400	0.107600	10000	2.00	1.00	0.30	0.15
28	16	17	C17	0.008733	0.017933	10000	8.00	1.00	0.50	0.35
29	17	18	C18	0.031440	0.064560	10000	13.00	1.00	0.70	0.40
30	18	19	C19	0.039300	0.080700	10000	14.00	1.00	1.20	1.00
31	19	20	C20	0.085150	0.174850	10000	16.00	1.00	0.30	0.30
32	20	21	C21	0.235800	0.484200	10000	17.00	1.00	0.40	0.35
33	21	22	C22	0.157200	0.322800	10000	18.00	1.00	0.05	0.02

Linha	De	Para	Cabo	R ( $\Omega/\text{km}$ )	XL ( $\Omega/\text{km}$ )	I <sub>max</sub> (A)	X (km)	Y (km)	P(MW)	Q(Mvar)
34	21	23	C23	0.196500	0.403500	10000	17.00	2.00	0.05	0.02
35	23	24	C24	0.131000	0.269000	10000	17.00	3.00	0.05	0.01
36	D	25	C25	0.056700	0.193200	10000	1.00	2.00	0.05	0.03
37	25	26	C26	0.104800	0.215200	10000	2.00	2.00	0.10	0.06
38	26	27	C27	0.248900	0.511100	10000	3.00	2.00	0.10	0.07
39	27	28	C28	0.024300	0.082800	10000	5.00	2.00	1.80	1.30
40	28	29	C29	0.065500	0.134500	10000	7.00	2.00	0.20	0.12
41	E	30	C30	0.196500	0.396000	10000	1.00	3.00	0.00	0.00
42	30	31	C31	0.131000	0.269000	10000	2.00	3.00	1.80	1.60
43	31	32	C32	0.131000	0.269000	10000	3.00	3.00	0.20	0.15
44	32	33	C33	0.026200	0.053800	10000	4.00	3.00	0.20	0.10
45	33	34	C34	0.170300	0.349700	10000	5.00	3.00	0.80	0.60
46	34	35	C35	0.052400	0.107600	10000	6.00	3.00	0.10	0.06
47	35	36	C36	0.497800	1.022200	10000	7.00	3.00	0.10	0.06
48	36	37	C37	0.039300	0.080700	10000	8.00	3.00	0.02	0.01
49	37	38	C38	0.039300	0.080700	10000	9.00	3.00	0.02	0.01
50	38	39	C39	0.078600	0.161400	10000	9.00	2.00	0.02	0.01
51	39	40	C40	0.104800	0.215200	10000	11.00	2.00	0.02	0.01
52	38	41	C41	0.196500	0.403500	10000	10.00	3.00	0.20	0.16
53	41	42	C42	0.209600	0.430400	10000	11.00	3.00	0.05	0.03
54	F	43	C43	0.048600	0.165600	10000	1.00	4.00	0.00	0.00
55	43	44	C44	0.019650	0.040350	10000	3.00	4.00	0.03	0.02
56	44	45	C45	0.131000	0.269000	10000	4.00	4.00	0.80	0.70
57	45	46	C46	0.235800	0.484200	10000	5.00	4.00	0.20	0.15
58	G	47	C47	0.040500	0.138000	10000	6.00	12.00	0.00	0.00
59	47	48	C48	0.065500	0.134500	10000	7.00	12.00	0.00	0.00
60	48	49	C49	0.065500	0.134500	10000	8.00	12.00	0.00	0.00
61	49	50	C50	0.039300	0.080700	10000	9.00	12.00	0.20	0.16
62	50	51	C51	0.078600	0.161400	10000	10.00	12.00	0.80	0.60
63	51	52	C52	0.006550	0.013450	10000	16.00	12.00	0.50	0.30
64	52	53	C53	0.078600	0.161400	10000	16.00	11.00	0.50	0.35
65	53	54	C54	0.010480	0.021520	10000	11.00	11.00	0.50	0.30
66	54	55	C55	0.131000	0.269000	10000	10.00	11.00	0.20	0.08
67	H	56	C56	0.028350	0.096600	10000	8.00	9.00	0.00	0.00
68	56	57	C57	0.537100	1.102900	10000	9.00	9.00	0.03	0.02
69	57	58	C58	0.052400	0.107600	10000	10.00	9.00	0.60	0.42
70	58	59	C59	0.040500	0.138000	10000	11.00	9.00	0.00	0.00
71	59	60	C60	0.039300	0.080700	10000	12.00	9.00	0.02	0.01
72	60	61	C61	0.026200	0.053800	10000	13.00	9.00	0.02	0.01
73	61	62	C62	0.104800	0.215200	10000	14.00	9.00	0.20	0.13

