

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**  
Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial

---

**DISSERTAÇÃO**

apresentada à UTFPR  
para obtenção do grau de  
**MESTRE EM CIÊNCIAS**

por

**PAULO NEIS**

---

**PROCESSAMENTO INTELIGENTE DE ALARMES  
EMPREGANDO ALGORITMOS GENÉTICOS**

---

Banca Examinadora:

Presidente e Orientadora:

**PROF.<sup>a</sup> DR.<sup>a</sup> MYRIAM REGATTIERI DELGADO**                      **UTFPR**

Examinadores:

**PROF.<sup>a</sup> DR.<sup>a</sup> AURORA TRINIDAD RAMIREZ POZO**                      **UFPR**

**PROF. DR. RICARDO LÜDERS**    **UTFPR**

**PROF. DR. VÓLDI COSTA ZAMBENEDETTI**    **LACTEC**

Curitiba, agosto de 2006

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.



**PAULO NEIS**

***PROCESSAMENTO INTELIGENTE DE  
ALARMES EMPREGANDO ALGORITMOS  
GENÉTICOS***

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial da Universidade Tecnológica Federal do Paraná como requisito parcial para a obtenção do grau de Mestre em Ciências.  
Área de Concentração: Informática Industrial.

Orientadora:  
Prof.<sup>a</sup> Dr.<sup>a</sup> MYRIAM REGATTIERI DELGADO

Curitiba

2006

Ficha catalográfica elaborada pela Biblioteca da UTFPR – Campus Curitiba

N416p Neis, Paulo  
Processamento inteligente de alarmes empregando algoritmos genéticos /  
Paulo Neis. Curitiba. UTFPR, 2006  
XXII, 114 f. : il.; 30 cm

Orientadora: Prof. Dr<sup>a</sup>. Myriam Regattieri Delgado  
Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná. Curso de Pós-Graduação em Engenharia Elétrica e Informática Industrial. Curitiba, 2006  
Bibliografia: f. 105-109

1. Sistemas eletrônicos de potência. 2. Algoritmos genéticos. 3. Processamento inteligente de alarmes. 4. Engenharia elétrica. I. Delgado, Myriam Regattieri, orient. II. Universidade Tecnológica Federal do Paraná. Curso de Pós-Graduação em Engenharia Elétrica e Informática Industrial. III. Título.

CDD: 519.72

## *Agradecimentos*

**Meus sinceros agradecimentos às seguintes pessoas, pela contribuição a este trabalho:**

**Prof.<sup>a</sup> Dra.<sup>a</sup> Myriam Regattieri Delgado**

Pela orientação e pelos ensinamentos, pela paciência e compreensão.

**Prof. Dr. Ricardo Lüders e Prof. Dr. Heitor Silvério Lopes**

Pelas lições e sugestões.

**Eng. Arlenio Carneiro Frisch e Eng. Carlos César Krauss**

Pelo apoio, encaminhamento e viabilização deste trabalho.

**Eng. Carlos Baggetti, Eng. Paulo Stocker,**

**Eng. Rafael Ramirez e Eng. Miguel Correia Netto**

Pela contribuição técnica.

**Aos membros da Banca Examinadora**

Pelas importantes observações e sugestões.



*“... e os tais acentos a torto e a direito que os reformadores oficiais impuseram à nova ortografia vêm complicar, vêm contrariar a lei da evolução! São, pois, uma coisa incientífica, tola, imbecil, cretinizante e que deve ser violentamente repelida por todas as pessoas decentes.*

*... Se alguém me escreve uma carta cheia de acentos, encosto-a. Não leio. E se vem alguma com trema, devolvo-a, nobremente enojado...”*

**Monteiro Lobato**



# *Sumário*

<b>Lista de Figuras</b>	p. xiii
<b>Lista de Tabelas</b>	p. xv
<b>Lista de Abreviaturas e Siglas</b>	p. xvii
<b>Resumo</b>	p. xix
<b>Abstract</b>	p. xxi
<b>1 Introdução</b>	p. 1
1.1 Caracterização do Problema . . . . .	p. 1
1.2 Motivações . . . . .	p. 3
1.3 Objetivos . . . . .	p. 4
1.3.1 Uma Ferramenta para Processamento de Alarmes . . . . .	p. 5
1.3.2 Investigações e Aplicação dos Algoritmos Genéticos . . . . .	p. 5
1.3.3 Estudo de Caso: O Sistema SASE . . . . .	p. 6
1.4 Estrutura da Dissertação . . . . .	p. 7
<b>2 Fundamentação Teórica</b>	p. 9
2.1 Algoritmos Genéticos . . . . .	p. 9
2.1.1 Teoria da Evolução . . . . .	p. 9
2.1.2 Algoritmos Genéticos: “Darwinismo Computacional” . . . . .	p. 10
2.1.3 Parâmetros do AG . . . . .	p. 11
2.1.3.1 Codificação . . . . .	p. 11

2.1.3.2	Determinação da Função de <i>Fitness</i> . . . . .	p. 12
2.1.3.3	Métodos de Seleção . . . . .	p. 12
2.1.3.4	Operadores Genéticos . . . . .	p. 15
2.1.3.5	Método de Inicialização e Critério de Parada . . . . .	p. 17
2.1.3.6	Determinação dos demais parâmetros . . . . .	p. 17
2.1.4	Funcionamento do AG Canônico . . . . .	p. 17
2.1.5	Teorias Sobre o Funcionamento do AG . . . . .	p. 20
2.1.6	Algoritmos Genéticos Paralelos . . . . .	p. 21
2.1.6.1	AG Paralelo Global de População Única (Mestre-Escravo) . . . . .	p. 22
2.1.6.2	AG massivamente Paralelo de População Única . . . . .	p. 23
2.1.6.3	AG paralelo multipopulacional . . . . .	p. 24
2.1.7	Algoritmos Genéticos “ <i>Messy</i> ” . . . . .	p. 25
2.1.7.1	Representação do AGm . . . . .	p. 25
2.1.7.2	Principais Operações no AGm . . . . .	p. 26
2.1.7.3	Funcionamento do AGm . . . . .	p. 28
2.1.8	Principais Vantagens do AG . . . . .	p. 29
2.1.9	Desvantagens do AG . . . . .	p. 30
2.1.10	Algoritmos Genéticos em Aplicações de Tempo Real . . . . .	p. 31
2.2	O Problema da Cobertura de Conjuntos . . . . .	p. 32
<b>3</b>	<b>Processamento de Alarmes</b> . . . . .	p. 35
3.1	Alarmes e Eventos . . . . .	p. 35
3.2	Processamento de Alarmes e Diagnóstico de Falhas . . . . .	p. 36
3.3	Processamento Inteligente de Alarmes . . . . .	p. 36
3.4	Arquitetura do Sistema . . . . .	p. 37
3.5	Justificativas do Processamento Inteligente de Alarmes . . . . .	p. 38
3.5.1	Objetivos do Processamento de Alarmes . . . . .	p. 38

3.6	Métodos Empregados no Processamento Inteligente de Alarmes . . . . .	p. 39
3.6.1	Sistemas Baseados em Regras / Baseados em Conhecimento . . . . .	p. 39
3.6.2	Reconhecimento de Padrões . . . . .	p. 41
3.6.3	Redes Neurais Artificiais (RNA) . . . . .	p. 41
3.6.4	Sistemas de Inferência <i>Fuzzy</i> . . . . .	p. 41
3.6.5	Algoritmos Genéticos . . . . .	p. 42
3.6.5.1	Relação entre Eventos e Alarmes . . . . .	p. 42
3.6.5.2	Critério para descrição do problema . . . . .	p. 43
3.6.5.3	Implementação Apresentada Pelos Autores . . . . .	p. 45
3.6.5.4	Deficiências da Implementação . . . . .	p. 45
<b>4</b>	<b>Uma Nova Abordagem para o Processamento de Alarmes</b>	p. 47
4.1	Processamento de Alarmes e Cobertura de Conjuntos . . . . .	p. 47
4.2	Solucionando o Problema do Processamento de Alarmes com AG . . . . .	p. 49
4.2.1	O Modelo "Clássico" . . . . .	p. 49
4.2.2	Problemas no Modelo . . . . .	p. 50
4.2.3	Melhorias Necessárias . . . . .	p. 50
4.2.4	Modelo Melhorado 1: Vetor de Eventos Inteiros . . . . .	p. 51
4.2.4.1	Exemplo Numérico . . . . .	p. 52
4.2.5	Modelo Melhorado 2: Palavra de Eventos Sobre o Alfabeto $E_s$ . . . . .	p. 54
4.2.5.1	Uma Nova Interpretação para o Problema . . . . .	p. 54
4.2.5.2	Considerações Sobre o Modelo . . . . .	p. 55
4.2.5.3	Solucionando o Problema com AG Convencional . . . . .	p. 56
4.2.5.4	Inconvenientes do Modelo . . . . .	p. 56
4.2.5.5	Exemplo Numérico . . . . .	p. 57
4.2.6	Algoritmo Genético com Cromossomo de Tamanho Variável . . . . .	p. 57
4.2.6.1	Resolvendo uma Instância do Problema . . . . .	p. 60

4.2.7	Modelo Melhorado 3: Palavra de Eventos com SOE . . . . .	p. 60
4.3	Restrição do Espaço de Busca . . . . .	p. 61
<b>5</b>	<b>Implementação do Sistema</b>	<b>p. 63</b>
5.1	Materiais e Infra-estrutura . . . . .	p. 63
5.1.1	Hierarquia de Controle e Supervisão do Sistema Elétrico . . . . .	p. 63
5.1.2	Sistema de Automação de Subestações da COPEL . . . . .	p. 64
5.1.2.1	O Conceito de Alarme no SASE . . . . .	p. 65
5.1.3	A Subestação Escolhida . . . . .	p. 66
5.2	Construção da Base de Conhecimento . . . . .	p. 67
5.3	Ferramentas Computacionais Empregadas . . . . .	p. 69
5.4	Construção do Protótipo . . . . .	p. 69
5.4.1	Integração com o Sistema SASE . . . . .	p. 69
5.4.2	Disparo do Algoritmo . . . . .	p. 71
5.5	Aquisição de Dados . . . . .	p. 71
<b>6</b>	<b>Resultados</b>	<b>p. 73</b>
6.1	Critérios de Avaliação . . . . .	p. 73
6.1.1	Métricas . . . . .	p. 73
6.1.1.1	Erros e Acertos . . . . .	p. 73
6.1.1.2	Capacidade de Síntese . . . . .	p. 75
6.1.2	Estatísticas . . . . .	p. 76
6.1.2.1	Medida de Tendência . . . . .	p. 76
6.1.2.2	Medida de Dispersão . . . . .	p. 76
6.2	Metodologia de Testes . . . . .	p. 77
6.2.1	Fluxograma de Testes . . . . .	p. 77
6.2.2	Parâmetros de Avaliação . . . . .	p. 79

6.2.2.1	Tipo do AG . . . . .	p. 79
6.2.2.2	Número de Eventos Simultâneos . . . . .	p. 80
6.2.2.3	Tamanho da População . . . . .	p. 80
6.2.2.4	Ruído . . . . .	p. 80
6.2.3	Outros Parâmetros do AG . . . . .	p. 81
6.2.3.1	Critérios de Parada . . . . .	p. 81
6.2.3.2	Método de Seleção . . . . .	p. 81
6.2.3.3	Probabilidade de Mutação e Cruzamento . . . . .	p. 81
6.2.3.4	Operador de Cruzamento . . . . .	p. 81
6.2.3.5	AG Multipopulacional . . . . .	p. 81
6.3	Considerações Sobre “AG1” e “AG2” . . . . .	p. 82
6.4	Avaliação Segundo os Diversos Parâmetros . . . . .	p. 84
6.4.1	Desempenho em Função do Tamanho da População . . . . .	p. 84
6.4.1.1	Acertos . . . . .	p. 84
6.4.1.2	Erros . . . . .	p. 86
6.4.1.3	Tempo de Cálculo . . . . .	p. 88
6.4.2	Desempenho em Função do Ruído . . . . .	p. 90
6.4.2.1	Acertos . . . . .	p. 90
6.4.2.2	Erros . . . . .	p. 92
6.4.3	Desempenho em Função do Número de Eventos . . . . .	p. 93
6.4.4	Desempenho Geral em Função do Tipo do AG . . . . .	p. 94
6.4.4.1	Representatividade dos Resultados . . . . .	p. 95
6.5	Redução do Número de Mensagens . . . . .	p. 96
6.6	Comparação com Outros Trabalhos . . . . .	p. 96
<b>7</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>p. 101</b>
7.1	Conclusões . . . . .	p. 101

7.1.1	Considerações Sobre Desempenho em Tempo Real . . . . .	p. 103
7.2	Trabalhos Futuros . . . . .	p. 103
7.2.1	Implantação de um Sistema Piloto . . . . .	p. 103
7.2.2	Melhorias nos Algoritmos . . . . .	p. 104
	<b>Referências</b>	p. 105
	<b>Apêndice A – Diagrama de Sequência</b>	p. 111
	<b>Apêndice B – Diagrama de Classes</b>	p. 113

## *Lista de Figuras*

1	Desempenho do operador em função do <i>stress</i> . . . . .	p. 3
2	Representação esquemática do COS, COEs e SEs. . . . .	p. 7
3	Seleção pelo método da roleta. . . . .	p. 13
4	Seleção pelo método do ordenamento linear. . . . .	p. 14
5	Cruzamento de um ponto e de dois pontos. . . . .	p. 16
6	Cruzamento uniforme. . . . .	p. 16
7	Operador de mutação. . . . .	p. 17
8	Fluxograma de execução do AG. . . . .	p. 18
9	AG paralelo com arquitetura mestre-escravo. . . . .	p. 22
10	AG massivamente paralelo. . . . .	p. 23
11	AG paralelo insular. . . . .	p. 24
12	Operador de “cortar e colar”. . . . .	p. 27
13	Fluxograma do AG <i>Messy</i> . . . . .	p. 28
14	Uma instância do problema da cobertura de conjuntos. . . . .	p. 33
15	Listagem de alarmes. . . . .	p. 46
16	Processamento de alarmes e cobertura de conjuntos. . . . .	p. 48
17	Alarmes reportados. . . . .	p. 53
18	Fluxograma do AG com cromossomo variável. . . . .	p. 58
19	Alarmes reportados para os eventos $e_1$ e $e_3$ . . . . .	p. 60
20	Alarmes reportados para os eventos $e_1$ e $e_3$ , ocorrência simultânea. . . . .	p. 61
21	Hierarquia de Controle e Supervisão do Sistema Elétrico na COPEL. . . . .	p. 64
22	Diagrama unifilar da SE Batel. . . . .	p. 67

23	Regra no formato XML. . . . .	p. 68
24	Arquitetura do sistema. . . . .	p. 70
25	Interface com o operador. . . . .	p. 71
26	Diagramas dos conjuntos Esperado e Diagnosticado. . . . .	p. 75
27	Fluxograma dos testes executados. . . . .	p. 78
28	Topologia utilizada para o AG de múltiplas populações. . . . .	p. 82
29	Percentual médio de acertos em função do tamanho da população - AG1 e AG2. . . . .	p. 83
30	Distribuição de frequência de acertos - AG1 e AG2. . . . .	p. 83
31	Percentual médio de erros em função do tamanho da população - AG1 e AG2. . . . .	p. 84
32	Percentual médio de acertos em função do tamanho da população. . . . .	p. 85
33	Distribuição de frequência de acertos para diferentes tamanhos de população. . . . .	p. 86
34	Percentual médio de erros em função do tamanho da população. . . . .	p. 87
35	Distribuição de frequência de erros para diferentes tamanhos de população. . . . .	p. 88
36	Tempo médio de cálculo (s) em função do tamanho da população. . . . .	p. 89
37	Percentual médio de acertos em função do ruído. . . . .	p. 91
38	Percentual médio de erros em função do percentual de ruído. . . . .	p. 92
39	Percentual médio de acertos em função do número de eventos. . . . .	p. 93
40	Percentual médio de erros em função do número de eventos. . . . .	p. 94
41	Percentual médio de erros e acertos de cada algoritmo. . . . .	p. 94
42	Percentual médio de acertos em função do tamanho da população - Modelo binário. . . . .	p. 98
43	Percentual médio de erros em função do tamanho da população - Modelo binário. . . . .	p. 99
44	Diagrama de Sequência do AG com Cromossomo de Tamanho Variável. . . . .	p. 112
45	Diagrama de Classes do AG com Cromossomo de Tamanho Variável. . . . .	p. 114

## *Lista de Tabelas*

1	População hipotética, e seus valores de adequabilidade. . . . .	p. 13
2	Tabela-verdade da operação (-). . . . .	p. 44
3	Conjunto de eventos e alarmes do sistema. . . . .	p. 52
4	Relação eventos/alarmes na forma tabular. . . . .	p. 62
5	Correlação entre eventos e alarmes. . . . .	p. 67
6	Diferentes parâmetros de simulação. . . . .	p. 79
7	Comparação entre os diferentes AGs utilizados. . . . .	p. 80
8	Percentual de acertos em função do tamanho da população. . . . .	p. 85
9	Percentual de erros em função do tamanho da população. . . . .	p. 87
10	Tempo médio de cálculo (s) em função do tamanho da população. . . . .	p. 89
11	Percentual médio de acertos em função do percentual de ruído. . . . .	p. 91
12	Percentual médio de erros em função do percentual de ruído. . . . .	p. 92
13	Percentual de acertos e erros em função do tipo de AG. . . . .	p. 95
14	Valores “ <i>p</i> ” retornados pelo teste de Kruskal-Wallis. . . . .	p. 95
15	Percentual médio de redução do número de mensagens. . . . .	p. 96
16	Percentual de acertos em função do tamanho da população - Modelo binário. . . . .	p. 98
17	Percentual de erros em função do tamanho da população - Modelo binário. . . . .	p. 98
18	Comparação com o melhor resultado. . . . .	p. 99



## *Lista de Abreviaturas e Siglas*

**AG:** Algoritmo Genético

**AGC:** *Automatic Generation Control*

**AGm:** Algoritmo Genético *Messy*

**COD:** Centro de Operação da Distribuição

**COE:** Centro de Operação de Estações

**COPEL:** Companhia Paranaense de Energia

**CORBA:** *Common Object Request Broker Architecture*

**COS:** Centro de Operação do Sistema

**EMS:** *Energy Management System*

**GALOPPS:** *The Genetic ALgorithm Optimized for Portability and Parallelism System*

**GCC:** *GNU Compiler Collection*

**GPL:** *GNU General Public License*

**IA:** Inteligência Artificial

**MFD:** *Multiple Fault Diagnosis*

**NA:** Normalmente Aberto

**NF:** Normalmente Fechado

**NP:** Non-deterministic Polynomial time

**PGA:** Ponto Gera Alarme

**RNA:** Rede Neural Artificial

**SASE:** Sistema de Automação de Subestações da COPEL

**SCADA:** *Supervisory Control and Data Acquisition*

**SE:** Subestação

**SOE:** *Sequence Of Events*

**STL:** *Standard Template Library*

**TAO:** *The Ace ORB*

**UAC:** Unidade de Aquisição e Controle

**UCC:** Unidade Central de Controle

**XML:** *eXtensible Markup Language*

## *Resumo*

Em um centro de operação de um sistema elétrico de potência, um pequeno grupo de operadores é responsável pela supervisão de uma área que geralmente compreende dezenas de subestações, englobando vários milhares de pontos de medida. Durante situações de emergência, como tempestades ou falhas em equipamentos, avalanches de mensagens de alarme são produzidas, sobrecarregando os operadores e criando sérias dificuldades de interpretação destas informações.

Ferramentas computacionais capazes de auxiliar o operador, provendo categorização, priorização, síntese e diagnóstico das mensagens de alarme tendem a diminuir a probabilidade de falhas causadas por interpretações equivocadas. O objetivo deste trabalho é propor um método de processamento inteligente de alarmes para ser utilizado associado aos aplicativos de controle e supervisão do sistema elétrico de potência. Este método emprega um tradicional paradigma da Computação Evolucionária, o “Algoritmo Genético”.

A modelagem do problema do processamento de alarmes é formulada de maneira semelhante ao problema da cobertura de conjuntos, e solucionada segundo um critério de otimização. Diversas formas de codificação e diferentes variações de algoritmos genéticos são utilizadas na solução, entre elas o AG de população única, de múltiplas populações, e um algoritmo inspirado no AG “*messy*”, proposto neste trabalho.

Um protótipo do processador de alarmes foi construído para validação do método proposto. Foram realizados testes utilizando uma base de dados de uma subestação real. O desempenho dos algoritmos foi avaliado considerando-se a variação de uma série de parâmetros, incluindo o tamanho das populações, número de eventos simultâneos ocorridos e percentual de ruído no conjunto de alarmes reportados. Os resultados obtidos revelam bom nível de exatidão, apresentando taxas de erro percentual aceitáveis, mesmo para casos ruidosos. O processador de alarmes mostrou-se capaz de operar “em tempo real”, ou seja, foi capaz de produzir o diagnóstico dentro de um intervalo de tempo consideravelmente menor do que a taxa máxima recomendada para a apresentação de informações ao operador.

**Palavras-chave:** Processamento Inteligente de Alarmes; Sistemas Elétricos de Potência; Algoritmos Genéticos.



## *Abstract*

In an electrical power system's control center, a small group of operators is responsible for the supervision of an area that usually comprehends several substations, including several thousands of measuring points. During emergency situations, like storms or equipment malfunctions, avalanches of alarm messages are issued, overloading the operators and creating serious difficulties of interpretation of this information.

Software tools capable of aiding the operator, providing categorization, prioritization, synthesis and diagnosis of the alarm messages, tend to reduce the probability of flaws caused by mistaken interpretations. The objective of this work is to propose a method of intelligent alarm processing for being used together with the SCADA/EMS that carries through the control and supervision of the electrical power system. This method uses a traditional paradigm of the Evolutionary Computation, the "Genetic Algorithm".

The modeling of the alarm processing problem is formulated in similar way to the set covering problem, and solved according to an optimization criterion. Several coding schemes and different variations of genetic algorithms are used in the solution, among them the single population GA, the multiple populations GA, and an algorithm inspired in the "messy" GA, proposed in this work.

A prototype of the alarm processor was built for the validation of the proposed method. Tests were carried through a database of a real substation. The algorithms performance was evaluated according to several parameters, such as the size of the populations, the number of simultaneous events that occurred and the percentile of noise in the set of alarms that were reported. The results revealed a good level of accuracy, and presented low error rates, even for noisy cases. The alarm processor was shown capable to operate "in real time", that is equivalent to say it could produce the diagnosis in an interval considerably shorter than the maximum update rate recommended for displaying information to the operator.

**Key-words:** Intelligent Alarm Processing; Electrical Power Systems; Genetic Algorithms



# 1 *Introdução*

## 1.1 **Caracterização do Problema**

A operação do sistema elétrico interligado requer a correta execução e coordenação de um grande conjunto de complexas funções por parte das empresas participantes (SMATHERS; AKHIL, 2001). É necessário que o fornecimento de energia aos consumidores seja mantido estável e confiável. Nesta tarefa, são empregadas diversas ferramentas de *software* para auxiliar o tratamento do grande volume de informações que continuamente converge para um centro de operação. Um centro de operação moderno geralmente é dotado de um certo número de computadores interconectados. À medida que novos dados confluem para o centro de operação, os sistemas de *software* em execução nestes computadores têm acesso imediato às informações sobre as condições operacionais da rede elétrica. Este pacote de *software* que presta suporte ao operador é chamado EMS (*Energy Management System* - Sistema de Gerenciamento de Energia). Algumas das ferramentas que tipicamente compõem um EMS são (SMATHERS; AKHIL, 2001)<sup>1</sup>:

- SCADA (*Supervisory Control and Data Acquisition*):

A função SCADA coleta, em tempo real, dados sobre as condições atuais do sistema elétrico. Estes dados consistem em medidas analógicas (tensão, corrente, etc), e pontos de estado (estado de disjuntores, dispositivos de proteção, etc). O SCADA permite também que o operador execute ações sobre equipamentos remotamente controlados (abertura de disjuntores, chaveamento de tapes de transformadores, etc).

- Controle Automático de Geração (AGC)

De modo simplista, pode-se dizer que a função AGC controla a potência gerada em sua área de atuação. Este controle é feito com base no monitoramento da diferença entre a

---

<sup>1</sup>Estes componentes não são obrigatoriamente parte de qualquer EMS. Um determinado EMS, responsável pelo gerenciamento de um sistema de transmissão, por exemplo, não é dotado de controle de geração.

potência consumida pela carga, a potência gerada e intercambiada com outras áreas, bem como dos valores de frequência no sistema.

- Comunicação com outros centros de operação:

Para a operação coordenada entre diversas áreas, é imperativo que exista intercâmbio eficiente de informações entre centros de operação, bem como entre estes e seus níveis superiores. Protocolos de comunicação digital foram desenvolvidos para gerenciar este intercâmbio.

- Funções elétricas automáticas:

As funções elétricas automáticas são algoritmos de controle que permitem um alto grau de independência da intervenção humana na operação do sistema elétrico. Estes automatismos são responsáveis pelo controle de tensão nas barras de carga, controle de reativo, recomposição de subestações e barras de carga, esquema regional de alívio de carga por subfrequência, etc (MARQUEZ et al., 2003).

- Geração de alarmes:

Considerando o tamanho e a complexidade do sistema elétrico, não se pode esperar que o operador consiga detectar todas as situações anormais apenas pela análise de pontos de estado e medidas analógicas. Para isso foram desenvolvidos programas para automatizar esta tarefa e chamar a atenção do operador em situações notáveis. Estes programas desempenham a função de geração de alarmes. A função de geração de alarmes verifica mudanças de estado em pontos digitais e violação de limites em pontos analógicos. Se alguma destas condições é detectada, uma mensagem é emitida, indicando que o operador deve tomar providências. Pode-se dizer, de maneira simplificada, que o papel do sistema de alarmes é alertar o operador sobre eventuais desvios nos processos (KANG; SEONG, 1999).

Com o advento dos recursos de informática, os serviços relacionados à geração, transmissão e distribuição de energia elétrica passaram a depender dramaticamente de computadores e redes de telecomunicações para o monitoramento e controle. Um pequeno grupo de operadores em um centro de operação, auxiliado por um EMS, é responsável pela supervisão de uma área que compreende dezenas de subestações, englobando vários milhares de pontos de medida.

Durante a operação normal, os operadores encarregam-se de ações rotineiras e ajustes para otimizar a segurança e o desempenho do sistema. Tempestades, flutuações de carga ou falhas em equipamentos podem levar o sistema a uma condição insegura ou instável. Em tais situações, o

operador encarregado da região atingida deve atuar de forma a restabelecer condições aceitáveis de operação. As ações envolvidas neste restabelecimento podem ser:

- Diagnóstico da situação/Identificação do problema.
- Localização das possíveis falhas e acionamento de equipes de manutenção.
- Recomposição de subestações ou redes desligadas por dispositivos de proteção.

Equívocos ou omissões ocorridas nestas situações podem levar a uma deterioração do estado do sistema, implicando em desligamento de cargas, danificação de equipamentos e possivelmente ao colapso do sistema (apagão).

## 1.2 Motivações

Pesquisas revelam que, sob condições de *stress*, o operador sente dificuldades em interpretar o grande número de alarmes gerado pelo sistema supervisorio (PRICE; WOLLENBERG; BERTAGNOLLI, 1989; VALE; MOURA, 1993). Esta dificuldade é mais aparente durante situações de emergência, ocasionadas por tempestades ou contingências do sistema elétrico, quando avalanches de mensagens são produzidas. Diversos alarmes ainda podem ser disparados, em decorrência de efeitos de cascadeamento (a causa inicial provoca falhas em outros pontos do sistema) (CHEON; CHANG; CHUNG, 1993). Esta inundação de informação provoca uma queda brusca no desempenho do operador (KIRSCHEN; WOLLENBERG, 1992), de maneira semelhante à sugerida na Figura 1<sup>2</sup>. Nesta figura percebe-se que a velocidade de resposta do operador aumenta à medida que o *stress* aumenta, até um ponto limite de sobrecarga. A partir deste ponto, a velocidade de resposta do operador cai rapidamente.

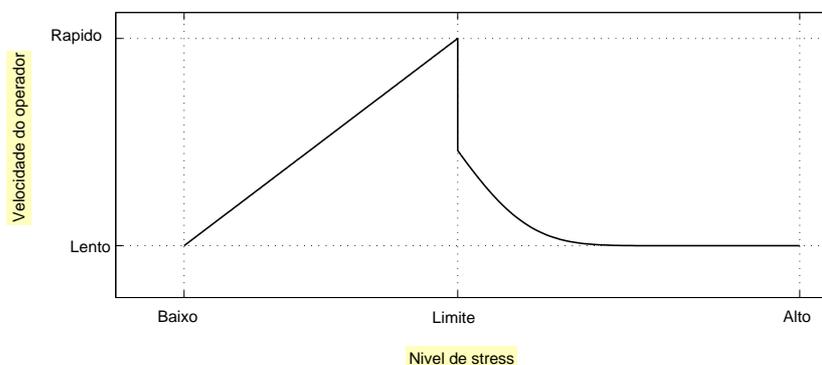


Figura 1: Desempenho do operador em função do *stress*.

<sup>2</sup>Baseado no gráfico de Kirschen e Wollenberg (1992).

Na mesma situação, o operador deve procurar entre uma grande quantidade de mensagens, determinando quais são úteis para diagnosticar a causa do problema. Um período de tempo significativo tende a ser desperdiçado neste processo, tempo este que pode ser crucial para prevenir uma deterioração da situação. Finalmente, um operador trabalhando sob *stress* e inundado com quantidades excessivas de dados pode ser levado a conclusões errôneas acerca da natureza do problema.

Como exemplo, pode ser citado o caso extremo da Hydro-Québec, onde o derretimento de neve em uma SE causou a falha simultânea de três linhas de transmissão que forneciam 3600 MW de potência (BERNARD; DUROCHER, 1993). Os operadores do centro de controle receberam 1200 alarmes apenas no primeiro segundo, e mais 3800 no minuto seguinte. A localização da falha só foi possível após 2:30 h de trabalho, durante as quais toda a área em questão ficou sem energia.

As consequências sociais e econômicas de uma interrupção no fornecimento de energia elétrica podem ser tão graves que grandes esforços são promovidos para reduzir o impacto de um distúrbio. Uma abordagem eficiente para solucionar o problema do processamento de alarmes pode contribuir para estes esforços, melhorando a forma e o conteúdo das mensagens apresentadas ao operador. O objetivo principal destas ferramentas é reduzir a quantidade de informação que deve ser processada pelo ser humano, acelerando o processo de tomada de decisões e reduzindo a probabilidade de erros operacionais (KIRSCHEN; WOLLENBERG, 1992).

### 1.3 Objetivos

O objetivo deste trabalho é propor um método de processamento inteligente de alarmes para ser utilizado associado ao SCADA/EMS que realiza o controle e supervisão de um sistema elétrico de potência. Este método será baseado em um tradicional paradigma da Computação Evolucionária, o “Algoritmo Genético”.

Um protótipo do processador de alarmes será construído para validação do método proposto. Este protótipo deverá ser independente do SCADA/EMS, porém deverá comunicar-se diretamente com ele, para obter os dados relativos às mensagens de alarme (fila de alarmes).

O processador de alarmes deverá ser capaz de operar “em tempo real”. A definição de “tempo real”, neste caso, significa que o sistema garante a obtenção do resultado dentro de um intervalo de tempo fixo (BRUNNER et al., 1993). O efeito prático desta imposição é que o operador poderá obter um diagnóstico da situação em tempo hábil, mesmo para casos complexos.

### 1.3.1 Uma Ferramenta para Processamento de Alarmes

Como a função do sistema supervisor é coletar informações pertinentes aos eventos do sistema, diferentes tipos de alarmes produzidos por vários mecanismos acabam justapostos. Uma ferramenta para processamento de alarmes é, portanto, de grande utilidade principalmente nas situações de contingência. Como características interessantes de um processador de alarmes, enfatiza-se a capacidade de:

- Eliminar informações redundantes.
- Informar sobre estados anormais da topologia da rede/SE (alimentadores fora de serviço, linhas de transmissão desligadas, etc).
- Informar os operadores quando do início de condições anormais causadas, por exemplo, pela atuação da proteção ou afundamento de tensão.
- Apresentar, quando possível, conclusões cronológicas sobre o incidente.
- Informar aos operadores quando do fim de condições anormais.
- Eventualmente, sugerir ao operador ações corretivas.

### 1.3.2 Investigações e Aplicação dos Algoritmos Genéticos

Diversos aspectos dos Algoritmos Genéticos deverão ser explorados a fim de determinar o conjunto de parâmetros mais adequado ao problema em questão. Em particular, a escolha de uma codificação adequada é fator determinante para o sucesso de uma aplicação do AG. Portanto, diversas formas de codificação deverão ser experimentadas.

Diferentes modalidades ou variações do AG tradicional podem apresentar diferentes características quando comparadas em relação ao desempenho, qualidade dos resultados, possibilidade de expansão ou uso de processamento paralelo. Estas variações deverão, portanto, ser exploradas, permitindo que se compare o desempenho de cada uma delas.

Qualquer ambiente real sofre a incidência de ruído. No sistema de processamento de alarmes, em particular, este ruído manifesta-se na forma de alarmes falsos e alarmes não reportados. O comportamento do AG deverá ser avaliado quando submetido à presença de ruído na base de informação, podendo este ser composto de alarmes falsos, alarmes não reportados ou ambos.

### 1.3.3 Estudo de Caso: O Sistema SASE

A base para o desenvolvimento deste trabalho será o Sistema de Automação de Subestações da COPEL (SASE). A partir das características do SASE (banco de dados, fila de alarmes, relatórios, etc), será analisada a viabilidade e aplicabilidade do método proposto.

O escopo de utilização do processador de alarmes será o centro de operação regional. Na área de Transmissão da COPEL, estes centros regionais são denominados Centros de Operação de Estações (COE). Na área de Distribuição, estes centros são denominados Centros de Operação da Distribuição (COD).

O sistema SASE foi construído para automatizar subestações, COEs e CODs. A supervisão e controle do sistema elétrico da COPEL, nas áreas de Transmissão e Distribuição, é realizado de maneira distribuída, por diversos destes centros regionais. Cada centro regional supervisiona e controla um determinado número de subestações. As subestações subordinadas a um determinado centro de operação podem estar interligadas entre si, ou ainda interligadas com subestações supervisionadas por outros centros.

Em razão destas características, o processador de alarmes deverá realizar uma análise “orientada à estação”, ou seja para uma subestação isolada, ou, eventualmente, para redes de distribuição. A análise de ocorrências em linhas de transmissão nem sempre será possível nos COEs, pois geralmente envolve dados de mais de uma subestação. Se a linha em questão realiza a interligação entre SEs supervisionadas por diferentes centros regionais de operação, não haverão dados suficientes disponíveis para realizar a análise. Neste caso, apenas o nível mais alto na hierarquia de controle e supervisão (COS - Centro de Operação do Sistema) tem acesso a todos os dados pertinentes (para um melhor entendimento da hierarquia de operação do sistema elétrico, ver seção 5.1.1). A Figura 2 ilustra esta situação. Estão ali representados esquematicamente o COS, dois COEs, quatro SEs e três linhas de transmissão. Neste caso, para um processador de alarmes em execução no “COE1” seria possível realizar o diagnóstico de ocorrências na linha “LT1”, porém o mesmo não seria possível para a linha “LT2”, pois os dados referentes a esta linha coletados na “SE3” não estão disponíveis no “COE1”.

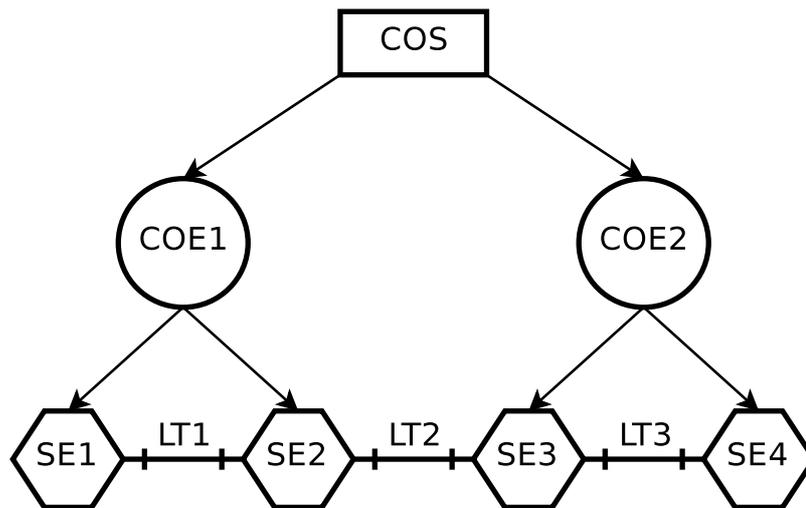


Figura 2: Representação esquemática do COS, COEs e SEs.

## 1.4 Estrutura da Dissertação

Esta dissertação divide-se em sete capítulos. O Capítulo 1 identifica e caracteriza o tema da dissertação, delineando os objetivos e plano de ação básico. No Capítulo 2 são apresentados os principais conceitos teóricos utilizados no desenvolvimento do trabalho, incluindo os fundamentos sobre Algoritmos Genéticos, suas principais variações, vantagens, desvantagens e escopo de aplicação.

