



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências

Faculdade de Engenharia

Daniel dos Santos Morim

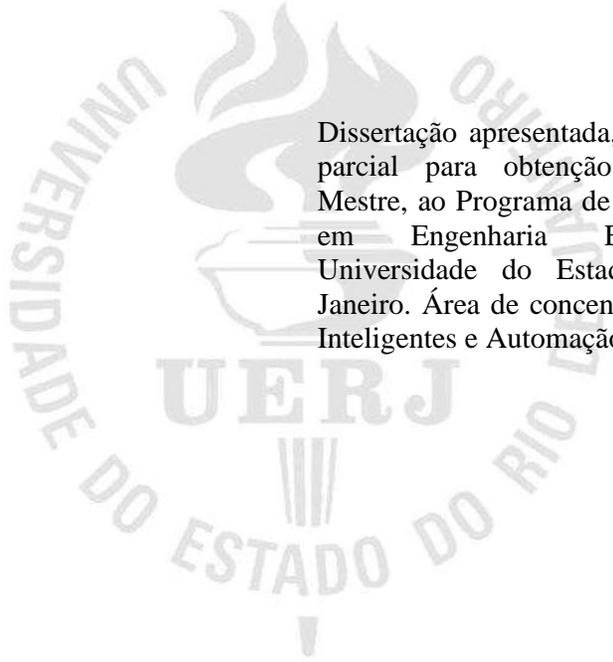
Uso de detectores de dimensões variáveis aplicados na detecção de anomalias através de sistemas imunológicos artificiais

Rio de Janeiro

2009

Daniel dos Santos Morim

Uso de Detectores de Dimensões Variáveis Aplicados na Detecção de Anomalias Através de Sistemas Imunológicos Artificiais



Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós- Graduação em Engenharia Eletrônica da Universidade do Estado do Rio de Janeiro. Área de concentração: Sistemas Inteligentes e Automação.

Orientador: Prof. Dr. Jorge Luís Machado do Amaral

Rio de Janeiro

2009

CATALOGAÇÃO NA FONTE
UERJ/REDE SIRIUS/CTCB

M858 Morim, Daniel dos Santos.

Uso de detectores de dimensões variáveis aplicados na detecção de anomalias através de sistemas imunológicos artificiais./ Daniel dos Santos Morim. – 2009.

92 f.

Orientador : Jorge Luís Machado do Amaral

Dissertação (mestrado) – Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia.

1. Algoritmos genéticos. 2. Análise numérica. I. Amaral, Jorge Luís Machado do. II. Universidade do Estado do Rio de Janeiro. Faculdade de Engenharia. III. Título.

CDU 681.51:556.18

Autorizo, apenas para fins acadêmicos e científicos, a reprodução total ou parcial desta dissertação.

Assinatura

Data

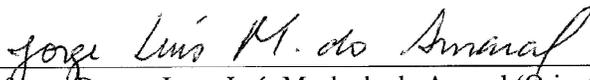
Daniel dos Santos Morim

**Uso de Detectores de Dimensões Variáveis Aplicados na Detecção de Anomalias
Através de Sistemas Imunológicos Artificiais**

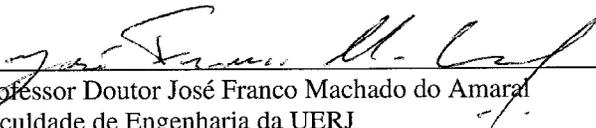
Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós- Graduação em Engenharia Eletrônica da Universidade do Estado do Rio de Janeiro. Área de concentração: Sistemas Inteligentes e Automação.

Aprovado em 15/07/2009

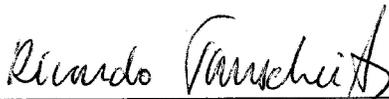
Banca Examinadora:



Professor Doutor Jorge Luís Machado do Amaral (Orientador)
Faculdade de Engenharia da UERJ



Professor Doutor José Franco Machado do Amaral
Faculdade de Engenharia da UERJ



Professor Doutor Ricardo Tanscheit
Departamento de Engenharia Elétrica da PUC-Rio

Rio de Janeiro

2009

DEDICATÓRIA

À minha amada namorada Brunna, aos meus pais e irmão, e à toda a minha família, que sempre me motivaram nos momentos mais difíceis desta trajetória.

AGRADECIMENTOS

À Deus por ter me dado paz e sabedoria para enfrentar todos os problemas;

Aos meus queridos pais, Katia e José Carlos, por não terem deixado faltar nada em minha vida e por toda a paciência;

À minha doce namorada Brunna pelo amor, carinho e apoio intermitentes;

Ao meu irmão, Felipe, por ter me feito sorrir;

Aos meus queridos tios, Dr. Fernando e Dra. Rosa, por toda a ajuda e conhecimento a mim passado com tanta dedicação e carinho;

Aos meus primos, Igor, Fábio, Renata, Paula, Mariana e Renata, Isabel e Júlio por todo o apoio e carinho;

À Universidade do Estado do Rio de Janeiro, minha segunda casa por alguns anos;

Aos funcionários e professores do Programa de Pós-Graduação em Engenharia Eletrônica por todos os ensinamentos a mim passados. Um agradecimento especial ao mais que competente funcionário Alex, que me ajudou muito com toda a parte burocrática;

Ao meu orientador, professor Jorge Amaral, pelos conselhos, opiniões e ensinamentos durante essa jornada, assim como pela paciência e por um inesquecível suco de manga. Estas palavras não são suficientes para agradecer um dos melhores educadores que tive;

À Sutech Engenharia por ter me dado a oportunidade de iniciar este mestrado e trabalhar ao mesmo tempo, particularmente ao Sr. Dr. Eutíquio Calazans;

À Micromed Biotecnologia e a todos seus funcionários por toda a compreensão e por ter me dado a oportunidade de concluir este mestrado, em especial aos diretores Fernando, Renato e Neto, ao gerente Raul e à funcionária Flávia;

Ao professor Márcio e aos alunos do Programa de Engenharia Biomédica da UFRJ pela paciência e ensinamentos a mim passados;

Ao grande amigo de longa data, Dr. Gustavo Medeiros, por me escutar sempre;

Ao colega de graduação e grande amigo, Rafael Fortes, por seus sábios ensinamentos, pelos seus endereços eletrônicos nas manhãs antes das 8h e por me fazer rir sempre;

Ao colega de graduação e empresa e grande amigo, Felipe Henriques, por todo o incentivo e palavras de apoio em todos os momentos;

Aos colegas do mestrado, Marcos, Joaquim e Renato, pela força;

Aos amigos Mike, Jas e Zeke, por terem me feito rir e esquecer meus problemas.

RESUMO

MORIM, Daniel S. *Uso de detectores de dimensões variáveis aplicados na detecção de anomalias através de sistemas imunológicos artificiais*, Brasil. 2009. 92f. Dissertação (Mestrado em Engenharia Eletrônica) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2009.

O presente trabalho investiga um método de detecção de anomalias baseado em sistemas imunológicos artificiais, especificamente em uma técnica de reconhecimento próprio/não-próprio chamada algoritmo de seleção negativa (NSA). Foi utilizado um esquema de representação baseado em hiperesferas com centros e raios variáveis e um modelo capaz de gerar detectores, com esta representação, de forma eficiente. Tal modelo utiliza algoritmos genéticos onde cada gene do cromossomo contém um índice para um ponto de uma distribuição quasi-aleatória que servirá como centro do detector e uma função decodificadora responsável por determinar os raios apropriados. A aptidão do cromossomo é dada por uma estimativa do volume coberto através uma integral de Monte Carlo. Este algoritmo teve seu desempenho verificado em diferentes dimensões e suas limitações levantadas. Com isso, pode-se focar as melhorias no algoritmo, feitas através da implementação de operadores genéticos mais adequados para a representação utilizada, de técnicas de redução do número de pontos do conjunto próprio e de um método de pré-processamento baseado em bitmaps de séries temporais. Avaliações com dados sintéticos e experimentos com dados reais demonstram o bom desempenho do algoritmo proposto e a diminuição do tempo de execução.

Palavras-Chave: Sistemas Imunológicos Artificiais. Seleção Negativa. Detecção de Anomalias

ABSTRACT

This work investigates a novel detection method based on Artificial Immune Systems, specifically on a self/non-self recognition technique called negative selection algorithm (NSA). A representation scheme based on hyperspheres with variable center and radius and a model that is able to generate detectors, based on that representation scheme, have been used. This model employs Genetic Algorithms where each chromosome gene represents an index to a point in a quasi-random distribution, which serves as a detector center, and a decoder function that determines the appropriate radius. The chromosome fitness is given by an estimation of the covered volume, which is calculated through a Monte Carlo integral. This algorithm had its performance evaluated for different dimensions, and more suitable genetic operators for the used representation, techniques of reducing self-points number and a pre-processing method based on bitmap time series have been therefore implemented. Evaluations with synthetic data and experiments with real data demonstrate the performance of the proposed algorithm and the decrease in execution time.