Linha	De	Para	Cabo	R ( $\Omega$ /km)	XL ( $\Omega$ /km)	Imax (A)	X (km)	Y (km)	P(MW)	Q(Mvar)
74	62	63	C63	0.235800	0.484200	10000	15.00	9.00	0.30	0.24
75	63	64	C64	0.024300	0.082800	10000	16.00	9.00	0.30	0.20
76	I	65	C65	0.048600	0.165600	10000	1.00	6.00	0.00	0.00
77	65	66	C66	0.170300	0.349700	10000	2.00	6.00	0.05	0.03
78	66	67	C67	0.121500	0.414000	10000	3.00	6.00	0.00	0.00
79	67	68	C68	0.218700	0.745200	10000	4.00	6.00	0.40	0.36
80	68	69	C69	0.048600	0.165600	10000	5.00	6.00	0.00	0.00
81	69	70	C70	0.072900	0.248400	10000	6.00	6.00	0.00	0.00
82	70	71	C71	0.056700	0.193200	10000	7.00	6.00	2.00	1.50
83	71	72	C72	0.004367	0.008800	10000	13.00	6.00	0.20	0.15
84	J	73	C73	0.324000	1.102400	10000	1.00	7.00	0.00	0.00
85	73	74	C74	0.032400	0.110400	10000	2.00	7.00	0.00	0.00
86	74	75	C75	0.056700	0.193200	10000	3.00	7.00	1.20	0.95
87	75	76	C76	0.004418	0.015055	10000	14.00	7.00	0.30	0.18
88	K	77	C77	0.050220	0.171120	10000	5.00	8.00	0.00	0.00
89	77	78	C78	0.064800	0.220800	10000	7.00	8.00	0.40	0.36
90	78	79	C79	0.024300	0.082800	10000	9.00	8.00	2.00	1.30
91	79	80	C80	0.065500	0.132000	10000	11.00	8.00	0.20	0.14
92	80	81	C81	0.065500	0.132000	10000	13.00	8.00	0.50	0.36
93	81	82	C82	0.091700	0.188300	10000	14.00	8.00	0.10	0.03
94	82	83	C83	0.157200	0.322900	10000	16.00	8.00	0.40	0.36
95	5	55	C84	0.131000	0.269000	10000				
96	7	60	C85	0.131000	0.269000	10000				
97	11	43	C86	0.131000	0.269000	10000				
98	12	72	C87	0.340600	0.699400	10000				
99	13	76	C88	0.229250	0.470750	10000				
100	14	18	C89	0.179033	0.360800	10000				
101	16	26	C90	0.091700	0.188300	10000				
102	20	83	C91	0.011229	0.023057	10000				
103	28	32	C92	0.023434	0.048120	10000				
104	29	39	C93	0.039300	0.080700	10000				
105	34	46	C94	0.026200	0.053800	10000				
106	40	42	C95	0.196500	0.403500	10000				
107	53	64	C96	0.019650	0.040350	10000				

### A.5 Sistema de Mantovani et al (2000)

Este sistema é composto por 137-nós, sendo 2 subestações com tensão nominal 13.8 kV. Apresentam-se na Tabela A.5 seus dados de entrada.

Tabela A.5 – Dados de entrada referentes ao sistema 137-nós.