O Capítulo 3 aponta as principais funções de um processador de alarmes e posiciona este componente em relação às demais ferramentas empregadas na operação do sistema elétrico. Alguns trabalhos previamente publicados na área de processamento de alarmes são brevemente analisados neste capítulo.

A descrição formal do problema e a elaboração dos modelos de solução são expostos no Capítulo 4. A implementação do protótipo e descrição de suas características são apresentados no Capítulo 5.

Os resultados obtidos são expostos e analisados no Capítulo 6. Por fim, as conclusões são apresentadas no Capítulo 7, seguidas de sugestões para futuros aperfeiçoamentos do trabalho.



## 2 *Fundamentação Teórica*

Este capítulo trata dos principais tópicos teóricos necessários para o desenvolvimento e compreensão deste trabalho. Estes tópicos, porém, são tratados de maneira objetiva, sem que o assunto seja esgotado. Para uma abordagem mais completa de cada assunto, o leitor deverá consultar as respectivas referências bibliográficas.

### 2.1 Algoritmos Genéticos

Algoritmos Genéticos são procedimentos computacionais de busca e otimização, cujo funcionamento é inspirado nos processos naturais de seleção e refinamento genético (GOLDBERG, 1989). Estes algoritmos foram inicialmente propostos por John H. Holland e sua equipe da Universidade de Michigan, na década de 60. As origens destes princípios remontam ao século 19, nos trabalhos de Charles Robert Darwin. A observação da natureza dos organismos vivos leva a concluir que estes são “consumados solucionadores de problemas” (HOLLAND, 1992).

#### 2.1.1 Teoria da Evolução

Por volta de 1800, Jean-Baptiste Peirre Antoine de Monet, “*Chevalier de Lamarck*”, propôs sua teoria da evolução (que ficaria conhecida como Lamarckismo) (WIKIPEDIA, 2005b). Lamarck defendia a idéia de que “as plantas e os animais evoluem pela necessidade de se adaptar às mudanças do meio ambiente e que transmitem essa mutação às futuras gerações”. De Lamarck vem o famoso exemplo das girafas, por vezes utilizado para ridicularizar este autor (SAKALL, 2005). Segundo Lamarck, no princípio, todas as girafas eram dotadas de pescoços curtos. No ambiente, porém, as melhores opções de alimento encontravam-se no alto das árvores. Isto obrigava estes animais a esticarem o pescoço em direção à copa das árvores. O constante “esticamento” provocaria a incorporação desta característica na espécie. As novas girafas nasceriam com o pescoço levemente mais longo, geração após geração, até a estabilidade (a girafa como conhecemos hoje).

Em 1859, Charles Darwin publicou um livro intitulado “A Origem das Espécies”, no qual foi proposta a idéia da seleção natural (WIKIPEDIA, 2005a). Utilizando o mesmo exemplo das girafas, Darwin afirma que, inicialmente existiam girafas de pescoço curto e também de pescoço comprido. As girafas de pescoço mais longo possuíam uma vantagem estratégica, que lhes garantia acesso à alimentação farta, tornando-as mais fortes - estavam mais adaptadas ao meio. Assim sendo, animais com estas características tinham maior probabilidade de sobreviver e reproduzir-se, passando adiante seu material genético (tendo maior número de descendentes). Por outro lado, as girafas de pescoço curto tinham menor probabilidade de sobreviver e reproduzir-se, acabando por desaparecer.

Um detalhe, porém, passa despercebido pela teoria de Darwin: como surgiu a girafa de pescoço comprido? Uma teoria proposta como um complemento à teoria original, conhecida como Neo-Darwinismo (SAKALL, 2005) afirma que, dentro de uma população homogênea, mudanças aleatórias são constantemente introduzidas. Estas mudanças provocam o aparecimento de indivíduos levemente diferentes. Se as características destes indivíduos forem positivas, estes tendem a tornar-se mais fortes, tendo maior probabilidade de sobreviver e reproduzir-se, exatamente como descrito por Darwin. Era introduzido ali o conceito de mutação.

### **2.1.2 Algoritmos Genéticos: “Darwinismo Computacional”**

A maioria dos organismos vivos evolui por meio de dois processos: seleção natural e reprodução sexuada. O primeiro determina quais membros de uma população sobrevivem e se reproduzem. O segundo assegura que existe troca de informação e recombinação entre os indivíduos (HOLLAND, 1992). Ocorre ainda um terceiro processo, responsável por introduzir características novas nos seres vivos: a mutação.

A evolução das espécies pode ser vista como um mecanismo adaptativo de otimização que envolve certa aleatoriedade (TANOMARU, 1995). Este mecanismo é emulado pelos algoritmos genéticos. Os AGs empregam diversas metáforas biológicas para descrever conceitos computacionais associados.

Os AGs pertencem à classe dos métodos probabilísticos de busca e otimização, embora não sejam uma mera busca aleatória. Como características que diferenciam o AG de outros métodos de otimização, destacam-se (TANOMARU, 1995):

- O AG trabalha com um conjunto de pontos (soluções-candidatas) para um dado problema, e não sobre pontos isolados. Este conjunto é denominado “população”. Em contrapartida, cada ponto da população é denominado “indivíduo”.

- Normalmente o AG opera em um espaço de soluções codificadas, e não diretamente no espaço de busca. Estas soluções são codificadas em sequências denominadas “cromossomos” ou “*strings*”. Cada elemento do cromossomo é denominado “gene”. A codificação da solução é frequentemente denominada “genótipo”, enquanto a solução real é denominada “fenótipo”.
- O AG requer apenas informação sobre a função-objetivo a ser otimizada, para cada membro da população. Informações adicionais, como derivadas e gradiente são desnecessárias. A função-objetivo, na metáfora biológica, representa o meio no qual o indivíduo está inserido. Através dela é determinado o quanto um indivíduo está “adaptado ao ambiente”, ou seja, quão boa é uma solução. Esta medida é chamada de *fitness* ou adequabilidade.
- O AG utiliza regras de transição probabilísticas, e não determinísticas. Cada indivíduo tem uma certa probabilidade de ser selecionado e passar adiante seu material genético. Normalmente esta probabilidade é proporcional ao valor do *fitness*.

## 2.1.3 Parâmetros do AG

### 2.1.3.1 Codificação

A aplicação do algoritmo genético inicia-se pela codificação. A codificação especifica um mapeamento que transforma uma possível solução para o problema em uma estrutura contendo um conjunto de variáveis de decisão (LOBO, 2000). O conjunto de todas as possíveis combinações destas variáveis de decisão constitui o “espaço de busca”, e uma combinação particular representa um ponto neste espaço. Na prática, a representação mais comum para estas estruturas é uma simples concatenação (*string*) das variáveis de decisão. Existe ainda uma variedade de possíveis valores assumidos pelas variáveis de decisão (tipos). Entre os mais largamente utilizados estão o binário, código *Gray*, inteiro e real (LOBO, 2000).

As variáveis de decisão no AG são denominadas simplesmente “genes”, enquanto um valor associado à mesma é denominado “alelo”. Uma concatenação ou *string* de genes é chamado “cromossomo”. A posição de um determinado gene no cromossomo é chamada *locus gênico* (TANOMARU, 1995).

A solução-candidata, representada neste formato codificado, é geralmente denominada “genótipo”. Sua representação no domínio não-codificado é denominada “fenótipo” (VIANNA, 2004). Esta nomenclatura é uma metáfora da terminologia biológica.

### 2.1.3.2 Determinação da Função de *Fitness*

Cada indivíduo da população deve receber uma “pontuação” que classifica seu desempenho em relação à função-objetivo correspondente ao problema. Este valor é uma medida de adequabilidade da solução-candidata (quanto maior a adequabilidade, melhor a solução). Uma designação largamente aceita para esta medida é a palavra da língua inglesa “*fitness*”. O *fitness*, por definição, é uma quantidade positiva (GOLDBERG, 1989), e deve ser tanto maior quanto mais a solução candidata aproxima-se do ponto ótimo. Desta forma, é normalmente necessário realizar uma transformação na função-objetivo, de forma a adequá-la a estas restrições.

### 2.1.3.3 Métodos de Seleção

O processo de seleção simula o “princípio da sobrevivência do melhor”, encontrado na natureza. Existem diversas maneiras de implementar este procedimento no AG, mas a idéia geral é que os indivíduos mais adaptados (com maior valor de *fitness*) tenham preferência (LOBO, 2000). Por outro lado, um certo grau de aleatoriedade está inerentemente envolvido neste processo.

O método de seleção empregado é um dos principais fatores determinantes da velocidade de convergência do algoritmo, e por consequência da qualidade das soluções obtidas. O termo “pressão seletiva” ou “intensidade seletiva” é utilizado para caracterizar diferentes métodos de seleção em relação à mudança imposta pela aplicação do método à adequabilidade média da população (BLICKLE; THIELE, 1995). Em outras palavras, a pressão seletiva é uma medida do “progresso” da população devido ao esquema de seleção. Em geral, quanto maior a pressão seletiva imposta, mais rápido o AG converge, porém a qualidade das soluções obtidas tende a piorar.

Alguns dos métodos de seleção mais conhecidos são:

#### 1. Seleção Proporcional (Método da Roleta)

Um dos métodos mais tradicionais de seleção de indivíduos para reprodução é a seleção proporcional, ou “método da roleta”. O algoritmo básico para a implementação deste método consiste em criar uma roleta viciada, onde cada indivíduo da população é associado a um setor desta roleta. O cálculo do tamanho do setor é feito somando-se a adequabilidade de todos os membros da população (que perfaz 100%), e computando-se o valor percentual com que cada indivíduo contribuiu para esta totalização. O tamanho do setor associado a cada membro é correspondente a esta contribuição (GOLDBERG, 1989).

Supondo que a população seja composta dos indivíduos ilustrados na Tabela 1, onde são transcritos os respectivos valores de adequabilidade bruta e percentual, a roleta resultante seria equivalente àquela ilustrada na Figura 3. Girando-se esta, retornar-se-ia um membro aleatório da população, porém, a chance que cada membro possui de ser selecionado é proporcional ao seu valor de adequabilidade.

A seleção proporcional, no entanto, é considerada um método ineficiente, pois beneficia demasiadamente os indivíduos com alta adequabilidade (impõe uma alta pressão seletiva), podendo provocar convergência prematura para máximos locais.

Tabela 1: População hipotética, e seus valores de adequabilidade.

Indivíduo	Adequabilidade	Percentual do Total
Ind 1	0,42	12 %
Ind 2	0,7	20 %
Ind 3	0,53	15 %
Ind 4	0,95	27 %
Ind 5	0,28	8 %
Ind 6	0,63	18 %

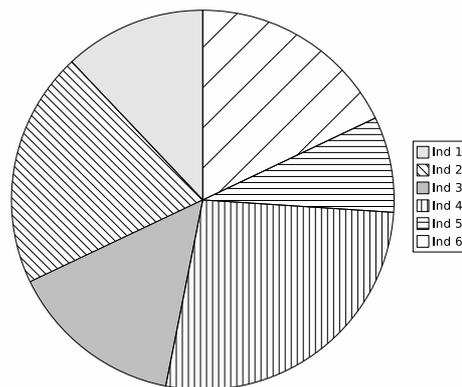


Figura 3: Seleção pelo método da roleta.

## 2. Método do Ordenamento Linear

O método do ordenamento linear foi inicialmente proposto com o objetivo de eliminar as sérias desvantagens da seleção proporcional. Considerando a roleta exemplificada anteriormente, na hipótese de o melhor indivíduo ser responsável por 90% do valor total

da adequabilidade, é fácil perceber que os demais indivíduos terão apenas uma remota chance de serem selecionados, como pode ser visto na Figura 4 (a).

A seleção por ordenamento linear consiste em ordenar os  $N$  membros da população de acordo com o valor de adequabilidade. Uma “pontuação” de valor igual a  $N$  é atribuída para o melhor indivíduo,  $N - 1$  para o segundo melhor, e assim sucessivamente, até que o pior indivíduo receba apenas um ponto. A seguir, esta pontuação é utilizada para determinar o tamanho dos setores da roleta. Esta situação é ilustrada na Figura 4.

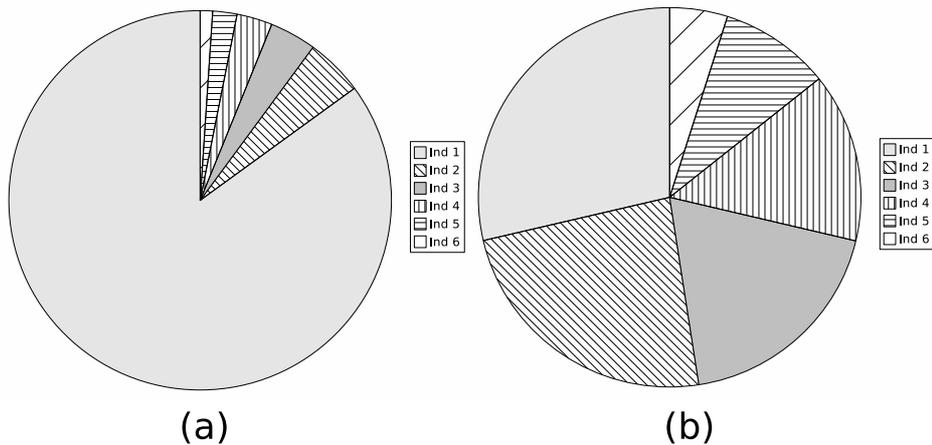


Figura 4: Seleção pelo método do ordenamento linear.

### 3. Método do Torneio Estocástico

O método do torneio estocástico (ou “torneio de tamanho  $K$ ”) consiste em selecionar aleatoriamente  $K$  indivíduos da população, ordená-los de acordo com a adequabilidade e tomar apenas o melhor (ou os dois melhores) para a reprodução. Uma das vantagens imediatas deste método é a simplicidade e eficiência, visto que não é necessário ordenar todos os membros da população para executá-lo (BLICKLE; THIELE, 1995). O fator  $K$  por sua vez possui forte influência no esquema de seleção. Valores elevados de  $K$  impõem uma alta pressão seletiva na população, tendendo a levar a uma convergência prematura, enquanto valores demasiadamente pequenos fazem com que o método se aproxime de uma busca aleatória.

### 4. Estratégias Elitistas

O procedimento elitista força o algoritmo genético a preservar os melhores indivíduos a cada geração, enquanto os filhos substituem os indivíduos menos adaptados. Os melhores indivíduos poderiam ser perdidos caso o processo de seleção não os contemplasse ou poderiam ser destruídos pelos operadores cruzamento ou mutação (FURTADO, 1998).

### 2.1.3.4 Operadores Genéticos

Para produzir novas soluções, o AG utiliza essencialmente dois operadores: cruzamento e mutação. O operador de cruzamento (ou recombinação) funciona basicamente associando pedaços de duas soluções candidatas, de forma a gerar novos indivíduos. Os indivíduos que compõem o par original são denominados “pais”, enquanto os novos indivíduos gerados são denominados “filhos”. Este operador é normalmente aplicado com alta probabilidade, e é responsável pela maior parte da busca realizada pelo AG (LOBO, 2000). As principais técnicas de recombinação encontradas na literatura são:

#### 1. Cruzamento de um ponto

Dados um par de indivíduos selecionados para a recombinação (pais), uma posição de cruzamento aleatória  $p$  é escolhida ( $1 \leq p < \ell$ , onde  $\ell$  é o tamanho do cromossomo). Esta posição é denominada “ponto de corte”. A seguir, dois novos indivíduos são produzidos, trocando-se os genes entre as posições  $p + 1$  e  $\ell$ , inclusive (GOLDBERG, 1989). Esta situação é ilustrada na Figura 5 (a).

#### 2. Cruzamento multi-ponto

Diversos algoritmos de cruzamento são propostos na literatura, alguns deles envolvendo mais de um ponto de corte. Foi observado, porém, que o aumento indiscriminado do número de pontos de corte degrada a performance do algoritmo, por causar a destruição de possíveis blocos construtivos (GOLDBERG, 1989). O cruzamento de dois pontos, todavia, é considerado adequado, pois em geral produz melhores resultados. Esta técnica consiste em estabelecer dois pontos de corte aleatórios, e então realizar a troca da sequência de genes que se encontra entre as secções, conforme ilustrado na Figura 5 (b).

#### 3. Cruzamento uniforme

O cruzamento uniforme é radicalmente diferente dos métodos expostos anteriormente. Cada gene no indivíduo filho é criado através da cópia do correspondente gene nos indivíduos pais, de acordo com uma máscara binária, gerada aleatoriamente. Caso exista o valor 1 na posição correspondente da máscara, o gene é copiado do primeiro pai, e caso exista o valor 0, o gene é copiado do segundo pai, conforme ilustrado na Figura 6. A seguir, o processo é repetido, com os indivíduos pais trocados, para gerar o segundo descendente. Uma nova máscara aleatória é gerada para cada par de indivíduos pais. Esta técnica é considerada mais robusta que as anteriores (BEASLEY; BULL; MARTIN, 1993).

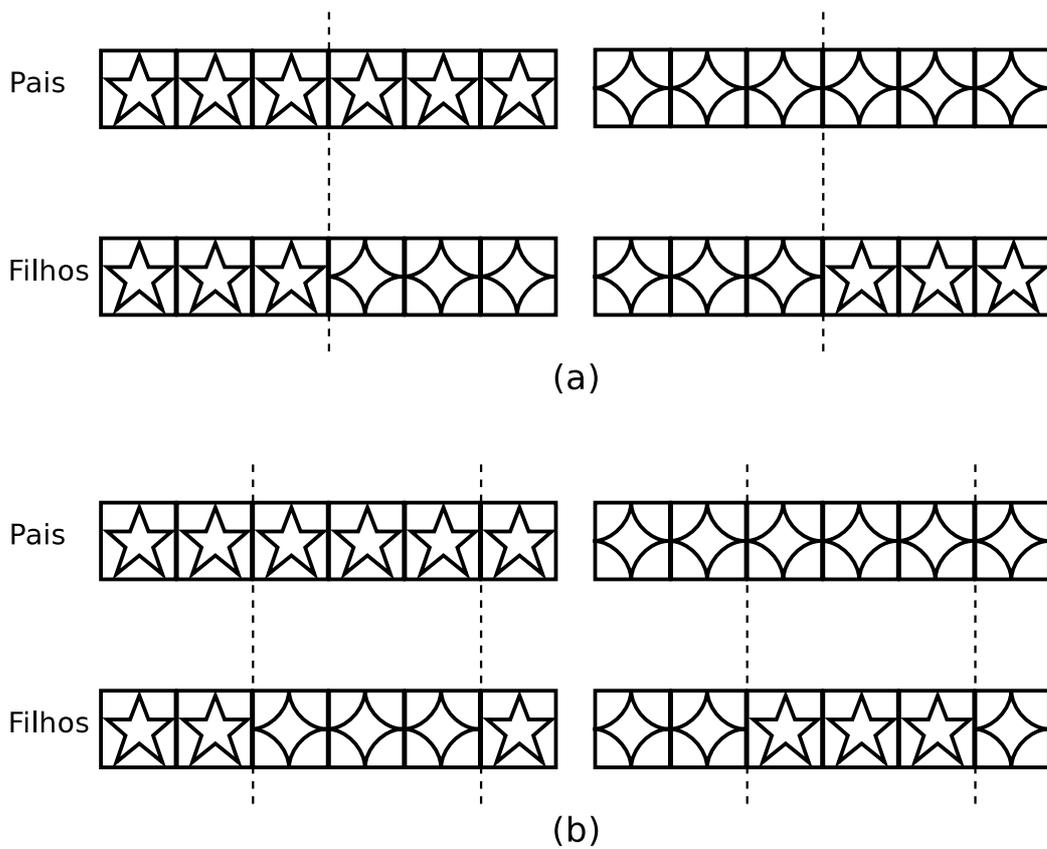


Figura 5: Cruzamento de um ponto e de dois pontos.

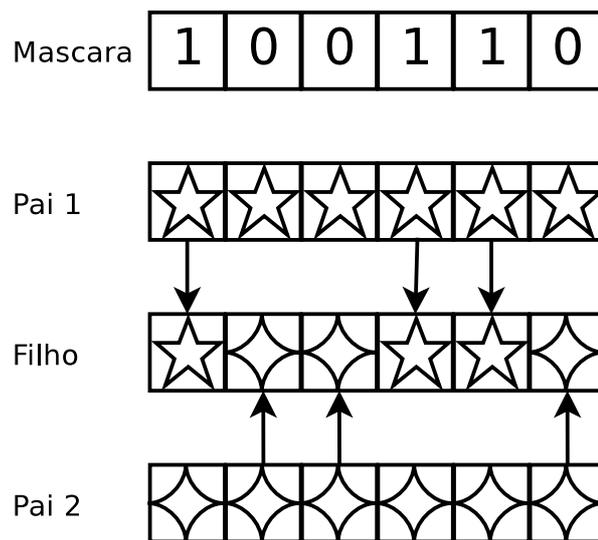


Figura 6: Cruzamento uniforme.

A mutação é tradicionalmente vista como um operador auxiliar, responsável por alterar aleatoriamente o valor de uma variável de decisão. A mutação é normalmente aplicada com

baixa probabilidade, sendo responsável por manter a diversidade entre a população, ou ainda, sob outra interpretação, realizar uma exploração global ou busca aleatória (TANOMARU, 1995).

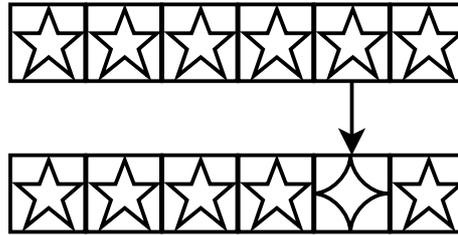


Figura 7: Operador de mutação.

### 2.1.3.5 Método de Inicialização e Critério de Parada

A população inicial do AG é normalmente composta por indivíduos gerados aleatoriamente, embora em alguns casos seja possível inicializar parte da população com indivíduos previamente escolhidos (LOBO, 2000). Sobre esta população inicial, são aplicados iterativamente a seleção de indivíduos e os operadores genéticos. Cada uma destas iterações (seleção + operadores genéticos) é denominada “geração”. Este processo é repetido até que algum critério de parada especificado seja atingido. O critério de parada pode ser um máximo número de gerações, um determinado tempo máximo de processamento, a obtenção de um valor de *fitness* pré-estabelecido ou a estagnação, ou seja, a convergência da população para alguma região do espaço de busca, impedindo que qualquer melhoria seja observada com o passar das gerações (TANOMARU, 1995).

### 2.1.3.6 Determinação dos demais parâmetros

Para que o AG possa entrar em operação, resta ainda determinar um certo número de parâmetros, tais como o tamanho da população, probabilidade de aplicação de operadores, taxa de seleção, entre outros.

## 2.1.4 Funcionamento do AG Canônico

A operação do algoritmo genético canônico (GOLDBERG, 1989) é bastante simples, consistindo nos passos descritos no fluxograma da Figura 8 (a). Para esclarecer utilizando um exemplo, considere-se a função  $f(x) = -x^2 + 8x + 4$ , para  $x$  inteiro no intervalo  $[0, 15]$  ( $x \in \mathbb{Z}$ ,  $0 \leq x \leq 15$ ). Sem utilizar o AG, descobre-se facilmente que o máximo desta função encontra-se em  $x = 4$  (Figura 8 (c)). Este exemplo, porém, é meramente ilustrativo. Existem maneiras mais

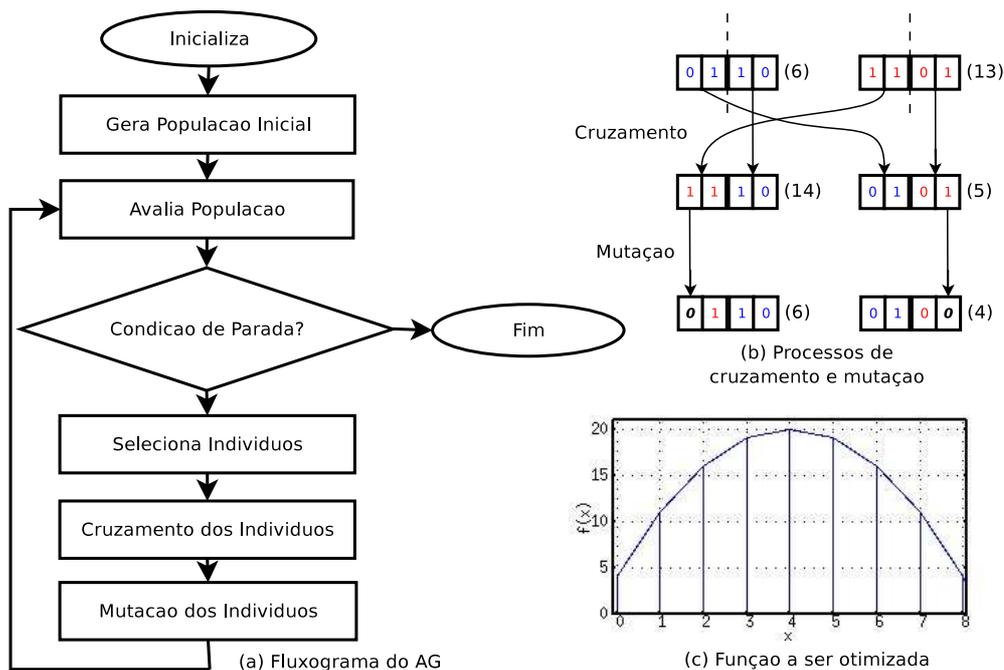


Figura 8: Fluxograma de execução do AG.

simples e eficientes de encontrar o máximo desta função, por exemplo, através de sua derivada, ou enumerando todos os possíveis valores de "x" no intervalo (visto que são apenas 15). Mas supondo-se que este problema esteja sendo solucionado com AG, utilizando os seguintes parâmetros:

- Codificação: Neste caso, codifica-se a variável como número binário.
- Função de *fitness*: Como visto, esta medida deve ser sempre positiva. Para que isto aconteça para todos os valores da variável  $x$ , será somado um número suficientemente grande à função-objetivo, de acordo com a equação 2.1. Este número é equivalente ao valor absoluto do menor valor assumido por  $f(x)$  no intervalo de interesse.

$$F_{fit} = f(x) + 101 \quad (2.1)$$

### 1. Inicialização e geração da população inicial

Nesta fase, é gerada aleatoriamente uma população de indivíduos, cada um correspondendo a uma solução-candidata à instância do problema sendo resolvida. A população é composta de um número razoavelmente grande de indivíduos (geralmente em torno de centenas). A cada iteração sobre o laço de execução do AG, esta população será substituída por um novo grupo de indivíduos, obtidos através da aplicação dos operadores

genéticos sobre elementos selecionados da população original. Para este exemplo, será considerado que a população ( $Pop$ ) é composta de seis membros ( $I_1, \dots, I_6$ ). Estes membros são inicializados através de um processo aleatório, resultando no seguinte:

$$Pop = \{I_1, I_2, I_3, I_4, I_5, I_6\} = \{[0110], [0001], [1101], [1100], [0000], [1000]\}$$

## 2. Avaliação

O AG necessita de uma informação que mede o quanto cada indivíduo está adaptado ao meio, ou seja, quão boa cada solução é em relação a uma função-objetivo. Nesta fase, cada indivíduo é avaliado, recebendo uma pontuação correspondente ao *fitness*. Utilizando a equação 2.1 como medida de adequabilidade, a população gerada no passo anterior seria avaliada conforme os seguintes valores:

$$Fit(I_1, I_2, I_3, I_4, I_5, I_6) = \{117, 112, 40, 57, 105, 105\}$$

## 3. Seleção

O mecanismo de seleção no AG emula o processo natural de seleção e reprodução. Neste processo, indivíduos bem adaptados possuem alta probabilidade de sobreviverem e passarem adiante sua informação genética, enquanto indivíduos com baixa adequabilidade possuem grande probabilidade de extinção. Supondo agora que, da população inicial, dois indivíduos,  $I_1$  e  $I_3$ , tenham sido selecionados, conforme a Figura 8 (b). Estes indivíduos representam soluções-candidatas correspondentes aos valores  $x = 6$  e  $x = 13$ .

## 4. Recombinação ou cruzamento

A recombinação é um processo sexuado (envolve mais de um indivíduo) que emula o fenômeno natural do cruzamento. Através dele, fragmentos de informação são trocados para gerar novos indivíduos. Aplicando o operador de cruzamento aos indivíduos selecionados anteriormente, duas novas soluções são produzidas ( $x = 14$  e  $x = 5$ ), conforme a Figura 8 (b). Percebe-se que logo após o cruzamento, houve uma melhoria genética (a solução ficou mais próxima do máximo).

## 5. Mutação

Este processo, no AG, equivale a uma busca aleatória. A ação deste operador consiste basicamente em escolher um gene no cromossomo e alterar aleatoriamente seu valor. O operador de mutação é responsável por introduzir variedade genética na população, proporcionando uma melhor exploração do espaço de busca. Observando a Figura 8 (b) percebe-se o operador de mutação atuando sobre os dois novos indivíduos gerados pelo

cruzamento (produzindo  $x = 6$  e  $x = 4$ ). Igualmente ao caso anterior, percebe-se que após a mutação o espaço de busca foi melhor explorado, e o máximo foi atingido.

## 6. Condições de parada

O AG é um processo iterativo, portanto, é necessário estabelecer um critério para o término de sua execução. O ideal seria que a execução fosse encerrada ao atingir o ponto ótimo, porém, geralmente não é possível afirmar que o ponto ótimo foi atingido. Utiliza-se portanto como critério de parada um número máximo de gerações ou um tempo máximo de processamento. Outro possível critério de parada é o estabelecimento de um determinado valor da função objetivo a ser atingido. Para este exemplo simples, poderia-se dizer que o critério de parada é a chegada ao ponto máximo.

### 2.1.5 Teorias Sobre o Funcionamento do AG

A teoria original sobre o funcionamento dos algoritmos genéticos, proposta por Holland, baseia-se na suposição de que uma boa solução para uma dada instância de um problema pode ser obtida através da combinação de pedaços de soluções de boa qualidade (LOBO, 2000). Holland introduziu o conceito de “esquema” (do inglês, *schema*) para analisar o efeito dos operadores genéticos sobre estes pedaços de soluções.

Um esquema é um “padrão de similaridade”, ou simplesmente um “molde” (TANOMARU, 1995) que representa um conjunto de soluções no espaço de busca, possuindo similaridades em determinadas posições da cadeia que codifica a solução (GOLDBERG, 1989). Considerando-se o alfabeto binário  $\{0, 1\}$ , e cromossomos de comprimento  $\ell$ , um esquema seria qualquer sequência de  $\ell$  caracteres gerados a partir do alfabeto estendido  $\{0, 1, *\}$ , onde o símbolo “\*” desempenha o papel de um “curinga”, podendo representar tanto o valor “0” quanto “1” (TANOMARU, 1995). Como exemplo, tomando  $\ell = 4$ , o esquema 000\* representa todos os cromossomos que possuem os três primeiros elementos iguais a zero, ou seja, 0000 e 0001. Este conceito pode ser estendido para alfabetos de qualquer ordem. Um resultado importante, conhecido como “Teorema dos Esquemas”, demonstra que esquemas com adequabilidade superior à média da população recebem um número exponencialmente crescente de representantes na próxima geração do AG (TANOMARU, 1995).

O funcionamento do AG fica melhor representado através da teoria dos esquemas. O AG é capaz de selecionar esquemas simples, possuindo elevada adequabilidade, e recombina-los em sequências mais complexas, com adequabilidade possivelmente maior. Trabalhando com estes esquemas em particular, a complexidade do problema é reduzida. Ao invés de construir uma

solução boa testando exaustivamente todas as possíveis combinações de símbolos do alfabeto para cada posição do cromossomo, soluções cada vez melhores são construídas a partir de soluções parciais de gerações passadas. Estas soluções parciais, de “pequena ordem” (poucos genes com valor definido) e “pequeno comprimento” (distância entre o primeiro e o último gene definido) são denominadas “blocos construtivos” (GOLDBERG, 1989).

Desta forma, o AG é capaz de encontrar soluções cada vez mais próximas do ótimo, combinando blocos construtivos de ordem inferior em *strings* de ordem superior. Em determinados problemas, porém, esquemas de ordem inferior que possuem alta adequabilidade, quando combinados entre si produzem soluções de baixa qualidade. Goldberg introduziu a noção de “problemas enganadores” (do inglês, *deceptive problems*) para ilustrar este conceito (GOLDBERG, 1989). Estes problemas possuem duas características que degradam o desempenho do AG: a solução global é isolada e a avaliação dos blocos construtivos tende a levar a uma solução sub-ótima (LOBO, 2000).

Nenhuma destas teorias, porém, é universalmente aceita. Thornton (1997), por exemplo, contesta a teoria dos esquemas de Holland e afirma que esta é contraditória à hipótese dos blocos construtivos de Goldberg. Na realidade, existe um lapso entre a teoria e a prática no que concerne os algoritmos genéticos. Empiricamente, comprovou-se que o AG é eficiente em uma vasta gama de aplicações onde não se conhece um método analítico ou numérico eficiente, que garanta a obtenção da solução ótima.

De maneira geral, o AG apresenta bom desempenho em problemas cuja solução pode ser obtida através da combinação de frações de uma solução completa. Cada uma destas frações corresponde a um pequeno conjunto de variáveis de decisão, que interagem entre si (LOBO, 2000). Por outro lado, problemas envolvendo interações complexas entre as variáveis de decisão (epistasia) tendem a degradar o desempenho do algoritmo. Lobo (2000) cita como exemplo o problema de encontrar uma agulha em um palheiro. Este problema envolve complexas interações entre todas as variáveis de decisão, não podendo ser resolvido através da combinação de soluções parciais. Um algoritmo genético certamente apresentaria desempenho insatisfatório resolvendo uma instância deste problema. Por outro lado, a única forma de encontrar a agulha no palheiro seria através da enumeração completa do espaço de busca.

### **2.1.6 Algoritmos Genéticos Paralelos**

A idéia básica de qualquer algoritmo paralelo é dividir uma única tarefa em pedaços e resolver cada um destes pedaços simultaneamente utilizando diversas unidades de processamento - o método de dividir para conquistar (CANTÚ-PAZ, 1997).

As motivações que levam a aplicar métodos de processamento paralelo aos AGs são: aumentar a velocidade de execução e melhorar a exploração global do espaço de busca, produzindo eficientemente resultados de melhor qualidade (TANOMARU, 1995).

Estes métodos podem ser aplicados aos AGs de diversas maneiras diferentes. Existem três categorias principais de algoritmos genéticos paralelos (CANTÚ-PAZ, 1997): O AG mestre-escravo de população única, o AG massivamente paralelo de população única, e o AG multi-populacional insular. Combinações destas categorias básicas podem ser utilizadas para compor arranjos mais sofisticados, denominados AGs paralelos hierárquicos (CANTÚ-PAZ, 1997).

### 2.1.6.1 AG Paralelo Global de População Única (Mestre-Escravo)

Nesta categoria de algoritmo, uma única população é empregada. A avaliação dos indivíduos e/ou a aplicação dos operadores genéticos é feita em paralelo. Assim como no AG convencional, cada indivíduo compete e pode trocar informação genética com qualquer outro integrante da população. AGs paralelos globais são geralmente implementados em arquitetura mestre-escravo, onde o processador mestre armazena a população e os escravos calculam a adequabilidade dos indivíduos, conforme ilustrado na Figura 9 (CANTÚ-PAZ, 1997).

A operação mais frequentemente distribuída é a avaliação dos indivíduos, pois, em geral, independe do restante da população e não requer comunicação entre os processadores durante sua execução. O processamento paralelo é feito atribuindo-se uma fração da população para cada escravo. A comunicação entre processadores ocorre somente enquanto cada escravo recebe sua parcela de indivíduos para avaliação e quando o resultado é retornado ao mestre (CANTÚ-PAZ, 1997).

O AG paralelo global, em geral, possui as mesmas propriedades do AG simples, diferenciando-se apenas pelo ganho de velocidade.

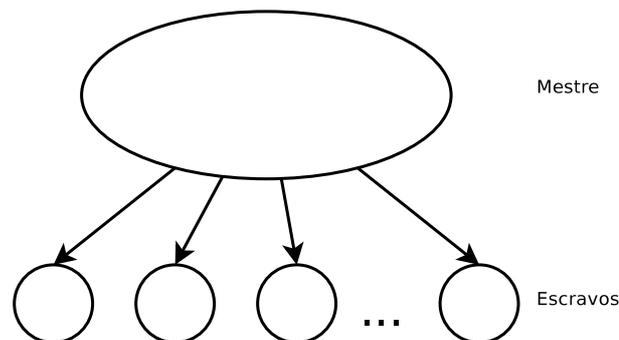


Figura 9: AG paralelo com arquitetura mestre-escravo.

### 2.1.6.2 AG massivamente Paralelo de População Única

Esta categoria de AGs utiliza uma única população, porém esta é dotada de uma distribuição espacial, que limita as interações entre indivíduos. A um determinado indivíduo é permitido competir e procriar apenas entre seus vizinhos na estrutura. As vizinhanças de cada indivíduo, porém, sobrepõem-se, permitindo que soluções de boa qualidade sejam disseminadas entre a população (CANTÚ-PAZ, 1997). Esta estrutura é ilustrada esquematicamente no diagrama da Figura 10. Neste diagrama, os círculos representam unidades de processamento, enquanto as linhas retas representam possíveis comunicações entre os processadores. A unidade representada em cinza escuro tem sua respectiva vizinhança destacada.