Keywords: Artificial Immune Systems. Negative Selection. Novelty Detection

LISTA DE FIGURAS

Figura 1 - Conceito básico do Algoritmo de Seleção Negativa.....	32
Figura 2 - Algoritmo de Seleção Negativa com representação real.....	35
Figura 3 – Comparação entre ambos os princípios de seleção	36
Figura 4 – Comparação entre o algoritmo de detectores de tamanho fixos e variáveis.....	37
Figura 5 - A função Peaks.....	39
Figura 6 – A decomposição Quadtree da função PEAKS	39
Figura 7 – Fluxograma para particionamento do detector	41
Figura 8 – Funcionamento de um Algoritmo Genético	44
Figura 9 – Exemplo da codificação utilizada.....	45
Figura 10 – Cálculo do raio em função do parâmetro α	46
Figura 11 – Posicionamento dos detectores.....	47
Figura 12 - Conjuntos de dados	51
Figura 13 – Resultados para o conjunto da cruz	53
Figura 14 - Resultados para o conjunto do triângulo.....	54
Figura 15 – Exemplos de Funcionamento do Operador de Seleção com Algoritmo de Poda..	61
Figura 16 – Comparação entre algoritmo original e algoritmo de poda de detectores	62
Figura 17 – Ponto ideal para a redução do conjunto próprio.....	64
Figura 18 – Seleção negativa utilizando algoritmo original (1000 pontos).....	71
Figura 19 – Seleção positiva utilizando algoritmo original (1000 pontos).....	72
Figura 20 – Seleção positiva utilizando algoritmo original (84 grupos)	73
Figura 21 – Circuito do filtro Sallen Key	76
Figura 22 – Representação SAX.....	77
Figura 23 – Geração do <i>Bitmap</i>	77
Figura 24 – Circuito do filtro universal	79
Figura 25 – Circuito do filtro Leapfrog	80

LISTA DE TABELAS

Tabela 1 – Volume Coberto no Teste de Ordenação	52
Tabela 2 – Resultados do algoritmo original	55
Tabela 3 – Comparação entre o algoritmo original e o algoritmo com os operadores genéticos mais adequados	59
Tabela 4 – Comparação entre o algoritmo original e o algoritmo com poda de detectores	62
Tabela 5 – <i>k-means</i> com raio ao elemento mais próximo.....	65
Tabela 6 – <i>k-means</i> com raio pelo centro mais próximo	65
Tabela 7 – <i>k-means</i> com limiar 0.68.....	65
Tabela 8 – <i>k-means</i> com limiar 0.95.....	65
Tabela 9 – Volume do espaço não-próprio para cada dimensão	67
Tabela 10 – Resultados do algoritmo original	68
Tabela 11 – Resultados do algoritmo com melhorias	68
Tabela 12 – Resultados para algoritmo original	69
Tabela 13 – Resultados para algoritmo com melhorias	69
Tabela 14 – Resultados do algoritmo original utilizando representação positiva e negativa (1000 pontos)	70
Tabela 15 – Resultados do algoritmo com melhorias utilizando representação positiva e negativa (1000 pontos).....	71
Tabela 16 – Resultados do algoritmo original utilizando representação positiva e negativa (84 grupos).....	72
Tabela 17 – Resultado do algoritmo original.....	74
Tabela 18 – Resultado do algoritmo com melhorias	74
Tabela 19 – Resultados do algoritmo genético para filtro Sallen Key	78
Tabela 20 – Resultados do classificador para filtro Sallen Key	78
Tabela 21 – Resultados do algoritmo genético para filtro universal	79
Tabela 22 – Resultados do classificador para filtro universal	80
Tabela 23 – Resultados do algoritmo genético para filtro Leapfrog	81
Tabela 24 – Resultados do classificador para filtro Leapfrog	81
Tabela 25 – Resultados obtidos para experimento com rolamentos.....	82
Tabela 26 – Resultados obtidos para experimento com eletrocardiograma	83

SUMÁRIO

INTRODUÇÃO.....	12
1 TÉCNICAS PARA DETECÇÃO DE ANOMALIAS.....	15
1.1 <i>Introdução.....</i>	15
1.2 <i>Abordagens Estatísticas.....</i>	16
1.2.1 Método Paramétrico.....	16
1.2.2 Método Não-Paramétrico.....	18
1.3 <i>Abordagens Baseadas em Redes Neurais.....</i>	20
1.3.1 Abordagens Através de MLP.....	21
1.3.2 Abordagens Através de Máquinas de Vetores de Suporte.....	22
1.3.3 Abordagens Através da Teoria da Ressonância Adaptativa.....	23
1.3.4 Abordagens Através de RBF.....	23
1.3.5 Abordagens de Redes Auto-Associativas.....	24
1.3.6 Abordagens Através de Redes de Hopfield.....	24
1.3.7 Abordagens Através de SOM.....	25
2 SISTEMAS IMUNOLÓGICOS ARTIFICIAIS.....	26
2.1 <i>Introdução.....</i>	26
2.2 <i>Visão Geral do Sistema Imunológico.....</i>	26
2.2.1 Mecanismos Inatos ou Não Específicos.....	27
2.2.2 Sistema Imunológico Adaptativo.....	27
2.3 <i>Sistemas Imunológicos Artificiais.....</i>	28
2.3.1 Redes Imunológicas Artificiais.....	29
2.3.2 Algoritmo de Seleção Clonal.....	30
2.3.3 Algoritmo de Seleção Negativa.....	30
2.4 <i>Estudo Detalhado na Geração de Detectores.....</i>	38
3 MODELO UTILIZADO NA GERAÇÃO DE DETECTORES.....	43
3.1 <i>Descrição do Método.....</i>	43
3.2 <i>Identificação de Limitações do Método.....</i>	51
3.2.1 Ordenação dos Detectores.....	51
3.2.2 Variação na Sobreposição.....	52
3.2.3 Comportamento em Altas Dimensões.....	55
3.3 <i>Otimizações do Método.....</i>	56
3.3.1 Operadores Genéticos Mais Adequados.....	56
3.3.2 Operadores de Seleção de Detectores.....	59
3.3.3 Redução do Número de Pontos do Conjunto Próprio.....	63
4 ESTUDOS DE CASOS.....	66
4.1 <i>Avaliações.....</i>	66
4.1.1 Desempenho do Algoritmo.....	67
4.1.2 Métodos de Redução do Conjunto Próprio.....	69

4.1.3	Comparação entre Métodos de Seleção Positiva e Negativa.....	70
4.2	<i>Estudo de Casos</i>	74
4.2.1	Filtro Sallen Key.....	75
4.2.2	Filtro Universal.....	78
4.2.3	Filtro Leapfrog.....	80
4.2.4	Detecção de Anomalias em Rolamentos.....	81
4.2.5	Detecção de Anomalias em Eletrocardiogramas.....	82
5	CONCLUSÃO	84
	REFERÊNCIAS	86

INTRODUÇÃO

O desenvolvimento de estratégias para a detecção de anomalias é ainda fortemente dependente da perícia dos engenheiros e do conhecimento que se tem sobre as características operacionais do sistema. Como resultado, a detecção e diagnóstico de anomalias ainda é um processo interativo e lento. Uma investigação na área mostra que, nas últimas décadas, uma boa quantidade de pesquisas sobre detecção e diagnóstico de anomalias e falhas tem-se concentrado no desenvolvimento de ferramentas inteligentes para tornar a tarefa de detecção mais fácil. Embora tenha havido alguns desenvolvimentos importantes, estas novas tecnologias não têm sido amplamente aceitas (Fenton, 2001). Isto deve motivar os investigadores a procurar outros modelos e desenvolver novas estratégias de detecção e diagnóstico de anomalias.

O Sistema Imunológico é um dos sistemas mais complexos que existem na natureza e sua complexidade pode ser comparada à do Sistema Nervoso (Varela et al., 1988). A principal função do Sistema Imunológico é fornecer proteção ao organismo contra uma infinidade de agentes patogênicos (vírus, bactérias, etc) (Hofmeyr, 1997). A grande diversidade de agentes patogênicos a ser combatida torna esta tarefa extremamente difícil e exige que o Sistema Imunológico utilize uma variedade de mecanismos que se completam e se sobrepõem, tornando-o bastante complexo. Dependendo do tipo de agressor, da maneira como invade o corpo e do dano que causa, o sistema imunológico usa diferentes mecanismos para destruir o invasor ou para neutralizar os seus efeitos. De fato, muitos destes mecanismos ainda hoje não são totalmente compreendidos (Janeway, 1993). Por outro lado, esta complexidade pode ser uma fonte de inspiração bastante frutífera para a computação inspirada na natureza.

Os sistemas imunológicos artificiais (Castro, 2001) são inspirados no funcionamento do sistema imunológico humano no que diz respeito à criação de soluções inovadoras para resolver problemas. Alguns algoritmos apresentados são implementações bem sucedidas de sistemas imunológicos artificiais: a rede imunológica, o Algoritmo de Seleção Clonal e o Algoritmo de Seleção Negativa (Castro & Timmis, 2002), (Dasgupta & Gonzalez, 2003), (Forrest et al, 1994).

Como foco desta pesquisa, o Algoritmo de Seleção Negativa pode ser aplicado em diversas áreas e é especialmente útil em detecção de anomalias. Este algoritmo faz uso da propriedade dos sistemas imunológicos de distinguir as células estranhas ao corpo (não-próprias) das células que pertencem ao corpo (próprias) (Forrest et al, 1994). Essa distinção é

feita através de um conjunto de detectores, sendo primeiramente gerados candidatos aleatoriamente e consecutivamente descartando aqueles que reconhecem os dados como próprios (conjunto *próprio*). Posteriormente, estes detectores podem ser utilizados para apontar padrões normais e anormais de um sistema, fornecendo assim um mecanismo de detecção de anomalias.

Apesar do algoritmo de Seleção Negativa ter sido utilizado com sucesso em aplicações de detecção de falhas (Amaral et al., 2004) (Dasgupta et al., 2004), é desejável que algoritmos de geração dos conjuntos de detectores sejam continuamente melhorados.

Neste trabalho, são propostas alterações no método de geração de detectores utilizados de Algoritmos de Seleção Negativa existentes com o objetivo de implementar um sistema imunológico artificial voltado para a detecção de anomalias, principalmente em circuitos eletrônicos. Assim sendo, a investigação de alguns algoritmos de Seleção Negativa (Ji e Dasgupta, 2004), (Ji e Dasgupta, 2005), (Amaral et al, 2006) e (Amaral, 2007) e a identificação dos principais problemas deste último algoritmo para a aplicação desejada com enfoque especial na escalabilidade, na escolha dos parâmetros de controle, na complexidade computacional e na modelagem de dados, assim como a utilização de pré-processamento dos dados de entrada, se tornam pontos importantes deste estudo.