Linha	De	Para	Cabo	R ( $\Omega$ /km)	XL ( $\Omega$ /km)	I <sub>max</sub> (A)	X (km)	Y (km)	P(MW)	Q(Mvar)
1		0E					1.00	7.00		
2		0D					26.00	7.00		
3	0E	1	C1	0.065120	0.150329	10000	2.00	12.00	0.00	0.00
4	1	2	C2	0.001880	0.004330	10000	3.00	12.00	0.05	0.02
5	2	3	C3	0.223240	0.515350	10000	4.00	12.00	0.04	0.02
6	3	4	C4	0.099430	0.229530	10000	5.00	12.00	0.09	0.03
7	4	5	C5	0.155710	0.359450	10000	6.00	12.00	0.31	0.12
8	5	6	C6	0.163210	0.376770	10000	7.00	12.00	0.15	0.06
9	6	7	C7	0.114440	0.264170	10000	7.00	13.00	0.24	0.09
10	6	9	C8	0.056750	0.056660	10000	8.00	12.00	0.06	0.02
11	9	10	C9	0.521240	0.274180	10000	8.00	11.00	0.12	0.05
12	9	12	C10	0.108770	0.108600	10000	9.00	12.00	0.14	0.06
13	12	13	C11	0.796060	0.418740	10000	9.00	12.50	0.12	0.05
14	12	15	C12	0.917440	0.314690	10000	9.00	11.00	0.25	0.10
15	12	17	C13	0.118230	0.118050	10000	10.00	12.00	0.29	0.12
16	17	18	C14	0.502280	0.264210	10000	11.00	12.00	0.30	0.12
17	17	20	C15	0.056750	0.056660	10000	10.00	11.00	0.22	0.09
18	20	21	C16	0.293790	0.154540	10000	10.00	10.00	0.20	0.08
19	0E	24	C17	0.234795	0.542019	10000	2.00	8.00	0.00	0.00
20	24	25	C18	0.001880	0.004330	10000	3.00	8.00	0.00	0.00
21	25	26	C19	0.223240	0.515350	10000	3.00	9.00	0.00	0.00
22	26	27	C20	0.108810	0.251180	10000	4.00	9.00	0.03	0.01
23	27	28	C21	0.710780	0.373880	10000	4.00	10.00	0.23	0.11
24	27	29	C22	0.181970	0.420079	10000	5.00	9.00	0.06	0.03
25	29	30	C23	0.303260	0.159520	10000	5.00	10.00	0.23	0.11
26	29	32	C24	0.024390	0.056300	10000	6.00	9.00	0.12	0.06
27	32	33	C25	0.045020	0.103940	10000	7.00	9.00	0.00	0.00
28	33	34	C26	0.018760	0.043310	10000	8.00	9.00	0.06	0.03
29	34	35	C27	0.118230	0.118050	10000	9.00	9.00	0.36	0.18
30	35	36	C28	0.023650	0.023610	10000	10.00	9.00	0.00	0.00
31	36	37	C29	0.189540	0.099700	10000	11.00	9.00	0.12	0.06
32	37	38	C30	0.398030	0.209370	10000	12.00	9.00	0.06	0.03
33	36	39	C31	0.056750	0.056660	10000	10.00	8.00	0.00	0.00
34	39	40	C32	0.094770	0.049850	10000	9.00	8.00	0.09	0.04
35	40	41	C33	0.416990	0.219340	10000	8.00	8.00	0.00	0.00
36	41	42	C34	0.227440	0.119640	10000	7.50	8.00	0.40	0.19
37	39	43	C35	0.075660	0.075550	10000	11.00	8.00	0.00	0.00
38	43	44	C36	0.369600	0.194420	10000	11.00	7.00	0.18	0.09
39	44	45	C37	0.265360	0.139580	10000	12.00	7.00	0.24	0.12

Linha	De	Para	Cabo	R ( $\Omega/\text{km}$ )	XL ( $\Omega/\text{km}$ )	I <sub>max</sub> (A)	X (km)	Y (km)	P(MW)	Q(Mvar)
40	43	46	C38	0.056750	0.056660	10000	12.00	8.00	0.08	0.04
41	0E	48	C39	0.234795	0.542019	10000	2.00	6.00	0.00	0.00
42	48	49	C40	0.118190	0.272830	10000	3.00	6.00	0.00	0.00
43	49	50	C41	2.962880	1.016280	10000	4.00	6.00	0.01	0.00
44	49	51	C42	0.001880	0.004330	10000	3.00	5.00	0.00	0.00
45	51	52	C43	0.069410	0.160240	10000	3.00	4.00	0.12	0.05
46	52	53	C44	0.815020	0.428720	10000	3.00	3.00	0.06	0.03
47	52	54	C45	0.063780	0.147240	10000	4.00	4.00	0.17	0.07
48	54	55	C46	0.131320	0.303150	10000	5.00	4.00	0.46	0.19
49	55	56	C47	0.061910	0.142910	10000	6.00	4.00	0.26	0.11
50	56	57	C48	0.114440	0.264170	10000	7.00	4.00	0.24	0.10
51	57	58	C49	0.283740	0.283310	10000	8.00	4.00	0.00	0.00
52	58	59	C50	0.283740	0.283310	10000	9.00	4.00	0.11	0.05
53	57	61	C51	0.022510	0.051970	10000	7.00	6.00	0.00	0.00
54	61	62	C52	0.026260	0.060630	10000	8.00	6.00	0.07	0.03
55	62	63	C53	0.060030	0.138580	10000	9.00	6.00	0.26	0.11
56	63	64	C54	0.030020	0.069290	10000	10.00	6.00	0.07	0.03
57	64	65	C55	0.020640	0.047640	10000	11.00	6.00	0.02	0.01
58	62	67	C56	0.108810	0.251180	10000	8.00	5.00	0.00	0.00
59	67	68	C57	0.255880	0.134600	10000	9.00	5.00	0.02	0.01
60	68	69	C58	0.416990	0.219340	10000	10.00	5.00	0.15	0.06
61	69	70	C59	0.502280	0.264210	10000	11.00	5.00	0.22	0.09
62	70	71	C60	0.331700	0.174480	10000	12.00	5.00	0.09	0.04
63	71	72	C61	0.208490	0.109670	10000	13.00	5.00	0.00	0.00
64	56	73	C62	0.098161	0.226607	10000	7.00	3.00	0.23	0.10
65	0D	75	C63	0.003354	0.007746	10000	25.00	9.00	0.00	0.00
66	75	76	C64	0.040270	0.092964	10000	19.00	12.00	0.29	0.12
67	76	77	C65	0.382700	0.883460	10000	18.00	12.00	0.08	0.03
68	77	78	C66	0.330180	0.762200	10000	17.00	12.00	0.08	0.03
69	78	80	C67	0.328300	0.757870	10000	16.00	12.00	0.10	0.04
70	80	81	C68	0.170720	0.394090	10000	15.00	12.00	0.18	0.07
71	81	82	C69	0.559140	0.294120	10000	15.00	11.00	0.08	0.03
72	81	83	C70	0.058160	0.134250	10000	14.00	12.00	0.22	0.09
73	83	84	C71	0.701300	0.368900	10000	14.00	11.00	0.02	0.01
74	84	85	C72	1.023520	0.538390	10000	13.00	11.00	0.01	0.00
75	83	86	C73	0.067540	0.155910	10000	13.00	12.00	0.07	0.03
76	86	87	C74	1.323520	0.453970	10000	13.00	13.00	0.41	0.16
77	0D	121	C75	0.007962	0.018371	10000	25.00	8.00	0.00	0.00
78	121	122	C76	0.243253	0.561547	10000	22.00	8.00	0.10	0.04
79	122	123	C77	0.225120	0.519680	10000	21.00	8.00	0.14	0.06