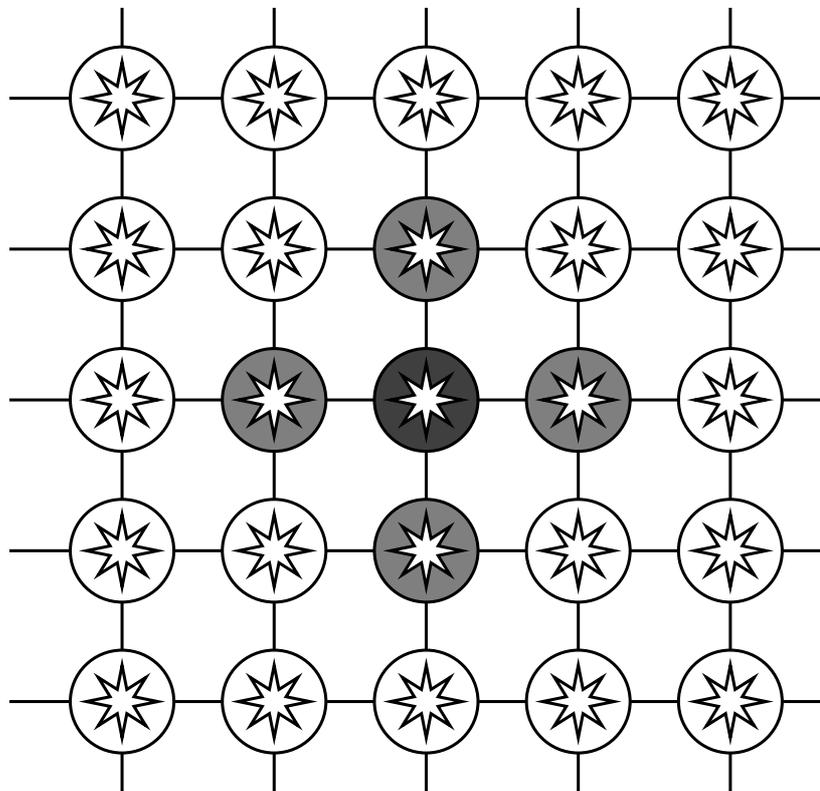


Figura 10: AG massivamente paralelo.

Os AGs massivamente paralelos são muitas vezes referidos como “AGs celulares” ou “AGs de granularidade fina” (TANOMARU, 1995). AGs de granularidade fina são apropriados para o uso em computadores massivamente paralelos. O cenário ideal para a aplicação é o caso onde atribui-se apenas um indivíduo por processador disponível. Na Figura 10 esta situação pode ser visualizada se considerarmos que o círculo corresponde a um processador e a estrela corresponde a um indivíduo.

### 2.1.6.3 AG paralelo multipopulacional

Estes são os modelos mais simples e apelativos de AGs paralelos, onde a população é particionada em subpopulações, que por sua vez são atribuídas a processadores distintos em um computador paralelo. Cada processador executa essencialmente um AG convencional, porém, envia periodicamente cópias de alguns de seus indivíduos (geralmente os melhores) para os processadores vizinhos, recebendo em troca cópias de indivíduos pertencentes àqueles processadores (TANOMARU, 1995). Este processo de troca de indivíduos entre subpopulações é denominado migração. O AG paralelo multipopulacional também é conhecido como “AG insular” ou “de granularidade grossa”.

Na Figura 11 é esquematizado o funcionamento de um AG paralelo insular, contendo quatro subpopulações, representadas pelos octógonos. O processo de migração de indivíduos também está ilustrado.

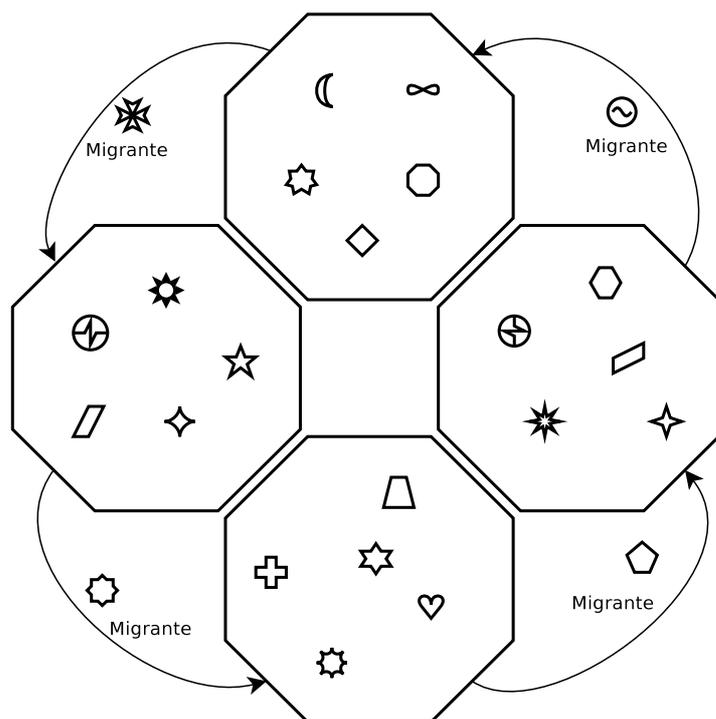


Figura 11: AG paralelo insular.

O tempo de processamento em um AG cresce com o tamanho da população (TANOMARU, 1995). Por outro lado, a taxa de crescimento média da adequabilidade dos indivíduos é maior em populações menores, mas em contrapartida tende a estagnar em valores mais baixos do que aqueles atingidos em populações maiores (CANTÚ-PAZ, 1997). As principais vantagens apontadas para o uso de AGs paralelos insulares são:

- Simplicidade conceitual: AGs paralelos insulares podem ser vistos como uma simples extensão do AG convencional. Basicamente, é necessário apenas que se execute um determinado número de AGs seriais (um em cada nodo de um computador paralelo, por exemplo), e promova-se uma troca periódica de indivíduos entre eles (CANTÚ-PAZ, 1997).
- Baixo custo de implementação: É necessário pouco esforço adicional para converter um AG serial (convencional) em um AG paralelo multipopulacional (CANTÚ-PAZ, 1997). A maior parte do programa permanece inalterada, sendo necessário alterar ou adicionar apenas algumas subrotinas, responsáveis pela inicialização e pelo processo de migração (GOODMAN, 1996).
- Disponibilidade de infra-estrutura: A infra-estrutura necessária para a execução do AG paralelo insular é relativamente simples e significativamente mais barata do que os computadores massivamente paralelos. Esta infra-estrutura pode ser composta de uma simples rede de computadores do tipo estação de trabalho utilizando programas livremente distribuídos, ou mesmo simulada em um único computador (CANTÚ-PAZ, 1997; GOODMAN, 1996).

### 2.1.7 Algoritmos Genéticos “*Messy*”

Os Algoritmos Genéticos “*Messy*”, ou AGms, são uma classe de algoritmos iterativos de otimização que fazem uso de um padrão de busca local, de uma representação adaptativa e de uma estratégia de amostragem baseada em decisão (KARGUPTA, 1996). Esta classe foi criada com o objetivo de contornar algumas limitações dos AGs simples: a limitada capacidade de busca por relações (KARGUPTA, 1996) e a susceptibilidade à epistasia em problemas enganadores (TANOMARU, 1995). Algumas das vantagens do AGm sobre o AG tradicional são: boa imunidade à convergência prematura (evitando máximos locais) e a rapidez de execução (KOSTKA; SKVOR, 2002).

#### 2.1.7.1 Representação do AGm

O AGm relaxa a premissa de *locus*-fixo, característica da maioria dos AG's simples. Isto é possível a partir da definição do “gene *messy*” como um par ordenado (*locus*, *valor*), que o identifica por uma posição e um valor, e da definição do “cromossomo *messy*” como uma coleção de genes *messy* (GOLDBERG et al., 1993).

O AGm permite a utilização de cromossomos de tamanho variável, que podem ser sub-especificados ou super-especificados em relação ao problema em questão (GOLDBERG et al.,

1993). Isto significa que, para um dado problema de comprimento  $\ell = 3$ , por exemplo, ambos os cromossomos  $\{(0, 1)(2, 0)(1, 1)(2, 1)\}$  e  $\{(0, 1)(2, 1)\}$  são válidos. O primeiro cromossomo está super-especificado, enquanto o segundo está sub-especificado. Cromossomos super-especificados são mapeados em um espaço de dimensão  $\ell$ , tomando-se elementos da esquerda para a direita, sob o critério de “primeiro encontrado, primeiro servido”. Por outro lado, um cromossomo sub-especificado de comprimento  $k$  define uma classe de equivalência de ordem  $k$ , e é tratada através de uma busca local (KARGUPTA, 1996). Esta busca local consiste em determinar esquemas localmente ótimos, chamados “padrões competitivos” e preencher os genes não especificados a partir destes. Estes padrões competitivos podem ser determinados, por exemplo, através do método da “subida de encosta” (*hill climbing*) (FURTADO, 1998) ou selecionados durante a própria execução do AGm (GOLDBERG et al., 1993). Como exemplo, considere-se o padrão  $\{(0, 1)(1, 0)(2, 0)\}$ , e o cromossomo sub-especificado  $\{(0, 1)(2, 1)\}$ . O gene faltante é preenchido com o respectivo valor do padrão competitivo, resultando em  $\{(0, 1)(1, 0)(2, 1)\}$ .

Apesar do AGm permitir genes redundantes ou ausentes, esta representação é mais rica do que a representação de sequências simples, utilizada em outras versões do AG. Devido ao fato de o *locus* de cada gene ser explicitamente especificado, genes relacionados podem ser adaptativamente agrupados durante a busca, compondo uma “ligação forte” entre elementos do mesmo bloco construtivo (GOLDBERG et al., 1993). Esta ligação contribui para minimizar o “efeito de ruptura” (*operator disruption*) (KARGUPTA, 1996), ou seja, a destruição dos blocos construtivos pelos operadores genéticos. Adicionalmente, a existência destas ligações fortes facilita a recombinação de diferentes blocos construtivos, beneficiando desta forma a seleção das melhores estruturas, que por fim tende a levar à obtenção da solução ótima (GOLDBERG et al., 1993).

### 2.1.7.2 Principais Operações no AGm

**Seleção de Limiar** (*Thresholding Selection*): O papel do processo de seleção no AG é produzir um maior número de cópias dos melhores indivíduos, comparando os respectivos valores de adequabilidade. No AGm, esta comparação nem sempre pode ser feita diretamente.

Como exemplo, considerem-se  $\{(1, 0)(0, 0)\}$ ,  $\{(1, 1)(0, 1)\}$  e  $\{(1, 0)(2, 1)\}$  como supostos cromossomos *messy* em um problema de comprimento  $\ell = 3$ . Os dois primeiros definem as classes de equivalência  $00^*$  e  $11^*$  sobre a relação  $ff^*$ , onde  $f$  define um gene com valor estabelecido. O terceiro define a classe  $*01$ , sobre a relação  $*ff$ . As duas primeiras sequências podem ser comparadas diretamente, porém a comparação de qualquer uma destas duas com a terceira não faz sentido, e deve ser evitada. A seleção de limiar tem

como objetivo restringir a competição entre sequências pertencentes a diferentes relações, além de selecionar os indivíduos de maior adequabilidade (KARGUPTA, 1996).

**Operador de “Cortar e Colar”** (*Cut and Splice*): Este operador é análogo ao operador de cruzamento no AG tradicional, porém pode ser executado sobre sequências de diferentes comprimentos. Dados dois cromossomos, a operação de “cortar” determina pontos de corte aleatórios e então dissocia cada cromossomo em duas partes. A operação de “colar” permuta as partes cortadas de cada sequência e gera novos cromossomos (KARGUPTA, 1996).

Considerem-se as sequências  $\{(1,1)(1,0)(2,1)\}$  e  $\{(0,1)(2,0)(1,1)(0,0)\}$ . Supondo-se que os pontos de corte sejam respectivamente 2 e 1, operação de “cortar” produziria  $\{(1,1)(1,0)\} \dots \{(2,1)\}$  e  $\{(0,1)\} \dots \{(2,0)(1,1)(0,0)\}$ . Como resultado da operação de “colar”, ter-se-iam os cromossomos  $\{(1,1)(1,0)(2,0)(1,1)(0,0)\}$  e  $\{(0,1)(2,1)\}$ . Estas operações são ilustradas na Figura 12.

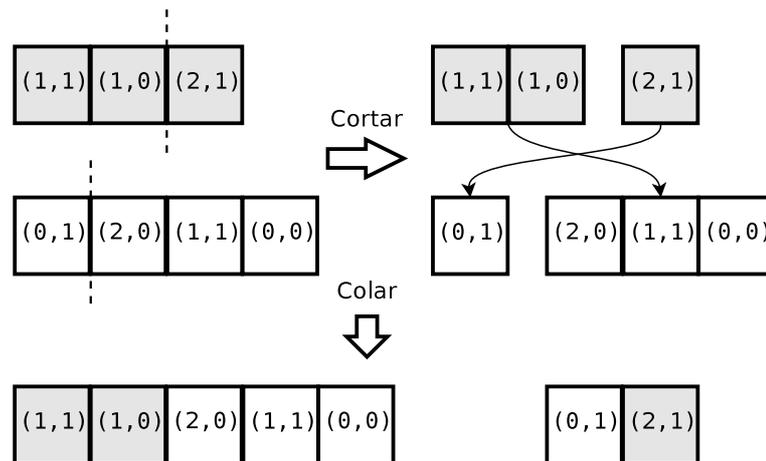


Figura 12: Operador de “cortar e colar”.

### 2.1.7.3 Funcionamento do AGm

O fluxograma de operação do Algoritmo *messy* é ilustrado na Figura 13.

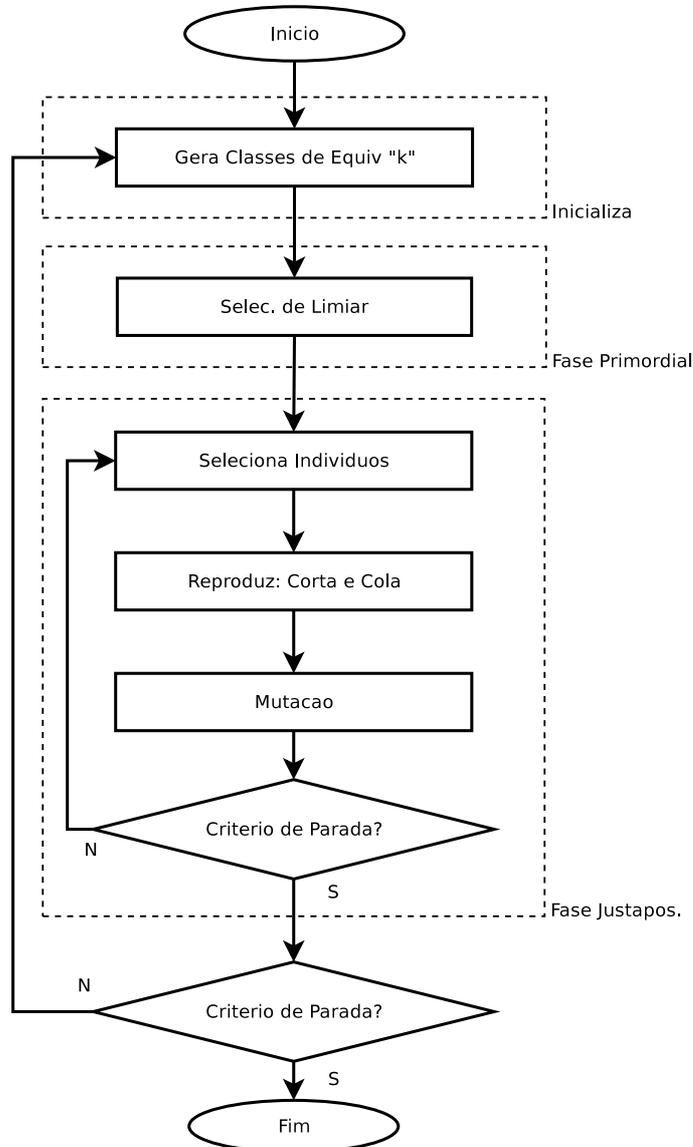


Figura 13: Fluxograma do AG *Messy*.

O AGm opera entre dois laços de execução: O laço externo e o laço interno. O laço interno do AGm original é composto de três fases (GOLDBERG et al., 1993):

1. Inicialização: A inicialização é realizada através da criação de uma população que contenha uma única cópia de todas as sequências de comprimento  $k$ . Fazendo isto, garante-se que todos os blocos construtivos do tamanho desejado estejam presentes, permitindo que sejam combinados pelo algoritmo para formar boas soluções finais. O grande problema de

representar todos estes blocos construtivos é o tamanho da população: em um problema de comprimento  $\ell$ , existe um total de  $\binom{\ell}{k}$  combinações de genes, e para cada combinação de genes,  $2^k$  combinações de alelos (supondo codificação binária). Isto resulta em uma população de  $2^k \times \binom{\ell}{k}$  indivíduos.

2. Fase primordial: Nesta fase, o processo de seleção de limiar é realizado, fazendo cópias de cadeias, de acordo com sua adequabilidade (sem recombinação ou mutação), de forma a manter uma boa porção de blocos construtivos promissores. É também possível ajustar o tamanho da população gerada na fase anterior, para que este se torne adequado à aplicação dos operadores na fase de justaposição.
3. Fase de justaposição: O processamento executado nesta fase é semelhante ao realizado no AG canônico. Durante esta fase, indivíduos são selecionados e submetidos a um operador de “cortar e colar”. Operadores de mutação também podem ser aplicados, porém não é comum a sua utilização.

O laço externo consiste na repetição do laço interno a cada “nível de blocos construtivos” (GOLDBERG et al., 1993), ou seja, a variável que controla a execução do laço externo é a ordem ( $k$ ) das classes de equivalência consideradas (KARGUPTA, 1996). Ao final de cada iteração do laço interno, o padrão competitivo assume o valor da melhor sequência encontrada na fase de justaposição. O algoritmo é encerrado quando a ordem das classes de equivalência atinge um determinado valor ou algum outro critério de parada é satisfeito (KARGUPTA, 1996).

### 2.1.8 Principais Vantagens do AG

Os algoritmos genéticos são métodos robustos, apresentando bons resultados em espaços de busca complexos. Diversas classes de problemas podem ser solucionadas via estes métodos, inclusive problemas de otimização multiobjetivos. A aplicação do AG não se restringe a problemas onde são impostas determinadas condições a respeito da função objetivo ou do espaço de busca, tais como continuidade, existência de derivadas ou unimodalidade (GOLDBERG, 1989). Restrições quanto aos valores das variáveis de decisão também podem ser facilmente integradas no algoritmo.

O AG é computacionalmente simples, apresentando baixo custo de implementação (GOLDBERG, 1989). É possível ainda construir sistemas híbridos, onde associa-se o AG a outros métodos de otimização a fim de obter resultados mais aprimorados (TANOMARU, 1995). Em virtude da natureza do AG, é possível ainda executá-lo utilizando técnicas de processamento

paralelo, que tendem a melhorar tanto a eficiência quanto a qualidade das soluções obtidas (CANTÚ-PAZ, 1997).

Para construir uma aplicação com AG, não é necessário conhecimento matemático profundo a respeito do problema em questão. Basta apenas que se saiba como classificar as soluções candidatas em relação a uma função-objetivo, ou, na falta desta, que se conheça uma maneira de classificar as soluções candidatas, uma em relação à outra (LOBO, 2000).

### **2.1.9 Desvantagens do AG**

Por conta da sua natureza probabilística, o AG “tende” a encontrar soluções ótimas. Porém, devida a esta mesma natureza, não há garantias de obtenção da solução ótima. Para o AG, a população representa um conjunto de possíveis resultados, diferenciando-se apenas pelo seu nível de adequabilidade (MARDLE; PASCOE, 1999). Uma solução-candidata pode ser considerada melhor apenas em comparação com outra. Não existe o conceito de “solução ótima”, e em geral não é possível determinar se um ponto no espaço de busca é o máximo (ou mínimo) global (ou mesmo local). Isto também implica em um critério de parada pouco objetivo para o algoritmo. Este critério é geralmente baseado em número de iterações executadas, tempo decorrido ou estagnação da evolução.

O AG tipicamente demanda mais recursos computacionais quando comparado com outras técnicas, como por exemplo o método de subida do gradiente (MCNELIS; DUFFY, 1998). Esta demanda por recursos pode manifestar-se tanto em consumo de memória (AG opera em uma população, ao invés de uma solução individual) quanto em tempo de processamento (AG itera por um determinado número de gerações sobre todos os indivíduos da população).

A literatura sobre Algoritmos Genéticos, embora vasta, possui pouco embasamento teórico. Os resultados e formulações estabelecidos são em grande maioria obtidos empiricamente. As teorias que tentam explicar matematicamente seu funcionamento são insatisfatórias (GOLDBERG, 1989; LOBO, 2000; THORNTON, 1997).

O ajuste de parâmetros do AG (tamanho da população, taxa de aplicação de operadores, critério de convergência, etc) é feito de maneira quase totalmente empírica. Embora existam trabalhos na área de determinação automática de parâmetros (LOBO, 2000; MARUO; LOPES; DELGADO, 2005; EIBEN; HINTERDING; MICHALEWICZ, 1999), esta tarefa geralmente depende da experiência do usuário e do conhecimento específico no domínio do problema em questão.

O AG não é intrinsecamente melhor do que qualquer outro algoritmo de otimização. Sem-

pre que exista a alternativa, deve-se dar preferência a métodos analíticos ou numéricos que garantam a obtenção da solução ótima.

### **2.1.10 Algoritmos Genéticos em Aplicações de Tempo Real**

Ciesielski e Scerri (1998) caracterizam uma aplicação de tempo real como sendo aquela que requer um certo processamento e a obtenção de um resultado dentro de um intervalo de tempo limitado, caso contrário, a informação relacionada perde a validade. O AG, em consequência de sua natureza iterativa e probabilística, é geralmente associado a longos períodos de processamento e incertezas sobre a obtenção do máximo global. Por estes motivos, o AG dificilmente é considerado como alternativa para a solução de problemas de tempo real. Por outro lado, estes autores apresentam justificativas para a aplicação de AG em determinadas tarefas de tempo real:

- Uma população de soluções potenciais está sempre disponível. Persistindo-se na execução do algoritmo, estas soluções tendem a tornar-se melhores (evoluir). Em qualquer instante, porém, existe uma solução “melhor” disponível dentro da população. O AG pode, portanto, ser considerado um “algoritmo de qualquer tempo”.
- O AG é inerentemente paralelo. Caso seja necessário adicionar maior capacidade computacional para obter um resultado aceitável, pode-se simplesmente adicionar um maior número de processadores, com um mínimo impacto sobre a implementação.

Além disso, não existem algoritmos eficientes que garantam a obtenção da solução ótima para problemas da classe de complexidade NP (SKIENA, 1997). Nestes casos, a única maneira de garantir a obtenção do máximo global é a enumeração de todos os pontos do espaço de busca. Um método capaz de encontrar uma solução, mesmo que sub-ótima, dentro de um intervalo de tempo aceitável, pode ser de grande utilidade em aplicações com requisitos temporais bem definidos.

Podem ser citados como exemplos do emprego de AG em aplicações de tempo real: o escalonamento de pouso e decolagem de aviões em aeroportos (CIESIELSKI; SCERRI, 1998), a geração dinâmica de planos de produção otimizados em linhas de produção (ROSSI; DINI, 2000) e o cancelamento de ruído em equipamentos de eletrocardiograma (KUMARAVEL; VISWANATHAN, 1999).

## 2.2 O Problema da Cobertura de Conjuntos

O problema da cobertura de conjuntos é uma questão clássica da área de informática e pesquisa operacional (BEASLEY; CHU, 1996; WIKIPEDIA, 2005c). Diversas aplicações práticas podem ser modeladas através desta teoria, por exemplo, simplificação de funções lógicas (SKIENA, 1997; BEASLEY; CHU, 1996), determinação da área de cobertura para uma rede de sensores (QUINTAO; MATEUS; NAKAMURA, 2004), cobertura de vértices (SKIENA, 1997), logística (BEASLEY; CHU, 1996) e processamento de alarmes (CARDOZO; TALUKDAR, 1988; WEN; CHANG; SRINIVASAN, 1995; WEN; CHANG, 1996; NEIS et al., 2005).

Dado um conjunto finito  $A$  (conjunto universo), e uma coleção  $S = \{S_1, S_2, \dots, S_n\}$  de subconjuntos do conjunto  $A$ , estes podem possuir alguns elementos em comum entre si. O problema aparece quando é desejado adquirir ou representar itens empacotados em um conjunto fixo de lotes. É necessário obter todos os itens, e ao mesmo tempo adquirir o menor número possível de lotes. A solução ótima para este problema possui duas propriedades (WIKIPEDIA, 2005c):

1. Inclui todos os elementos contidos no conjunto universo;
2. Não é possível escolher um menor número de conjuntos de entrada e ainda assim cobrir todos os elementos.

Na Figura 14 é ilustrada uma instância do problema da cobertura de conjuntos. Em (a) é representado o conjunto universo, contendo os pontos enumerados pelos algarismos de 1 a 12. Em (b) é dado um conjunto de subconjuntos do universo (a). Em (c) é ilustrada uma possível cobertura para o conjunto (a). Em (d) é ilustrada a cobertura ótima. A Figura 14 (d) permite que se visualize claramente as propriedades acima enumeradas, ou seja: todos os elementos do conjunto universo estão contemplados, e não é possível escolher uma combinação menor de subconjuntos que contenha todos os elementos do universo.

O problema da cobertura de conjuntos é comprovadamente NP-completo <sup>1</sup> (KARP, 1972), portanto não são conhecidos algoritmos eficientes para resolvê-lo. A única maneira de encontrar a solução ótima é enumerar todas as possíveis combinações dos conjuntos de entrada (método da “força-bruta”). Esta abordagem, porém, torna-se computacionalmente inviável quando o espaço de busca assume dimensões elevadas.

Diversos métodos alternativos são propostos para encontrar soluções da cobertura de conjuntos. Métodos heurísticos como o “*greedy set cover*”, métodos probabilísticos como “recozimento simulado” (*simulated annealing*) (SKIENA, 1997), e algoritmos genéticos (BEASLEY; CHU, 1996; QUINTAO; MATEUS; NAKAMURA, 2004) são frequentemente empregados.

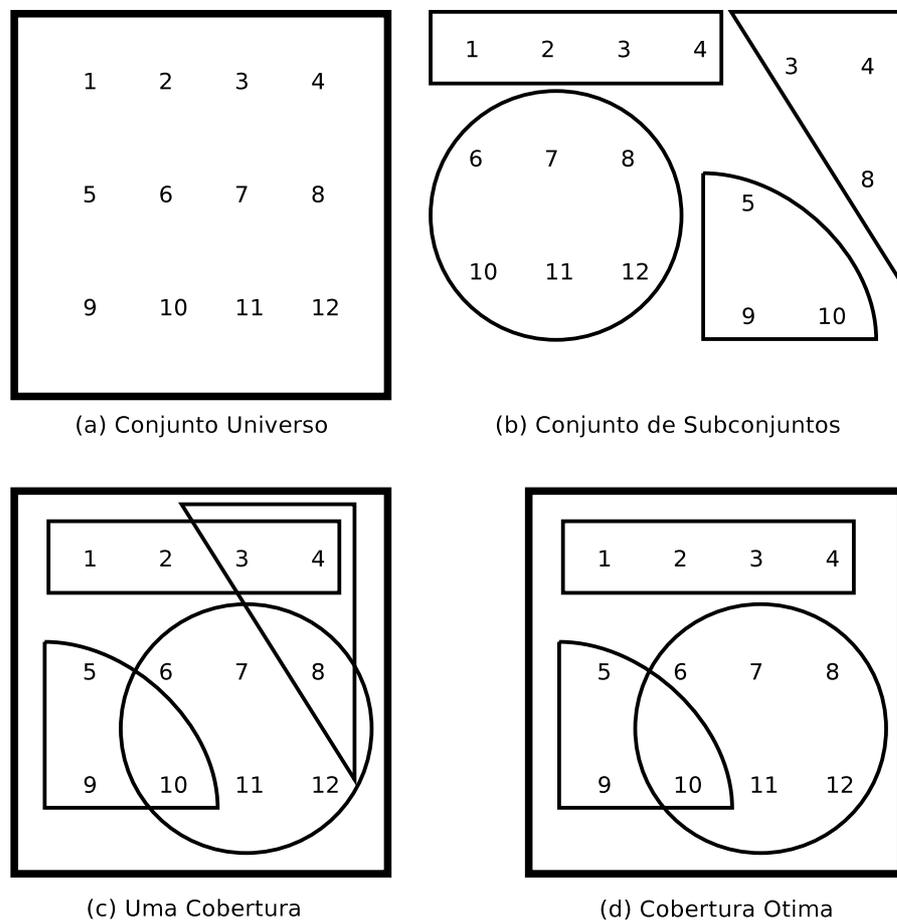


Figura 14: Uma instância do problema da cobertura de conjuntos.

<sup>1</sup>A esta classe pertencem problemas considerados “difíceis” do ponto de vista de consumo de recursos computacionais. De maneira simplificada, pode-se dizer que a solução de instâncias de problemas NP-completos demanda um tempo computacional que aumenta exponencialmente em relação ao tamanho da instância, tornando-as intratáveis. Para mais detalhes, ver (BOVET; CRESCENZI, 1994).



## 3 *Processamento de Alarmes*

Este capítulo posiciona o processador de alarmes em relação às demais ferramentas empregadas na supervisão e operação do sistema elétrico, apontando as principais funções deste componente. Alguns trabalhos previamente publicados na área de processamento de alarmes são brevemente analisados, salientando as abordagens usualmente empregadas na solução do problema.

### 3.1 Alarmes e Eventos

Considerando o tamanho e a complexidade de um sistema elétrico de potência típico, um operador humano pode não ser capaz de diagnosticar corretamente todas as condições anormais. Sofisticados programas computacionais foram desenvolvidos para automatizar este processo e chamar a atenção do operador para eventos notáveis, na forma de mensagens de alarme. Alarmes podem ser gerados pelo sistema supervisório em diversas condições, por exemplo quando:

- Um valor medido por um transdutor excede um limite (por exemplo, uma sobretensão);
- Um ponto digital muda de estado (por exemplo, a abertura de um disjuntor);
- Ocorre um erro em algum processo (por exemplo, quando ocorre um problema na execução de uma função automática);
- Uma falha de comunicação é detectada.

A definição de alarme é um pouco subjetiva, variando de um aplicativo para outro. De uma maneira simplista, pode-se definir alarmes como sintomas apresentados pelo sistema em consequência de eventos ocorridos (CARDOZO; TALUKDAR, 1988; WEN; CHANG; SRINIVASAN, 1995). Outros autores preferem denominar eventos como “causas” ou “perturbações”, e alarmes como “efeitos”, ou ainda “manifestações” (WEN; CHANG, 1996).

Neste trabalho, o termo “alarme” refere-se à manifestação de uma mensagem (textual e/ou sonora), produzida pelo sistema SCADA, que visa alertar o operador sobre uma condição anormal detectada no sistema elétrico. O termo “evento” refere-se a uma perturbação (ou ocorrência) no sistema elétrico que produz um determinado conjunto de alarmes.

## **3.2 Processamento de Alarmes e Diagnóstico de Falhas**

Sistemas frequentemente descritos na literatura como “processadores inteligentes de alarmes” utilizam mensagens de alarme para apontar a localização de falhas que ocasionalmente ocorrem na rede elétrica. Embora estes sistemas também busquem uma melhor apresentação das informações para o operador, não possuem o mesmo foco ou objetivo dos verdadeiros processadores de alarmes. Estas ferramentas são mais propriamente designadas por “sistemas de diagnóstico de falhas” (KIRSCHEN; WOLLENBERG, 1992).

Processadores de alarmes são projetados para tratar todos os possíveis tipos de alarmes. Sistemas de diagnóstico de falhas analisam apenas o conjunto de alarmes necessário para localizar a falha. Enquanto o objetivo do processador de alarmes é apresentar um panorama global claro da situação, sistemas de diagnóstico de falhas concentram-se em encontrar uma justificativa exata para um conjunto de sintomas. Em uma definição ainda mais simplista, poderíamos dizer que a função de um processador de alarmes é descrever “o que está acontecendo”, enquanto que o papel de um sistema de diagnóstico de falhas é “explicar por que alguns eventos estão ocorrendo” (KIRSCHEN; WOLLENBERG, 1992).

Processadores de alarmes devem fornecer uma resposta rápida (em “tempo real”), e portanto devem basear-se em modelos mais superficiais do sistema elétrico. Sistemas de diagnóstico de falhas realizam uma análise mais complexa, e portanto requerem um modelo mais detalhado. Por outro lado, um tempo de resposta maior é aceitável para estes.

## **3.3 Processamento Inteligente de Alarmes**

Os pacotes EMS empregados na operação de sistemas elétricos geralmente apresentam alguma ferramenta de tratamento de alarmes já incorporada (SMATHERS; AKHIL, 2001). Estas ferramentas são capazes de realizar operações básicas, como filtragem/inibição, priorização estática, agrupamento e roteamento de mensagens. Operações deste tipo constituem claramente uma forma de “processamento de alarmes” (HASAN et al., 1994). Este processamento reduz o número de mensagens e melhora a visibilidade de alarmes mais importantes, porém não pode

ser usada para sintetizar mensagens com maior conteúdo de informação. Além do mais, priorização estática não reflete a real importância de determinado alarme em qualquer contexto (KIRSCHEN; WOLLENBERG, 1992).

Por outro lado, programas que utilizam recursos de Inteligência Computacional ou sistemas baseados em regras (Inteligência Artificial clássica) para realizar esta tarefa, oferecem consideravelmente maior flexibilidade. É possível, através destes, combinar e correlacionar alarmes com informações obtidas de outras fontes (KIRSCHEN; WOLLENBERG, 1992), capturar o conhecimento de especialistas (HEIN, 1986), incorporar capacidade de generalização e tratamento de informações incompletas ou ruidosas (CHOI et al., 1995; SOUZA et al., 2004; WEN; CHANG; SRINIVASAN, 1995). Neste trabalho, por questões de clareza, será utilizada a expressão “processamento inteligente de alarmes” para referir-se a tais sistemas.

### 3.4 Arquitetura do Sistema

O processador inteligente de alarmes pode ser associado ao EMS de duas maneiras diferentes: embutido ou anexado (TESCH et al., 1990). Na arquitetura embutida, o processador é incorporado no EMS. Na arquitetura anexada, os alarmes são processados por um sistema de *software* independente, geralmente executado em um computador distinto, comunicando-se com o EMS via rede.

Os processadores inteligentes atuais são projetados para complementar ou estender os sistemas já existentes nos EMS. É assumido que o operador deve ter acesso às mensagens de alarme “cruas” (não processadas), caso seja necessário. Por estes motivos, a arquitetura mais utilizada é a anexada. Acredita-se que, em sistemas mais novos, a função de processamento inteligente de alarmes venha a ser integrada ao EMS (KIRSCHEN; WOLLENBERG, 1992). Por outro lado, a arquitetura anexada apresenta algumas vantagens importantes (TESCH et al., 1990):

- Geralmente o processador inteligente é executado em outro computador, portanto não compete por recursos com o SCADA/EMS.
- A implantação, manutenção e testes do sistema não perturbam o funcionamento do SCADA/EMS.

Porém, alguns inconvenientes também podem ser observados:

- Deve ser estabelecida uma conexão entre os computadores.
- Não há acesso direto ao banco de dados do SCADA, nem à sua interface com o operador.

## 3.5 Justificativas do Processamento Inteligente de Alarmes

Estudos revelam que os operadores de centros de controle percebem diversos problemas ao trabalhar com as mensagens de alarmes geradas durante a operação do sistema elétrico (HASAN et al., 1994), como por exemplo, o fato de que:

- Existem mensagens pouco específicas ou demasiadamente específicas;
- Uma quantidade muito grande de alarmes é gerada durante um distúrbio;
- Alarmes falsos são gerados;
- Uma multiplicidade de alarmes é gerada para um único evento;
- Mensagens de alarmes são modificadas muito rapidamente para serem lidas.

Estes problemas revelam a necessidade de ferramentas capazes de auxiliar o operador, provendo categorização, priorização, síntese e diagnóstico das mensagens de alarme. Abordagens inovadoras para o problema do processamento de alarmes devem contribuir para melhorar a forma e o conteúdo das mensagens de alarme apresentadas. O objetivo principal destas ferramentas é reduzir a quantidade de informação que o operador deve assimilar. A importância de cada pedaço de informação deve ser avaliada antes da sua apresentação, e apenas as mensagens que são realmente importantes devem ser mostradas. Este limite de importância pode ser determinado combinando mensagens de alarme primitivas. Estes programas também devem prender a atenção do operador e ajudá-lo a acompanhar a evolução do estado do sistema elétrico através de sumário de condições anormais. “Espera-se que fornecendo melhores informações ao invés de mais informações permita que o operador venha a reagir mais rápida e efetivamente” (KIRSCHEN; WOLLENBERG, 1992).