Foram estudadas algumas técnicas utilizadas para gerar os detectores, em particular o método proposto por Gonzalez, o Algoritmo de Seleção Negativa com representação real (RNSA) (Gonzalez et al, 2002), no qual os detectores são hipersferas cujos raios são fixos e determinados antes da execução do algoritmo de geração de detectores.

Posteriormente ao estudo do RNSA foi analisado um algoritmo com a mesma representação real, porém que utiliza um sistema de representação baseado em hipersferas de centro e raio variáveis, proposto por Amaral (2006a). Este método faz uso de algoritmos genéticos para gerar conjuntos de detectores através de sucessivas evoluções. Cada cromossomo representa um conjunto de detectores e a função de avaliação dá a porção de espaço não-próprio que é coberto por estes conjuntos.

Por sua vez, este algoritmo, chamado nesta dissertação de algoritmo original, teve seu desempenho verificado em diferentes dimensões e foram levantadas algumas de suas limitações.

Para gerar os detectores, um conjunto de pontos que representa o funcionamento normal é utilizado. Uma vez que se deve testar que o detector gerado não invada o conjunto próprio, quanto maior o número de pontos próprios maior será o tempo necessário para testar este detector, sugerindo que um pequeno número de pontos próprios representativos deva ser

encontrado. Para atingir este objetivo foram desenvolvidos métodos de redução do conjunto próprio, constituindo a primeira implementação para aperfeiçoar o desempenho do algoritmo.

A segunda implementação consiste em utilizar operadores genéticos de cruzamento e mutação mais adequados à representação utilizada. Com isso, ambas as implementações compõem uma contribuição para o avanço da pesquisa na área de detecção de anomalias utilizando sistemas inteligentes.

Após as melhorias propostas terem sido implementadas e avaliadas em conjuntos de dados sintéticos, foi ainda executado um estudo de casos envolvendo diferentes aplicações. Os casos estudados foram: a detecção de falhas em três filtros ativos, a detecção de falhas em rolamentos e em eletrocardiogramas.

Esta dissertação possui cinco capítulos, cujos conteúdos são descritos a seguir.

O Capítulo 1 apresenta uma visão geral das diferentes abordagens utilizadas para a detecção e diagnósticos de falhas.

O Capítulo 2 apresenta de forma sucinta um material de apoio em imunologia e discute alguns conceitos necessários à compreensão deste trabalho. São apresentados os principais modelos usados em Sistemas Imunológicos Artificiais: Redes Imunológicas, Algoritmo de Seleção Clonal e o Algoritmo de Seleção Negativa, e são mencionados alguns artigos relevantes. Esta pesquisa procura demonstrar as potencialidades dos Sistemas Imunológicos Artificiais e fornece subsídios para o trabalho desenvolvido.

O Capítulo 3 descreve o modelo proposto para geração do conjunto de detectores segundo a representação proposta, assim como suas limitações e melhorias realizadas.

No Capítulo 4 são apresentados alguns estudos para avaliar o desempenho do algoritmo melhorado. Também é apresentado o estudo de casos, com a utilização de casos reais, e efetuados comentários sobre os resultados obtidos.

A conclusão deste trabalho é apresentada no Capítulo 5 juntamente com uma relação de possíveis trabalhos futuros.

1 TÉCNICAS PARA DETECÇÃO DE ANOMALIAS

1.1 Introdução

A detecção de anomalias consiste na identificação de padrões em um conjunto de dados que não estão de acordo com o padrão normal estabelecido. Estes padrões detectados são chamados de anomalias e normalmente representam uma informação crítica sobre a qual deve-se tomar alguma ação (Chandola, 2009). Por exemplo, anomalias encontradas nos padrões de tráfego em redes de computadores podem indicar que a rede está sofrendo algum tipo de ataque e que medidas de proteção devem ser tomadas. Do mesmo modo, quando se encontra uma anomalia no sinal de saída de um circuito eletrônico, isto pode significar que o circuito apresenta problemas e deve ser reparado. Na literatura, as anomalias também podem ser chamadas de *outliers*, desvios, peculiaridades, aberrações, etc (Chandola, 2009).

Diversos estudos sobre detecção de anomalias foram realizados e este assunto ainda é bastante discutido. Por esta razão existem diversos modelos de detecção de anomalias que têm apresentado bons desempenhos em diferentes tipos de dados, como em sinais de eletrocardiogramas (Akhavan et al., 1998; Chuah, 2007) ou em redes IP (Thottan, 2003).

Não há um modelo único para a melhor detecção e seu sucesso depende não somente do tipo de método utilizado, como também das propriedades dos dados manipulados.

Existem três grandes categorias de técnicas de detecção de anomalias (Chandola, 2009); as supervisionadas, as semi-supervisionadas e as não supervisionadas. Nas técnicas supervisionadas, utiliza-se um classificador treinado com exemplos pertencentes à classe normal e à classe anormal, para indicar se um ponto de teste é normal ou não. Neste tipo de técnica só é possível detectar anomalias que foram usadas no treinamento do classificador. Nas técnicas semi-supervisionadas, constrói-se um modelo representando o modelo normal a partir de um conjunto de dados de treinamento e então é testada a probabilidade do ponto de teste ser gerado por este modelo. Nas técnicas não supervisionadas, a detecção de anomalias em um conjunto de dados ocorre a partir da hipótese de que a maioria dos dados seja normal; caso contrário haverá uma alta taxa de falsos alarmes.

A seguir serão apresentadas algumas técnicas de detecção de anomalias e uma breve revisão bibliográfica sobre suas aplicações.

1.2 Abordagens Estatísticas

As abordagens estatísticas são, em grande parte, baseadas em modelagens de dados estatísticos e na utilização desta informação para estimar se as amostras de teste são provenientes da mesma distribuição ou não (Markou et al., 2003a).

Uma abordagem bastante simples pode ser baseada na construção de uma função densidade de probabilidade para dados de uma classe conhecida e, então, calcular a probabilidade de uma amostra de pertencer a essa classe.

Existem dois métodos principais para a estimativa de uma função densidade de probabilidade: o paramétrico e o não-paramétrico.

No método paramétrico supõe-se que as distribuições das variáveis são feitas de acordo com distribuições probabilísticas usuais, e certos parâmetros do modelo são calculados para adequarem a estas distribuições. Entretanto, existem casos em que a forma geral da função densidade de probabilidade associada aos dados não é conhecida. A alternativa nestes casos consiste na utilização dos chamados métodos não-paramétricos, onde a forma da função densidade e seus parâmetros são obtidos a partir dos dados. Isso faz com que este método seja mais flexível, porém, mais custoso computacionalmente.

1.2.1 Método Paramétrico

O método paramétrico faz uma suposição de que os dados são distribuições gaussianas e podem ser modelados estatisticamente com base na média e covariância dos mesmos. As taxas de erro e de rejeição são comumente usadas para descrever o nível de desempenho de um sistema de reconhecimento de padrões.

Alguns métodos avançados de detecção serão apresentados, como o modelo de misturas gaussianas (GMM), modelos ocultos de Markov (HMM) e teste de hipóteses.

1.2.1.1 Modelo de Misturas Gaussianas

Modelos de mistura finita e seus métodos típicos de estimativa podem aproximar uma ampla gama de funções densidade de probabilidade e são, portanto, soluções atraentes para os casos onde funções de formas únicas, tais como uma distribuição normal, falham. Geralmente, a função de distribuição básica pode ser de qualquer tipo, mas a distribuição normal multivariada, a distribuição gaussiana, é sem dúvida uma das mais famosas e úteis em distribuições estatísticas, desempenhando um papel predominante em diversas áreas de aplicações (Paalanen et al., 1995).

Entre estas aplicações pode-se citar a utilização do GMM como classificador em um sistema de detecção de falhas em motores a gás. Neste sistema foram utilizadas, como conjunto de treinamento, amostras de 31 tipos de falha, onde cada amostra foi composta de valores monitorados de 9 sensores (Donat et al., 2007).

1.2.1.2 Modelos Ocultos de Markov

Um modelo de Markov é um sistema de transições de estados, onde a probabilidade do sistema estar em um certo estado futuro depende apenas do estado atual do sistema. As transições de estados são controladas por processos estocásticos para formar cadeias de Markov.

Em um modelo de Markov normal, o estado é diretamente visível para o observador, de modo a que a probabilidade de transição entre os estados são os únicos parâmetros. Num modelo oculto de Markov (HMM), o estado não é diretamente visível, mas sim suas variáveis, que são influenciadas pelo estado corrente. Cada estado possui uma distribuição de probabilidade sobre cada saída, conseqüentemente, a seqüência de saídas geradas por um HMM fornece algumas informações sobre a seqüência de estados.

Este tipo de método pode ser visto em aplicações nas áreas de detecção de intruso (Qiao et al., 2002) e reconhecimento de padrão para monitoramento de falhas de sistemas dinâmicos (Smyth, 1993). Em Li et al. (2005) utiliza-se um modelo HMM para diagnóstico de falhas em processos de aceleração e desaceleração de máquinas rotativas.

1.2.1.3 Teste de Hipóteses

Um teste de hipóteses é um método estatístico para tomada de decisões usando dados experimentais. Esta técnica estatística consiste em determinar se a amostra para ensaio provém ou não da mesma distribuição dos dados de treinamento.

O teste de hipóteses é encontrado como parte do algoritmo de detecção de falhas, como em circuitos integrados lineares (Epstein et al., 1993) e em sistemas de potência (Chowdhury et al.; 1991).

1.2.2 Método Não-Paramétrico

Nos métodos não-paramétricos, a forma da função densidade de probabilidade e seus parâmetros são obtidos a partir dos dados. Para a abordagem não-paramétrica serão considerados quatro métodos. São eles: método baseado no vizinho mais próximo (kNN), método de Parzen, métodos baseados em *strings* e em *clusters*.