Linha	De	Para	Cabo	R ( $\Omega/\text{km}$ )	XL ( $\Omega/\text{km}$ )	I <sub>max</sub> (A)	X (km)	Y (km)	P(MW)	Q(Mvar)
80	123	124	C78	0.208240	0.480710	10000	20.00	8.00	0.10	0.04
81	124	125	C79	0.020974	0.048420	10000	19.00	10.00	0.30	0.13
82	125	127	C80	0.619500	0.618570	10000	18.00	10.00	0.14	0.06
83	127	128	C81	0.340490	0.339980	10000	17.00	10.00	0.28	0.12
84	128	129	C82	0.568620	0.299110	10000	17.00	11.00	0.09	0.04
85	128	130	C83	0.054385	0.054300	10000	15.00	10.00	0.24	0.10
86	130	131	C84	0.568620	0.299110	10000	14.00	10.00	0.25	0.11
87	0D	133	C85	0.003561	0.008216	10000	25.00	4.00	0.00	0.00
88	133	134	C86	0.209175	0.482875	10000	23.00	4.00	0.09	0.04
89	134	135	C87	0.104990	0.136410	10000	23.00	5.00	1.14	0.48
90	134	136	C88	0.438980	1.013380	10000	22.00	4.00	0.46	0.19
91	136	137	C89	0.075200	0.025790	10000	21.00	4.00	0.39	0.16
92	137	138	C90	0.076920	0.177560	10000	20.00	4.00	0.00	0.00
93	138	139	C91	0.332050	0.766530	10000	19.00	4.00	0.08	0.03
94	139	141	C92	0.084420	0.194880	10000	18.00	4.00	0.09	0.04
95	141	142	C93	0.133200	0.307480	10000	17.00	4.00	0.00	0.00
96	142	143	C94	0.293200	0.292760	10000	16.00	4.00	0.07	0.03
97	143	144	C95	0.217530	0.217210	10000	15.00	4.00	0.23	0.10
98	144	145	C96	0.264820	0.264430	10000	14.00	4.00	0.14	0.06
99	142	146	C97	0.036480	0.084213	10000	15.00	6.00	0.00	0.00
100	146	147	C98	0.135070	0.311810	10000	14.00	6.00	0.08	0.03
101	0D	148	C99	0.002275	0.005251	10000	25.00	3.00	0.00	0.00
102	148	149	C100	0.168840	0.389760	10000	24.00	3.00	0.05	0.02
103	149	150	C101	0.059095	0.136415	10000	22.00	3.00	0.06	0.03
104	150	152	C102	2.286080	0.784140	10000	22.00	2.00	0.01	0.00
105	150	153	C103	0.227935	0.526180	10000	20.00	3.00	0.00	0.00
106	153	154	C104	0.696000	1.606690	10000	19.00	3.00	0.02	0.01
107	154	155	C105	0.045547	0.105145	10000	9.00	2.00	1.51	0.64
108	155	156	C106	0.202980	0.263730	10000	8.00	2.00	0.31	0.13
109	156	157	C107	0.213480	0.277370	10000	7.00	2.00	0.08	0.03
110	157	158	C108	0.549670	0.289140	10000	7.00	1.00	0.05	0.02
111	158	159	C109	0.540190	0.284150	10000	6.00	1.00	0.00	0.00
112	157	160	C110	0.045500	0.059110	10000	6.00	2.00	0.20	0.09
113	160	162	C111	0.473850	0.249260	10000	5.00	2.00	0.06	0.03
114	162	163	C112	0.862410	0.453640	10000	4.00	2.00	0.05	0.02
115	163	164	C113	0.568620	0.299110	10000	3.00	2.00	0.00	0.00
116	158	200	C114	0.777110	0.408780	10000	8.00	1.00	0.16	0.07
117	200	201	C115	1.080380	0.568300	10000	9.00	1.00	0.00	0.00
118	159	202	C116	1.099330	0.578270	10000	5.00	1.00	0.25	0.11
119	202	203	C117	0.473850	0.249260	10000	4.00	1.00	0.00	0.00