### 3.5.1 Objetivos do Processamento de Alarmes

Kirschen e Wollenberg (1992) estabelecem como objetivos principais de um sistema de processamento de alarmes:

1. Reduzir a quantidade de informação apresentada ao operador.
2. Apresentar uma idéia mais clara a respeito da condição que causou os alarmes.
3. Eventualmente, recomendar ações corretivas.

Cada empresa do setor elétrico possui suas peculiaridades e distintos problemas de operação. Estas diferenças influenciam as expectativas em relação ao processador de alarmes, e podem levar a uma definição um pouco diferente para os objetivos.

## **3.6 Métodos Empregados no Processamento Inteligente de Alarmes**

Os primeiros sistemas de alarmes utilizavam processamento combinatorial. Tabelas lógicas eram usadas para definir, de forma Booleana, estados particulares que mereceriam atenção do operador. A principal desvantagem deste método é a falta de flexibilidade, visto que cada estado é definido em termos de instâncias específicas de dispositivos e pontos do banco de dados. Desenvolver e manter tal programa requer um esforço enorme (KIRSCHEN; WOLLENBERG, 1992).

Por outro lado, o emprego de métodos baseados em Inteligência Computacional (ou mesmo IA clássica) podem apresentar diversas vantagens. Capacidade de generalização, possibilidade de tratamento de conjuntos de informações incompletos ou ruidosos, e facilidade de agregação do conhecimento de especialistas são benefícios imediatos.

A seguir, é feito um apanhado dos métodos empregados no processamento de alarmes mais frequentemente encontrados na literatura. Vale lembrar que é comum encontrar-se a descrição de sistemas híbridos. A finalidade da hibridização é obter melhores resultados explorando os pontos fortes de cada método associado.

### **3.6.1 Sistemas Baseados em Regras / Baseados em Conhecimento**

Sistemas especialistas baseados em regras oferecem consideravelmente mais flexibilidade que os métodos convencionais de tratamento de alarmes. Estes sistemas podem filtrar e priorizar, além de correlacionar informações. O raciocínio do operador pode ser emulado por este processo. A separação entre o conhecimento procedural (regras) e o conhecimento factual (dados) facilita o desenvolvimento e manutenção do sistema (KIRSCHEN; WOLLENBERG, 1992).

Pfau-Wagenbauer e Nejd (1993) propõem um sistema que utiliza uma combinação de modelos do sistema de proteção e regras heurísticas. O processador de alarmes é composto de diversas camadas de diagnóstico hierárquicas, utilizando regras heurísticas e também modelos compilados. Em uma primeira camada, regras de modelos qualitativos são utilizadas para eliminar soluções absurdas do universo de possíveis diagnósticos e localizar a área do distúrbio.

A segunda camada, utiliza tanto regras heurísticas quanto modelos para localizar o ponto exato da falha. A terceira camada utiliza informações topológicas para realizar uma análise global do distúrbio e seu impacto na rede.

Em (CHEON; CHANG; CHUNG, 1993) é proposto um sistema tratamento de alarmes em usinas nucleares. O método de diagnóstico adotado é um paradigma “hipótese e teste”. Isto consiste em gerar todas as possíveis hipóteses sobre as possíveis causas e então determinar sua validade ou não através de busca em uma base de regras causa-inferência. Os casos compostos por sintomas incertos ou insuficientes são tratados com um fator de certeza. O fator de certeza representa o grau com que se acredita que uma evidência é verdadeira.

Em (VALE; MOURA, 1993) é descrito um sistema especialista composto por um pré-processador e uma máquina de inferência. O pré-processador transforma as mensagens de alarme em fatos Prolog e os insere em uma base. Estes fatos são então processados pela máquina de inferência. A base de regras utilizada foi construída colhendo-se conhecimento técnico e empírico junto aos especialistas na operação do sistema elétrico. O aspecto temporal é também considerado: cada fato da base recebe uma estampa de tempo (correspondente a da mensagem de alarme). Esta estampa de tempo é utilizada para tratar de questões temporais.

Em (TESCH et al., 1990) é descrito o protótipo de um processador de alarmes baseado em conhecimento. Este sistema é capaz de substituir um grande número de alarmes por algumas mensagens de diagnóstico que descrevem os eventos que ocasionaram os alarmes. A arquitetura descrita é “orientada a estação”. Esta orientação permite a utilização de regras genéricas (válidas para todo o sistema), e também regras específicas (válida para estações particulares). O método de diagnóstico também utiliza o paradigma de hipótese e teste.

Uma das maiores desvantagens dos sistemas a base de regras é o esforço inerente necessário para construir e manter regras que cubram todas as possíveis condições anormais (EL-SAYED; ALFUHAID, 2000). Para tratar este problema, (DUARTE et al., 2003) propõe um sistema que combina o raciocínio baseado em regras com o raciocínio baseado em modelos. Esta abordagem tende a atenuar os impactos das constantes alterações na topologia das redes. Outro ganho importante obtido é a produção de uma base de regras mais compacta.

Outra grande limitação destes sistemas é que os mesmos costumam produzir bons resultados apenas para as situações que foram consideradas durante o desenvolvimento da base de conhecimento (MEZA et al., 2001). Diferentes eventos, com padrões de alarmes não previstos durante a fase de projeto, ou mesmo conjuntos de informações incompletas podem facilmente levar a um resultado incorreto.

### 3.6.2 Reconhecimento de Padrões

Hein (1986) descreve um sistema especialista que utiliza como entradas sinais em tempo real, vindos de disjuntores e dispositivos de proteção. Os incidentes são detectados pela presença de *clusters* destes sinais no conjunto de informações de entrada. Esta detecção é feita por técnicas de reconhecimento de padrões.

### 3.6.3 Redes Neurais Artificiais (RNA)

Redes neurais podem ser utilizadas para encontrar determinados tipos de problemas topológicos, mas ainda não foi provado que o método pode ser expandido para qualquer tipo de distúrbio (KIRSCHEN; WOLLENBERG, 1992).

Além da grande quantidade de sinais de entrada, a topologia do sistema elétrico é frequentemente modificada. Estas modificações não podem ser facilmente refletidas em uma rede neural com estrutura fixa e possuindo grande número de neurônios. Por isso, a configuração da RNA deve ser atualizada dinamicamente, refletindo as mudanças da topologia (EL-SAYED; ALFUHAID, 2000). Para reduzir o efeito das alterações topológicas, cada parte do sistema é tratada por uma rede paralela independente, aproveitando a característica paralela vista pelo centro de controle em relação às subestações (EL-SAYED; ALFUHAID, 2000). Isto também contribui para a redução do tempo de treinamento das redes, visto que, em sistemas de grande escala, este processo consome muito tempo. Uma abordagem semelhante é utilizada em (TESCH et al., 1990) para sistemas baseados em regras.

O treinamento e teste das redes é geralmente feito com dados históricos de operação (EL-SAYED; ALFUHAID, 2000; FRISCH; CARDOSO; ARRUDA, 1997; BATISTA, 2005; COUTTO FILHO et al., 1999). Um grande problema pode surgir no caso de um conjunto incompleto de dados históricos. Por exemplo, se uma família de eventos que constitui uma determinada região no espaço não consta na base (nunca ocorreu nenhum evento deste tipo), a RNA não será capaz de “aprender” este padrão, produzindo um resultado incorreto. Em (KHOSLA; DILLON, 1997) são discutidas diversas estratégias de treinamento de RNA's.

### 3.6.4 Sistemas de Inferência *Fuzzy*

A teoria dos conjuntos *fuzzy* foi desenvolvida para modelar imprecisões, ambiguidade e incerteza nas informações. O processamento de alarmes em sistemas elétricos é um problema que envolve diversas incertezas. A aplicação de sistemas *fuzzy* nesta tarefa permite modelar estas

incertezas e tratar conjuntos de dados incompletos, levando em conta informações qualitativas capturadas junto aos operadores (especialistas) (MEZA et al., 2001).

Em (CHOI et al., 1995), é descrito um processador de alarmes para usinas nucleares utilizando sistemas *fuzzy*. O processo de diagnóstico utiliza inferência *fuzzy* composicional, por ser apropriada para manipular incertezas ou dados insuficientes.

Uma metodologia que combina RNA's e sistemas *fuzzy* para processamento de alarmes e identificação de componentes defeituosos é apresentada por Souza et al. (2004). Relações *fuzzy* são estabelecidas, e formam uma base de dados empregada para treinar RNA's. Esta, por sua vez, recebe como entradas padrões de alarmes, produzindo como saída a estimativa do grau de pertinência de um equipamento específico à classe dos componentes defeituosos.

### 3.6.5 Algoritmos Genéticos

Métodos de processamento de alarmes baseados em algoritmos genéticos (WEN; CHANG; SRINIVASAN, 1995; WEN; CHANG, 1996; NEIS et al., 2005) têm a vantagem de poder encontrar múltiplas soluções globais ótimas (ou próximas das soluções ótimas) de maneira direta e eficiente, especialmente em casos de alarmes falsos e/ou não reportados, onde diferentes combinações de eventos podem produzir o mesmo conjunto de alarmes.

Em (WEN; CHANG; SRINIVASAN, 1995), foi proposto pela primeira vez um método de processamento de alarmes baseado na otimização de uma função-objetivo através de um algoritmo genético. Esta função-objetivo reflete um critério matemático formal que descreve o problema do processamento de alarmes. Posteriormente o problema é analisado sob a luz da teoria de MFD (*Multiple Fault Diagnosis* - Diagnóstico de Múltiplas Falhas) (WEN; CHANG, 1996). Para esta ciência, um evento do sistema elétrico corresponde a uma “desordem”, enquanto um alarme corresponde a uma “manifestação”. A teoria de cobertura de conjuntos e o princípio da parcimônia<sup>1</sup> constituem o fundamento teórico para a determinação da função-objetivo. Por fim, o AG é empregado para encontrar múltiplas soluções desta função.

#### 3.6.5.1 Relação entre Eventos e Alarmes

Em um sistema SCADA/EMS, todas as possíveis mensagens de alarme constituem o conjunto de alarmes do sistema. Este conjunto pode ser representado como  $A_s = \{a_1, a_2, \dots, a_{na}\}$ ,

---

<sup>1</sup>O “Princípio da Parcimônia”, ou “Navalha de Occam” afirma que a explicação para um dado fenômeno deve fazer tão poucas suposições quanto possível sobre suas causas, eliminando hipóteses desnecessárias. Isto implica na opção por uma solução mais simples em detrimento de uma mais complexa, quando ambas apresentarem o mesmo desempenho.

sendo  $n_a$  o número de possíveis alarmes.

Todos os possíveis eventos diagnosticados constituem o conjunto de eventos do sistema. Este conjunto pode ser descrito como  $E_s = \{e_1, e_2, \dots, e_{n_e}\}$ , onde  $n_e$  é o número de possíveis eventos. A relação entre um evento e seu correspondente padrão de alarmes pode ser descrita como (WEN; CHANG; SRINIVASAN, 1995):

$$e_i \rightarrow A_i \quad i = 1, 2, \dots, n_e \quad (3.1)$$

Onde  $e_i \in E_s$  representa o  $i$ -ésimo evento e  $A_i \subset A_s$  é o conjunto característico de alarmes associado a este evento. O símbolo “ $\rightarrow$ ” denota a relação entre  $e_i$  e  $A_i$ . A equação 3.1 estabelece que um evento pode produzir um conjunto característico de alarmes, portanto, o problema do processamento de alarmes pode ser descrito como:

“Dado um conjunto de alarmes denotado por  $A_r, A_r \subset A_s$ , determinar quais eventos podem explicar  $A_r$ .”

O subscrito “ $r$ ” denota “reportado”, indicando que este é o conjunto de alarmes reportados ao sistema supervisor. Quando o conjunto  $A_r$  coincide exatamente com um dos subconjuntos  $A_i$  predefinidos na equação 3.1, a solução torna-se trivial. Em situações reais, porém, o conjunto  $A_r$  não coincide com nenhum dos  $A_i$  em particular. É possível, por outro lado, escrever  $A_r$  como uma combinação de subconjuntos  $A_i$ . Esta combinação deve ser parcimoniosa, ou seja, deve conter o menor número possível de subconjuntos. Este problema consiste em uma variação da cobertura de conjuntos (seção 2.2).

### 3.6.5.2 Critério para descrição do problema

O critério que descreve o processamento de alarmes segundo um problema de otimização pode ser expresso como (WEN; CHANG; SRINIVASAN, 1995; WEN; CHANG, 1996): “Encontrar todos os vetores de eventos  $E_i$  que minimizam  $f(E_i)$ ”

$$f(E_i) = |\Delta A| \quad (3.2)$$

onde:

- $E_i$  são hipóteses de eventos, expressas na forma vetorial de  $n_e$  elementos binários. Cada elemento do vetor  $E_i$  representa o estado de um evento do sistema, e recebe o valor 1 se assume-se que o evento ocorreu, e 0 caso contrário.
- $\Delta A$  é um vetor de  $n_a$  elementos, que determina a diferença entre o conjunto de alarmes

reportado e o conjunto simulado no caso da ocorrência de  $E_i$ .  $\Delta A$  pode ser escrito como:

$$\Delta A = A_r - A_m(E_i) \quad (3.3)$$

sendo que:

- $A_r$  representa o vetor dos alarmes reportados ao sistema. Cada elemento de  $A_r$  assume o valor 1 se o respectivo alarme foi reportado ao sistema, e 0 caso contrário.
- $A_m(E_i)$  representa o vetor de estado dos alarmes do sistema, simulado no caso da ocorrência de  $E_i$ .
- O operador (-) denota “subtração binária”<sup>2</sup>, elemento a elemento dos vetores  $A_r$  e  $A_m(E_i)$ , de acordo com a Tabela 2.
- $|\Delta A|$  denota o número de elementos não-zero em  $\Delta A$ .

Tabela 2: Tabela-verdade da operação (-).

X1	X2	X1-X2
0	0	0
0	1	1
1	0	1
1	1	0

Este critério pode, inapropriadamente, produzir múltiplas soluções, mesmo para casos simples. Um método razoável para reduzir o número de soluções é aplicar o Princípio da Parcimônia, tomando-se apenas as soluções mais simples (com o menor número de eventos).

Para a aplicação do AG, a codificação do problema é direta, ou seja, o próprio vetor  $E_i$  pode ser utilizado na representação cromossômica dos indivíduos que compõem a população de soluções-candidatas. A função-objetivo, a ser minimizada, pode então ser escrita segundo a equação 3.2. A função de *fitness*, por sua vez, deve ser escrita segundo um critério de maximização. Isto pode ser feito escolhendo-se um valor positivo  $W$ , tão grande quanto necessário, e escrevendo:

$$f_{fit} = W - |\Delta A| \quad (3.4)$$

<sup>2</sup>A tabela-verdade desta operação coincide com a tabela-verdade da operação “ou-exclusivo”. Por razões de coerência com a notação original empregada em (WEN; CHANG; SRINIVASAN, 1995), e pelo fato de o símbolo (-) enfatizar o sentido de “diferença” entre os conjuntos, optou-se por mantê-la desta forma.

### 3.6.5.3 Implementação Apresentada Pelos Autores

Wen, Chang e Srinivasan (1995) descrevem um protótipo do processador de alarmes construído para testes. As principais características deste protótipo são:

**Codificação:** Segundo apresentado anteriormente, é desejado encontrar um ou mais vetores  $E_i$  que minimizam a equação 3.2. Para conveniência,  $E_i$  está representado na forma binária, podendo ser diretamente utilizado no algoritmo genético, sem codificações adicionais. O tamanho do espaço de busca fica também determinado, sendo igual a  $2^{10}$ .

**Tamanho da população:** O tamanho da população utilizada é de 100 indivíduos.

**Número de gerações:** O máximo número de gerações estabelecido é  $MG = 6$  (WEN; CHANG; SRINIVASAN, 1995) e  $MG = 8$  (WEN; CHANG, 1996). O critério de parada empregado no algoritmo é atingir o máximo número de gerações.

**Tipo de seleção:** Seleção por torneio estocástico.

**Operadores de cruzamento:** Algumas variações do protótipo são descritas, empregando cruzamento de um ponto, multi-ponto e uniforme.

Os testes foram feitos em um sistema fictício, constituído de 11 alarmes e 10 eventos. A codificação destes parâmetros produz um cromossomo binário com 10 elementos, resultando em um espaço de busca com tamanho  $2^{10} = 1024$  pontos.

### 3.6.5.4 Deficiências da Implementação

A implementação apresentada por Wen, Chang e Srinivasan (1995) apresenta diversas deficiências que inviabilizam sua aplicação no processamento de alarmes em um sistema supervisor real. Dentre estas, podem ser enumeradas:

1. O modelo empregado é demasiadamente simples. A representação dos alarmes/eventos através de quantidades binárias não modela corretamente o comportamento do sistema. Um único alarme pode ser reportado mais de uma vez para uma mesma varredura, no caso de ocorrências sequenciais.

Para ilustrar esta situação, pode ser citado o caso de um curto-circuito em uma linha de transmissão. Este evento poderia provocar, por exemplo, a atuação dos dispositivos de proteção de distância por zona 1 (relé 21Z1), que por sua vez atuaria sobre o relé 94

(*buffer* de acionamento do disjuntor), acarretando a abertura do disjuntor da linha. Dependendo da configuração do sistema, um esquema de religamento automático poderia ser acionado (relé 79), fechando o disjuntor. No caso da persistência do curto, uma nova atuação da proteção causaria a abertura definitiva do disjuntor e o bloqueio da função religamento automático. Uma típica listagem de alarmes para esta ocorrência seria semelhante à ilustrada na Figura 15.

```

09:16:38:195 SE-? LINHA 69 RELE 21Z1 DESLIGAMENTO
09:16:38:195 SE-? LINHA 69 RELE 94X DESLIGAMENTO
09:16:38:195 SE-? LINHA 69 DJ 52-?? ABERTO
09:16:38:195 SE-? LINHA 69 RELE 94X NORMAL
09:16:38:195 SE-? LINHA 69 RELE 79 RELIGAMENTO
09:16:38:195 SE-? LINHA 69 DJ 52-?? FECHADO
09:16:38:195 SE-? LINHA 69 RELE 21Z1 DESLIGAMENTO
09:16:38:195 SE-? LINHA 69 RELE 94X DESLIGAMENTO
09:16:38:195 SE-? LINHA 69 DJ 52-?? ABERTO
09:16:38:195 SE-? LINHA 69 BLOQ RELIG BLOQUEADO

```

Figura 15: Listagem de alarmes.

Percebe-se que determinadas mensagens de alarme aparecem repetidas vezes. Esta característica deve ser prevista no modelo do sistema de processamento de alarmes.

2. Não é levada em conta a sequência em que os alarmes são reportados e/ou o momento da ocorrência dos eventos.
3. O sistema elétrico empregado pelo autor nas simulações possui dimensões bastante reduzidas, implicando em um espaço de busca muito pequeno ( $2^{10} = 1024$  pontos). Um espaço de busca com estas dimensões não justifica o emprego de um método computacional como o AG. Problemas desta ordem de grandeza podem ser resolvidos de maneira rápida e eficiente por métodos enumerativos, garantindo a obtenção da solução ótima em tempo razoável. Considerando os parâmetros utilizados pelos autores, como o tamanho da população (100 indivíduos) e o número de gerações (8 gerações), ao final do processo 800 indivíduos teriam sido avaliados, ou seja,  $\frac{800}{1024} \times 100 = 78,12\%$  de todo o universo de possíveis soluções, indicando claramente que a busca exaustiva teria sido mais apropriada.
4. Não é apresentada justificativa para a determinação de alguns parâmetros do AG, como por exemplo o máximo número de gerações empregadas ( $MG = 6$  (WEN; CHANG; SRINIVASAN, 1995) e  $MG = 8$  (WEN; CHANG, 1996)).

## 4 *Uma Nova Abordagem para o Processamento de Alarmes*

Neste capítulo o problema do processamento de alarmes é descrito formalmente, segundo um critério de otimização. Diversos modelos de solução, baseados em algoritmos genéticos, são propostos e discutidos.

### 4.1 **Processamento de Alarmes e Cobertura de Conjuntos**

A questão do processamento de alarmes pode ser descrita como um caso particular do problema de cobertura de conjuntos. Cardozo e Talukdar (1988) tratam o problema como tal. Considerando que:

- Cada evento  $e_i$  do sistema está relacionado a um subconjunto de alarmes do sistema, denotado por  $A_i$ ;
- Podem ocorrer simultaneamente (ou quase simultaneamente) diversos eventos no sistema elétrico;
- Os alarmes disparados pela ocorrência destes diversos eventos aparecem justapostos no sistema supervisorio. Este conjunto de alarmes é denotado por  $A_r$ .
- Dado um conjunto  $A_r$  de alarmes (sintomas), é possível determinar de forma inequívoca os eventos (causas) que os dispararam;

É possível estabelecer o seguinte critério:

“Dado um conjunto  $A_r$  de alarmes reportados ao sistema, é desejado encontrar o menor conjunto de subconjuntos  $A_i$  que contém todos os elementos de  $A_r$ .”

Este critério reflete exatamente a descrição do problema da cobertura de conjuntos. É interessante notar, porém, que o sistema elétrico, assim como outros sistemas reais, não é um

ambiente ideal, estando sujeito a falhas e submetido à presença de ruído. Deve-se, portanto, considerar que:

- O conjunto de alarmes reportado pode conter mensagens não estritamente correlacionadas a nenhum evento. Estas mensagens (alarmes falsos) podem ser provocadas por fenômenos não-determinísticos, como, por exemplo, o “repique” de contatos em relés eletromecânicos.
- Mensagens de alarme podem perder-se por conta de falhas em canais de comunicação, sensores ou equipamentos de medida. Nesta situação, o conjunto de alarmes reportado  $A_r$  não contém todas as mensagens correspondentes ao conjunto de perturbações que as desencadeou.

Assim sendo, algum critério deve ser especificado para refletir quão bem um evento ou combinação de eventos explica o conjunto de alarmes reportados. Deve-se considerar a hipótese da existência de alarmes não reportados e de estarem presentes alarmes falsos em  $A_r$ . Na Figura 16 é feita uma representação pictórica do problema do processamento de alarmes. Nesta figura, o conjunto  $A_r$  aparece parcialmente coberto pelo conjunto de subconjuntos  $A_i$ ,  $i \in \{1, 2, 3, 4, 5\}$ . As áreas não cobertas de  $A_r$  correspondem aos alarmes falsos, enquanto as áreas dos subconjuntos  $A_i$  que extrapolam os limites de  $A_r$  correspondem aos alarmes não reportados.

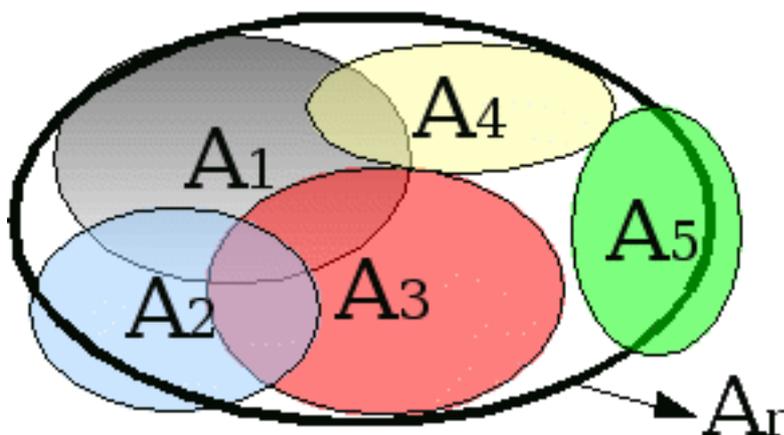


Figura 16: Processamento de alarmes e cobertura de conjuntos.

O problema assim exposto não pode ser chamado de “cobertura de conjuntos”, pois  $A_r$  não é totalmente coberto pelos subconjuntos  $A_i$ . As características do problema, porém, ainda são bastante semelhantes, diferindo apenas no critério de avaliação da solução. No problema

original, é desejado encontrar o menor conjunto de subconjuntos  $A_i$  que contém todos os elementos de  $A_r$ . No caso do processamento de alarmes, é desejado encontrar o menor conjunto de subconjuntos  $A_i$  para o qual a soma do número de alarmes falsos (área não coberta de  $A_r$ ) e não reportados (área que extrapola os limites de  $A_r$ ) seja mínima. Nas seções seguintes, são descritos critérios para avaliar estas soluções.

## 4.2 Solucionando o Problema do Processamento de Alarmes com AG

Para que um problema possa ser resolvido com AG, é necessário escrever uma função que avalie quão boa é uma determinada solução. Isto é feito através de uma função de avaliação responsável por fornecer uma medida de “diferença” entre dois conjuntos de alarmes. Para obter-se a solução ótima, esta função deve ser minimizada. A seguir, são expostos critérios que modelam a questão do processamento de alarmes segundo um problema de otimização.

### 4.2.1 O Modelo “Clássico”

Este é o modelo proposto por Wen, Chang e Srinivasan (1995), brevemente discutido na seção 3.6.5.2. Para fins de comparação, o modelo original será transcrito nesta seção, porém uma notação ligeiramente diferente será adotada. Escrevendo-se na forma vetorial:

$V_{AS} = [a_1, a_2, \dots, a_{n_a}]$ : O vetor de alarmes do sistema.

$V_{ES} = [e_1, e_2, \dots, e_{n_e}]$ : O vetor de eventos do sistema.

$V_{Ar} = [x_{a_1}, x_{a_2}, \dots, x_{a_{n_a}}]$ : O vetor de alarmes reportados ao sistema. A variável  $x_{a_n}$ ,  $n \in \{1, 2, \dots, n_a\}$  representa um número binário, que assume o valor 1 se o alarme foi reportado, e 0 caso contrário.

$E_i = [x_{e_1}, x_{e_2}, \dots, x_{e_{n_e}}]$ : Um vetor de eventos simulado. A variável  $x_{e_n}$ ,  $n \in \{1, 2, \dots, n_e\}$  representa um número binário, que assume o valor 1 se o evento supostamente ocorre, e 0 caso contrário.

$V_m(E_i)$ : Um vetor de alarmes binário, de mesma dimensão de  $V_{Ar}$ . Este vetor representa os alarmes característicos correspondentes aos eventos  $E_i$ .

$\Delta A = V_{Ar} - V_m(E_i)$ : O vetor diferença entre  $V_{Ar}$  e  $V_m(E_i)$ . O operador “-” denota “subtração binária”, elemento a elemento dos vetores, segundo a Tabela 2 (seção 3.6.5.2).

Sendo  $|\Delta A|$  o número de elementos não-zero em  $\Delta A$ , a função-objetivo do problema con-

siste em minimizar  $|\Delta A|$ . Para obter uma função de *fitness* a ser maximizada, sempre positiva, escolhe-se uma constante  $W$  apropriada, da qual será reduzido o valor da função-objetivo. Para o caso em questão, basta que  $W$  seja igual ou maior do que  $n_a$ . Outra conveniência em relação à função de *fitness* é que ela tenha seus valores normalizados no intervalo  $[0, 1]$ . Assim sendo, pode-se escrever:

$$f_{fit} = \frac{n_a - |\Delta A|}{n_a} \quad (4.1)$$

O tamanho do espaço de busca é determinado pelo número máximo de valores que podem ser assumidos pelo vetor  $E_i$ . Considerando-se que o número de diferentes valores que cada elemento do vetor  $E_i$  pode assumir (tamanho do alfabeto) é  $\alpha = 2$ , e que o número de elementos é  $n_e$ , o tamanho do espaço de busca é dado pela equação 4.2:

$$S_b = \alpha^{n_e} \quad (4.2)$$

Para fins de comparação, em uma SE típica seriam mapeados cerca de 200 possíveis eventos. O espaço de busca seria então composto de  $2^{200} \simeq 1,61 \times 10^{60}$  pontos. Deste total, um AG com parâmetros típicos (população de 100 indivíduos, rodando por 100 gerações) exploraria somente cerca de  $100 \times 100 = 10^4$  pontos. Este número representa uma parcela muito pequena do espaço total, tornando improvável a obtenção de bons resultados.

## 4.2.2 Problemas no Modelo

O modelo descrito anteriormente apresenta pelo menos duas grandes deficiências que dificultam sua aplicação no processamento de alarmes em um sistema supervisório real. São elas:

1. Um único alarme pode ser reportado mais de uma vez para uma mesma varredura, no caso de ocorrências sequenciais. Da mesma forma, um determinado evento pode ocorrer repetidas vezes.
2. Não é levada em conta a sequência em que os alarmes são reportados e/ou o momento da ocorrência dos eventos.

## 4.2.3 Melhorias Necessárias

Uma melhoria aparentemente óbvia seria introduzir as variáveis "tempo da ocorrência" e "número de ocorrências" no modelo. A introdução do "número de ocorrências" pode ser realizada sem grandes dificuldades. Há uma grande limitação, porém, no caso da ausência da

estampa SOE<sup>1</sup>, quando a estampa de tempo associada ao alarme não corresponde ao momento da ocorrência, nem mesmo reflete a sequência real dos acontecimentos. É necessário, portanto, criar modelos capazes de operar em ambas as situações.

#### 4.2.4 Modelo Melhorado 1: Vetor de Eventos Inteiros

Uma evolução natural do modelo da seção 4.2.1 é representar os alarmes e eventos através de números inteiros. Desta forma, resultariam:

$V_{Ar} = [x_{a_1}, x_{a_2}, \dots, x_{a_{na}}]$ : O vetor de alarmes reportados ao sistema. A variável  $x_{a_n} \in \mathbb{Z}$ ,  $n \in \{1, 2, \dots, n_a\}$  representa a contagem do número de vezes que um determinado alarme do sistema foi reportado.

$E_i = [x_{e_1}, x_{e_2}, \dots, x_{e_{ne}}]$ : Um vetor de eventos simulado. A variável  $x_{e_n} \in \mathbb{Z}$ ,  $n \in \{1, 2, \dots, n_e\}$  representa o número de vezes que um determinado evento do sistema ocorreu.

$V_m(E_i) = [y_{a_1}, y_{a_2}, \dots, y_{a_{na}}]$ : Representa o vetor de alarmes obtido a partir do vetor de eventos  $E_i$ . A variável  $y_{a_n} \in \mathbb{Z}$ ,  $n \in \{1, 2, \dots, n_a\}$  representa o número de vezes que um determinado alarme do sistema seria reportado, no caso da ocorrência do vetor de eventos  $E_i$ .

$\Delta A = V_{Ar} - V_m(E_i)$ : O vetor diferença entre  $V_{Ar}$  e  $V_m(E_i)$ . O operador “-” denota subtração aritmética elemento a elemento dos vetores  $V_{Ar}$  e  $V_m(E_i)$ .

A função-objetivo, a ser minimizada, pode ainda ser escrita segundo a equação 3.2, com a ressalva de que o operador “|” representa a somatória dos valores absolutos dos elementos de  $\Delta A$ . De uma maneira mais formal:

$$|\Delta A| = \sum_{j=1}^{n_a} abs(\Delta A[j]) \quad (4.3)$$

Onde o operador “[ ]” indica acesso ao  $j$ -ésimo elemento do vetor  $\Delta A$ .

A codificação do problema consiste em representar o vetor  $E_i$  por uma sequência de números binários<sup>2</sup>. A função de *fitness*, escrita na forma de maximização, pode ainda ser definida de maneira idêntica à equação 3.4. Neste caso, porém,  $W$  deve ser redefinido de forma a garantir que seja sempre maior ou igual a  $|\Delta A|$ . Se for considerado que cada evento dispara uma quantidade máxima de alarmes igual a  $M_a$ , e assumido que cada evento pode ocorrer no máximo  $M_e$

<sup>1</sup>*Sequence Of Events* - mudanças de estado são estampadas no exato momento da ocorrência, no próprio equipamento de aquisição de dados.

<sup>2</sup>Neste trabalho, optou-se por utilizar codificação binária. Eventualmente, outro tipo de codificação poderia ser utilizado.

vezes no intervalo de tempo analisado,  $W$  torna-se:

$$W = M_e \times n_e \times M_a \quad (4.4)$$

Os valores destes parâmetros podem ser determinados empiricamente. Baseado em características de equipamentos como os religadores, por exemplo, que normalmente são configurados para atuar por no máximo três vezes seguidas, pode-se estabelecer um valor de  $M_e = 3$ . Para os casos observados, cada evento costuma gerar não mais que cinco alarmes. Pode-se arbitrar, portanto,  $M_a = 5$ .

A função de *fitness* normalizada no intervalo  $[0, 1]$  assume a forma da equação 4.5:

$$f_{fit} = \frac{W - |\Delta A|}{W} \quad (4.5)$$

O parâmetro  $M_e$  exposto anteriormente determina também o tamanho do alfabeto dos elementos de  $E_i$ . Considerando que cada evento pode ocorrer de 0 (nenhuma vez) até  $M_e$  vezes, pode-se determinar:

$$\alpha = M_e + 1 \quad (4.6)$$

O tamanho do espaço de busca é dado pela equação 4.2. Para efeitos de comparação, em uma subestação típica podem ser mapeados cerca de 200 eventos, resultando em  $n_e = 200$ . Assumindo  $M_e = 3$  resultaria em um espaço de busca com  $4^{200} \simeq 2,58 \times 10^{120}$  pontos.

#### 4.2.4.1 Exemplo Numérico

Considerando-se um sistema elétrico fictício, composto pela subestação “XYZ”, possuindo 5 alarmes ( $A_s = \{a_1, a_2, a_3, a_4, a_5\}$ ) e 3 eventos ( $E_s = \{e_1, e_2, e_3\}$ ). Estes alarmes e eventos estão descritos detalhadamente na Tabela 3.

Tabela 3: Conjunto de eventos e alarmes do sistema.

Alarme	Mensagem	Evento	Descrição
$a_1$	XYZ 13 DJ 52- 01 ABERTO	$e_1$	Operação de Religamento no alimentador XYZ
$a_2$	XYZ 13 DJ 52- 01 FECHADO	$e_2$	Religamento mal sucedido no alimentador XYZ
$a_3$	XYZ 13 RELIG BLOQ	$e_3$	Sobretensão no alimentador XYZ
$a_4$	XYZ 13 RELE 50/51 F/N DESLIG		
$a_5$	XYZ 13 SOBRETENSAO		

Cada evento do sistema está relacionado a um conjunto característico de alarmes. Esta relação, na forma vetorial, é descrita como:

$$e_1 \rightarrow [1, 1, 0, 1, 0] \quad e_2 \rightarrow [2, 1, 1, 2, 0] \quad e_3 \rightarrow [0, 0, 0, 0, 1]$$

O significado destas relações é o seguinte: Cada evento é caracterizado por uma série de operações de dispositivos de proteção, chaves ou determinados valores de grandezas analógicas. Considerando o evento  $e_1$ , por exemplo, a “Operação de Religamento no alimentador XYZ” é caracterizada por:

1. Atuação de um dispositivo de proteção ( $a_4$ );
2. Abertura do disjuntor ( $a_1$ );
3. Fechamento do disjuntor ( $a_2$ );

Supondo que tenham sido reportados ao sistema os alarmes ilustrados na Figura 17. Na forma vetorial, ter-se-ia:

$$V_{Ar} = [1, 1, 0, 1, 1]$$

```
XYZ 13 SOBRETENSAO
XYZ 13 RELE 50/51 F/N DESLIG
XYZ 13 DJ 52- 01 ABERTO
XYZ 13 DJ 52- 01 FECHADO
```

Figura 17: Alarmes reportados.

Procura-se então um vetor de eventos que minimize a equação 3.2. Neste caso, o vetor de eventos possui três elementos. Supondo  $M_e = 4$ , cada elemento pode assumir valores inteiros no intervalo  $[0, 4]$ , resultando em  $\alpha = 5$ . O tamanho do espaço de busca seria determinado pela equação 4.2, portanto:

$$S_b = 5^3 = 125$$

Analisando o conjunto de alarmes reportado, é possível concluir facilmente que a solução ótima para esta instância do problema consiste em:

$$E = [1, 0, 1]$$

ou seja, ocorreram os eventos  $e_1$  e  $e_3$ , pois:

$$V_m(E) = V_m(e_1) + V_m(e_3)$$

que por sua vez é igual a  $V_{Ar}$ . Para este caso, a solução torna-se trivial, porém, em situações reais, o número de possíveis eventos seria consideravelmente maior, tornando o problema intratável por métodos enumerativos, mesmo empregando um grande volume de recursos computacionais.

#### 4.2.5 Modelo Melhorado 2: Palavra de Eventos Sobre o Alfabeto $E_s$

Uma descrição linguística da função-objetivo utilizada no modelo anterior seria: “encontrar um vetor  $E_i$ , de dimensão  $n_e$ , que gera um vetor  $V_m$ , de dimensão  $n_a$ , e que seja o mais “parecido” possível com o vetor de alarmes  $V_{Ar}$  reportado ao sistema”. Sabe-se que o vetor  $V_{Ar}$  foi produzido em decorrência de alguns eventos que se passaram no sistema elétrico. O que não se sabe é QUANTOS eventos e QUAIS eventos.

O número máximo de eventos ocorridos em um determinado intervalo de tempo não possui um limite estabelecido (teoricamente, pode ser infinito). Pode-se, porém, afirmar que geralmente não ocorre um número muito grande de eventos ao mesmo tempo. Este número costuma ser pequeno em relação ao número total de eventos do sistema ( $n_e$ ). Um resultado prático que comprova esta afirmação é que os vetores encontrados no processo de otimização da função 3.2 são esparsos - possuem muitos zeros.