1.2.2.1 Método kNN

O algoritmo kNN é utilizado para calcular a função densidade de probabilidade de dados (Markou et al., 2003a). Sua abordagem utiliza o volume da região que contém o vizinho mais próximo de um ponto para estimar o valor da densidade de probabilidade neste ponto (Fukunaga, 1975).

Na área de detecção de anomalias, o kNN determina se um ponto reside em uma região do espaço de busca, calculando as distâncias para o vizinho mais próximos do ponto. Em densas regiões, por exemplo, um ponto terá muitos vizinhos próximos a ele e, com isso, a distância será menor. Com isso, o algoritmo identifica os vizinhos mais próximos e mais distantes deste ponto e, assim, avalia estes vizinhos como anomalia ou não.

Como é um método muito utilizado, o kNN pode ser facilmente encontrado na área de detecção de anomalias de circuitos eletrônicos analógicos, como filtros universal e Sallen Key (Amaral, 2006); e detecção de falhas em motores de indução, através da extração de características de tensão e corrente medidos (Ondel et al., 2006).

1.2.2.2 Método de Parzen

O método da Janela de Parzen (Parzen, 1972) é outro método não-paramétrico para estimar a função densidade de probabilidade (Stibor et al., 2005) a partir dos dados rotulados. A classificação é realizada, portanto, a partir dessa densidade estimada. É utilizada uma variável aleatória que determina uma janela fixa de largura determinada. Em outras palavras, dada uma entrada, os rótulos são observados entre os dados concentrados na janela e é atribuído o rótulo de maior densidade local a esta entrada.

Usualmente, utiliza-se a distribuição gaussiana (Abe et al., 2006) em torno de cada um dos dados de treinamento, pois a mesma possui numerosas propriedades estatísticas.

É comum autores se basearem em estimar a densidade de probabilidade dos dados de treinamento e padrões de rejeição, fazendo, assim, a detecção de anomalias (Markou et al., 2003a); como em Bishop (1994), onde foi utilizado a Janela de Parzen como classificador em dados de fluxo de óleo em dutos.

1.2.2.3 Abordagens Baseadas em *Strings*

Este método é baseado em tratamento de dados como modelos representados por uma *string* (vetor de recursos) e, em seguida, calculando medidas de dissimilaridade entre dados de treinamento e teste (Markou et al., 2003a).

Assim como neste trabalho, Forrest et al. (1994) apresentam um método para resolver o problema da distinção entre o funcionamento normal e anormal de um sistema usando um algoritmo de detecção de mudança, baseado na maneira como o sistema imunológico humano se comporta realizando a mesma tarefa.

1.2.2.4 Abordagens Baseadas em *Clusters*

Esse tipo de abordagem tem por objetivo a partição de dados em uma série de *clusters*, onde um grau de inclusão é associado a cada padrão. Se o grau de inclusão ultrapassa um limiar, sugere-se que um padrão de dados pertence ou não a um *cluster*; a anomalia pode ser detectada quando uma amostra não pertence a nenhuma das categorias disponíveis (Markou et al., 2003a).

1.3 Abordagens Baseadas em Redes Neurais

As Redes Neurais Artificiais são inspiradas no funcionamento do cérebro humano e, nas últimas décadas, têm se mostrado um importante paradigma de programação. Esta área foi desenvolvida para resolver problemas de reconhecimento de padrões, como processamento de imagens e voz, que eram insolúveis ou demasiadamente custosos computacionalmente quando implementados usando os tradicionais métodos de computação (Barbosa et al., 2000).

As redes neurais se baseiam em unidades de processamento simples (neurônios) densamente interconectadas. Estes padrões de interconexão podem variar desde redes *feed-forward* de uma única camada, como redes perceptron, até topologias mais complexas, formadas por várias camadas com retropropagação de erros como no caso das Perceptrons com várias camadas (*Multilayer Preceptrons*), ou mesmo redes recorrentes que apresentam um comportamento dinâmico, como as Redes de Hopfield.

As RNAs têm sido amplamente utilizadas para a detecção de anomalias. Comparadas aos métodos estatísticos, algumas questões na detecção de anomalias são mais críticas para redes neurais, tais como sua capacidade de generalização, custo computacional durante o treinamento e custo computacional quando a rede precisar ser retreinada. Por sua vez, elas têm a vantagem de que um número muito reduzido de parâmetros deve ser otimizado para o treinamento da rede e que não é preciso fazer nenhuma consideração a priori a respeito das propriedades dos dados (Markou et al., 2003b).

Os principais critérios de avaliação de detecção de anomalias através das RNA é a maximização na detecção das verdadeiras falhas minimizando, ao mesmo tempo, os falsos positivos.

Diferentes métodos e arquiteturas são empregados para a detecção de anomalias por das RNA. A seguir serão descritas alguns destes métodos: perceptrons de múltiplas camadas (MLP), mapas auto-organizáveis (SOM), teoria da ressonância adaptativa (ART), redes de funções de base radial (RBF), máquinas de vetores de suporte (SVM) e redes de Hopfield (Markou et al., 2003b).

1.3.1 Abordagens Através de MLP

As redes perceptrons de múltiplas camadas (MLP) são as mais conhecidas e largamente utilizadas classes de redes neurais. Possuem uma camada de entrada, pelo menos uma camada intermediária e uma camada de saída, onde a quantidade de neurônios em cada camada depende do problema a ser resolvido. As camadas são interligadas por conexões entre seus neurônios, ou seja, um neurônio de uma camada é conectado a todos da outra camada quando a rede é completamente conectada.

Como essas redes não geram limites entre classes, conceber métodos de detecção de anomalias é uma tarefa bastante desafiadora, especialmente no que tange a garantir que a generalização da rede não interfira com a sua capacidade de detecção (Markou et al., 2003b).

Em Börner et al. (2002) é abordada a detecção de falhas na suspensão de veículos através de uma rede MLP, entre outros métodos. Foi utilizada uma rede com duas camadas escondidas e função sigmóide como função de ativação em todas as camadas. O número de neurônios na saída foi três, uma vez que oito classes precisavam ser distinguidas. Em cada uma das camadas escondidas foram utilizados cinco neurônios. O treinamento foi realizado utilizando o algoritmo de Levenberg-Marquardt (Ranganathan, 2004). Após o treinamento, todos os conjuntos de dados foram classificados corretamente e o erro de generalização foi inferior a 4%.

Em Maidon et al. (1997) é demonstrado como várias falhas nos circuitos integrados analógicos podem ser diagnosticadas a partir da magnitude dos harmônicos de Fourier do espectro da resposta do circuito a um sinal de entrada senoidal, usando uma rede MLP de duas camadas para classificar as respostas às correspondentes falhas.

Em Ye et al. (2001) é tratado o diagnóstico de falhas mecânicas em motores de indução com base em redes neurais do tipo MLP. Esta rede é constituída de quatro camadas: uma de entrada, duas camadas escondidas, e uma camada de saída.

1.3.2 Abordagens por Máquinas de Vetores de Suporte

As máquinas de vetores de suporte (SVM) são baseadas no conceito de determinação ótima de hiperplanos para separar os dados provenientes em diferentes classes (Markou et al., 2003b).

As SVMs encaram o problema de classificação como um problema de otimização quadrática, combinando controle de generalização com uma técnica para evitar o excesso de dimensões do problema, colocando um limite superior sobre a margem entre as diferentes classes, tornando-se uma ferramenta prática e dinâmica para grandes conjuntos de dados. As SVMs classificam os dados, determinando um conjunto de vetores, que são membros do conjunto de entradas de treinamento que esboçar um hiperplano no espaço.

As SVMs são baseadas na idéia de minimização de risco estrutural, que minimiza o erro de generalização. O número de parâmetros livres utilizado pelas SVMs depende da margem que separa os pontos de dados e não do número de entradas, a fim de evitar *overfitting*. As SVMs fornecem um mecanismo genérico para ajustar a superfície do hiperplano aos dados através da utilização de uma função kernel. O usuário pode fornecer uma função (linear, polinomial ou curva sigmóide) às SVMs durante o processo de formação, onde são selecionados os vetores de suporte ao longo da superfície dessa função. Esta capacidade permite classificar um vasto leque de problemas (Mukkamala et al., 2002).

Em Sun et al. (2007) é apresentado um método de diagnóstico de falhas baseado em SVM aplicado a circuitos analógicos. Como exemplo foi utilizado um filtro biquadrático, obtendo taxas de acerto que variam de 95,5% a 98,5%.

1.3.3 Abordagens Através da Teoria da Ressonância Adaptativa

Uma característica fundamental dos sistemas baseados na teoria da ressonância adaptativa (ART) é o processo de correspondência de padrões, cuja função é comparar uma entrada externa com a memória interna de um código ativo. A correspondência ART leva a um estado ressonante, onde permanece tempo suficiente para permitir a aprendizagem, ou uma busca na memória. Se a pesquisa termina em um código estabelecido, a representação da memória pode permanecer a mesma, ou incorporar novas informações a partir de trechos correspondentes à entrada atual. Este recurso, baseado no processo de aprendizagem, é a base da estabilidade do sistema ART (Carpenter, 2002).

A metodologia ART tem se mostrado eficaz para gerar classificadores para a detecção de anomalias, inclusive sendo superiores a outros classificadores, como o SOM (Markou et al., 2003b).

Uma aplicação de diagnóstico de falhas em turbinas de gás é abordada em Qingyang et al. (2008). Já em Amini (2004), são utilizadas redes ART na detecção de anomalias para detecção de intrusos.

1.3.4 Abordagens Através de RBF

As redes RBF podem ser vistas como redes neurais *feedforward* com uma única camada escondida. Cada unidade escondida é uma função não-linear (normalmente Gaussiana), com saída relacionada à distância entre os vetores de entrada e do centróide da função base. A camada de saída forma uma combinação linear que calcula a soma ponderada dos resultados das unidades escondidas (Mak, 1993).