Linha	De	Para	Cabo	R ( $\Omega/\text{km}$ )	XL ( $\Omega/\text{km}$ )	Imax (A)	X (km)	Y (km)	P(MW)	Q(Mvar)
120	154	204	C118	0.322670	0.744880	10000	18.00	3.00	0.07	0.03
121	204	205	C119	0.146330	0.337790	10000	17.00	3.00	0.03	0.01
122	205	206	C120	0.123820	0.285830	10000	16.00	3.00	0.06	0.03
123	0D	207	C121	0.007962	0.018371	10000	25.00	6.00	0.00	0.00
124	207	208	C122	0.649100	1.498420	10000	24.00	6.00	0.09	0.05
125	208	209	C123	0.045020	0.103940	10000	23.00	6.00	0.05	0.02
126	209	210	C124	0.526400	0.180560	10000	23.00	7.00	0.12	0.06
127	209	211	C125	0.020640	0.047640	10000	22.00	6.00	0.08	0.04
128	211	212	C126	0.530710	0.279170	10000	22.00	7.00	0.15	0.07
129	211	214	C127	0.097550	0.225200	10000	21.00	6.00	0.02	0.01
130	214	215	C128	0.118190	0.272830	10000	21.00	7.00	0.07	0.04
131	214	217	C129	0.138820	0.320470	10000	20.00	6.00	0.23	0.11
132	217	218	C130	0.043150	0.099610	10000	19.00	6.00	0.04	0.02
133	218	219	C131	0.029068	0.067104	10000	18.00	9.00	0.25	0.12
134	219	220	C132	0.161340	0.372440	10000	17.00	9.00	0.32	0.15
135	220	221	C133	0.378320	0.377750	10000	16.00	9.00	0.33	0.16
136	221	222	C134	0.397240	0.396640	10000	15.00	9.00	0.25	0.12
137	222	223	C135	0.293200	0.292760	10000	14.00	9.00	0.00	0.00
138	7	86	C136	0.021589	0.049838	10000				
139	10	32	C137	0.093819	0.049349	10000				
140	20	130	C138	0.027823	0.027782	10000				
141	46	223	C139	0.038067	0.038009	10000				
142	33	61	C140	0.015007	0.034647	10000				
143	59	145	C141	0.028374	0.028332	10000				
144	65	147	C142	0.047290	0.047220	10000				
145	73	206	C143	0.004378	0.010104	10000				
146	78	125	C144	0.045764	0.105649	10000				
147	125	219	C145	0.011936	0.027563	10000				
148	131	223	C146	0.331700	0.174480	10000				
149	139	154	C147	0.141870	0.141660	10000				
150	138	217	C148	0.038460	0.088780	10000				
151	138	153	C149	0.076920	0.177560	10000				
152	141	154	C150	0.054391	0.125554	10000				
153	141	220	C151	0.015085	0.034822	10000				
154	145	206	C152	0.118431	0.118257	10000				
155	160	56	C153	0.248480	0.322835	10000				
156	212	122	C154	0.170590	0.089730	10000				
157	215	123	C155	0.052530	0.121260	10000				
158	223	147	C156	0.097733	0.097587	10000				