Em termos de execução do algoritmo, isto também significa que se está procurando soluções em um espaço de busca demasiadamente grande. Como descrito na seção 4.2.4, o espaço de busca explorado seria, tipicamente, da ordem de  $10^{120}$  pontos. Para efeitos de comparação, o número estimado de átomos no universo está entre  $10^{72}$  e  $10^{87}$  (WIKIPEDIA, 2006). É interessante, portanto, buscar outra representação para o problema, a fim de limitar o espaço de busca, e por consequência melhorar a eficiência de exploração.

##### 4.2.5.1 Uma Nova Interpretação para o Problema

Seja  $E_s = \{e_1, e_2, \dots, e_{n_e}\}$  o conjunto de eventos do sistema.  $E_s$  é um conjunto finito, não vazio, portanto pode ser considerado um alfabeto composto de  $n_e$  símbolos. Uma palavra sobre o alfabeto  $E_s$  é uma tupla  $x = \langle e_{i1}, e_{i2}, \dots, e_{iL} \rangle$  de símbolos pertencentes a  $E_s$  - uma palavra de eventos. Este raciocínio pode ser estendido para o conjunto de alarmes do sistema  $A_s = \{a_1, a_2, \dots, a_{n_a}\}$ . Desta forma,  $A_s$  pode ser visto como um alfabeto composto de  $n_a$  símbolos, e qualquer tupla  $y = \langle a_{i1}, a_{i2}, \dots, a_{iK} \rangle$  constitui uma palavra sobre este alfabeto - uma palavra de alarmes.

A cada símbolo  $e_i \in E_s$  está relacionada uma palavra de alarmes característica, sobre o alfa-

betto  $A_s$ ,  $y_i = \langle a_{i1}, a_{i2}, \dots, a_{in} \rangle$ . Dada uma palavra  $x_i$  de eventos sobre  $E_s$ , é possível encontrar uma palavra  $y_m$  sobre  $A_s$ , correlacionada à palavra  $x_i$ , de forma que  $y_m$  é a concatenação das palavras  $y_i$ :

$$y_m = y_1 y_2 \dots y_n$$

É conveniente que cada símbolo de  $E_s$  seja representado por um número inteiro maior que zero, de forma que  $e_1 = 1, e_2 = 2, \dots, e_{n_e} = n_e$ , ou seja, o símbolo que representa o evento é numericamente igual ao índice deste evento. Símbolos que assumem um valor  $i \in \{1, 2, \dots, n_e\}$  indicam a ocorrência do evento  $e_i$ . Também é conveniente criar o símbolo “0” e inserí-lo no alfabeto de eventos, de forma a obter  $E'_s = E_s \cup \{0\}$ . O símbolo “0”, ou “evento nulo”, está relacionado à palavra vazia sobre  $A_s$ . O tamanho do alfabeto  $E'_s$  (número de diferentes símbolos) é, portanto,  $\alpha = n_e + 1$ .

O problema do processamento de alarmes pode então ser descrito como: “Encontrar todas as palavras  $x_i$ , sobre o alfabeto  $E'_s$  que minimizam a equação 4.7”:

$$f(x_i) = |\Delta A| \quad (4.7)$$

onde  $\Delta A$  é um vetor de dimensões  $n_a$  obtido a partir de dois outros vetores:

$$\Delta A = V_{Ar} - V_{Am}(x_i)$$

$V_{Ar}$  é o vetor de alarmes reportados ao sistema<sup>3</sup>. O elemento da  $i$ -ésima posição de  $V_{Ar}$  representa o número de vezes que o alarme “ $i$ ” foi reportado ao sistema (zero caso o alarme não tenha sido reportado). O operador  $||$  é definido segundo a equação 4.3

$V_{Am}(x_i)$  é o vetor de alarmes simulado no caso da ocorrência da palavra de eventos  $x_i$ . O vetor  $V_{Am}$  pode ser obtido a partir da palavra de alarmes  $y_m$  da seguinte forma:

”Para cada símbolo  $a_i$  do alfabeto  $A_s$ , verifica-se quantas vezes  $a_i$  aparece em  $y_m$  e atribui-se este valor a  $V_{Am}[i]$ . Se o símbolo de índice “ $i$ ” não aparece em nenhuma palavra do conjunto  $y_m$ , o  $i$ -ésimo elemento do vetor recebe o valor zero”.

#### 4.2.5.2 Considerações Sobre o Modelo

O critério previamente proposto aparentemente não traz nenhum ganho. Em uma consideração inicial, poder-se-ia argumentar que o problema tornou-se ainda mais complexo, pois o conjunto de todas as palavras sobre o alfabeto  $E'_s$ , denominado  $E'_s*$ , é infinito, logo o espaço de busca tornar-se-ia infinito.

<sup>3</sup>Para o cálculo do  $\Delta A$  é conveniente que as palavras de alarme sejam escritas na forma vetorial.

Porém, assumindo que, para um determinado intervalo de tempo, não ocorrem mais do que  $n_L$  eventos no sistema elétrico, pode-se restringir a busca a uma linguagem  $L$ , composta por todas as palavras de comprimento menor ou igual a  $n_L$ , que é um subconjunto finito de  $E'_s$ . Desta forma, o critério para a solução do problema seria: “Encontrar todas as palavras  $x_i$  sobre o alfabeto  $E'_s$ , de comprimento  $n \leq n_L$ , que minimizam a equação 4.7”.

#### 4.2.5.3 Solucionando o Problema com AG Convencional

Surge um inconveniente ao realizar-se a representação cromossômica da palavra de eventos  $x_i$ , visto que a mesma pode assumir qualquer comprimento no intervalo  $\{1, 2, \dots, n_L\}$ . Em um AG convencional, o cromossomo utilizado possui comprimento fixo. Assim sendo, é interessante restringir ainda mais o critério, de forma a tornar-se: “Encontrar todas as palavras  $x_i$  sobre o alfabeto  $E'_s$ , de comprimento  $n = n_L$ , que minimizam a equação 4.7”.

Caso ocorra no sistema elétrico, durante o intervalo de tempo em questão, um número de eventos  $n$ , menor do que  $n_L$ , esta condição pode ainda ser representada por uma palavra  $x$  de comprimento  $n_L$  sobre  $E'_s$ . Neste caso, a palavra  $x$  contém os  $n$  símbolos correspondentes aos eventos ocorridos e  $n_L - n$  símbolos iguais a 0 (evento nulo).

A palavra de eventos  $x$  pode agora ser codificada em uma representação cromossômica e utilizada no AG convencional. A função de *fitness* normalizada no intervalo  $[0, 1]$  assume a mesma forma da equação 4.5. A constante  $W$  pode ser calculada segundo a equação 4.4.

O tamanho do espaço de busca no universo de possíveis soluções fica determinado pela equação 4.8:

$$S_b = (n_e + 1)^{n_L} \quad (4.8)$$

Para efeitos de comparação, assumindo  $n_e = 200$  e  $n_L = 10$ , resultaria em um espaço de busca com  $201^{10} \simeq 1,08 \times 10^{23}$  pontos, que é consideravelmente menor que a alternativa anterior ( $4^{200} \simeq 2,58 \times 10^{120}$ ).

#### 4.2.5.4 Inconvenientes do Modelo

Ao codificar este modelo em um AG convencional, criam-se artificialmente múltiplos máximos na função objetivo. Assumindo, por exemplo, que em dado momento tenham ocorrido no sistema elétrico os eventos  $\langle e_1, e_2, e_3 \rangle$ , para os quais foi reportada a correspondente palavra de alarmes. Nestas condições, estaria-se procurando entre todas as palavras de comprimento  $n_L$  aquelas que minimizam a equação 4.7. Utilizando-se  $n_L = 5$ , por exemplo, as palavras  $\langle e_1, e_2, e_3, 0, 0 \rangle$ ,  $\langle 0, 0, e_1, e_2, e_3 \rangle$ ,  $\langle 0, e_1, e_2, e_3, 0 \rangle$ , dentre outras, seriam soluções ótimas.

#### 4.2.5.5 Exemplo Numérico

Considerando o mesmo sistema elétrico fictício utilizado em 4.2.4.1, teríamos as relações na forma:

$$e_1 \rightarrow \langle a_4, a_1, a_2 \rangle \quad e_2 \rightarrow \langle a_4, a_1, a_2, a_4, a_1, a_3 \rangle \quad e_3 \rightarrow \langle a_5 \rangle$$

Embora a informação sobre o instante da ocorrência dos alarmes ainda não esteja sendo utilizada, é interessante perceber que esta notação permite que se expresse a palavra de alarmes na ordem desejada. Esta característica poderá permitir utilizar tal informação (quando disponível) para a emissão de diagnósticos com informação cronológica.

Supondo agora que ocorreram os alarmes enunciados na Figura 17, resultando na palavra de alarmes reportada:

$$y_r = \langle a_5, a_4, a_1, a_2 \rangle$$

Mapeando a palavra de eventos procurada em um vetor de dimensão fixa  $n_L = 3$  (para que possa ser empregado o AG), por exemplo, sendo que o tamanho do alfabeto de eventos é  $\alpha = 3 + 1 = 4$  (três possíveis eventos diferentes, mais o evento nulo), o tamanho do espaço de busca torna-se:

$$S_b = 4^3 = 64$$

Desta vez, porém, possíveis soluções ótimas para o problema seriam:  $E_1 = \langle e_1, e_3, 0 \rangle$ ,  $E_2 = \langle 0, e_3, e_1 \rangle$  e  $E_3 = \langle e_1, 0, e_3 \rangle$ , entre outras.

#### 4.2.6 Algoritmo Genético com Cromossomo de Tamanho Variável

Para contornar o inconveniente exposto anteriormente, uma nova categoria de algoritmos genéticos foi desenvolvida. Esta categoria consiste em uma variação inspirada no AG *Messy*, porém fortemente dedicada à solução do problema do processamento de alarmes. As principais diferenças em relação ao algoritmo original são:

- Foi eliminado o conceito de superespecificação e subespecificação. Como não se sabe de antemão quantos eventos ocorreram, não se pode afirmar se um cromossomo está com o tamanho ideal, se está superespecificado ou subespecificado. O tamanho da palavra de eventos é claramente uma variável do problema, e será determinada pelo processo de otimização.

- Foi eliminado o mapeamento em um espaço de dimensão fixa, pelos mesmos motivos expostos anteriormente.
- Pelos mesmos motivos, foi dispensado o conceito de busca local.
- A representação do gene na forma de par ordenado foi dispensada. O conceito de posição absoluta no problema do processamento de alarmes não existe, dispensando este artifício.

**Laço Externo:** A iteração sobre o laço externo empregada no AGm degenera-se neste algoritmo em apenas uma passagem, consistindo na geração de todas as classes de equivalência de ordem 1. Estas classes de equivalência resumem-se a eventos singulares, aqui denominados “blocos construtivos básicos”.

**Blocos construtivos:** A unidade mínima representativa de um bloco construtivo nesta categoria de AG é o evento, representado exatamente em um gene. Um conjunto de blocos construtivos mínimos é mantido permanentemente durante a execução do algoritmo, constituindo uma população auxiliar, atuando como um complemento à população principal do AG.

A execução do algoritmo consiste nos passos ilustrados na Figura 18, e descritos a seguir:

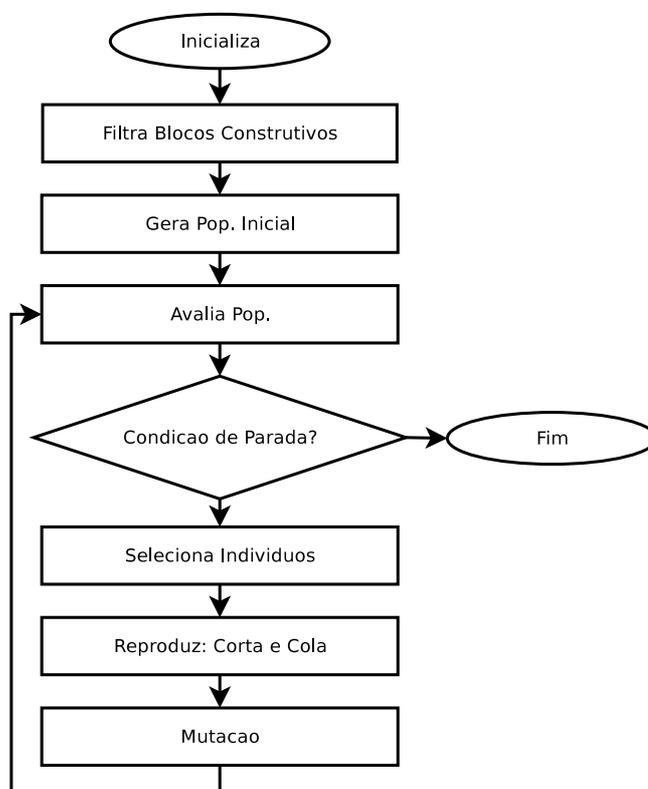


Figura 18: Fluxograma do AG com cromossomo variável.

1. Inicialização: Esta etapa consiste na inicialização básica do programa, onde realizam-se operações como leitura de configuração e instanciação de objetos. Do processo de leitura da configuração, deriva-se a geração das classes de equivalência de ordem 1.
2. Filtragem dos blocos construtivos: Nesta fase, de todos os eventos do sistema, apenas aqueles que pontuam segundo o critério de uma função-filtro são separados e mantidos em um conjunto chamado “conjunto de blocos construtivos básicos”. Esta função-filtro consiste em verificar se um dado evento está correlacionado a algum dos alarmes reportados ao sistema. Caso um dado evento não esteja correlacionado à nenhum dos alarmes reportados, este pode ser descartado, o que resultará numa diminuição do espaço de busca. Esta fase é semelhante à “fase primordial” do AGm.
3. Geração da população inicial: Nesta etapa, uma população aleatória de indivíduos é gerada. Cada indivíduo é composto da concatenação de blocos construtivos básicos, tomados aleatoriamente do conjunto filtrado, obtido na fase anterior. O comprimento do cromossomo de cada indivíduo é aleatório, porém está limitado ao intervalo  $[2, X]$ , onde  $X$  é um parâmetro determinado pelo usuário, e que pode ser estimado em função do número de alarmes presentes no conjunto reportado. O limite inferior do intervalo (2) é desta forma fixado por já estarem representadas todas os possíveis soluções candidatas de tamanho igual a um no conjunto de blocos construtivos básicos.
4. Elitismo: O melhor indivíduo da geração anterior é copiado para a nova população.
5. Geração do *pool* para o processo de seleção: Um *pool* de indivíduos é gerado, composto da população atual unida ao conjunto de blocos construtivos.
6. Seleção: Pares de indivíduos do *pool* são selecionados segundo o critério do torneio estocástico.
7. Reprodução: Os pares de indivíduos selecionados são submetidos ao operador “cortar e colar” (descrito na seção 2.1.7.2), responsável por gerar os indivíduos que deverão compor a nova população. Este operador deve ser aplicado com alta probabilidade (nesta implementação, ele sempre é aplicado).
8. Mutação: Cada novo indivíduo, é submetido ao operador de mutação. Este operador consiste em substituir o valor de determinado gene por um dos eventos que compõem o conjunto de blocos construtivos. A probabilidade de aplicação deste operador é baixa.
9. Critério de parada: Se a adequabilidade do melhor indivíduo atingir o valor 1, ou o máximo número de gerações for atingido, o processamento é encerrado. Vale lembrar que

este algoritmo tende a convergir em um número menor de gerações que o AG convencional, pois o processo de filtragem de blocos construtivos é executado previamente.

#### 4.2.6.1 Resolvendo uma Instância do Problema

Retomando o exemplo apresentado em 4.2.4.1, a formulação seria idêntica àquela da seção 4.2.5.5, com a exceção de que o tamanho do espaço de busca não poderia mais ser determinado, pois não se fixou o tamanho da palavra. As possíveis soluções ótimas seriam reduzidas a:  $E_1 = \langle e_1, e_2 \rangle$  e  $E_2 = \langle e_2, e_1 \rangle$ , pois foi eliminado o evento nulo.

### 4.2.7 Modelo Melhorado 3: Palavra de Eventos com SOE

Outra vantagem do modelo baseado em palavra de eventos é a possibilidade de incorporação de informação temporal, a respeito da sequência em que os eventos ocorreram. Esta informação pode ser útil em determinadas situações, onde conclusões baseadas na ordem de ocorrência dos alarmes e eventos é necessária. Neste trabalho, porém, estas situações não foram abordadas. Embora um protótipo tenha sido implementado e alguns testes preliminares realizados, nenhum caso de teste mais elaborado foi realizado. A apresentação do modelo é feita conceitualmente, ficando seu teste e validação para uma etapa futura.

A palavra de alarmes produzida por uma determinada palavra de eventos não é necessariamente a concatenação das palavras de alarmes correspondentes a cada símbolo da palavra de eventos. Como exemplo, considere-se novamente o sistema apresentado na seção 4.2.4.1. Supondo-se que este sistema disponha de informações SOE, dois cenários distintos serão supostos:

1. O evento  $e_1$  ocorre, produzindo os respectivos alarmes, um determinado intervalo de tempo se passa e somente então o evento  $e_3$  ocorre. Neste caso, a palavra de alarmes reportada ao sistema é exatamente igual à concatenação das palavras características de  $e_1$  e  $e_3$ , respectivamente, segundo ilustrado na Figura 19.

```

XYZ 13 RELE 50/51 F/N DESLIG
XYZ 13 DJ 52- 01 ABERTO
XYZ 13 DJ 52- 01 FECHADO
XYZ 13 SOBRETENSAO

```

Figura 19: Alarmes reportados para os eventos  $e_1$  e  $e_3$ .

2. O evento  $e_1$  está em processo, e simultaneamente o evento  $e_3$  ocorre. Neste caso, uma possível palavra de alarmes reportada ao sistema seria conforme a Figura 20 <sup>4</sup>.

```

XYZ 13 RELE 50/51 F/N DESLIG
XYZ 13 SOBRETENSAO
XYZ 13 DJ 52- 01 ABERTO
XYZ 13 DJ 52- 01 FECHADO

```

Figura 20: Alarmes reportados para os eventos  $e_1$  e  $e_3$ , ocorrência simultânea.

O segundo cenário acima exposto evidencia que a emissão de diagnósticos baseados em informações temporais requer um tratamento mais elaborado. Como visto, a sequência de alarmes esperada para o evento  $e_1$  seria  $A_m(e_1) = \langle a_4, a_1, a_2 \rangle$ , porém a sequência reportada é  $A_r = \langle a_4, a_5, a_1, a_2 \rangle$ .

Por outro lado, pode-se reescrever a sequência esperada na forma :

$$A_m(e_1) = \langle *, a_4, *, a_1, *, a_2, * \rangle \quad (4.9)$$

onde o caracter (\*) representa qualquer palavra sobre o alfabeto  $A_s$ . A relação 4.9 pode ser facilmente incorporada ao AG na forma de uma restrição adicional. Neste caso, dada uma solução-candidata, calcular-se-ia  $|\Delta A|$  segundo a equação 4.3 e a seguir aplicar-se-ia esta restrição, penalizando a adequabilidade a cada violação encontrada.

### 4.3 Restrição do Espaço de Busca

Considere-se novamente o exemplo apresentado na Tabela 3. Colocando agora a relação entre eventos e alarmes em uma forma tabular, conforme a Tabela 4, e considerando que sejam reportados os alarmes  $\{a_4, a_1, a_2\}$ . Em um primeiro momento, buscar-se-ia uma solução na forma de uma palavra de eventos escrita como combinação dos eventos  $e_1, e_2$  e  $e_3$ . É fácil perceber, porém, que o evento  $e_3$  não possui nenhum de seus alarmes correlacionados integrando a lista de alarmes reportada, e portanto não é preciso considerá-lo como variável do problema. Para chegar a esta conclusão, basta que se compare a lista de alarmes reportada com as palavras características, na segunda coluna da Tabela 4. Caso a lista reportada possua algum alarme em comum com uma determinada palavra característica, o evento correspondente é selecionado

<sup>4</sup>Este exemplo é meramente ilustrativo, e não faria sentido em uma situação real, visto que se está sinalizando um desligamento do alimentador, e ao mesmo tempo uma sobretensão. No caso de um desligamento, o esperado seria que o valor da tensão no alimentador se anulasse.

para compor eventuais soluções-candidatas. Caso não exista nenhum alarme em comum, o respectivo evento não é considerado na composição das soluções-candidatas, reduzindo assim a dimensão do espaço de busca

Tabela 4: Relação eventos/alarmes na forma tabular.

Evento	Alarmes Característicos
$e_1$	$a_4, a_1, a_2$
$e_2$	$a_4, a_1, a_2, a_4, a_1, a_3$
$e_3$	$a_5$

É possível realizar este pré-processamento em tempo polinomial, reduzindo consideravelmente o tamanho do espaço de busca. Nos testes realizados, de 235 eventos contidos na base original, reduziu-se, em média, para 52 após o pré-processamento.

## **5    *Implementação do Sistema***

Neste capítulo é apresentada a implementação de um protótipo do processador de alarmes. Este protótipo é posicionado em relação à infra-estrutura de supervisão e controle do sistema elétrico na COPEL.

### **5.1    Materiais e Infra-estrutura**

Para o desenvolvimento deste trabalho, foi utilizado como caso de estudo o SASE - Sistema de Automação de Subestações da COPEL - Companhia Paranaense de Energia. Este sistema é empregado pela COPEL no controle e supervisão de subestações e redes de transmissão e distribuição de energia elétrica.

Baseado nas características do SASE, foi proposta uma ferramenta, utilizando Algoritmos Genéticos, para processamento de alarmes em centros de operação de subestações de transmissão e distribuição de energia elétrica. Isto, porém, não limita a aplicação destes conceitos apenas ao sistema SASE. O método proposto neste trabalho é genérico, e pode ser particularizado para execução associada a qualquer ambiente de supervisão de plantas industriais onde o problema de correlação de eventos e alarmes esteja presente.

#### **5.1.1    Hierarquia de Controle e Supervisão do Sistema Elétrico**

A coordenação, supervisão e controle da operação do sistema elétrico da COPEL são atribuições do Centro de Operação do Sistema (COS). O COS desempenha ações de supervisão e controle em usinas e subestações de tensão maior ou igual a 69 kV, agindo numa hierarquia operacional com os outros órgãos da empresa e do Sistema Interligado Nacional (COPEL, 2006).

Subordinados diretamente ao COS, estão os Centros de Operação de Estações (COEs). Estes centros são responsáveis pelo controle e supervisão, em nível regional, de subestações e linhas de transmissão. Os COEs são de responsabilidade da área de Transmissão. Na área de

Distribuição, no mesmo nível hierárquico dos COEs, estão os CODs (Centros de Operação da Distribuição). Estes centros são responsáveis pelo controle e supervisão de subestações e redes de distribuição.

A Figura 21 ilustra esquematicamente a organização hierárquica do modelo de supervisão e controle adotado atualmente na COPEL<sup>1</sup>. A orientação das setas representa o sentido da ação de comando executada por determinado centro de operação sobre o nível subordinado<sup>2</sup>. Nos COEs e CODs, bem como na maioria das subestações comandadas por estes, é empregado o sistema SASE.

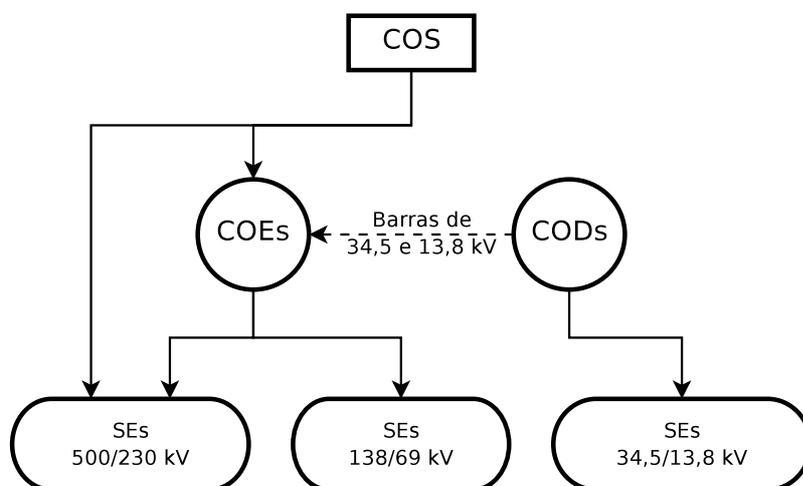


Figura 21: Hierarquia de Controle e Supervisão do Sistema Elétrico na COPEL.

### 5.1.2 Sistema de Automação de Subestações da COPEL

O Sistema de Automação de Subestações da Copel é um sistema de supervisão e controle em tempo real com arquitetura distribuída (MARQUEZ et al., 2003). O SASE é empregado em subestações de transmissão, subestações/redes de distribuição e centros regionais de controle, tanto de distribuição quanto de transmissão.

No nível da subestação, o SASE é composto por unidades de aquisição e controle (UAC) e por unidades centrais de controle (UCC), que se comunicam com as UACs através de fibras ópticas. As UCCs são responsáveis pela concentração dos dados, comunicação com os níveis superiores (Centros de Operação) e pela execução de processos inteligentes, denominados Funções Elétricas Automáticas (MARQUEZ et al., 2003).

<sup>1</sup>Este modelo encontra-se em fase de revisão.

<sup>2</sup>As SEs de tensão igual ou superior à 69 kV geralmente possuem alimentadores de 13,8 e/ou 34 kV (SEs mistas), os quais são comandados pelos CODs via COE.

No nível dos centros regionais, o SASE atua como elemento centralizador, estando conectado via canal de comunicação a cada uma das subestações subordinadas àquele centro. As principais funções desempenhadas pelo SASE nos centros regionais são a aquisição de dados e supervisão e controle à distância de todas as subestações desassistidas (que não possuem operador local) a ele ligadas (FRISCH; CARDOSO; ARRUDA, 1997).

Uma subestação é composta por uma série de equipamentos, dos quais são coletadas diversas informações, tais como:

- Medidas analógicas, representando valores de grandezas do sistema elétrico, tais como corrente, tensão, potência ativa, potência reativa, entre outras.
- Informações de estado, representadas por variáveis binárias. Exemplos destas informações são estados de chaves e disjuntores (aberto ou fechado) e estado de pontos relacionados à equipamentos de proteção (atuado ou não atuado).

Um centro de operação é responsável pela supervisão de várias subestações simultaneamente. Em determinadas situações de emergência, esse centro pode concentrar um número bastante grande de informações em curtos intervalos de tempo. Este volume de informações deve ser analisado constantemente pelo operador do centro regional para que as providências necessárias ao perfeito funcionamento do sistema elétrico sejam tomadas (FRISCH; CARDOSO; ARRUDA, 1997).

#### **5.1.2.1 O Conceito de Alarme no SASE**

No sistema SASE, a unidade básica de representação de informações do sistema elétrico é o “ponto de supervisão”, geralmente referido apenas como “ponto”. Os pontos supervisionados podem ser classificados em duas categorias básicas: pontos analógicos e pontos de estado (também chamados pontos digitais) (COPEL, 2003).

Pontos analógicos podem possuir os seguintes atributos: limite superior e limite inferior. Quando um destes limites é violado (valor da medida associada ao ponto extrapola o limite superior, ou cai abaixo do limite inferior), um alarme é gerado.

Pontos digitais podem possuir os seguintes atributos: Normalmente Aberto (NA), Normalmente Fechado (NF), Ponto Gera Alarme (PGA). A geração de alarmes para pontos de estado ocorre nas seguintes situações:

- Um ponto de estado que possui o atributo PGA é inserido na página de alarmes do sistema

quando seu estado muda para o estado que gera alarme, de acordo com o atributo NA/NF. Um exemplo da aplicação do atributo PGA é um ponto que monitora o estado de um relé de proteção contra sobrecorrente. Seu estado normal é aberto (NA), indicando que a corrente no circuito está dentro dos limites operacionais estabelecidos, quando obviamente o alarme correspondente não é gerado. Caso ocorra neste ponto uma transição de estado de aberto para fechado, um alarme deve ser exibido, alertando o operador da situação anormal, ou seja, a corrente no circuito ultrapassou o limite operacional estabelecido, provocando a atuação da proteção associada.

- Um ponto de estado que não possui o atributo PGA gera alarme sempre que houver uma transição de estado. Um bom exemplo desta categoria de pontos é um disjuntor. Toda vez que uma transição de estado ocorrer neste equipamento (desde que não tenha sido via comando do sistema), um alarme é gerado. Uma transição deste tipo geralmente é provocada pela proteção ou por equipamentos de religamento automático.

### 5.1.3 A Subestação Escolhida

Como caso de estudo para este trabalho, foi escolhida a SE Batel, localizada na cidade de Curitiba controlada pelo COE Uberaba. Esta SE possui interligações com duas subestações adjacentes, bem como doze alimentadores de redes de distribuição. Um diagrama unifilar simplificado da SE Batel é apresentado na Figura 22. Outras características relevantes da SE Batel são:

- Base de medidas analógicas composta por aproximadamente 600 pontos.
- Base de pontos de estado composta por aproximadamente 850 pontos.

A escolha desta SE em particular deve-se principalmente às suas características: porte médio em relação ao número de pontos de medida, recebendo linhas de 69kV e também possuindo alimentadores de 13,8kV.

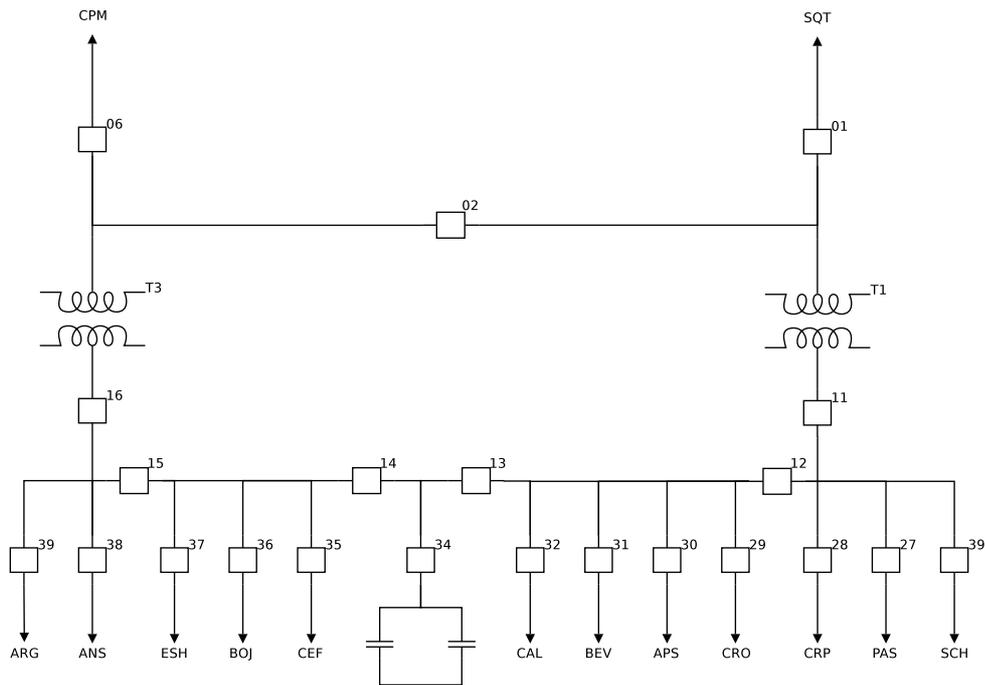


Figura 22: Diagrama unifilar da SE Batel.

## 5.2 Construção da Base de Conhecimento

Segundo visto na seção 3.6.5.1, cada evento pode ser relacionado a um conjunto característico de alarmes, de acordo com a equação 3.1. A formulação das regras básicas de correlação entre cada tipo de evento e seu conjunto característico de alarmes foi realizada por profissionais da equipe de Proteção da COPEL. Estas regras foram escritas na forma de uma tabela, semelhante à Tabela 5, onde é exemplificada a relação entre um evento e seu respectivo conjunto característico de alarmes.

Tabela 5: Correlação entre eventos e alarmes.

Evento		Alarmes			
Onde	O que	Mensagem	PTO	A/F	A/D
		SQT 69 RELE 94	05	1	1
Linha Transmissão BTE-SQT	Religamento por Trip de Distância	DJ01 69 DJ 52-01	13	0	1
		SQT 69 RELE 79	21	1	1
		DJ01 69 DJ 52-01	13	1	1

Nesta tabela, a coluna “PTO” identifica o índice do ponto no banco de dados do SASE, a coluna “A/F” define o estado do ponto na condição alarmada (Aberto ou Fechado) e a coluna “A/D” indica se o ponto é analógico ou de estado. Estas informações são suficientes para enumerar todos os possíveis alarmes gerados pelos pontos do banco, tanto de estado quanto analógicos. Neste trabalho, porém, não foram utilizadas informações de alarmes gerados por pontos analógicos, embora o sistema suporte este tipo de informação. Caso necessária, a incorporação de eventos envolvendo alarmes de pontos analógicos no sistema consiste apenas na inserção das respectivas regras na base.

A base de conhecimento, após particularizada para todos os eventos do sistema, foi representada em um arquivo no formato XML. Este arquivo pode ser empregado como entrada (na forma de arquivo de configuração) para o sistema de processamento de alarmes. Um trecho do arquivo XML que descreve a regra expressa na Tabela 5 foi transcrito na Figura 23. As vantagens de se utilizar o formato XML para esta finalidade são, entre outras:

- É facilmente processado por um programa de computador;
- É portátil e humanamente legível;
- É extensível;
- Pode ser validado segundo regras de formação;

```
<ocorrencia>
  <nome>Linha BTE-SQT: Religamento por trip de distancia.</nome>
  <alarme>
    <numpto>5</numpto>    <flagad>1</flagad>    <status>1</status>
  </alarme>
  <alarme>
    <numpto>13</numpto>    <flagad>1</flagad>    <status>0</status>
  </alarme>
  <alarme>
    <numpto>21</numpto>    <flagad>1</flagad>    <status>1</status>
  </alarme>
  <alarme>
    <numpto>13</numpto>    <flagad>1</flagad>    <status>1</status>
  </alarme>
</ocorrencia>
```

Figura 23: Regra no formato XML.

## 5.3 Ferramentas Computacionais Empregadas

No desenvolvimento deste trabalho, os principais pacotes de *software* utilizados foram:

- GALOPPS: *The Genetic ALgorithm Optimized for Portability and Parallelism System*. O GALOPPS é uma implementação genérica de algoritmos genéticos, escrita em linguagem “C” (GOODMAN, 1996).
- GCC/G++: *The Gnu Compiler Collection*, compilador do projeto GNU para linguagens “C” e “C++”, entre outras (Free Software Foundation, 2006).
- TAO: *The Ace Orb* (SCHMIDT, 2006), uma implementação aberta da especificação CORBA da OMG.
- QT: uma biblioteca para desenvolvimento de interfaces gráficas (TROLLTECH, 2006).

## 5.4 Construção do Protótipo

Para a implementação de alguns modelos de AG, foi utilizado o pacote GALOPPS. Este pacote dispõe de diversas funcionalidades, como diferentes operadores de cruzamento e mutação, vários métodos de seleção e possibilidade de evolução de populações em paralelo. Para utilizar o GALOPPS, basta que se escreva a função de *fitness* em linguagem “C”. O código-fonte do GALOPPS é aberto, e está disponível sob a licença GPL (Free Software Foundation, 1991).

Através do GALOPPS, foram implementados os modelos de processamento com “vetor de eventos inteiros” (seção 4.2.4), “palavra de eventos sobre o alfabeto  $E_s$ ” (seção 4.2.5) e “Palavra de Eventos sobre o Alfabeto  $E_s$  com SOE” (seção 4.2.7). Também foi implementado um algoritmo de múltiplas populações, que pode ser empregado em qualquer um destes modelos.

A implementação do AG com cromossomo de tamanho variável (seção 4.2.6) foi feita em linguagem C++, utilizando os *containers* da Biblioteca Padrão STL (*Standard Template Library*). No Apêndice 1 é apresentado um diagrama de sequência do algoritmo. O correspondente diagrama de classes é apresentado no Apêndice 2.

### 5.4.1 Integração com o Sistema SASE

Para a aquisição de dados (fila de alarmes) diretamente do sistema SASE, foi adotada uma arquitetura do tipo “anexada” (TESCH et al., 1990). No computador onde é executado o sistema

SASE, foi instalado um pequeno programa servidor, responsável por consultar a fila de alarmes atual e disponibilizar o conteúdo da mesma em um objeto servente CORBA. Em outro computador, conectado ao primeiro através de uma rede *ethernet*, é executado o programa processador de alarmes. As atribuições deste programa são:

- Consultar dados da fila de alarmes através da invocação de métodos deste objeto servente.
- Carga do arquivo XML, contendo as regras de correlação entre eventos e alarmes.
- Invocação do processo responsável pela execução do AG.
- Gerenciamento da interface com o operador.

Na Figura 24 é ilustrada esquematicamente a arquitetura adotada para o sistema. A interface com o operador é mostrada na Figura 25. Através desta interface é possível optar por qualquer uma das variações do AG implementadas.