Em Catelani (2002) foi utilizado uma rede RBF como parâmetro de comparação na detecção de falhas em circuitos analógicos. Foram considerados um filtro passa-baixa e um segundo circuito composto por dois filtros ativos passa-baixa, um filtro ativo passa-alta e um amplificador somador. Já em Mohammadi et al. (2002), os circuitos utilizados foram um circuito puramente resistivo e um amplificador com JFET.

1.3.5 Abordagens de Redes Auto-Associativas

As redes auto-associativas (AAN) são formadas para reconstruir suas entradas na camada de saída. Utilizando apenas os dados normais (sem anomalias) para treinamento, elas aprendem um mapeamento de reconstrução. Com isso, os dados normais são reconstruídos e os dados que possuem anomalias podem ser eliminados facilmente, já que produzem um erro de reconstrução muito maior.

Exemplos de redes AAN aplicadas à detecção de anomalias incluem detecção do batimento ectópico no eletrocardiograma (ECG), previsão de falhas em motores e detecção de danos em estruturas. Além disso, as redes AAN foram utilizadas em uma gama de aplicações onde a detecção de anomalia por reconhecimento de padrão é parte integrante do sistema, incluindo verificação de orador, detecção de faces e reconhecimento de sinais de rádio (Iversen et al., 2006).

Em Chan et al. (1999) é descrito o uso de uma rede neural auto-associativa para detecção de anomalia nos cabos da ponte Tsing Ma, localizada em Hong Kong, por medições das frequências dos cabos.

1.3.6 Abordagens Através de Redes de Hopfield

A Rede de Hopfield funciona como a memória humana, ou seja, como uma memória endereçada pelo conteúdo. Muitas vezes uma pessoa pode se lembrar de fatos inteiros tendo inicialmente uma pequena lembrança do acontecimento (Silva, 2003). O mesmo acontece com a rede neural de Hopfield, que possui uma única camada de neurônios realimentados e, deste modo, implementa uma memória auto-associativa. Isto significa que, ao ser apresentado a um determinado padrão, a rede retorna um padrão armazenado lhe é mais próximo (associado).

Em Crook et al. (2002), onde dois robôs fazem tarefas de inspeção utilizando um sonar e imagens de câmeras, foi utilizado o método de Hopfield para detecção de anomalias.

1.3.7 Abordagens Através de SOM

O mapa auto-organizável (SOM) é um modelo de rede neural de aprendizagem não-supervisionada que utiliza uma regra de aprendizagem competitiva. Este modelo consiste em modificar repetidamente os pesos sinápticos de uma rede neural em resposta a padrões de ativação de acordo com regras pré-estabelecidas, até que se desenvolva uma configuração final. Essa resposta é composta de um mapa dos padrões de entrada, sem que, a priori, houvesse informações de correlação entre os dados de entrada e saída.

Em Barreto et al. (2004) foi proposta uma nova abordagem para a detecção e diagnóstico de falhas em redes celulares de terceira geração (3G) utilizando algoritmos neurais competitivos, entre eles o SOM.

A detecção de vazamentos em oleodutos é outro exemplo de detecção de falha utilizando SOM, assim como a detecção precoce de falhas em máquinas rotativas mecânicas, com o intuito de aumentar a segurança do sistema e diminuir os custos manutenção (Ypma et al., 1998).

2 SISTEMAS IMUNOLÓGICOS ARTIFICIAIS

2.1 Introdução

O sistema imunológico biológico é um sistema adaptativo complexo que evoluiu nos vertebrados para protegê-los de agentes patogênicos invasores. Para realizar essas tarefas, o sistema imunológico desenvolveu mecanismos sofisticados de reconhecimento e resposta que se completam e se sobrepõem, tornando-o bastante complexo. Dependendo do tipo de agressor, da maneira como invade o corpo e do dano que causa, o sistema imunológico usa diferentes mecanismos ou para destruir o invasor ou para neutralizar seus efeitos (Dasgupta, 2006).

Esta característica de fornecer proteção ao organismo contra uma infinidade de agentes patogênicos dá origem ao interesse de pesquisadores das áreas de Engenharia de Ciência da Computação.

2.2 Visão Geral do Sistema Imunológico

O sistema imunológico possui a finalidade de proteger o organismo contra infecções. Um importante mecanismo envolve um conjunto de proteínas (anticorpos), que se ligam a potenciais fontes de infecção (antígenos), por mecanismo de afinidade. No sistema biológico, anticorpos competem por antígenos, e o vencedor, o mais apto a neutralizar este antígeno, é replicado. Assim, um grande número de cópias do anticorpo mais apto se liga aos antígenos, possibilitando sua remoção do sistema.

Existem dois tipos de mecanismos de defesa: os inatos ou não específicos, como a proteção da pele; e o sistema imunológico adaptativo, como a ação dos linfócitos e a sua produção de anticorpos específicos. É no sistema imunológico adaptativo que se concentra o foco desta pesquisa.

2.2.1 Mecanismos Inatos ou Não Específicos

O sistema inato é composto pelos mecanismos que defendem o organismo de forma não específica contra um invasor, respondendo da mesma forma, qualquer que seja ele. Este sistema se divide em barreiras físicas, fagócitos, sistema complemento e resposta inflamatória.

Entre as barreiras físicas destaca-se a pele como principal barreira. Sua superfície é constituída de células mortas ricas em queratina, uma proteína fibrilar, que impede a entrada de microorganismos.

Outro tipo de mecanismo inato são os fagócitos, que são células com a propriedade de englobar e digerir partículas ou microorganismos invasores por meio de um processo chamado fagocitose, fazendo deste mecanismo uma importante forma de defesa contra infecções bacterianas.

Já o sistema complemento consiste num grupo de proteínas produzidas pelo fígado e sua ativação é feita através dos anticorpos. Estas proteínas causam a destruição das células através de um processo denominado lise, caracterizado pela ruptura ou dissolução da parede bacteriana, levando à morte celular.

Quando uma região do corpo é lesada pelo ataque de microorganismos, ocorre uma resposta defensiva conhecida como inflamação. Os principais sintomas da inflamação são dor, calor, vermelhidão e inchaço no local atingido. Apesar dos sintomas, esta resposta inflamatória é um processo benéfico, que tem por função destruir ou remover os agentes causadores da lesão, além de reparar ou substituir o tecido danificado.

2.2.2 Sistema Imunológico Adaptativo

O sistema imunológico adaptativo se concentra na capacidade de as células imunologicamente competentes fazerem a distinção entre as proteínas produzidas pelas células do próprio corpo (antígenos "self" - ou seja, do próprio organismo) e as proteínas produzidas por invasores ou pelas células humanas sob o controle do invasor (antígenos "non-self" - ou seja, que não são reconhecidos como sendo do próprio organismo, ou ainda, não-

próprios do organismo). Este reconhecimento é feito através de linfócitos receptores, as células T e as células B.

Os linfócitos T e B são responsáveis pelo reconhecimento específico dos antígenos. Cada célula B está geneticamente apta a se diferenciar em novos tipos celulares para mecanismos de defesa específicos após reconhecer determinados antígenos por meio de um receptor de superfície, enquanto que os linfócitos T constituem várias subpopulações diferentes com a função de coordenar, por sua vez, a função de defesa imunológica contra vírus, bactérias e fungos.

Uma característica importante do sistema imunológico adaptativo é sua propriedade de memória, da qual nos utilizamos, por exemplo, ao formar defesas contra uma doença após realizar vacinação para a mesma. O sistema imune possui células que armazenam informações sobre contatos com determinados invasores e adquire mecanismos de defesa contra eles em possíveis contatos posteriores, sendo estas reações mais rápidas e eficientes.

Outra importante característica do sistema imunológico é sua auto-imunidade. Esta é a falha da auto-tolerância, uma divisão funcional do sistema imunológico. Para o bom funcionamento do sistema, os indivíduos próprios não devem ser detectados como anomalias, pois senão ocorre a destruição dos mesmos. Alguns exemplos, fazendo analogia ao sistema humano, incluem: a diabetes mellitus tipo 1, lúpus eritematoso sistêmico, tireoidite de Hashimoto, hepatite auto-imune, vitiligo, doença de Graves e artrite reumatóide.

Para maiores detalhes sobre sistema imunológico humano, tanto inato quanto adaptativo, consultar Goronzy e Weyand (2008).

2.3 Sistemas Imunológicos Artificiais

Os Sistemas Imunológicos Artificiais são compostos por metodologias inteligentes, inspiradas no sistema imunológico biológico, para a solução de problemas do mundo real.

A partir da visão geral do Sistema Imunológico foram criadas três principais abordagens, a partir das quais foram desenvolvidos algoritmos bem sucedidos que implementam os Sistemas Imunológicos Artificiais. São eles as Redes Imunológicas Artificiais, o Algoritmo de Seleção Clonal e o Algoritmo de Seleção Negativa, sendo este último o algoritmo utilizado neste trabalho, já que, por ser capaz de realizar a detecção

utilizando apenas os padrões normais, é o de maior interesse para a área de detecção de anomalias.

2.3.1 Redes Imunológicas Artificiais

Este método é baseado nas redes idiotípicas de Jerne (Jerne, 1984), onde há a sugestão de que o sistema imunológico mantenha uma rede de células B interligadas entre si. Nos modelos de rede imunológica artificial, uma população de células B é constituída por duas sub-populações: a população inicial e a população clonada. A população inicial é gerada a partir de um subconjunto de dados brutos para criar a rede. Os restantes dos dados são utilizados como antígeno, onde são então selecionados aleatoriamente e apresentados à rede de células B. Se a ligação for bem sucedida, a população de células B é clonada e sofre mutação. A mutação produz um conjunto de diversos anticorpos que podem ser usados neste processo de ligação. Quando uma nova célula B é criada, uma tentativa é feita para integrá-la à rede mais próxima. Se as novas células B não puderem ser integradas, são removidas da população. Se nenhuma integração for bem sucedida, uma nova rede de células é criada, utilizando o antígeno como modelo, e, em seguida, incorporada à rede.

Uma versão atualizada, chamada AINE (Timmis, 2000), usa o conceito *Artificial Recognition Ball* (ARB) para representar um determinado número de células B semelhantes. Isto remete à idéia do processo onde o anticorpo reconhece o antígeno inserido numa região do espaço. O ARB representa um único ponto n-dimensional que pode ser associado, através da distância euclidiana, a um antígeno ou a outro ARB. Uma ligação entre duas células B é criada quando a afinidade (distância) entre dois ARBs é inferior a um limiar de afinidade da rede. Os resultados mostram que a normalização dos níveis de estimulação ARBs na rede e no mecanismo de atribuição de recursos leva à polarização dos AINE em direção ao mais forte padrão dentro do conjunto de dados a surgir (Dasgupta, 2006).

2.3.2 Algoritmo de Seleção Clonal

O princípio da seleção clonal descreve as características básicas de uma resposta imunológica a um estímulo de um antígeno, onde apenas as células que reconhecem a proliferação do antígeno são selecionadas (Dasgupta, 2006). Em outras palavras, apenas as células B, que reconhecem um antígeno, são estimuladas. Quando isso ocorre, a célula B se multiplica, originando uma linhagem de células capaz de produzir anticorpos específicos contra o antígeno que induziu sua multiplicação. É, portanto, a presença do antígeno que seleciona o tipo de célula B que irá se multiplicar.

O Algoritmo de Seleção Clonal, intitulado CLONALG (Castro & Von Zuben, 2000), baseia-se neste princípio e nos de maturação por afinidade. Este algoritmo é semelhante a algoritmos evolucionários baseados em mutação e tem várias características interessantes:

- Tamanho da população dinamicamente ajustável;
- Aproveitamento e exploração do espaço de busca;
- Localização de múltiplas soluções ótimas;
- Capacidade de manter soluções locais ótimas, e;
- Critério de parada definido.

Este algoritmo sugere que um conjunto de anticorpos seja exposto seguidamente a uma população de antígenos, durante determinadas gerações com o intuito de criar, ao fim destas gerações, um conjunto de anticorpos mais sensíveis e, conseqüentemente, mais eficientes ao estímulo de antígenos.

2.3.3 Algoritmo de Seleção Negativa

Uma das finalidades do sistema imunológico é reconhecer todas as células dentro do corpo e classificá-las como *próprias* e *não-próprias*. As células não-próprias são classificadas em ordem, a fim de induzir um tipo apropriado de mecanismo de defesa. O sistema imunológico aprende, pela evolução, a distinguir os antígenos estranhos das células do próprio organismo. O objetivo da seleção negativa é fornecer tolerância às próprias células.

Ele lida com a capacidade do sistema imunológico de detectar antígenos desconhecidos enquanto não há reação às próprias células.

Durante a produção de células T, os receptores são criados através de um processo de rearranjo genético aleatório. Então, eles sofrem um processo de sensoriamento no timo, chamado seleção negativa. Lá, as células T que reagem com as proteínas do corpo (próprias) são destruídas e, portanto, só aquelas que não se ligam às proteínas são autorizadas a deixar o timo. Estas células T vencedoras, em seguida, circulam por todo o corpo para desempenhar funções imunológicas, protegendo o organismo contra os antígenos estranhos.

O Algoritmo de Seleção Negativa (Forrest et al., 1994) é um dos modelos computacionais de discriminação próprio/não-próprio (*self/non-self*), e o primeiro concebido como um método de detecção de mudança. É um dos primeiros algoritmos utilizado em diversas aplicações do mundo real (Dasgupta, 2006). Este algoritmo é utilizado no desenvolvimento desse trabalho e será abordado, com uma maior profundidade, a seguir.

Em seu trabalho pioneiro, Forrest et al. (1994) propôs um algoritmo de seleção negativa que detectasse a manipulação de dados causada por vírus computacionais. A idéia foi gerar uma série de detectores em uma região do espaço e, em seguida, aplicar esses detectores na classificação de novos dados, classificando-os como próprios/*self* (dados não manipulados) ou não-próprios/*non-self* (dados manipulados). O algoritmo de seleção negativa, como proposto originalmente, está resumido nos seguintes passos.

Dado um espaço U , o conjunto próprio S e o não próprio N , onde:

$$U = S \cup N \text{ e } S \cap N = \emptyset$$

1. Definir próprio como o conjunto S de elementos de tamanho L no espaço U ;
2. Gerar um conjunto D de detectores, de tal forma que cada falha seja associada a um elemento de S ;
3. Monitorar S para mudanças através de contínuas associações dos detectores D contra os elementos de S .

A Figura 1 ilustra o conceito básico do algoritmo de seleção negativa.

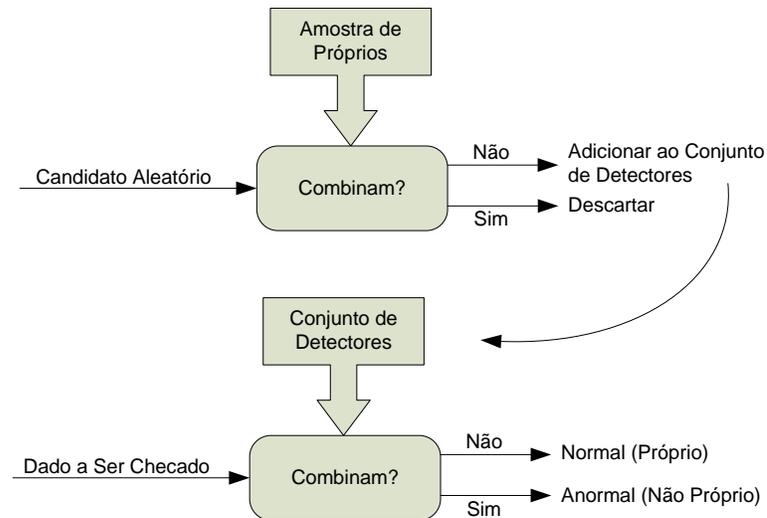


Figura 1 - Conceito básico do Algoritmo de Seleção Negativa

É importante destacar algumas das principais características do algoritmo (Ji e Dasgupta, 2007):

- Representação negativa da informação. Embora, do ponto de vista da teoria da informação, seja equivalente caracterizar o espaço próprio (positivo) ou o espaço não próprio (negativo), os mecanismos de aprendizado estabelecidos utilizam a representação do espaço positivo. Isso significa que, a robustez e a aplicabilidade da representação do espaço negativo ainda estão em fase de exploração.
- Detecção negativa é mais propícia a ambientes distribuídos. É possível dividir um conjunto de detectores negativos em subconjuntos e, então, aplicá-los em um espaço distribuído, desde que a ativação no único detector negativo seja suficiente para caracterizar a amostra como não própria.
- Classificação de uma única classe. Embora o objetivo seja fazer a discriminação entre o próprio e o não próprio, amostras oriundas de uma só classe (própria ou normal) são disponibilizadas para o treinamento do sistema.

Pelas características descritas anteriormente, pode ser observado que o NSA é propício à detecção de anomalias, desde que possa distinguir padrões de sistemas normais de padrões de sistemas defeituosos. Isto também é importante para identificar situações onde o uso do NSA em detecção de anomalias possa ser vantajoso (Martins, 2000):

- Quando o comportamento normal do sistema é definido por um conjunto de padrões complexo, em que é muito difícil obter suas correlações. Neste caso, pode ser mais fácil observar os padrões anormais do que os normais.
- Há sistemas onde o número de padrões anormais possíveis é muito maior do que o número de possíveis padrões normais. Desde que o treinamento de um sistema de detecção de anomalias com um grande número de situações de falha seja impraticável, é aconselhável, primeiramente, a detecção de qualquer comportamento anormal e, então, tentar identificar suas causas.

Existem várias implementações deste algoritmo utilizando a representação binária. Há dois motivos principais para este tipo de representação: primeiro, ela torna mais fácil a análise do espaço do problema. Em D'Haeseler et al. (1996), os autores foram capazes de fornecer uma estimativa exata da quantidade de detectores necessários para alcançar uma certa taxa de detecção e o número de lacunas no espaço não-próprio. O outro motivo é que a representação binária é mais natural quando se lida com dados categóricos (Gonzalez et al., 2002).

Em Gonzalez et al. (2002), os autores destacam algumas desvantagens do NSA com representação binária. A complexidade de tempo (e espaço) do algoritmo cresce exponencialmente com o número de bits utilizado na comparação entre duas cadeias binárias (janela de detecção). Isto representa um problema de escalabilidade, tendo em vista a expressão para interação de características complexas, podendo ser necessária uma grande janela de detecção. A outra desvantagem está relacionada com a inteligibilidade do espaço do problema. Algumas vezes a codificação binária é uma representação de nível muito baixo. Estas desvantagens encorajam a utilização da representação real para o algoritmo de seleção negativa.

2.3.3.1 Algoritmo de Seleção Negativa com Representação Real

A idéia de gerar detectores complementares no espaço para os dados contínuos foi proposta informalmente por Ebner et al. (2002), e formalmente por González et al. (2002, 2003).

O algoritmo de seleção negativa com representação real tem as mesmas três etapas do NSA original: definição do próprio, geração dos detectores e monitoramento. Entretanto,

emprega-se a representação real ao invés da codificação binária para a caracterização do próprio/não-próprio.

O algoritmo de seleção negativa com representação real opera em uma hipersfera unitária $[0,1]^n$. Um detector $d = (c_d, r_{ns})$ tem seu centro $c_d \in [0,1]^n$ e seu raio de reconhecimento de não-próprio $r_{ns} \in \mathfrak{R}$. Além disso, cada elemento próprio $s = (c_s, r_s)$ tem um centro c_s e um raio-próprio r_s .