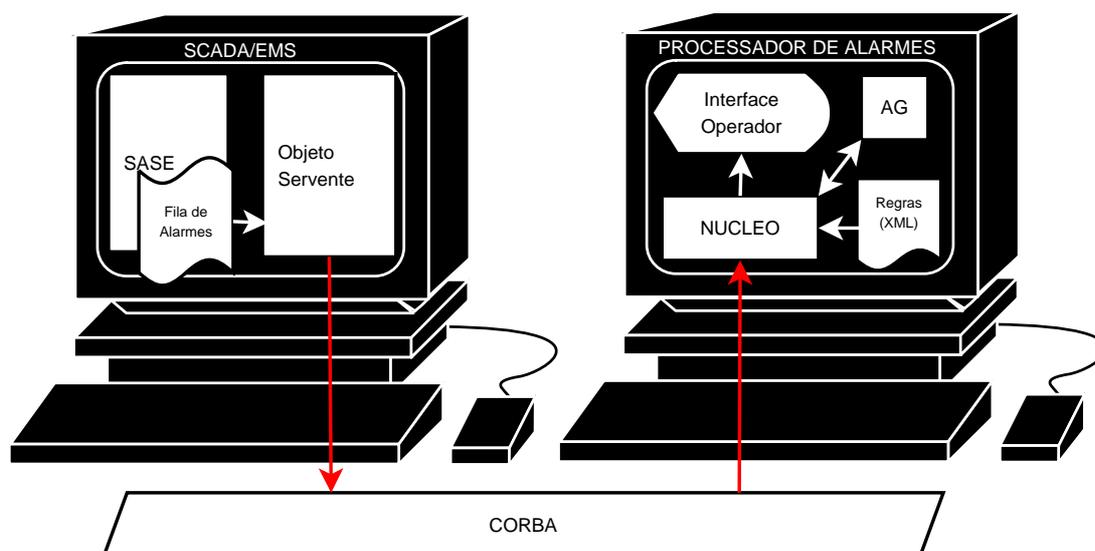


Figura 24: Arquitetura do sistema.

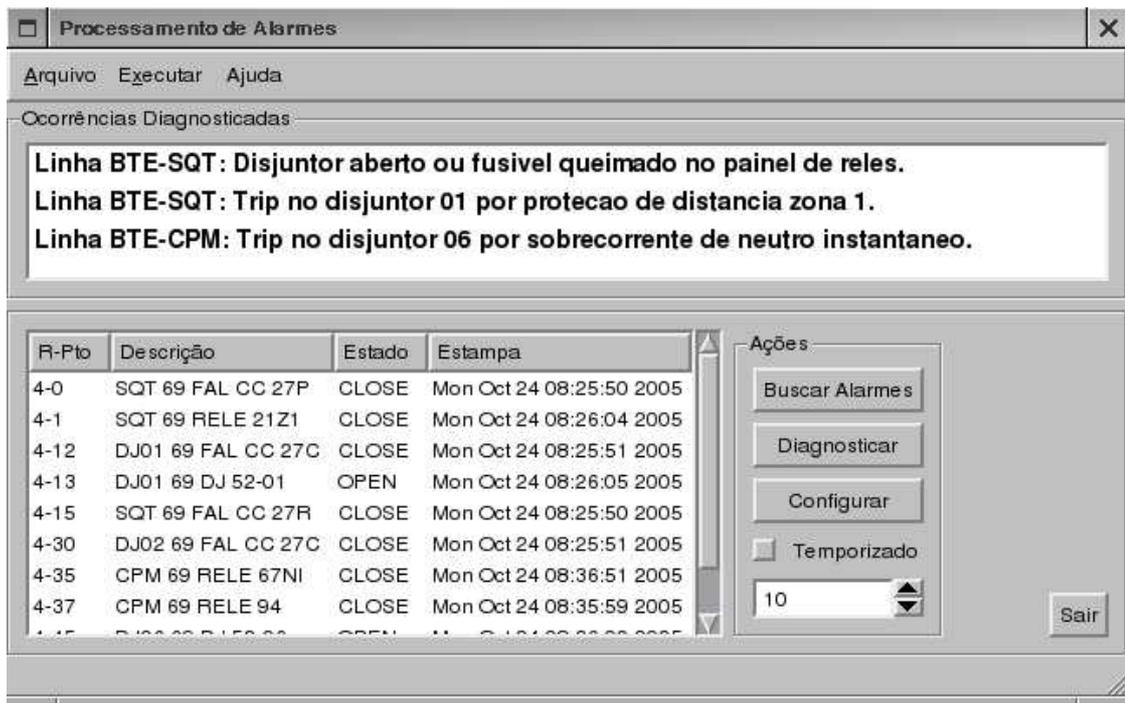


Figura 25: Interface com o operador.

### 5.4.2 Disparo do Algoritmo

Neste protótipo, o disparo do processo de diagnóstico pode ser realizado de duas maneiras:

1. Manualmente pelo operador, através dos botões “Buscar Alarmes” e “Diagnosticar” da interface (Figura 25).
2. Automaticamente, a cada intervalo de tempo configurado.

## 5.5 Aquisição de Dados

A aquisição de dados para testes e validação do sistema proposto pode ser feita de diversas maneiras:

1. Adquirindo-se dados reais de operação, diretamente do sistema SASE;
2. Utilizando-se dados simulados pelo SASE;
3. Gerando-se, através de um programa dedicado, eventos aleatórios e alimentando-se os alarmes correspondentes à entrada do processador de alarmes.

A aquisição de dados de operação em tempo real tem a inconveniência de depender da ocorrência de eventos no sistema elétrico, para que alarmes sejam gerados. A ocorrência destes eventos, porém, é de natureza estocástica, o que inviabiliza o uso desta abordagem, pois ficaria-se à mercê do imprevisto (eventos podem ocorrer a qualquer momento, assim como podem não ocorrer por horas ou até mesmo dias).

A utilização de dados simulados pelo sistema SASE é possível, e mais interessante do que a primeira abordagem. Como inconveniente, porém, exige mais recursos computacionais, visto que é necessário executar o programa de simulação associado ao SASE, e ainda a estrutura de integração do processador de alarmes. Por este motivo, a abordagem preferida para testes e validação é a terceira, por ser mais simples, rápida e demandar menos recursos.

## 6 *Resultados*

*"There are three kinds of lies:  
lies, damned lies and statistics."  
Mark Twain / Benjamin Disraeli*

Neste capítulo é apresentada a metodologia de testes empregada na avaliação do protótipo construído, bem como os resultados obtidos para os diferentes algoritmos propostos.

### 6.1 **Crítérios de Avaliação**

#### 6.1.1 **Métricas**

##### 6.1.1.1 **Erros e Acertos**

Para a avaliação do desempenho dos algoritmos em relação aos vários parâmetros, as seguintes métricas foram adotadas:

1. Percentual de Acertos por Rodada<sup>1</sup>: Esta quantia mede a razão entre o número de eventos corretamente diagnosticados pelo algoritmo e o número de eventos no conjunto esperado.
2. Percentual de Erros por Rodada: Esta medida é dada pela razão percentual entre a soma de dois fatores de erro e o número de eventos no conjunto esperado. Os fatores de erro em questão são o número de eventos não diagnosticados e o número de eventos espúrios gerados pelo algoritmo.

A medida percentual serve para que se possa comparar o resultado obtido para diferentes números de eventos ocorridos.

---

<sup>1</sup>Uma rodada consiste em uma execução completa do algoritmo para um dado conjunto de parâmetros de entrada e uma dada semente aleatória.

Considere-se o seguinte exemplo: supondo a ocorrência de quatro eventos, expressos pela palavra  $E_o = \langle e_1, e_2, e_3, e_4 \rangle$ . Submetendo-se a correspondente palavra de alarmes reportada ao AG, suponha-se que este tenha produzido como diagnóstico os eventos  $E_d = \langle e_1, e_2, e_4, e_5 \rangle$ .  $E_o$  consiste nos “eventos esperados”, enquanto  $E_d$  consiste nos “eventos diagnosticados”. Da diferença entre  $E_o$  e  $E_d$ , resultam:

- Eventos espúrios: Eventos produzidos como diagnóstico, porém não presentes no conjunto esperado. Neste caso,  $e_5$  é um evento espúrio, pois foi relatado no diagnóstico mas não está presente em  $E_o$ .
- Eventos não-diagnosticados: Eventos presentes no conjunto esperado porém não constantes no diagnóstico. Neste caso,  $e_3$  é um evento não-diagnosticado, pois consta em  $E_o$  e não está relatado em  $E_d$ .
- Acertos: Eventos presentes em ambos os conjuntos. Neste caso,  $\langle e_1, e_2, e_4 \rangle$  são diagnosticados corretamente.

Se fosse desejado comparar estes resultados com um caso onde o número de eventos de entrada fosse igual a dez, por exemplo, certamente não seria correto colocar lado a lado estas grandezas. Por outro lado, uma medida do percentual de acertos ou erros, tomados em relação ao tamanho da palavra de eventos esperada, representa melhor o desempenho do algoritmo. As métricas são, portanto, calculadas como:

- Percentual de Acertos por Rodada:  $\frac{\text{Numero de Acertos}}{\text{Num Eve Esperados}} \times 100 = \frac{3}{4} \times 100 = 75\%$
- Percentual de Erros por Rodada:  $\frac{\text{Nao Diag} + \text{Espurios}}{\text{Num Eve Esperados}} \times 100 = \frac{1+1}{4} \times 100 = 50\%$

Estas duas métricas, embora possam parecer redundantes, representam informações diferentes. Considere-se a Figura 26, onde são representados diagramas de possíveis conjuntos de erros e acertos. O conjunto universo corresponde ao conjunto de eventos do sistema. A área representada na cor cinza, delimitada por uma linha contínua, corresponde ao conjunto de eventos esperado. A área delimitada por uma linha tracejada corresponde aos eventos diagnosticados pelo sistema. A interseção das duas regiões determina o percentual de acertos. A soma das áreas não sobrepostas dos dois conjuntos determina o percentual de erros.

Na Figura 26 (a) a interseção do conjunto esperado com o conjunto diagnosticado é não-vazia, logo o percentual de acertos é maior do que zero. Em ambos os casos (a) e (b), o tamanho

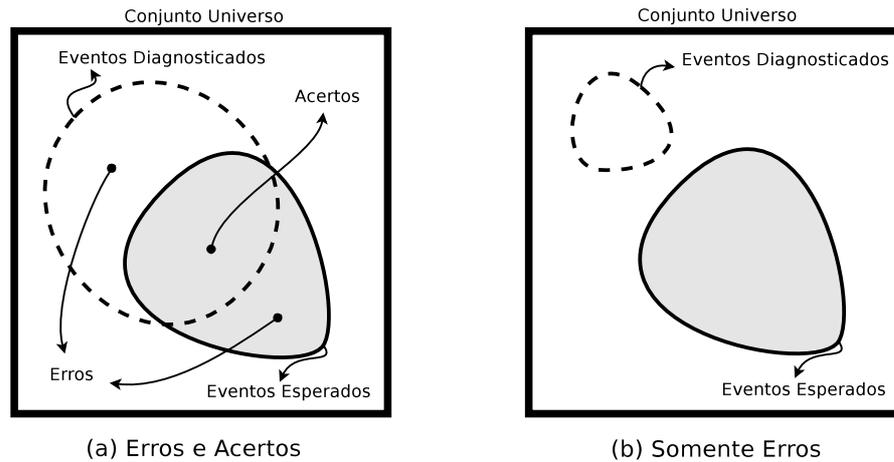


Figura 26: Diagramas dos conjuntos Esperado e Diagnosticado.

da área correspondente aos erros poderia ser semelhante, induzindo uma falsa idéia de equivalência. Em (b), porém, o percentual de acertos é nulo (as áreas são completamente disjuntas), indicando que o sistema produziu uma resposta localizada em uma região do espaço de busca que está distante do esperado. Logo o desempenho de (b) é pior do que (a).

É interessante notar também que o percentual de erro, em determinadas condições, pode ultrapassar 100%. Como exemplo, pode-se observar a Figura 26 (b). O percentual de erro é dado pela razão entre a soma das duas regiões e a região correspondente aos eventos esperados. Neste caso, claramente o percentual de erro é maior do que 100%. Por outro lado, o percentual de acerto nunca poderá ultrapassar 100%, pois é calculado pela razão entre a interseção das duas regiões e a região correspondente aos eventos esperados.

### 6.1.1.2 Capacidade de Síntese

Adicionalmente, foi avaliada a capacidade de síntese do sistema de processamento de alarmes, ou seja, a redução percentual média no número de mensagens apresentadas ao operador. Esta medida é feita tomando-se a razão entre o número de mensagens suprimidas e o número de mensagens no conjunto de alarmes reportado. Sendo  $N_{Ar}$  o número de mensagens reportadas originalmente e  $N_e$  o número de eventos diagnosticados (mensagens apresentadas ao operador), o número de mensagens suprimidas é dado pela diferença entre estas duas quantias ( $N_{Ar} - N_e$ ). A redução percentual do número de mensagens é então dada por:

$$\frac{N_{Ar} - N_e}{N_{Ar}} \times 100 = \left(1 - \frac{N_e}{N_{Ar}}\right) \times 100$$

## 6.1.2 Estatísticas

As métricas expostas anteriormente, embora expressas na forma de percentual, representam em sua essência uma variável discreta. Considere-se novamente o exemplo anterior, onde é suposta a ocorrência de quatro eventos. A variável “Percentual de Acertos por Rodada” pode assumir no máximo cinco valores: {0%, 25%, 50%, 75% e 100%}, caracterizando-se como variável discreta. Por outro lado, a distribuição de probabilidades desta variável é desconhecida, impondo limitações ao uso da estatística paramétrica na análise dos resultados.

### 6.1.2.1 Medida de Tendência

Recorrendo novamente a um exemplo, ao tomar-se o resultado de três rodadas do algoritmo (executadas a partir de diferentes populações iniciais), com supostos percentuais de acerto de {75%, 100% e 100%}, respectivamente, obter-se-ia uma média aritmética de 91,67% de acerto. Este valor, apesar de representar uma tendência, não tem significado prático, pois em uma dada rodada com quatro eventos de entrada, 91,67% de acertos representaria 3,67 eventos - um valor que não pode ser assumido pela variável discreta. Uma segunda alternativa para representar a tendência deste conjunto é a “moda”, ou seja, o valor que se repete com maior frequência no intervalo (neste caso 100%). A principal vantagem da moda é que esta sempre assume um valor válido para a variável. Neste trabalho, tanto a média quanto a moda são apresentadas como medida de tendência de um dado conjunto de experimentos.

### 6.1.2.2 Medida de Dispersão

Uma medida de dispersão frequentemente utilizada na literatura é o desvio padrão em torno da média aritmética, ou simplesmente “desvio padrão”. O desvio padrão mede a “dispersão média em torno da média aritmética” (LEVINE; BERENSON; STEPHAN, 2000), porém seu uso nem sempre faz sentido, e está restrito aos casos onde o conjunto de dados segue uma distribuição paramétrica conhecida (CAMPOS, 2002). Retornando ao exemplo anterior, onde o conjunto assumia os valores {75%, 100% e 100%}, com média 91,67%. Aplicando-se a conhecida fórmula para o desvio padrão (LEVINE; BERENSON; STEPHAN, 2000), obtém-se o valor de 14,43%. Para distribuições paramétricas, sabe-se que determinada porção dos dados concentra-se no intervalo da média  $\pm$  o desvio padrão. Neste exemplo, o valor da média mais um desvio padrão assumiria 106,10% - claramente fora do intervalo válido para a variável em questão. Por estas razões o desvio padrão não poderá ser utilizado como medida de dispersão neste trabalho.

Uma medida de dispersão simples e de fácil aplicação, embora limitada, é a amplitude.

A amplitude é definida como a diferença entre a maior e a menor observação feita em um conjunto de dados (LEVINE; BERENSON; STEPHAN, 2000). As principais desvantagens de se utilizar a amplitude é que esta não leva em conta a maneira como os dados estão efetivamente distribuídos entre os extremos do conjunto (LEVINE; BERENSON; STEPHAN, 2000) e sua vulnerabilidade a *outliers*. Outras medidas de tendência e dispersão, como os “quartis” e a “amplitude interquartil” (LEVINE; BERENSON; STEPHAN, 2000) produzem distorções demasiadamente grandes quando aplicadas a uma variável discreta. Nas medidas de dispersão realizadas neste trabalho será utilizada apenas a amplitude.

## 6.2 Metodologia de Testes

Para a aquisição de dados, validação e comparação de resultados, foi executada uma série de testes automatizados dos algoritmos propostos. Nestes testes, foi avaliado o desempenho de diferentes variações do AG propostas neste trabalho, bem como o uso de diferentes parâmetros. Foi também avaliado o desempenho do sistema na presença de ruído. A automatização dos testes foi feita por meio de programas escritos em linguagem C++ associados a *shell scripts*.

### 6.2.1 Fluxograma de Testes

Um fluxograma simplificado dos testes realizados, e os dados de entrada e saída associados a cada etapa são ilustrados na Figura 27. Uma iteração sobre este fluxograma é realizada para cada conjunto de parâmetros a serem avaliados (fatores de variação). Estes parâmetros são listados na Tabela 6. Os passos do fluxograma de teste são descritos a seguir:

1. Geração de uma palavra de eventos: para cada conjunto de parâmetros, é gerada uma palavra de eventos aleatória, denominada “palavra de eventos esperada”.
2. Geração da palavra de alarmes: à palavra de eventos esperada corresponde uma palavra de alarmes característica, obtida pela concatenação das palavras de alarmes características de cada evento esperado. Esta palavra foi denominada “palavra de alarmes esperada”.
3. Inserção de ruído: de acordo com os parâmetros “Alarmes Espúrios” e “Alarmes Não-reportados”, é subtraído um determinado número de alarmes e/ou somado um determinado número de alarmes falsos à palavra esperada, compondo assim a “palavra de alarmes reportada”.

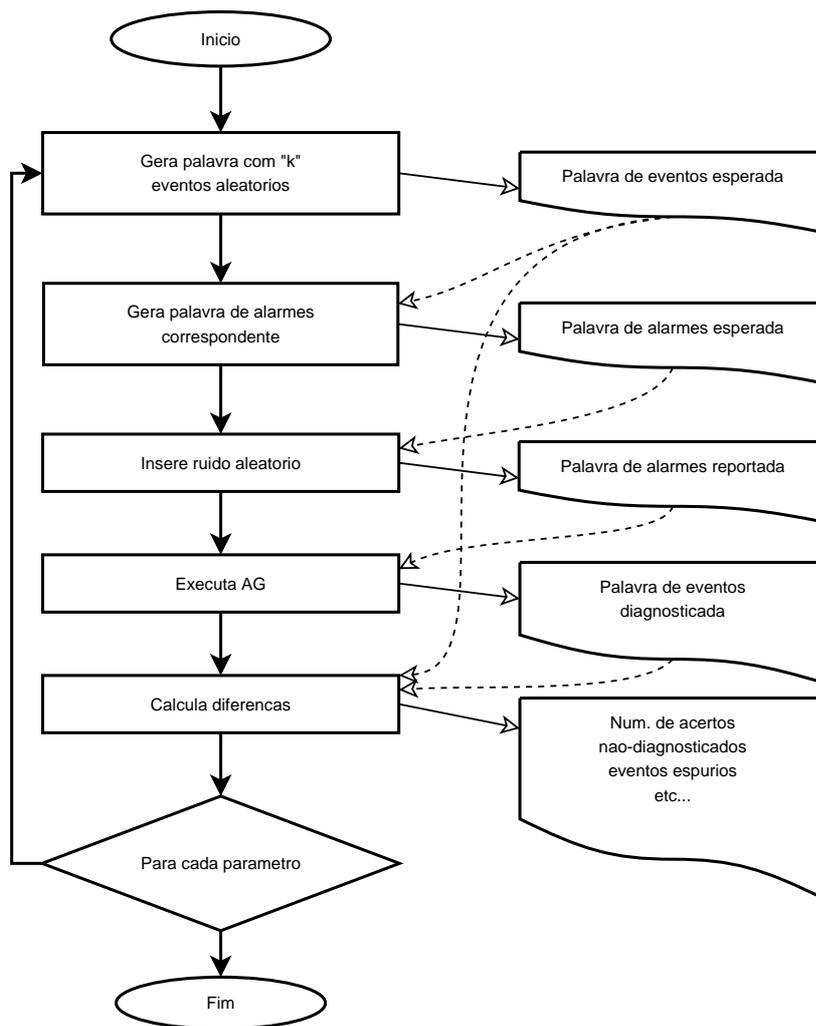


Figura 27: Fluxograma dos testes executados.

4. Execução do AG: nesta etapa, é invocado o processo que implementa o AG correspondente. A execução deste é repetida diversas vezes<sup>2</sup> para a mesma palavra de alarmes reportada, utilizando diferentes sementes aleatórias.
5. Cálculo das diferenças: as diferenças entre o diagnóstico fornecido pelo AG e a palavra de eventos esperada são calculadas. Os resultados desta diferença são dados em termos de “acertos”, “eventos não-diagnosticados” e “eventos espúrios”. Outros dados da execução do AG também são coletados e armazenados em um arquivo.

<sup>2</sup>Na realidade, o algoritmo de testes é um pouco mais sofisticado. Inicialmente são geradas dez diferentes palavras de eventos, e para cada uma delas o AG é executado por dez vezes (com diferentes sementes aleatórias), totalizando cem rodadas.

## 6.2.2 Parâmetros de Avaliação

O conjunto de parâmetros avaliados nos testes é ilustrado na Tabela 6. O número total de combinações resultante desta tabela é 405. O significado de cada um é descrito a seguir.

Tabela 6: Diferentes parâmetros de simulação.

Parâmetro	Valores Assumidos	Total
Tipo do AG	AG1, AG2, AG3, AG4, AG5	5
Número de Eventos	4, 7, 10	3
Tamanho da População	100, 200, 400	3
Alarmes Falsos (%)	0, 10, 20	3
Alarmes Não-Reportados (%)	0, 10, 20	3

### 6.2.2.1 Tipo do AG

Neste trabalho, diversos modelos matemáticos para o problema do processamento de alarmes foram estudados. Diversas variações de AGs também foram implementadas e avaliadas. Cada um destes modelos e variações é descrito a seguir, tendo suas características comparadas na Tabela 7.

1. AG1: Algoritmo genético de população única, implementando o modelo de “vetor de eventos inteiros” (Modelo Melhorado 1, descrito em 4.2.4).
2. AG2: Algoritmo genético multipopulacional, utilizando o modelo de “vetor de eventos inteiros”.
3. AG3: Algoritmo genético de população única, implementando o modelo de “palavra de eventos” (Modelo Melhorado 2, descrito em 4.2.5).
4. AG4: Algoritmo genético multipopulacional, utilizando o modelo de “palavra de eventos”.
5. AG5: Este algoritmo implementa a variação do “AG com cromossomo de tamanho variável”, proposta na seção 4.2.6 e implementada para uma única população.

Tabela 7: Comparação entre os diferentes AGs utilizados.

Algoritmo	Modelo	Tamanho Crom.	Software	Num. Pop.
AG1	Vetor de eventos inteiros	Fixo ( $= n_e$ )	GALOPPS	1
AG2	Vetor de eventos inteiros	Fixo ( $= n_e$ )	GALOPPS	4
AG3	Palavra de eventos inteiros	Fixo ( $= M_a$ )	GALOPPS	1
AG4	Palavra de eventos inteiros	Fixo ( $= M_a$ )	GALOPPS	4
AG5	Palavra de eventos inteiros	Variável	Próprio	1

### 6.2.2.2 Número de Eventos Simultâneos

O desempenho do algoritmo deve ser avaliado em função do número de eventos simultâneos ocorridos. Neste trabalho, foi simulada a ocorrência de diferentes números de eventos simultâneos.

### 6.2.2.3 Tamanho da População

O número de indivíduos presentes na população em um AG é fator determinante do desempenho. Populações menores tendem a convergir em menor número de gerações e consumindo menos recursos computacionais, porém às custas da qualidade do resultado. Para determinar um compromisso entre recursos de processamento e qualidade da solução, foram testadas populações com diferentes tamanhos.

### 6.2.2.4 Ruído

O ambiente do sistema elétrico, assim como a maioria dos ambientes reais, está sujeito à presença de ruído. No sistema de processamento de alarmes, este ruído se manifesta na forma de alarmes falsos e/ou alarmes não reportados. Para os testes de desempenho aqui realizados, foi inserido ruído, tanto na forma de “alarmes falsos” como de “alarmes não-reportados”, compondo assim a “palavra de alarmes reportada ao sistema”. O objetivo da inserção do ruído é realizar uma análise do impacto imposto por este sobre o desempenho do sistema de processamento de alarmes.

- Alarmes Falsos: Um determinado número de alarmes aleatórios é gerado e inserido na composição da palavra de alarmes reportada ao sistema. O número de alarmes falsos inserido é tomado percentualmente em relação ao tamanho da palavra de alarmes esperada (0, 10 ou 20%);

- Alarmes Não-Reportados: Um determinado número de alarmes é removido aleatoriamente da palavra de alarmes a ser reportada. Este número é tomado percentualmente em relação ao tamanho da palavra de alarmes esperada (0, 10 ou 20%);

## 6.2.3 Outros Parâmetros do AG

### 6.2.3.1 Critérios de Parada

Os critérios de parada empregados foram:

1. A adequabilidade de um indivíduo atinge o valor unitário, indicando que a equação 3.3 se anula, e por consequência uma solução exata foi encontrada.
2. O máximo número de gerações igual a cem é atingido;

### 6.2.3.2 Método de Seleção

O método de seleção empregado em todos os testes realizados foi o “torneio estocástico” (seção 2.1.3.3), com tamanho  $K = 4$ .

### 6.2.3.3 Probabilidade de Mutação e Cruzamento

Para os AGs implementados utilizando o GALOPPS, foi empregada probabilidade de 90% para cruzamento e 5% para mutação. No AG com cromossomo de tamanho variável, sempre é realizado o cruzamento (100%) e a probabilidade de mutação é de 5%.

### 6.2.3.4 Operador de Cruzamento

Para os AGs implementados utilizando o GALOPPS, foi empregado cruzamento do tipo uniforme. No AG com cromossomo de tamanho variável, o operador de “cortar e colar” padrão foi utilizado.

### 6.2.3.5 AG Multipopulacional

Na implementação do AG de múltiplas populações (AG paralelo insular), foi empregada uma topologia de quatro subpopulações. A política de migração aplicada foi a “injeção” de três para um, ou seja, uma das populações apenas recebe indivíduos advindos das demais, conforme sugerido na Figura 28. A taxa de migração empregada foi de quatro indivíduos por ciclo de

cinco gerações, sendo escolhidos os quatro melhores. O tamanho de cada subpopulação é igual ao do AG de população única dividido por quatro (perfazendo igual número de indivíduos). O paralelismo do algoritmo, porém, foi apenas simulado em um único processador, para fins de comparação dos resultados.

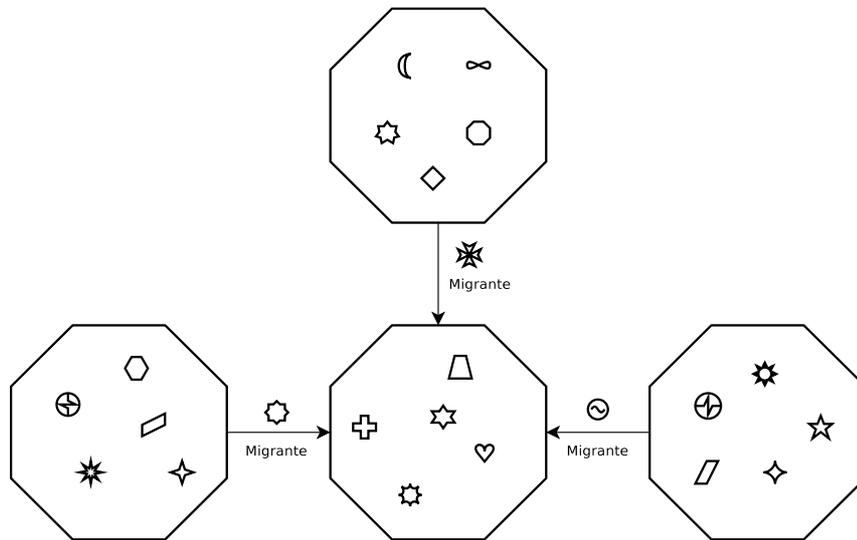


Figura 28: Topologia utilizada para o AG de múltiplas populações.

### 6.3 Considerações Sobre “AG1” e “AG2”

Nesta seção serão brevemente expostos os resultados obtidos para os algoritmos rotulados “AG1” e “AG2”, que implementam o modelo de “Vetor de Eventos Inteiros” para população única e múltipla, respectivamente.

Na Figura 29 é representada a variação da média de acertos de cada algoritmo em função do tamanho da população. Percebe-se que esta média é consideravelmente baixa (geralmente inferior à 65%). No histograma da Figura 30 é representada a distribuição da frequência de acertos do algoritmo. Cada um destes histogramas corresponde a um conjunto de 2700 amostras<sup>3</sup>. Percebe-se que em nenhum dos casos existe tendência em direção à 100%, ou seja, a frequência de acertos é baixa.

Na Figura 31 é representada a variação da média de erros cometida por cada algoritmo em função do tamanho da população. Percebe-se que esta média é demasiadamente alta (geralmente superior à 100%), evidenciando o desempenho pobre do modelo de “Vetor de Eventos Inteiros”, usado com estes algoritmos.

<sup>3</sup>Uma amostra consiste no resultado de uma execução do algoritmo.

O desempenho pobre destes algoritmos é devido, principalmente, à elevada dimensão do espaço de busca. Para o caso de teste em questão existem 235 eventos mapeados. Na equação 4.2, adotando  $\alpha = 4$ , resulta em  $4^{235} \simeq 3,05 \times 10^{141}$  possíveis soluções no espaço de busca. Por estes motivos, estes modelos não serão considerados nas seções futuras, onde apenas os demais, que apresentaram desempenho bastante superior, serão analisados.

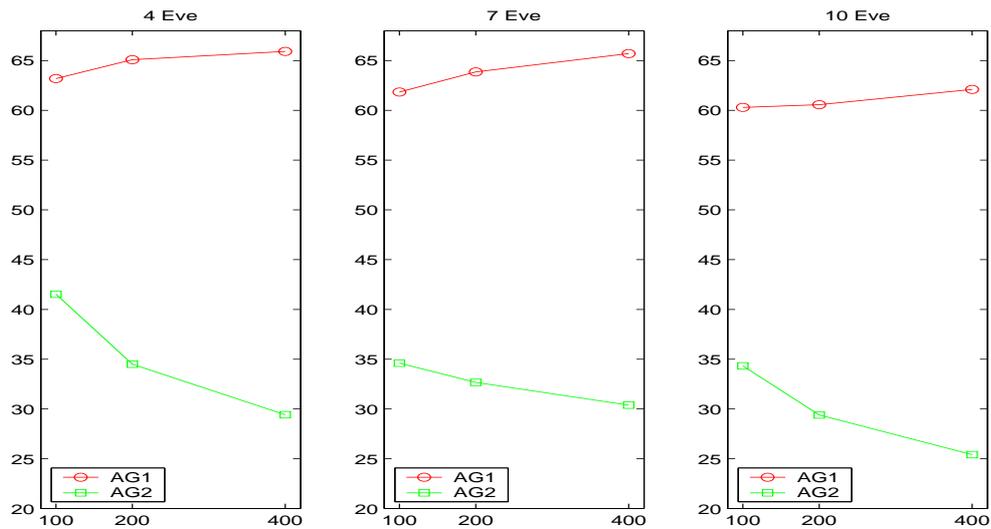


Figura 29: Percentual médio de acertos em função do tamanho da população - AG1 e AG2.

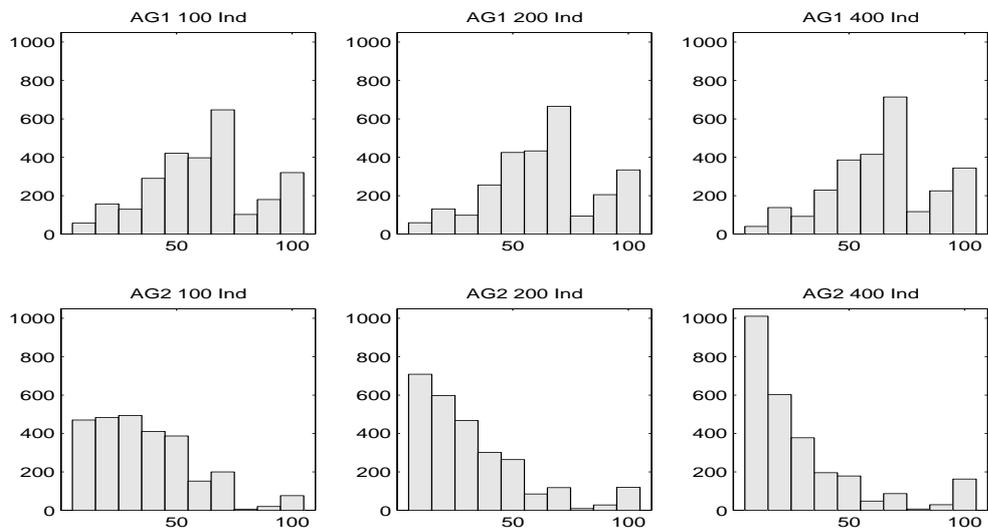


Figura 30: Distribuição de frequência de acertos - AG1 e AG2.

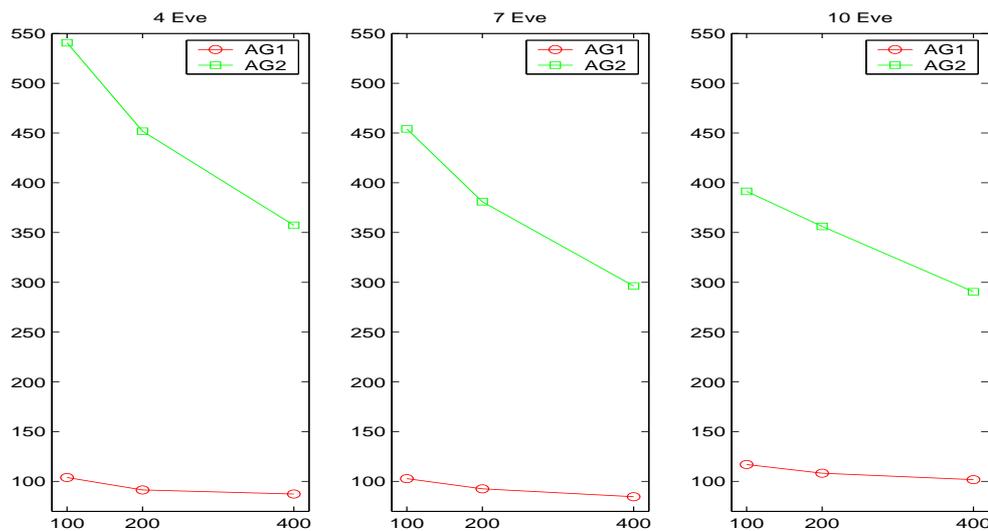


Figura 31: Percentual médio de erros em função do tamanho da população - AG1 e AG2.

## 6.4 Avaliação Segundo os Diversos Parâmetros

Nas seções seguintes, serão comparados apenas os resultados obtidos para os algoritmos “AG3” (modelo de palavra de eventos, implementado com AG tradicional), “AG4” (versão multipopulacional do “AG3”) e “AG5” (AG com cromossomo de tamanho variável). Para cada parâmetro avaliado, é feita uma estatística dos resultados obtidos em relação aos demais parâmetros, para todas as execuções do algoritmo. Para a avaliação do desempenho em função do tamanho da população, por exemplo, são tomados todos os resultados das execuções realizadas com população de tamanho 100 (cem), variando-se os demais parâmetros (percentual de ruído). A seguir aplica-se a estatística para os demais tamanhos de população (200 e 400).

### 6.4.1 Desempenho em Função do Tamanho da População

Esta estatística é feita para cada um dos algoritmos testados, e para cada um dos diferentes tamanhos da palavra de eventos esperada. Estes valores são calculados a partir de um conjunto de novecentas amostras (cem para cada possível combinação dos percentuais de ruído aplicado).

#### 6.4.1.1 Acertos

Na Tabela 8 foram transcritos os valores, respectivamente, da média aritmética, moda e amplitude do conjunto de dados referente ao percentual de acertos obtido por cada algoritmo. Cada uma das colunas principais refere-se ao tamanho da população empregada. Cada uma

das linhas principais refere-se ao algoritmo empregado. Os valores representados em negrito correspondem ao melhor resultado obtido (maior média), segundo cada conjunto de parâmetros.

Na Figura 32 a informação da média aritmética do percentual de acertos é representada graficamente. Na Figura 33 são representados os histogramas de distribuição da frequência de acertos atingidos por cada algoritmo para diferentes tamanhos de população. Cada um dos histogramas da Figura 33 compreende duas mil e setecentas (2700) amostras.

Tabela 8: Percentual de acertos em função do tamanho da população.