O raio-próprio foi introduzido para permitir que outros elementos sejam considerados elementos próprios, por se encontrarem perto do centro-próprio. Se um elemento reside dentro de um detector (hiperesfera), este então é classificado como não-próprio, caso contrário, como próprio. Um elemento e reside dentro de um detector $d = (c_d, r_{ns})$, se a distância euclidiana, mostrada na Equação 1, for menor que r_{ns} .

$$dist(c, e) = \left(\sum_{i=1}^n (c_i - e_i)^2 \right)^{1/2} < r_{ns} \quad (1)$$

Esta abordagem é descrita com detalhes em (González, 2002).

O algoritmo possui algumas contrapartidas: o número de detectores a ser usado tem que ser escolhido previamente, não há a possibilidade de determinar se o algoritmo está de fato melhorando o posicionamento dos detectores visando uma possível melhor distribuição, e todos os detectores têm o mesmo raio, o que pode causar uma limitação na escalabilidade.

Numa versão posterior e aperfeiçoada do algoritmo (González, 2003), o número de detectores pode ser estimado de antemão e suas posições podem ser determinadas através da minimização da sobreposição (*overlap*) entre os próprios detectores e entre os detectores e os pontos que representam o conjunto próprio. Uma vez concluída a geração de detectores, os detectores podem ser empregados na fase de monitoramento. A Figura abaixo mostra um desenho bidimensional com um conjunto de detectores e os pontos próprios.

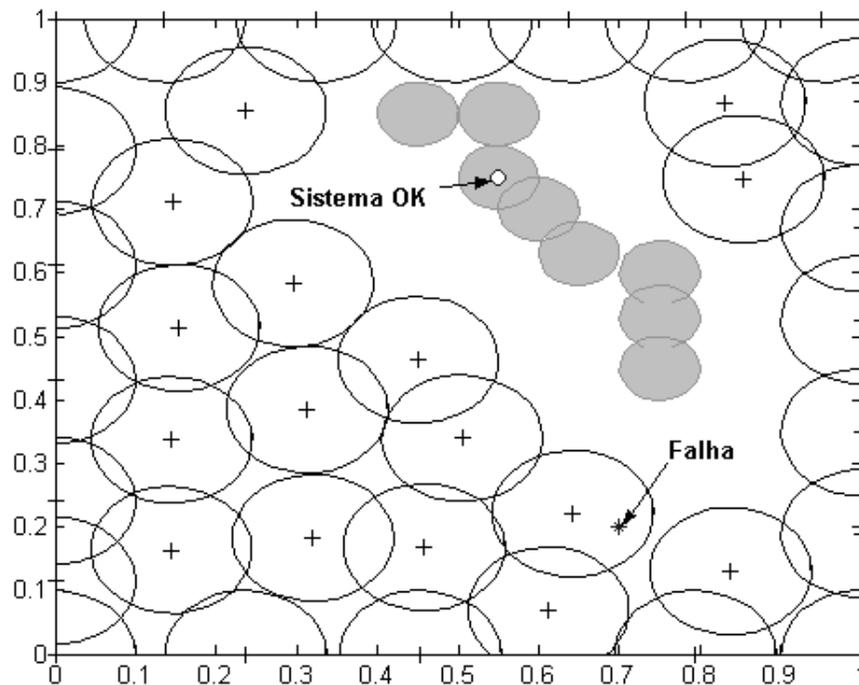


Figura 2 - Algoritmo de Seleção Negativa com representação real

A área cinza representa a região própria. Os círculos com um "+" no centro representam os detectores e o ponto indicado por "o" representa o estado atual do sistema. Nenhuma anomalia é detectada, pois o ponto não pertence ao espaço coberto pelos detectores. Entretanto, o ponto indicado por "*" é detectado como uma anomalia (falha), pois se encontra no espaço coberto pelos detectores. O algoritmo emprega um raio fixo para todos os detectores. Quando se escolhe um raio pequeno, um grande número de detectores será necessário para cobrir o espaço, o que representa um problema de escalabilidade. Se um raio grande é escolhido, é possível que partes do espaço não-próprio fiquem sem cobertura. Esta limitação encorajou o desenvolvimento de novas representações de detectores – que permitam geometria variável – e de novos esquemas de geração de detectores. Como exemplo de tais esquemas podem-se citar *V-Detectors* (Ji and Dasgupta, 2004), a geração automática de detectores por particionamento Quadtree (Amaral et al, 2006) e a geração de detectores através de algoritmos genéticos e integração de quasi-Monte Carlo (Amaral, et al, 2007).

2.3.3.2 Algoritmo de Seleção Positiva

O Algoritmo de Seleção Positiva foi informalmente descrito por Ebner et al. (2002), e formalmente por Stibor et al. (2005b). A principal diferença para a seleção negativa é que não existem detectores não-próprios. Em vez disso, cada elemento do conjunto próprio contém um detector que classifica quaisquer elementos. Um elemento que se situa dentro do detector é classificado como próprio, caso contrário, como não-próprio. Isto significa que nenhuma fase de geração de detectores é necessária, mas a classificação para cada elemento é computacionalmente custosa, quando comparado ao algoritmo de seleção negativa. A Figura 3 compara o Algoritmo Seleção Negativa com o Algoritmo de Seleção Positiva.

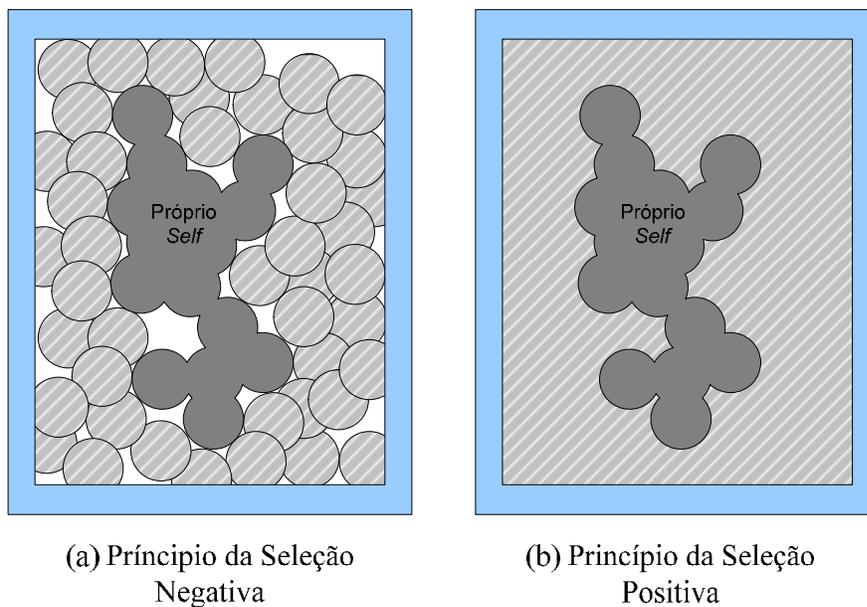


Figura 3 – Comparação entre ambos os princípios de seleção

2.3.3.3 Algoritmo de Seleção Negativa com Detectores Variáveis

Visando a diminuir as limitações do algoritmo com detectores de raio fixo, Ji e Dasgupta (2004) propuseram um algoritmo de seleção negativa utilizando detectores de tamanho variável (denominado *V-Detector*), o que, segundo os autores, otimiza o desempenho

dos detectores. No caso da utilização da representação real, os detectores admitem a forma de hiperesferas.

Este algoritmo determina aleatoriamente o centro de um detector que não deve se situar dentro da hiperesfera de um elemento do conjunto próprio nem dentro de um detector, como mostra a Figura 4. O limiar utilizado pela distância euclidiana define os raios de cada detector, ou seja, os raios são redimensionados dinamicamente até a fronteira da região entrar em contato com um elemento do conjunto próprio. O algoritmo termina quando um número predefinido de detectores é gerado, ou quando uma proporção pré-determinada de espaço não-próprio é coberta.

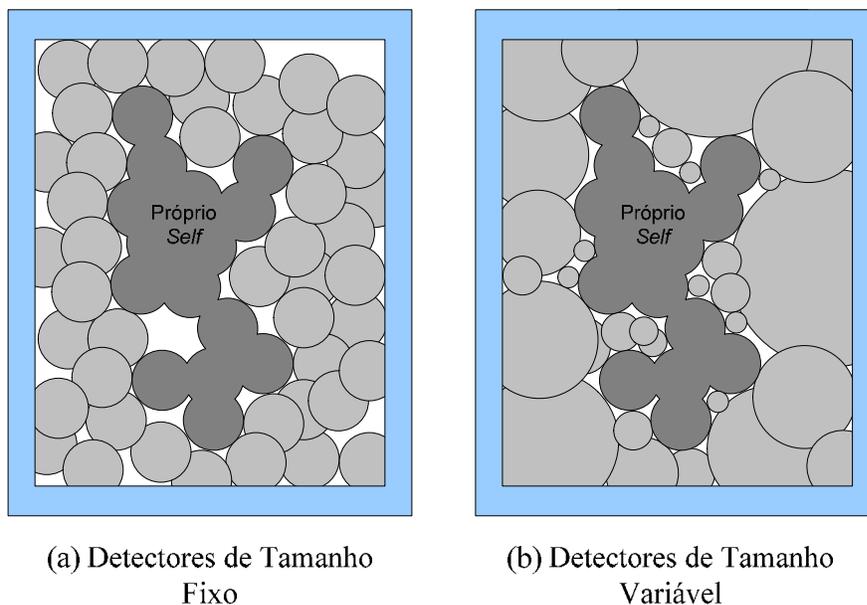


Figura 4 – Comparação entre o algoritmo de detectores de tamanho fixos e variáveis

Algumas das críticas apresentadas aos *V-Detectors* foram rebatidas (Ji e Dasgupta, 2006). Algumas deficiências apresentadas foram corrigidas no algoritmo e foram apontados alguns problemas nos experimentos feitos por Stibor et al. (2005), como, por exemplo, a representação dos dados utilizada. Procurou-se demonstrar que os algoritmos de Seleção Negativa não podem ser cegamente utilizados sem que haja uma representação de dados e uma escolha razoável dos parâmetros de controle. Os autores também evidenciam que existem problemas para os quais o algoritmo de Seleção Negativa não é aconselhado, como problemas com um pequeno número de amostras, ou um problema de classificação genérico, onde a distribuição de probabilidade tenha um grande impacto no desempenho do sistema. Eles afirmam que a aplicabilidade dos algoritmos de Seleção Negativa ainda é uma questão em

aberto e que existem desafios reais para seleção negativa, como, por exemplo, a escalabilidade para um grande número de dimensões, a escolha dos parâmetros de controle ótimos e uma boa modelagem de dados para o domínio de aplicação do problema.