		100 Indivíduos			200 Indivíduos			400 Indivíduos		
		Méd	Mod	Amp	Méd	Mod	Amp	Méd	Mod	Amp
AG3	4 Eve	<b>86,47</b>	100,00	75,00	<b>87,14</b>	100,00	50,00	86,86	100,00	50,00
	7 Eve	<b>83,98</b>	100,00	71,43	<b>84,73</b>	100,00	85,71	85,00	100,00	71,43
	10 Eve	<b>85,88</b>	100,00	60,00	<b>85,82</b>	100,00	60,00	85,80	100,00	60,00
AG4	4 Eve	85,06	100,00	75,00	86,56	100,00	75,00	<b>87,25</b>	100,00	75,00
	7 Eve	82,35	85,71	71,43	84,65	100,00	100,00	<b>85,55</b>	100,00	71,43
	10 Eve	81,96	80,00	90,00	85,61	80,00	60,00	<b>85,99</b>	100,00	90,00
AG5	4 Eve	86,11	100,00	50,00	86,36	100,00	50,00	86,67	100,00	50,00
	7 Eve	83,52	85,71	57,14	84,67	100,00	85,71	84,70	100,00	71,43
	10 Eve	74,13	70,00	70,00	82,19	80,00	80,00	85,03	100,00	60,00

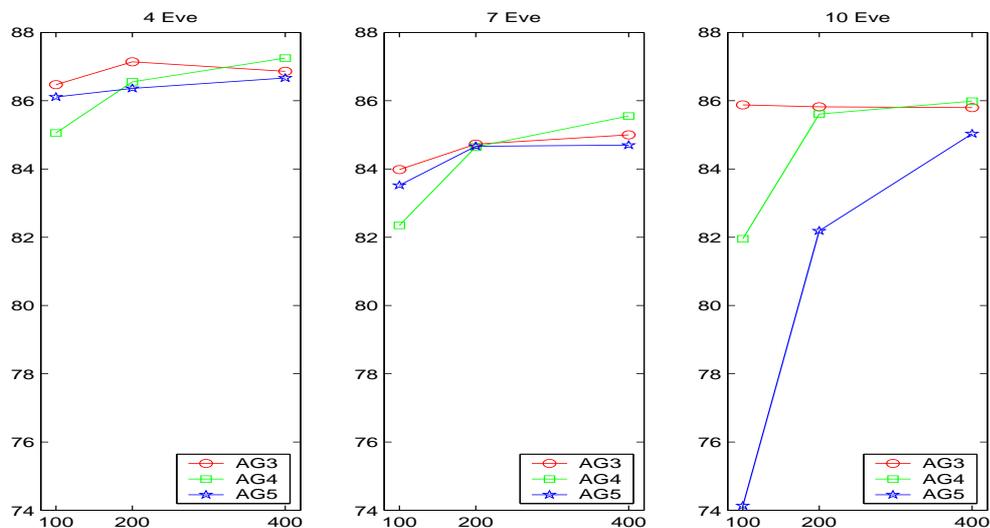


Figura 32: Percentual médio de acertos em função do tamanho da população.

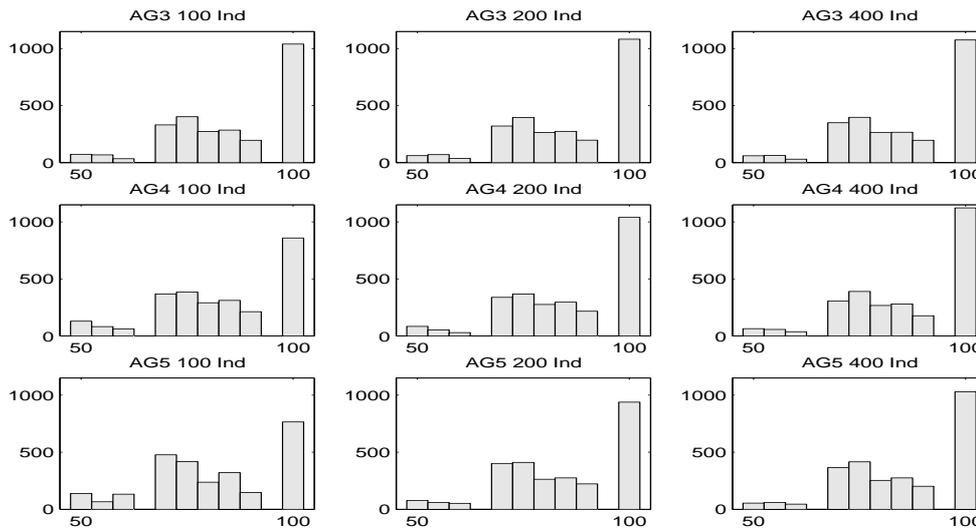


Figura 33: Distribuição de frequência de acertos para diferentes tamanhos de população.

Percebe-se que o desempenho do AG3 é menos sensível à variação do tamanho da população. O AG4 e o AG5 apresentam um percentual médio de acerto sensivelmente menor quando empregadas populações pequenas. No caso do AG4 (de múltiplas populações), este desempenho pobre com populações pequenas é justificável. Para o caso de 100 indivíduos, por exemplo, cada uma das quatro subpopulações é composta por apenas 25 soluções candidatas. Este pequeno número tende a provocar uma convergência mais rápida, porém a solução obtida tende a ser de pior qualidade. Por outro lado, com o aumento do tamanho da população, o AG4 sobrepõe os demais em termos de percentual médio de acerto.

#### 6.4.1.2 Erros

Na Tabela 9 foram transcritos os valores, respectivamente, da média aritmética, moda e amplitude do conjunto de dados referente ao percentual de erros resultantes da execução de cada algoritmo. Os valores representados em **negrito** correspondem ao melhor resultado obtido (menor média de erros), segundo cada conjunto de parâmetros.

Tabela 9: Percentual de erros em função do tamanho da população.

		100 Indivíduos			200 Indivíduos			400 Indivíduos		
		Méd	Mod	Amp	Méd	Mod	Amp	Méd	Mod	Amp
AG3	4 Eve	23,33	0,00	150,00	21,72	0,00	100,00	20,89	0,00	100,00
	7 Eve	29,05	0,00	157,14	26,05	0,00	142,86	24,83	0,00	114,29
	10 Eve	<b>22,42</b>	0,00	100,00	<b>21,79</b>	0,00	110,00	21,99	0,00	100,00
AG4	4 Eve	32,06	0,00	175,00	22,97	0,00	150,00	20,69	0,00	175,00
	7 Eve	35,76	0,00	157,14	27,27	0,00	142,86	25,10	0,00	142,86
	10 Eve	29,37	-	120,00	22,71	0,00	110,00	<b>21,60</b>	0,00	100,00
AG5	4 Eve	<b>21,11</b>	0,00	100,00	<b>20,61</b>	0,00	100,00	<b>19,89</b>	0,00	125,00
	7 Eve	<b>25,44</b>	0,00	128,57	<b>23,81</b>	0,00	142,86	<b>23,92</b>	0,00	185,71
	10 Eve	34,90	30,00	110,00	25,96	20,00	110,00	22,36	20,00	100,00

Na Figura 34 a informação da média aritmética do percentual de acertos é representada graficamente. Na Figura 35 são representados os histogramas de distribuição da frequência de erros de cada algoritmo para diferentes tamanhos de população.

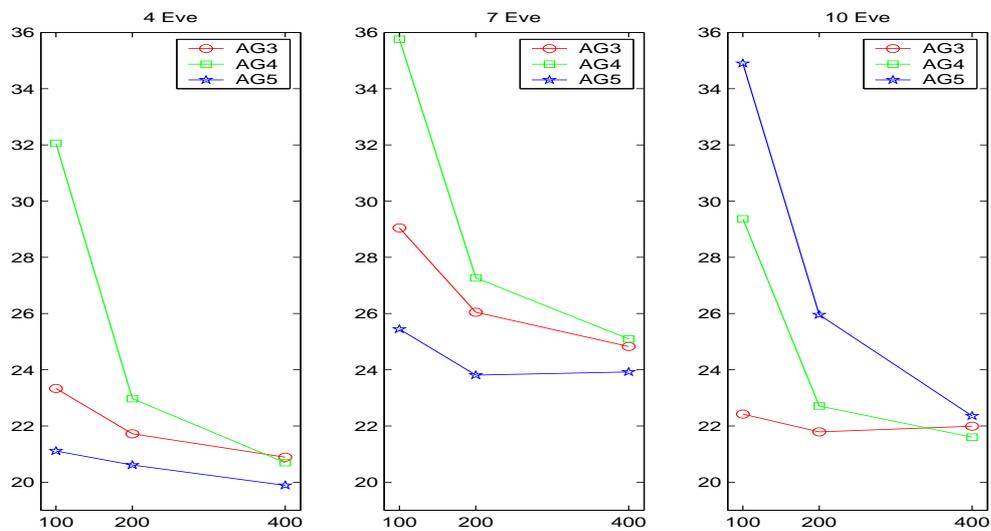


Figura 34: Percentual médio de erros em função do tamanho da população.

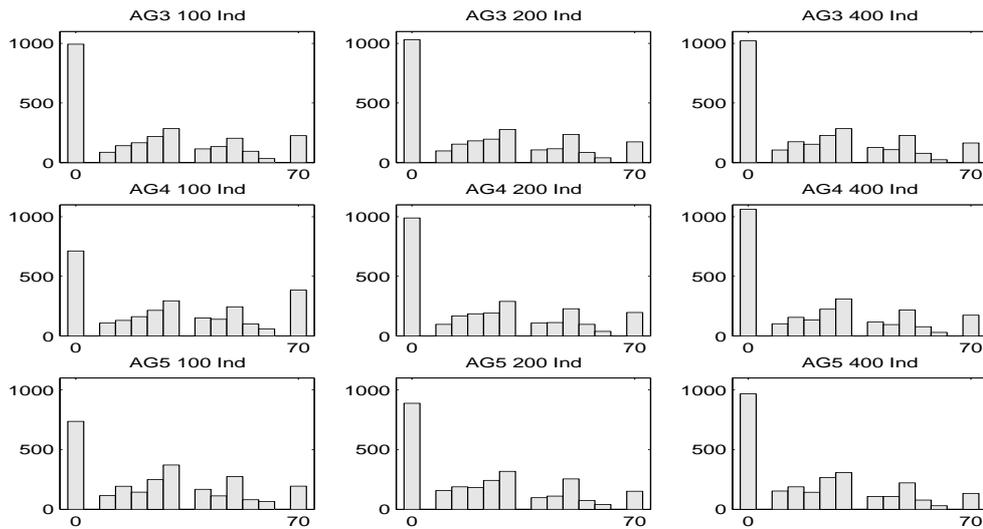


Figura 35: Distribuição de frequência de erros para diferentes tamanhos de população.

A robustez do AG3 em relação à variação do tamanho da população pode ser novamente observada ao avaliar-se o percentual médio de erros. A tendência do AG4 em superar os demais quando utilizando populações maiores parece confirmar-se, segundo sugerido na Figura 34 pela maior inclinação da curva correspondente a este. O AG5, por sua vez, costuma apresentar menor percentual médio de erros que os demais, pelo menos nos casos mais simples (menor número de eventos).

#### 6.4.1.3 Tempo de Cálculo

Alguns resultados a respeito do consumo de recursos computacionais foram coletados, com a finalidade de demonstrar a viabilidade da operação em “tempo real” e o impacto de determinados parâmetros sobre esta. Estes resultados, porém, devem ser analisados com certo cuidado, pois o sistema é apenas um protótipo. O equipamento e software utilizados nas simulações foram:

- Processador: Intel Pentium 4, 2.80GHz, 1024 KB *cache*, 512 MB RAM;
- Sistema Operacional: Debian 3.1, Linux 2.6.8-2-686;

Na Tabela 10 foram transcritos os valores médios de tempo gasto para a execução dos algoritmos de diagnóstico. A Figura 36 ilustra graficamente estes resultados. Como sugere a Figura 36, o tempo médio de cálculo aumenta de forma aproximadamente linear com o aumento do tamanho

da população. O número de eventos também provoca um pequeno aumento no tempo médio de cálculo, porém pouco significativo.

Tabela 10: Tempo médio de cálculo (s) em função do tamanho da população.

		100 Indivíduos	200 Indivíduos	400 Indivíduos
AG3	4 Eve	0,46	0,92	1,96
	7 Eve	0,53	1,01	2,19
	10 Eve	0,51	1,03	2,25
AG4	4 Eve	0,51	0,96	1,77
	7 Eve	0,57	1,02	1,99
	10 Eve	0,56	1,05	2,03
AG5	4 Eve	12,77	40,95	143,62
	7 Eve	17,50	51,04	168,66
	10 Eve	20,50	59,40	191,09

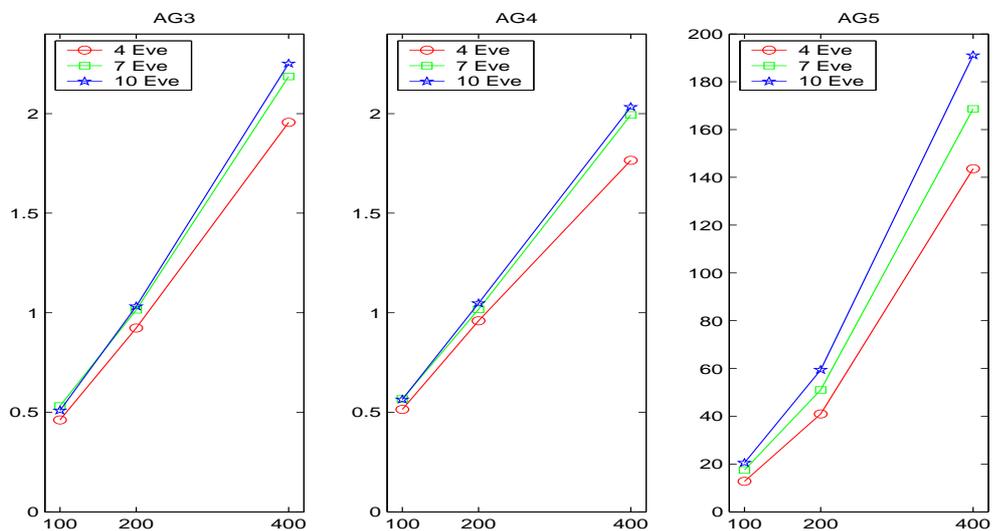


Figura 36: Tempo médio de cálculo (s) em função do tamanho da população.

O algoritmo paralelo de múltiplas populações (AG4) apresenta tempos de execução semelhantes ao algoritmo de população única, pois foi executado de maneira “sequencial”, ou seja, em um único processador. Se cada uma das subpopulações empregadas fosse executada em um processador dedicado, o ganho de desempenho seria próximo de quatro vezes.

O AG5, quando comparado ao AG3 e AG4, apresenta um elevado consumo de tempo computacional. Esta comparação, porém, pode ser chamada de “injusta”. A implementação dos algoritmos AG4 e AG5 é baseada no GALOPPS, um pacote altamente otimizado e já com vários anos de tradição. Por outro lado, o AG5 foi implementado “à moda caseira”, unicamente para o uso neste trabalho. Entre os fatores que causam a degradação do desempenho desta implementação, podem ser enunciados:

- Implementação pouco especializada: Para acelerar o processo de desenvolvimento, foram utilizados *containers* genéricos da biblioteca padrão C++ na construção do programa. Estes *containers* possuem desempenho insatisfatório para este tipo de aplicação, e, em etapas futuras, devem ser substituídos por implementações mais especializadas.
- Implementação pouco refinada: Nesta etapa do trabalho, não houve preocupação em otimizar o código do programa em relação ao uso de recursos. Um trabalho de refinamento e otimização de código tende a proporcionar ganhos sensíveis em relação ao uso de recursos.

## 6.4.2 Desempenho em Função do Ruído

Esta estatística é feita variando-se o percentual de ruído presente na palavra de alarmes reportada. O ruído apresenta-se na forma de alarmes falsos e alarmes não reportados. Estes valores são calculados a partir de um conjunto de trezentas amostras (cem para cada possível combinação de tamanhos de população).

### 6.4.2.1 Acertos

Na Tabela 11 são transcritos os valores médios do percentual de acertos obtidos por cada algoritmo, para diferentes números de eventos. A variação no percentual de alarmes falsos adicionados ao conjunto reportado é representada nas linhas da tabela. As colunas representam a variação no número de alarmes não-reportados. Os melhores resultados obtidos para cada conjunto de parâmetros estão representados em negrito.

Os dados desta tabela são representados graficamente na Figura 37. Este gráfico é feito fixando-se primeiramente o número de alarmes não-reportados e variando-se o número de alarmes falsos. A linha correspondente ao AG3 para quatro eventos, por exemplo, é obtida representando-se graficamente os três primeiros pontos da primeira coluna, seguidos pelos três primeiros pontos da segunda coluna e assim por diante. Da observação destes dados, percebe-se

que um aumento no número de alarmes não-reportados impõe grande influência no número de acertos do algoritmo, enquanto que um igual aumento no número de alarmes falsos impõe pequena influência nesta medida. Visualmente, isto provoca o aparecimento de uma “escada” descendente no gráfico. Em outras palavras, pode-se dizer que tirando-se informação do conjunto reportado à taxa de 10% (alarmes não-reportados), provoca-se uma queda de aproximadamente 10% (em média, 11,15%) na qualidade dos diagnósticos. Alarmes falsos, por sua vez, impõem menor influência (em média 0,26% de queda para cada 10% de alarmes falsos), como pode ser visto na leve inclinação descendente da curva entre cada degrau.

Tabela 11: Percentual médio de acertos em função do percentual de ruído.

Falsos↓ Não-Rep→	4 Eve			7 Eve			10 Eve			
	0 %	10 %	20 %	0 %	10 %	20 %	0 %	10 %	20 %	
0 %	99,25	<b>85,25</b>	<b>77,50</b>	96,48	<b>84,09</b>	72,76	<b>98,37</b>	<b>85,43</b>	<b>75,10</b>	
AG3	10 %	99,50	<b>83,58</b>	76,92	97,05	<b>84,90</b>	<b>73,33</b>	<b>97,43</b>	<b>84,67</b>	<b>75,30</b>
	20 %	98,83	<b>83,17</b>	<b>77,42</b>	<b>96,86</b>	<b>83,66</b>	72,00	<b>96,47</b>	<b>85,07</b>	<b>74,67</b>
	0 %	98,42	84,50	76,17	96,24	83,33	<b>73,48</b>	96,20	84,03	74,57
AG4	10 %	98,17	82,83	<b>78,00</b>	95,95	83,90	72,19	95,17	82,80	74,67
	20 %	98,08	83,42	77,00	95,71	83,28	<b>73,57</b>	95,03	84,17	74,03
	0 %	<b>100,00</b>	83,00	76,67	<b>98,19</b>	83,52	72,48	90,23	80,50	70,87
AG5	10 %	<b>99,83</b>	83,17	76,75	<b>97,24</b>	83,14	72,33	89,77	80,57	71,60
	20 %	<b>99,83</b>	82,83	75,33	96,71	82,24	72,81	90,20	78,70	71,63

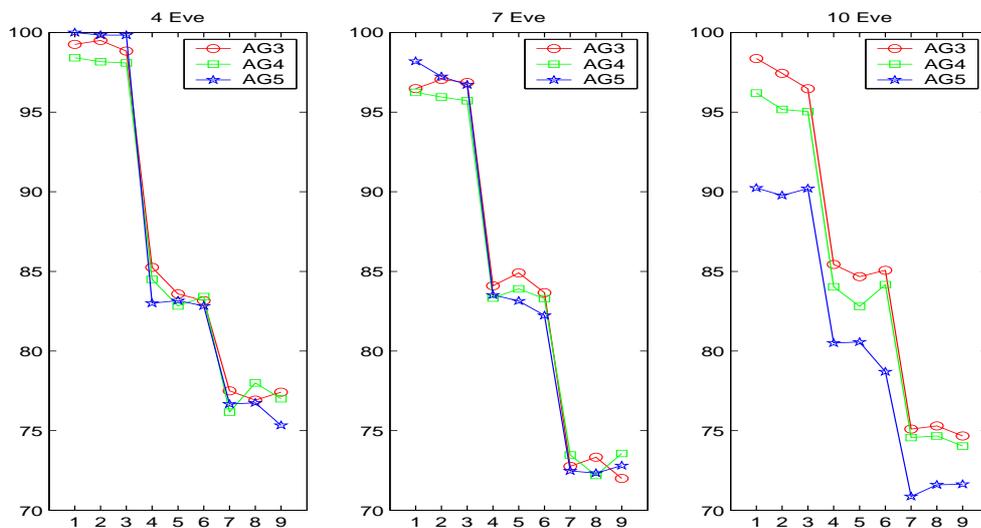


Figura 37: Percentual médio de acertos em função do ruído.

### 6.4.2.2 Erros

Na Tabela 12 são transcritos os valores médios do percentual de erro obtido por cada algoritmo, para diferentes números de eventos. Na Figura 38 é ilustrado o respectivo gráfico. Novamente percebe-se o aparecimento do efeito “escada”, causado pela maior dependência em relação ao percentual de alarmes não-reportados. Neste caso, 10% de aumento no número de alarmes não-reportados provoca aumento médio de 16,15% no percentual de erros. 10% de aumento no número de alarmes falsos provoca 1,40% de aumento médio no percentual de erros.

Tabela 12: Percentual médio de erros em função do percentual de ruído.

Falsos↓ Não-Rep→	4 Eve			7 Eve			10 Eve			
	0 %	10 %	20 %	0 %	10 %	20 %	0 %	10 %	20 %	
0 %	2,42	<b>21,92</b>	34,42	8,33	26,29	42,33	<b>3,10</b>	<b>21,73</b>	<b>36,97</b>	
AG3	10 %	3,17	26,58	39,50	8,43	<b>26,67</b>	42,19	<b>5,03</b>	<b>23,57</b>	<b>36,97</b>
	20 %	5,42	28,25	<b>36,17</b>	9,48	29,81	46,24	<b>6,97</b>	<b>24,03</b>	<b>40,23</b>
AG4	0 %	6,83	26,17	39,50	9,86	29,10	44,05	6,93	24,10	38,27
	10 %	9,00	30,50	35,83	12,14	30,33	47,62	9,27	27,57	39,33
	20 %	9,17	30,17	40,00	13,48	32,00	45,81	9,17	25,53	40,87
AG5	0 %	<b>0,00</b>	24,67	<b>33,58</b>	<b>3,24</b>	<b>25,05</b>	<b>39,52</b>	12,67	27,23	40,23
	10 %	<b>1,42</b>	<b>26,00</b>	<b>34,25</b>	<b>6,00</b>	<b>26,67</b>	<b>40,29</b>	14,70	27,13	39,97
	20 %	<b>2,00</b>	<b>26,42</b>	36,50	<b>8,19</b>	<b>29,38</b>	<b>41,19</b>	14,73	31,77	41,20

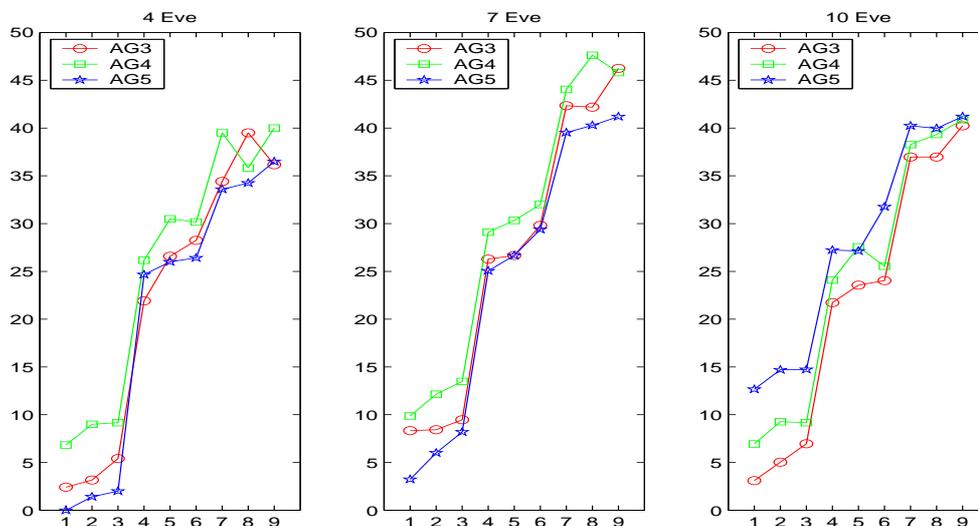


Figura 38: Percentual médio de erros em função do percentual de ruído.

### 6.4.3 Desempenho em Função do Número de Eventos

Na Figura 39 são apresentados os gráficos que ilustram o percentual médio de acertos em função do número de eventos simultâneos. Em contrapartida, na Figura 40 estão representados os correspondentes gráficos de erro percentual.

Para facilitar a análise das tendências, um conjunto extra de simulações foi realizado, sendo este composto por palavras de vinte eventos simultâneos<sup>4</sup>. Os valores numéricos dos pontos destes gráficos são os mesmos encontrados nas Tabelas 8 e 9, respectivamente, com exceção daqueles referentes às simulações para vinte eventos. Da análise destes gráficos, percebe-se que o AG5 é mais sensível ao aumento do número de eventos no conjunto de entrada que os demais algoritmos.

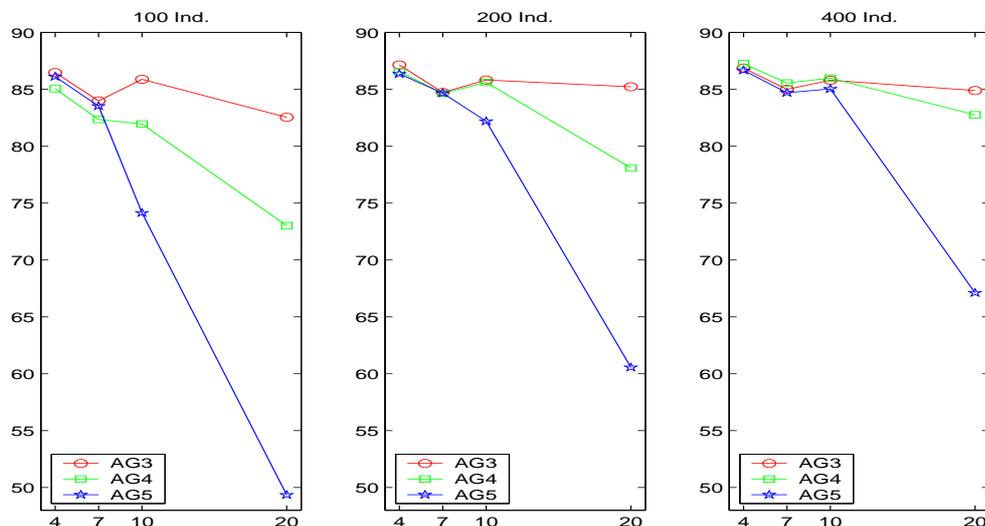


Figura 39: Percentual médio de acertos em função do número de eventos.

<sup>4</sup>Estas simulações extras foram utilizadas apenas nesta seção. As estatísticas e gráficos apresentados no restante do trabalho não levam em conta este conjunto.

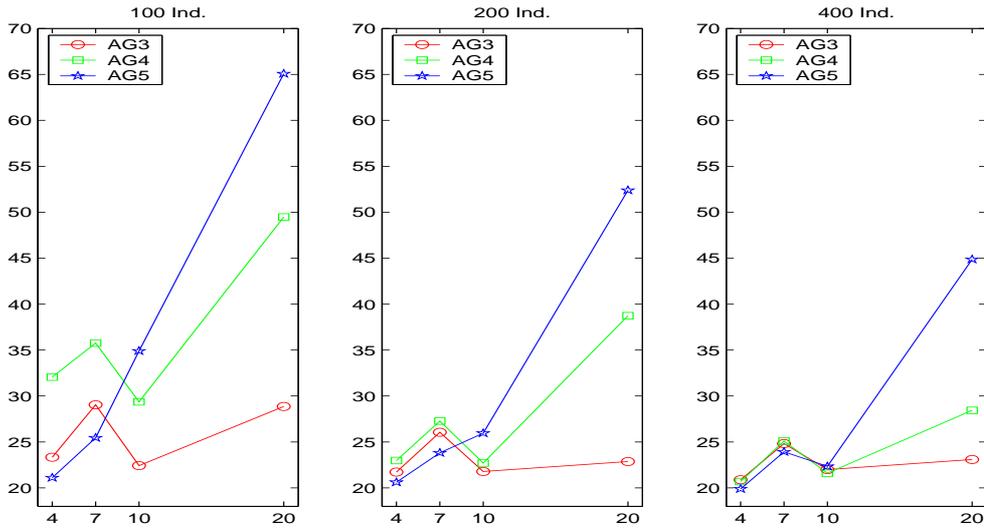


Figura 40: Percentual médio de erros em função do número de eventos.

### 6.4.4 Desempenho Geral em Função do Tipo do AG

Esta comparação foi feita para cada um dos três algoritmos em questão tomando-se uma estatística sobre todos os parâmetros avaliados (uma espécie de “média das médias”). Na Tabela 13 são apresentados os percentuais de erros e acertos obtidos para cada algoritmo. As colunas da tabela correspondem à média aritmética, mediana, moda e amplitude dos percentuais, respectivamente. Estas estatísticas são obtidas a partir de um conjunto de oito mil e cem (8.100) amostras correspondentes à diferentes execuções de cada algoritmo. Os valores da média aritmética são representados graficamente na Figura 41.

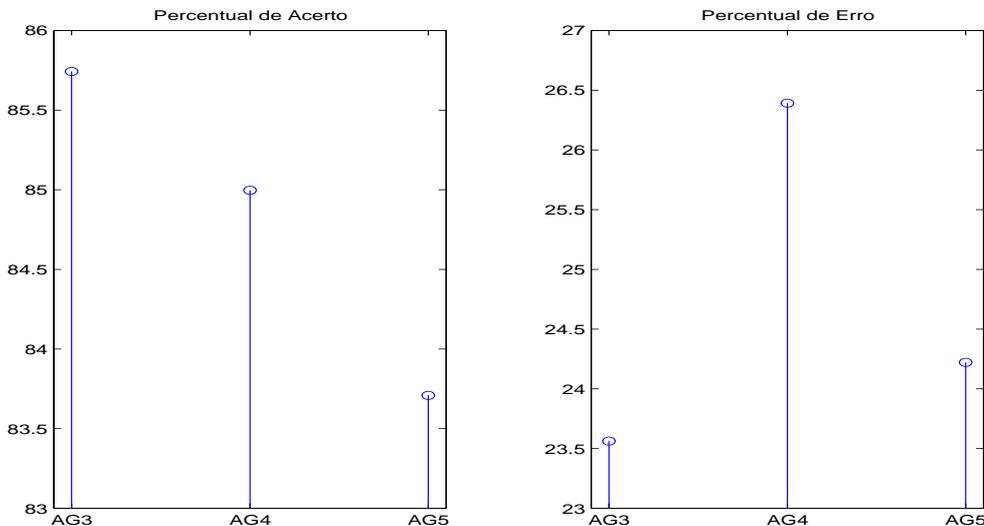


Figura 41: Percentual médio de erros e acertos de cada algoritmo.

Tabela 13: Percentual de acertos e erros em função do tipo de AG.

	AG3				AG4				AG5			
	Méd	Med	Mod	Amp	Méd	Med	Mod	Amp	Méd	Med	Mod	Amp
Ac.	<b>85,74</b>	85,71	100	85,71	85,00	85,71	100	100	83,71	85,71	100	85,71
Er.	<b>23,56</b>	20,00	0,00	157	26,39	25,00	0,00	175	24,22	20,00	0,00	185

Estes resultados sugerem que o AG3 possui desempenho superior aos demais. Deve-se, porém, fazer algumas observações a respeito desta conclusão, pois estas estatísticas foram tomadas em relação à média do desempenho segundo todos os parâmetros de avaliação.

- O AG3 demonstrou-se mais robusto em relação aos diferentes parâmetros de avaliação.
- Em relação ao percentual médio de erros, o AG5 apresentou desempenho superior, exceto para os casos com maior número de eventos, conforme sugerido pelos valores em negrito na Tabela 9.
- Em relação ao percentual médio de acertos, ao aumentar-se o tamanho da população, o AG4 apresentou desempenho superior, de acordo com o sugerido pelos valores em negrito na Tabela 8.

#### 6.4.4.1 Representatividade dos Resultados

Para a comprovação da representatividade estatística dos resultados da Tabela 13, foi empregado o teste não-paramétrico de Kruskal-Wallis. Se o valor “ $p$ ” retornado pelo teste for “próximo de zero”, as amostras são consideradas “estatisticamente significativas”, ou seja, foram tomadas de diferentes populações (ou de populações com diferentes distribuições) (The MathWorks Inc., 2006). Valores geralmente considerados aceitáveis são  $p \leq 0,05$  (95% ou mais de confiança). Os valores de “ $p$ ” resultantes das comparações feitas entre cada par de conjuntos de amostras foram transcritos na Tabela 14.

Tabela 14: Valores “ $p$ ” retornados pelo teste de Kruskal-Wallis.

Comparação	Valor “ $p$ ”
AG3 x AG4	$3,67 \times 10^{-9}$
AG3 x AG5	$3,67 \times 10^{-9}$
AG4 x AG5	$2,49 \times 10^{-2}$

## 6.5 Redução do Número de Mensagens

A redução percentual do número de mensagens de alarme apresentadas ao operador oferece uma medida da capacidade de síntese do sistema inteligente. Para o caso do algoritmo proposto, esta capacidade depende essencialmente da complexidade das regras de correlação entre eventos e alarmes formuladas. Regras mais complexas (envolvendo maior número de alarmes) proporcionam uma redução mais significativa no número de mensagens. Esta avaliação, portanto, diz respeito à base de conhecimento e não ao desempenho do AG em si. Na Tabela 15 é apresentado o percentual médio de redução do número de mensagens obtido.

Tabela 15: Percentual médio de redução do número de mensagens.

4 Eventos	7 Eventos	10 Eventos	Média Geral
50,00%	53,02%	52,61%	51,88%

## 6.6 Comparação com Outros Trabalhos

Existem grandes dificuldades em se comparar resultados, em função da inexistência de um padrão de testes universalmente aceito para estes sistemas. A inexistência de padrões é em partes explicada pela variedade de pacotes SCADA/EMS existentes no mercado, bem como das diferentes filosofias de operação adotadas pelas diversas empresas do setor elétrico. Como consequência, cada sistema descrito na literatura possui suas peculiaridades, requisitos distintos, e objetivos levemente diferentes. A própria definição de alarme pode variar de um sistema para o outro, e por consequência a forma com que esta informação é empregada na obtenção do diagnóstico pode variar. Outro grande obstáculo para a realização de uma comparação efetiva é a aparente relutância dos autores em detalhar os testes realizados e apresentar resultados numéricos significativos.

Em (COUTTO FILHO et al., 1999) é proposto o emprego de um sistema de localização de faltas em seções reduzidas do sistema elétrico, de modo a restringir a dimensão do problema. O método de diagnóstico empregado é baseado em RNAs. Os resultados apresentados neste trabalho referem-se a uma seção composta por sete barras, sendo que o sistema foi testado para 140 diferentes padrões de alarme. Para 90,00% dos casos de teste o sistema produziu o diagnóstico esperado. Em 1,43% dos casos, um diagnóstico incorreto foi produzido. Nos 8,57% dos casos restantes foram gerados diagnósticos indeterminados. O autor faz alusão à apresentação de padrões de alarmes corrompidos à entrada do sistema, porém o grau de corrupção não é es-

pecificado. Em (SOUZA et al., 2004), um sistema elétrico semelhante é utilizado, empregando no diagnóstico um sistema híbrido *neuro-fuzzy*. Os resultados obtidos são consideravelmente superiores ao trabalho anterior, sendo relatados diagnósticos corretos em 95,00% a 96,80% dos casos.

Batista (2005) utiliza em seu trabalho métodos baseados em RNAs associadas a árvores de decisão. O treinamento e a validação foram feitos utilizando uma base de dados de alarmes do sistema SCADA da Companhia Elétrica do Estado de Pernambuco, referentes a dois anos de operação. A dimensão exata do sistema elétrico em questão, porém, não é especificada. Este sistema proporciona uma redução de 85,40% no número de mensagens de alarme, com 74,70% de acerto na detecção de falhas e 92,50% de acerto na classificação das falhas. Os resultados obtidos por Faria et al. (2002) apresentam uma redução do número de mensagens da ordem de 84,10% à 92,70%.

Estes trabalhos, porém, assemelham-se mais à definição de sistemas de diagnóstico de falhas (KIRSCHEN; WOLLENBERG, 1992), pois utilizam outras informações, além das mensagens de alarme, para realizar o diagnóstico. Em contrapartida, a abordagem proposta nesta dissertação não utiliza nenhuma informação adicional, tal como o estado de pontos no banco de dados ou valores de medidas analógicas na composição das mensagens de saída. Apenas as informações presentes na fila de de alarme apresentadas ao operador são utilizadas como base de conhecimento factual.

O modelo de processamento de alarmes utilizando AG proposto por Wen, Chang e Srinivasan (1995) e descrito na seção 4.2.1 foi implementado e testado neste trabalho, para fins de comparação. O AG empregado nos testes foi o de população única, semelhante ao descrito no trabalho original. A mesma base de regras referente à SE Batel foi empregada, permitindo que se comparem os resultados lado a lado.

As taxas de acertos e erros percentuais médios obtidos foram transcritas nas Tabelas 16 e 17, respectivamente. Os gráficos representando a variação da média de acertos e erros em função do tamanho das populações utilizadas são ilustrados nas figuras 42 e 43 respectivamente.

Na Tabela 18 a média geral destes resultados é confrontada com aqueles obtidos para o AG3. Embora a média do percentual de acertos deste modelo aproxime-se daquela obtida para o AG3, o percentual de erros apresenta-se consideravelmente elevado, ressaltando as deficiências do modelo clássico.

Tabela 16: Percentual de acertos em função do tamanho da população - Modelo binário.

	100 Indivíduos			200 Indivíduos			400 Indivíduos		
	Méd	Mod	Amp	Méd	Mod	Amp	Méd	Mod	Amp
4 Eve	86,81	100,00	100,00	86,56	100,00	75,00	87,03	100,00	75,00
7 Eve	83,43	85,71	57,14	83,40	85,71	57,14	83,95	85,71	57,14
10 Eve	77,69	80,00	90,00	77,96	80,00	60,00	78,16	80,00	80,00

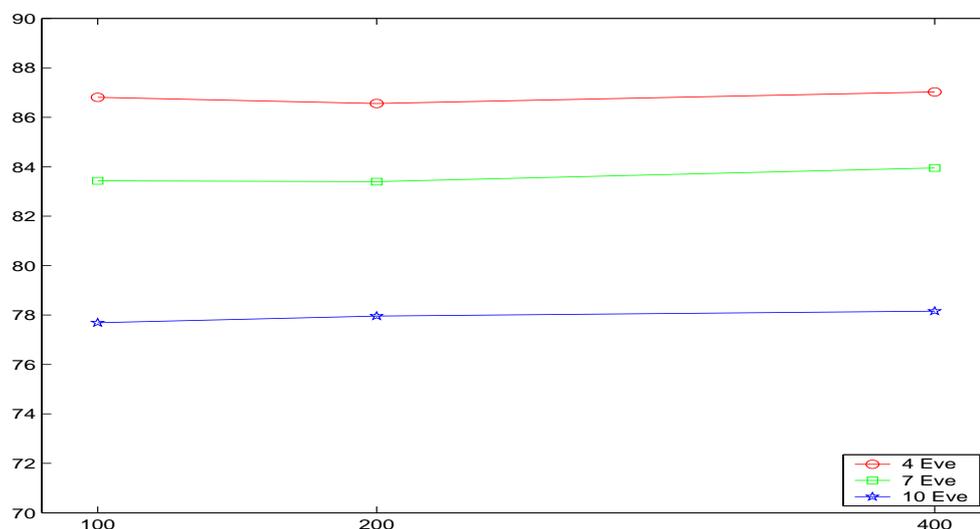


Figura 42: Percentual médio de acertos em função do tamanho da população - Modelo binário.

Tabela 17: Percentual de erros em função do tamanho da população - Modelo binário.

	100 Indivíduos			200 Indivíduos			400 Indivíduos		
	Méd	Mod	Amp	Méd	Mod	Amp	Méd	Mod	Amp
4 Eve	42,67	25,00	250,00	40,56	25,00	175,00	40,89	25,00	200,00
7 Eve	67,59	42,86	200,00	66,10	42,86	200,00	63,87	42,86	214,29
10 Eve	75,67	80,00	170,00	73,29	60,00	160,00	72,28	70,00	160,00

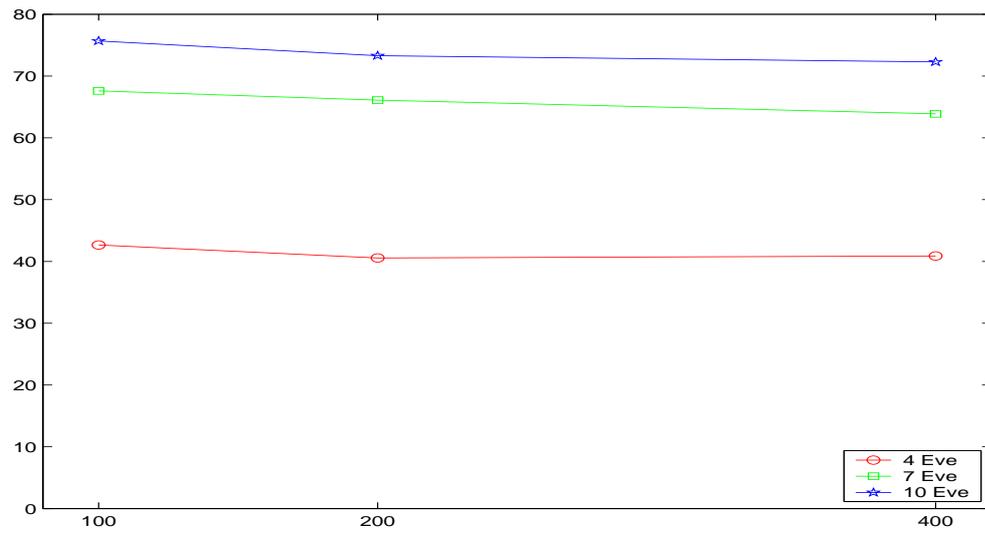


Figura 43: Percentual médio de erros em função do tamanho da população - Modelo binário.

Tabela 18: Comparação com o melhor resultado.

	AG3			Clássico		
	Méd	Mod	Amp	Méd	Mod	Amp
Acerto	<b>85,74</b>	100,00	85,71	82,77	100,00	100,00
Erro	<b>23,56</b>	0,00	157,14	60,32	50,00	250,00



## 7 *Conclusões e Trabalhos Futuros*

### 7.1 **Conclusões**

Esta dissertação oferece a fundamentação necessária para a implementação de uma ferramenta de tratamento de alarmes em centros de operação de um sistema elétrico. Foram abordados os principais aspectos do problema do processamento de alarmes e apresentados alguns dos possíveis modelos de solução utilizando Algoritmos Genéticos. Os frutos mais significativos deste trabalho foram:

#### 1. **Modelagem do problema segundo um critério de otimização**

A questão do processamento de alarmes foi tratada assumindo-se sua semelhança ao problema da cobertura de conjuntos, que é comprovadamente NP-completo. A obtenção da solução ótima para instâncias deste problema é garantida apenas através da enumeração completa do espaço de busca. Esta enumeração, porém, torna-se computacionalmente intratável quando a instância assume dimensões não triviais. Nestes casos, a aplicação de métodos mais robustos, como os Algoritmos Genéticos, revela-se interessante.

#### 2. **Proposição de diversas modalidades de AGs e diferentes codificações**

O espaço de soluções foi inicialmente codificado na forma de um vetor de eventos inteiros, de comprimento igual ao número de eventos do sistema. Esta codificação, porém, apresentou resultados insatisfatórios, em função da elevada dimensão do espaço de busca. Uma codificação na forma de uma palavra de eventos, que pode assumir um comprimento qualquer (geralmente bem menor do que o número de eventos do sistema) proporcionou um importante ganho de qualidade. Foram empregados na busca pela solução ótima o AG tradicional (de população única), o AG multipopulacional e um AG com cromossomo de tamanho variável, inspirado no AG “*messy*”.

#### 3. **Implementação dos algoritmos propostos e prova de conceito**

Protótipos dos algoritmos e codificações propostos foram implementados para testes. A

base de conhecimento necessária para a avaliação da função-objetivo foi construída com o auxílio de profissionais da área de proteção do sistema elétrico. Este conhecimento foi representado de uma forma padronizada, no formato XML, de modo a permitir seu uso como parâmetro de configuração. Um protótipo simplificado do processador de alarmes foi construído e associado ao sistema SASE, para demonstrar a viabilidade da integração com um SCADA/EMS existente.

#### 4. Avaliação dos resultados obtidos em função de diversos parâmetros

Um procedimento automatizado de testes foi elaborado, permitindo que o sistema seja facilmente avaliado em função de diversos parâmetros. Os resultados obtidos sugerem que:

- O AG tradicional, de população única, mostrou-se mais robusto que os demais em relação à variação dos parâmetros. Em contrapartida, possui menor flexibilidade, dificultando, por exemplo, o emprego de técnicas de processamento paralelo.
- A qualidade das soluções encontradas pelo AG multipopulacional tende a superar aquelas encontradas pelo AG de população única quando o tamanho das populações empregadas for adequado. Esta categoria de AG permite ainda que se introduzam no sistema recursos de processamento distribuído, de maneira fácil e transparente. Tais recursos podem proporcionar um considerável ganho de desempenho.
- O AG com cromossomo de tamanho variável proposto neste trabalho apresentou menores percentuais médios de erro em comparação com os demais. O desempenho deste, porém, degrada-se significativamente com o aumento da complexidade dos casos de entrada (maior número de eventos).
- A qualidade das soluções produzidas é sensivelmente influenciada pelo tamanho das populações empregadas. Em contrapartida, o volume de recursos computacionais demandado aumenta proporcionalmente ao aumento do número de indivíduos na população.
- O sistema apresenta-se pouco susceptível ao ruído na forma de alarmes falsos. Por outro lado, o ruído na forma de alarmes não reportados provoca uma degradação da qualidade dos diagnósticos proporcional ao percentual de alarmes ausentes.
- A complexidade dos casos de entrada (aumento do número de eventos, e por consequência do número de alarmes) influencia negativamente a qualidade dos diagnósticos fornecidos. Todos os algoritmos testados mostraram-se vulneráveis ao aumento de complexidade, especialmente o AG com cromossomo de tamanho variá-

vel. Esta perda de qualidade, porém, pode ser compensada por um correspondente aumento do tamanho das populações.

### **7.1.1 Considerações Sobre Desempenho em Tempo Real**

Um dos requisitos estabelecidos para um processador de alarmes é que este seja capaz de operar “em tempo real”. Em termos práticos, isto significa que o sistema deve ser capaz de processar uma grande quantidade de alarmes em tempo hábil, porém, a taxa com que esta informação é apresentada ao operador deve ser intencionalmente baixa. Kirschen e Wollenberg (1992) recomendam realizar uma atualização das informações apresentadas a cada vinte ou trinta segundos, para permitir que o operador leia e assimile esta informação. Um aumento nesta taxa de atualização não produz um correspondente aumento na efetividade do operador, e em contrapartida dificulta a interpretação das mensagens, aumentando o nível de *stress*. Desta forma, se o processador de alarmes for capaz de produzir o diagnóstico dentro de um intervalo de vinte segundos, pode-se afirmar que este opera “em tempo real”.

As implementações apresentadas neste trabalho, utilizando o GALOPPS, apresentaram, no pior caso, média de 2,25 segundos de tempo de cálculo (para uma subestação). Considerando o intervalo de atualização de vinte a trinta segundos, o sistema seria capaz de executar o processamento simultâneo para cerca de dez subestações, em um único computador. Se for considerado que, para um dado centro de operação, é pouco provável que existam alarmes reportados em todas as subestações ao mesmo tempo, pode ser estabelecido um “fator de utilização” do recurso, geralmente bem abaixo dos 100%. Com um fator de utilização de 50%, por exemplo, seria possível realizar o processamento para cerca de vinte SEs, e assim por diante. A determinação deste fator de utilização, porém, deve ser objeto de um estudo mais detalhado, onde deve-se tentar determinar situações de pior caso.

## **7.2 Trabalhos Futuros**

### **7.2.1 Implantação de um Sistema Piloto**

Uma evolução lógica deste trabalho seria a construção de um protótipo de avaliação, para uso em centros de operação. O objetivo deste protótipo seria verificar a aceitação do sistema por parte dos operadores, e coletar sugestões de possíveis aperfeiçoamentos e adequações. Uma melhor integração com o sistema SCADA/EMS também é altamente desejável, possibilitando, por exemplo, a realização do reconhecimento de alarmes diretamente via processador inteli-

gente.

Outra necessidade seria a disponibilização de um mecanismo de atualização automática da base de conhecimento (correlação alarmes/eventos). Esta atualização é necessária sempre que são realizadas alterações topológicas no sistema. Eventualmente, poderia ser estudada a possibilidade de permitir que o próprio operador adicione novas regras à base de conhecimento.

### 7.2.2 Melhorias nos Algoritmos

Segundo o exposto no decorrer deste trabalho, diversos parâmetros influenciam o desempenho do algoritmo genético. Em diferentes casos de entrada, diferentes conjuntos de parâmetros podem ser necessários para proporcionar melhor desempenho. A incorporação de técnicas de controle automático dos parâmetros do AG (LOBO, 2000; MARUO; LOPES; DELGADO, 2005; EIBEN; HINTERDING; MICHALEWICZ, 1999) certamente proporcionaria uma maior robustez ao sistema. Outras heurísticas também podem ser incorporadas no ajuste de parâmetros. O número de alarmes reportado ao sistema, por exemplo, pode ser utilizado para estimar o comprimento (de pior caso) da palavra de eventos que os originou. A partir de informações deste tipo pode-se limitar o tamanho do espaço de busca, ajustar o tamanho das populações (casos mais complexos requerem populações maiores) ou até mesmo optar por um algoritmo mais especializado.

O algoritmo genético multipopulacional apresentou resultados promissores, porém precisa ser melhor explorado. Técnicas de processamento distribuído podem ser facilmente incorporadas a este AG, facilitando expansões e proporcionando ganho de velocidade de processamento. Diferentes topologias e políticas de migração também devem ser avaliadas, podendo proporcionar ganho na qualidade das soluções.

O AG com cromossomo de tamanho variável revelou características interessantes, porém a perda de desempenho em casos mais complexos e a elevada demanda por recursos computacionais constituem uma desvantagem severa. Uma nova implementação das classes de entidade que compõem o programa, substituindo os *containers* da biblioteca padrão por estruturas mais especializadas, deve proporcionar a necessária redução do consumo de recursos. A incorporação de mais características do AG *messy* original também pode contribuir para o aperfeiçoamento dos resultados. A exploração do laço externo, iterando sobre classes de equivalência de ordem superior, pode melhorar a qualidade das soluções obtidas nos casos mais complexos.

## *Referências*

- BATISTA, L. B. Abordagem inteligente para tratamento de alarmes e diagnóstico de falhas em sistemas elétricos. In: VI SIMPASE. *Simpósio de Automação de Sistemas Elétricos*. São Paulo, SP, Brasil, 2005.
- BEASLEY, D.; BULL, D. R.; MARTIN, R. R. An overview of genetic algorithms: Part 2, research topics. *University Computing*, v. 15, n. 4, p. 170–181, 1993.
- BEASLEY, J. E.; CHU, P. A genetic algorithm for the set covering problem. *European Journal of Operational Research*, n. 94, p. 392–404, 1996.
- BERNARD, J. P.; DUROCHER, D. Expert system for fault diagnosis integrated in existing SCADA systems. In: IEEE. *Power Industry Computer Applications Conference*. Scottsdale, AZ, USA, 1993. p. 313–319.
- BLICKLE, T.; THIELE, L. *A Comparison of Selection Schemes Used in Genetic Algorithms*. 2. ed. Gloriosastrasse 35, 8092 Zurich, Switzerland, Dec. 1995. 67 p.
- BOVET, D. P.; CRESCENZI, P. *Introduction to the theory of complexity*. 1st. ed. Hertfordshire, UK: Prentice Hall International (UK) Ltd., 1994. 282 p. (Prentice-Hall International Series In Computer Science). ISBN 0-13-915380-2.
- BRUNNER, T. et al. On-line expert system for power system diagnosis and restoration. *Intelligent Systems Engineering*, v. 2, n. 1, p. 15–24, 1993.
- CAMPOS, G. M. *Estatística Prática para Docentes e Pós-Graduandos*. São Paulo: Faculdade de Odontologia de Ribeirão Preto da Universidade de São Paulo, 2002.
- CANTÚ-PAZ, E. *A Survey of Parallel Genetic Algorithms*. [S.l.], 1997. 28 p.
- CARDOZO, E.; TALUKDAR, S. N. A distributed expert system for fault diagnosis. *IEEE Transactions on Power Systems*, v. 3, n. 2, p. 641–646, May 1988.
- CHEON, S. W.; CHANG, S. H.; CHUNG, H. Y. Development strategies of an expert system for multiple alarm processing and diagnosis in nuclear power plants. *IEEE Transactions on Nuclear Science*, v. 40, n. 1, p. 21–30, Feb. 1993.
- CHOI, S. S. et al. Development of an on-line fuzzy expert system for integrated alarm processing in nuclear power plants. *IEEE Transactions on Nuclear Science*, v. 42, n. 4, p. 1406–1418, Aug. 1995.
- CIESIELSKI, V.; SCERRI, P. Real time genetic scheduling of aircraft landing times. In: ICEC98. *Proceedings of the 1998 IEEE International conference on Evolutionary Computation*. New York, USA, 1998. p. 360–364.

COPEL. *SASE - Sistema de Automação de Subestações*: Manual de configuração e referência. Curitiba, PR, Brasil, 2003. 390 p.

COPEL. *Operação do Sistema de Transmissão*. 2006. Disponível em: <<http://home.copel.com/PagCopel.nsf/docsap/96D119A5876063AC03256B7C00470272?OpenDocument>>. Acesso em: 17 jul. 2006.

COUTTO FILHO, M. B. et al. Localização de defeitos em sistemas de energia elétrica utilizando sistemas inteligentes. In: SNPTEE. *Seminário Nacional de Produção e Transmissão de Energia Elétrica*. Foz do Iguaçu - Paraná, 1999.

DUARTE, A. N. et al. Desenvolvimento de uma ferramenta de tratamento de eventos em redes elétricas. In: V SIMPASE. *Simpósio de Automação de Sistemas Elétricos*. Recife, PE, 2003.

EIBEN, A. E.; HINTERDING, R.; MICHALEWICZ, Z. Parameter control in evolutionary algorithms. *IEEE Trans. on Evolutionary Computation*, v. 3, n. 2, p. 124–141, 1999.

EL-SAYED, M. A. H.; ALFUHAID, A. S. ANN-based approach for fast fault diagnosis and alarm handling of power systems. In: APSCOM-00. *International Conference on Advances in Power System Control, Operation and Management*. [S.l.], 2000. v. 1, p. 54–58.

FARIA, V. R. et al. STA - sistema para tratamento de alarmes: Uma ferramenta de auxílio à operação. In: III SEPOCH. *Simpósio de Especialistas em Operação de Centrais Hidrelétricas*. Foz do Iguaçu, PR, Brasil, 2002.

Free Software Foundation. *GNU General Public License*. 1991. Disponível em: <<http://www.gnu.org/licenses/gpl.html>>. Acesso em: 16 mar. 2006.

Free Software Foundation. *GCC home page*. 2006. Disponível em: <<http://gcc.gnu.org/>>. Acesso em: 16 mar. 2006.

FRISCH, A. C.; CARDOSO, M. G.; ARRUDA, L. V. R. Processamento inteligente de alarmes em centros de operação de estações. In: III SIMPASE. *Simpósio de Automação de Sistemas Elétricos*. Rio de Janeiro, Brasil, 1997.

FURTADO, J. C. *Algoritmo Genético Construtivo na Otimização de Problemas Combinatoriais de Agrupamentos*. 112 p. Tese (Tese de Doutorado em Computação Aplicada) — INPE - Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 1998.

GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. 1st. ed. Boston, MA, USA: Addison-Wesley Professional, 1989. ISBN 0201157675.

GOLDBERG, D. E. et al. Rapid accurate optimization of difficult problems using fast messy genetic algorithms. In: FORREST, S. (Ed.). *Proc. of the Fifth Int. Conf. on Genetic Algorithms*. San Mateo, CA: Morgan Kaufmann, 1993. p. 56–64.

GOODMAN, E. D. *GALOPPS The Genetic Algorithm Optimized for Portability and Parallelism System R. 3.2*. East Lansing, Michigan, USA, 1996.

HASAN, K. et al. An object-oriented expert system for power system alarm processing and fault identification. In: IEEE. *7th Mediterranean Electrotechnical Conference*. Antalya, Turkey, 1994. v. 3, p. 909–912.

HEIN, F. Expert system using pattern recognition by real time signals. *International Conference on High Voltage Electric Systems*, Paris, Aug. 1986.

HOLLAND, J. H. Genetic algorithms: Computer programs that evolve in ways that resemble natural selection can solve complex problems even their creators do not fully understand. *Scientific American*, n. 267, p. 66–72, July 1992.

KANG, H. G.; SEONG, P. H. A methodology for evaluating alarm-processing systems using informational entropy-based measure and the analytic hierarchy process. *IEEE Transactions on Nuclear Science*, v. 46, n. 6, p. 2269–2280, Dec. 1999.

KARGUPTA, H. *SEARCH, Polynomial Complexity, and the Fast Messy Genetic Algorithm*. 188 p. Tese (PhD thesis) — University of Illinois, Urbana, IL, 1996.

KARP, R. M. Reducibility among combinatorial problems. In: PLENUM PRESS. *Complexity of Computer Computations*. New York, 1972. p. 85–103.

KHOSLA, R.; DILLON, T. S. Learning knowledge and strategy of a neuro-expert system architecture in alarm processing. *IEEE Transactions on Power Systems*, v. 12, n. 4, p. 1610–1618, Nov. 1997.

KIRSCHEN, D. S.; WOLLENBERG, B. F. Intelligent alarm processing in power systems. In: IEEE. *Proceedings of the IEEE*. [S.l.], 1992. v. 80, n. 5, p. 663–672.

KOSTKA, P.; SKVOR, Z. Scalable parallel optimisation using fast messy genetic algorithm. In: IEICE. *Asia Pacific Microwave Conference proceedings*. Kyoto, 2002. p. 309–312.

KUMARAVEL, N.; VISWANATHAN, S. Real time implementation of genetic algorithm cancellation of sinusoidal noise in ECG using TMS320C50 DSP processor. *Biomed. Sci. Instrum.*, n. 35, p. 110–115, 1999.

LEVINE, D. M.; BERENSON, M. L.; STEPHAN, D. *Estatística: Teoria e aplicações*. 1st. ed. Rio de Janeiro - RJ: LTC Editora S.A., 2000. ISBN 0-13-095071-8.

LOBO, F. M. P. da G. *The Parameter-Less Genetic Algorithm: Rational and automated parameter selection for simplified genetic algorithm operation*. 143 p. Tese (Doutorado em Engenharia do Ambiente) — Universidade Nova de Lisboa, Faculdade de Ciências e Tecnologia, Lisboa, Portugal, 2000.

MARDLE, S.; PASCOE, S. An overview of genetic algorithms for the solution of optimisation problems. *Computers in Higher Education Economics Review*, University of Bristol, BS8 1HH, United Kingdom, v. 13, n. 1, p. 16–20, 1999.

MARQUEZ, D. da C. et al. A experiência da COPEL na integração de proteções digitais ao sistema de automação de subestações. In: V SIMPASE. *Simpósio de Automação de Sistemas Elétricos*. Recife, PE, 2003.

MARUO, M. H.; LOPES, H. S.; DELGADO, M. R. de Biase da S. Self-adapting evolutionary parameters: Encoding aspects for combinatorial optimization problems. *Lecture Notes in Computer Science*, v. 3448, p. 154–165, 2005.

- MCNELIS, P.; DUFFY, J. *Approximating and Simulating the Stochastic Growth Model: Parameterized Expectations, Neural Networks, and the Genetic Algorithm*. [S.l.], Apr. 1998. Available at <http://ideas.repec.org/p/wpa/wuwpge/9804004.html>.
- MEZA, E. M. et al. Exploring fuzzy relations for alarm processing and fault location in electrical power systems. In: IEEE. *Proc. Porto Power Tech*. Porto, Portugal, 2001. v. 3.
- NEIS, P. et al. Processamento de alarmes em sistemas elétricos de potência utilizando algoritmos genéticos. In: VI SIMPASE. *Simpósio de Automação de Sistemas Elétricos*. São Paulo, SP, Brasil, 2005.
- PFAU-WAGENBAUER, M.; NEJDL, W. Model/heuristic-based alarm processing for power systems. *Journal on AI in Engineering, Design and Manufacturing (AI EDAM)*, v. 7, n. 1, p. 65–78, 1993.
- PRICE, W. R.; WOLLENBERG, B. F.; BERTAGNOLLI, D. B. Survey on excessive alarms. *IEEE Transactions on Power Systems*, v. 4, n. 3, p. 950–956, Aug. 1989.
- QUINTAO, F. P.; MATEUS, G. R.; NAKAMURA, F. G. Uma abordagem evolutiva para o problema de cobertura em redes de sensores sem fio. In: SBC. *Concurso de Trabalhos de Iniciação Científica do Congresso da Sociedade Brasileira de Computação*. São Paulo, 2004. v. 1, p. 440–450.
- ROSSI, A.; DINI, G. Dynamic scheduling of FMS using a real-time genetic algorithm. *International Journal of Production Research*, v. 38, n. 1, p. 1–20, Jan. 2000.
- SAKALL, S. *Jean-Baptiste de Monet, Cavaleiro de Lamarck*. 2005. Disponível em: [http://www.sergiosakall.com.br/artistas/personalidade\\_lamarck.htm](http://www.sergiosakall.com.br/artistas/personalidade_lamarck.htm). Acesso em: 11 jul. 2005.
- SCHMIDT, D. C. *Real-time CORBA with TAO*. 2006. Disponível em: <http://www.cs.wustl.edu/~schmidt/TAO.html>. Acesso em: 16 mar. 2006.
- SKIENA, S. S. *The Algorithm Design Manual*. New York, NY, USA: Springer-Verlag, 1997. 486 p. ISBN 0387948600.
- SMATHERS, D. C.; AKHIL, A. A. *Operating Environment and Functional Requirements for Intelligent Distributed Control in the Electric Power Grid*. Albuquerque, New Mexico - USA, Mar. 2001. 22 p.
- SOUZA, J. C. S. de et al. Alarm processing in electrical power systems through a neuro-fuzzy approach. *IEEE Transactions on Power Delivery*, v. 19, n. 2, p. 537–544, Apr. 2004.
- TANOMARU, J. Motivação, fundamentos e aplicações de algoritmos genéticos. In: *II Congresso Brasileiro de Redes Neurais, Anais*. Curitiba, PR, Brasil: [s.n.], 1995.
- TESCH, D. B. et al. A knowledge-based alarm processor for an energy management system. *IEEE Transactions on Power Systems*, v. 5, n. 1, p. 268–275, Feb 1990.
- The MathWorks Inc. *Statistics Toolbox For Use with MATLAB: User's guide*. Natick, MA, 2006. 1165 p.
- THORNTON, C. The building block fallacy. *Complexity International*, v. 4, 1997.

TROLLTECH. *Qt Reference Documentation Free Edition*. 2006. Disponível em: <<http://doc.trolltech.com/3.2/index.html>>. Acesso em: 16 mar. 2006.

VALE, Z. A.; MOURA, A. Machado e. An expert system with temporal reasoning for alarm processing in power system control centers. *IEEE Transactions on Power Systems*, v. 8, n. 3, p. 1307–1314, Aug. 1993.

VIANNA, D. S. *Heurísticas híbridas para o problema da filogenia*. Tese (Tese de doutorado) — Departamento de Informática, PUC-Rio, Rio de Janeiro, 2004.

WEN, F.; CHANG, C. S. A probabilistic approach to alarm processing in power systems using a refined genetic algorithm. In: *ISAP. Proc. International Conference on Intelligent Systems Applications to Power Systems*. Orlando, FL, USA, 1996. v. 1, p. 14–19.

WEN, F.; CHANG, C. S.; SRINIVASAN, D. Alarm processing in power systems using a genetic algorithm. In: *IEEE. Proc. International Conference on Evolutionary Computation*. Perth, WA, Australia, 1995. v. 1, p. 27–32.

WIKIPEDIA. *Charles Darwin From Wikipedia, the free encyclopedia*. 2005. Disponível em: <[http://en.wikipedia.org/wiki/Charles\\_Darwin](http://en.wikipedia.org/wiki/Charles_Darwin)>. Acesso em: 11 jul. 2005.

WIKIPEDIA. *Jean-Baptiste Lamarck From Wikipedia, the free encyclopedia*. 2005. Disponível em: <[http://en.wikipedia.org/wiki/Jean-Baptiste\\_Lamarck](http://en.wikipedia.org/wiki/Jean-Baptiste_Lamarck)>. Acesso em: 11 jul. 2005.

WIKIPEDIA. *Set cover problem From Wikipedia, the free encyclopedia*. 2005. Disponível em: <[http://en.wikipedia.org/wiki/Set\\_cover](http://en.wikipedia.org/wiki/Set_cover)>. Acesso em: 12 jul. 2005.

WIKIPEDIA. *Googol From Wikipedia, the free encyclopedia*. 2006. Disponível em: <<http://en.wikipedia.org/wiki/Googol>>. Acesso em: 06 mar. 2006.



## *APÊNDICE A – Diagrama de Sequência*

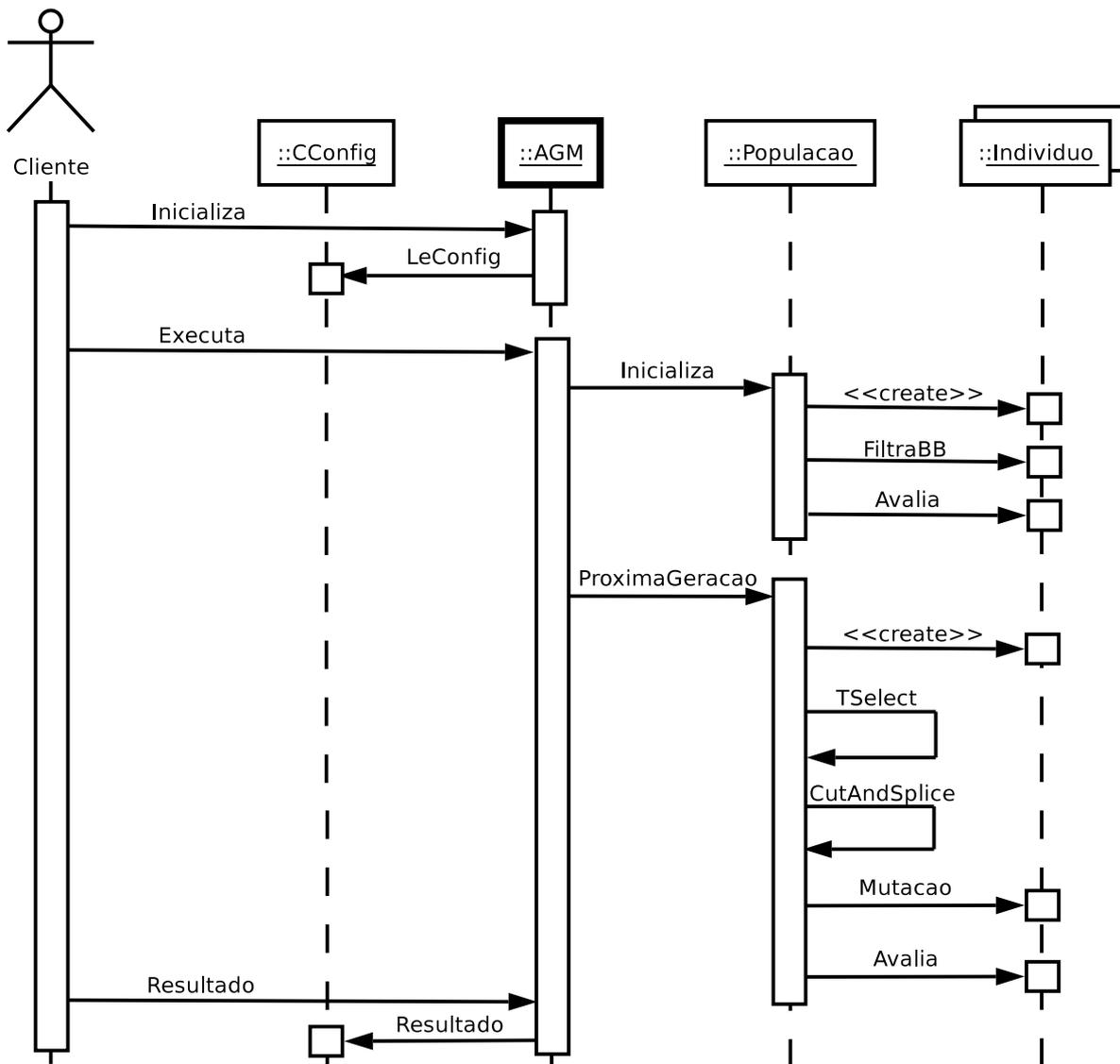


Figura 44: Diagrama de Sequência do AG com Cromossomo de Tamanho Variável.

## ***APÊNDICE B – Diagrama de Classes***

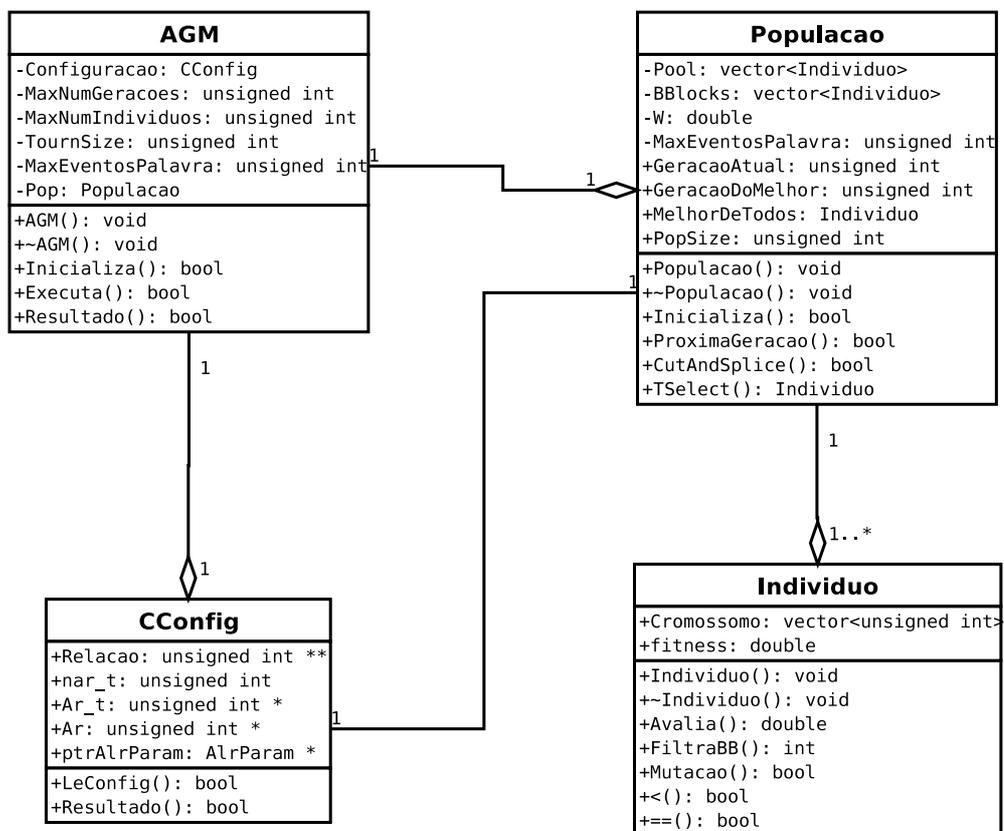


Figura 45: Diagrama de Classes do AG com Cromossomo de Tamanho Variável.

## RESUMO

Em um centro de operação de um sistema elétrico de potência, um pequeno grupo de operadores é responsável pela supervisão de uma área que geralmente compreende dezenas de subestações, englobando vários milhares de pontos de medida. Durante situações de emergência, como tempestades ou falhas em equipamentos, avalanches de mensagens de alarme são produzidas, sobrecarregando os operadores e criando sérias dificuldades de interpretação destas informações.

Ferramentas computacionais capazes de auxiliar o operador, provendo categorização, priorização, síntese e diagnóstico das mensagens de alarme tendem a diminuir a probabilidade de falhas causadas por interpretações equivocadas. O objetivo deste trabalho é propor um método de processamento inteligente de alarmes para ser utilizado associado aos aplicativos de controle e supervisão do sistema elétrico de potência. Este método emprega um tradicional paradigma da Computação Evolucionária, o “Algoritmo Genético”.

A modelagem do problema do processamento de alarmes é formulada de maneira semelhante ao problema da cobertura de conjuntos, e solucionada segundo um critério de otimização. Diversas formas de codificação e diferentes variações de algoritmos genéticos são utilizadas na solução, entre elas o AG de população única, de múltiplas populações, e um algoritmo inspirado no AG “*messy*”, proposto neste trabalho.

Um protótipo do processador de alarmes foi construído para validação do método proposto. Foram realizados testes utilizando uma base de dados de uma subestação real. O desempenho dos algoritmos foi avaliado considerando-se a variação de uma série de parâmetros, incluindo o tamanho das populações, número de eventos simultâneos ocorridos e percentual de ruído no conjunto de alarmes reportados. Os resultados obtidos revelam bom nível de exatidão, apresentando taxas de erro percentual aceitáveis, mesmo para casos ruidosos. O processador de alarmes mostrou-se capaz de operar “em tempo real”, ou seja, foi capaz de produzir o diagnóstico dentro de um intervalo de tempo consideravelmente menor do que a taxa máxima recomendada para a apresentação de informações ao operador.

## PALAVRAS-CHAVE

Processamento Inteligente de Alarmes; Sistemas Elétricos de Potência; Algoritmos Genéticos.

## ÁREA DE CONHECIMENTO

Grande Área: 3.00.00.00-9 - Engenharias  
Área: 3.04.00.00-7 - Engenharia Elétrica  
Sub-área: 3.04.04.00-2 - Sistemas Elétricos de Potência

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)