O debate acerca da aplicabilidade do Algoritmo de Seleção Negativa continuou em (Stibor e Timmis, 2007) com um novo conjunto de experiências apontando alguns problemas do algoritmo *V-Detectors* e afirmando a superioridade do SVM de classe única e da Seleção positiva para um problema de identificação com um grande número de dimensões.

2.4 Estudo Detalhado na Geração de Detectores

A representação multidimensional do espaço *próprio/não-próprio* abre a possibilidade de definição de diferentes tipos de geração de detectores.

Um sistema capaz de determinar automaticamente o número de detectores necessários para um bom desempenho do sistema é fundamental.

Em Amaral et al. (2006b) é proposto um método onde se podem gerar detectores automaticamente de maneira eficiente através da decomposição Quadtree. A decomposição Quadtree procura decompor um conjunto convexo cartesiano definido sobre um domínio reticulado convexo em subconjuntos cartesianos sucessivamente menores.

Como exemplo, a Figura 5 mostra a função PEAKS com a qual se deseja fazer uma decomposição Quadtree.

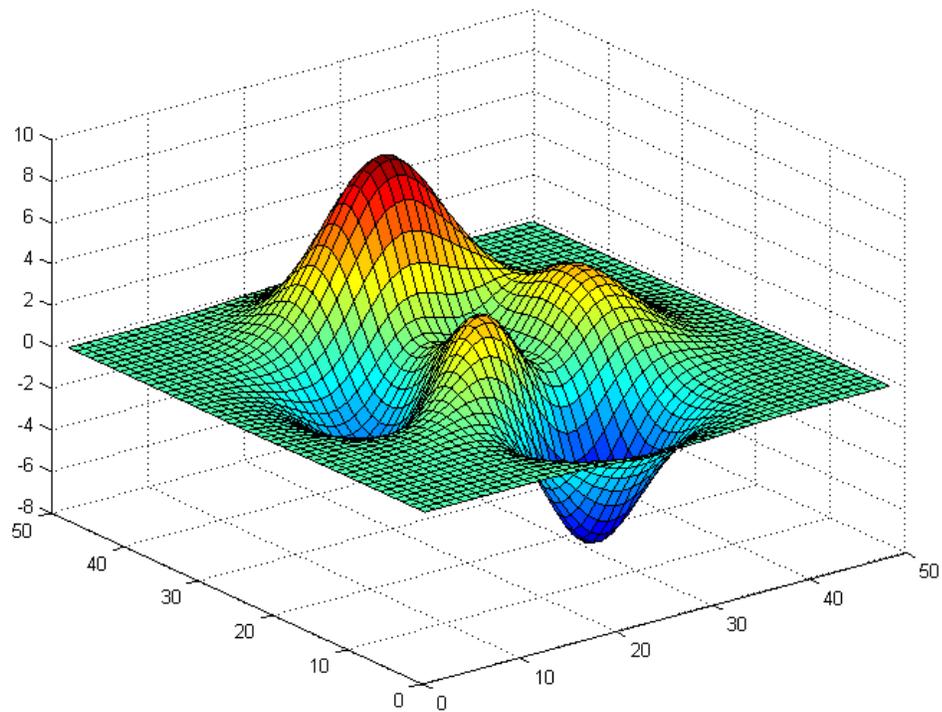


Figura 5 - A função Peaks

Neste caso, a região do domínio deve ser dividida sempre que a diferença entre o valor máximo e o valor mínimo do subconjunto for maior do que um certo valor de limite. O resultado pode ser visto na Figura 6.

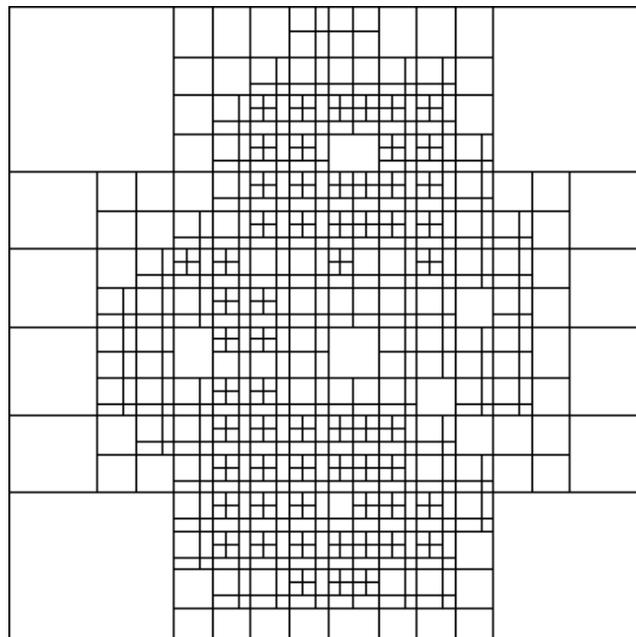


Figura 6 – A decomposição Quadtree da função PEAKS

O procedimento para gerar um conjunto de detectores utilizando a decomposição Quadtree pode ser implementado de duas maneiras diferentes: a primeira realiza o particionamento do domínio da chamada função de detecção e a segunda se baseia no particionamento do próprio detector (Amaral et al., 2006b).

No primeiro caso, particionamento Quadtree por meio da função de detecção, o procedimento para gerar um conjunto de detectores por meio da função de detecção é iniciado definindo-se um reticulado (*grid*) sobre o qual será feita a decomposição.

A seguir, a Função de Detecção (*FD*) em relação ao conjunto *próprio* é calculada para cada ponto do *grid*. Observa-se que esta função terá valores mais altos para os pontos do *grid* que estiverem mais próximos do *conjunto próprio*. A decomposição Quadtree permite que o espaço seja dividido em regiões nas quais o valor da função é praticamente constante. Analisando-se o valor da função em cada bloco, é possível determinar se esta região de pontos “ataca” o conjunto de pontos *próprios* ou não. Aquelas regiões nas quais os pontos não atacam o conjunto próprio são utilizadas para definir os detectores. A posição do detector e o seu raio de ação são definidas a partir das dimensões do bloco.

A geração de um conjunto de detectores por uma função de detecção tem como desvantagem a necessidade de cálculo desta função em cada ponto do *grid*, que em determinadas situações pode ser bastante custoso computacionalmente.

No segundo caso, particionamento Quadtree do detector, o procedimento de geração é descrito pelo fluxograma da Figura 7.

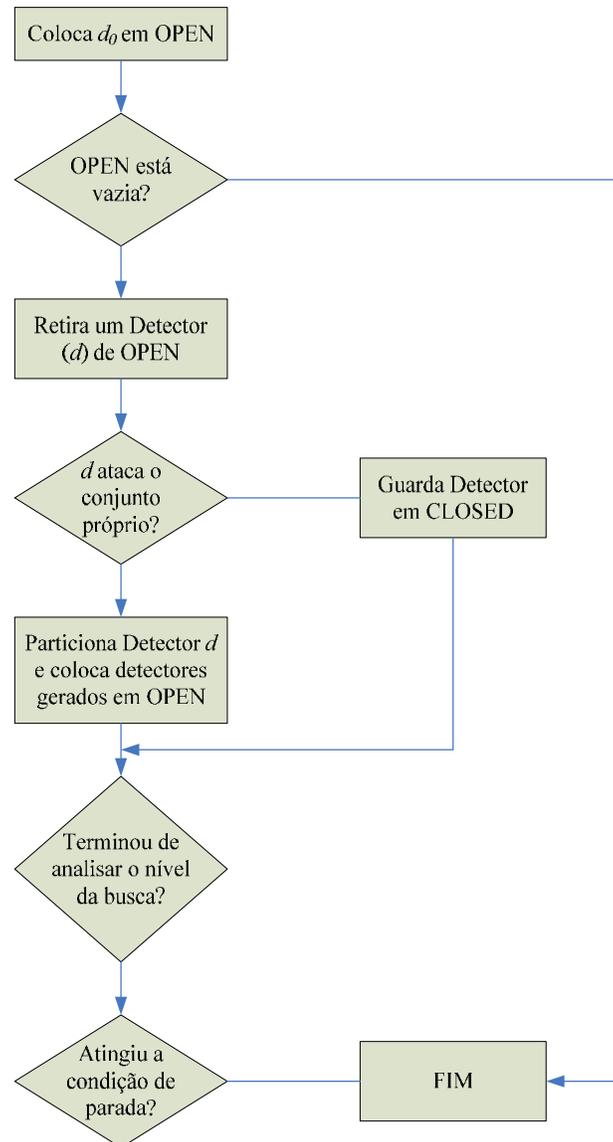


Figura 7 – Fluxograma para particionamento do detector

O algoritmo utiliza duas filas chamadas de OPEN e CLOSED. Na primeira são armazenados os detectores que serão testados e na segunda são armazenados os detectores que devem ser utilizados.

Antes de iniciar o algoritmo, o detector inicial d_0 deve ser colocado na fila OPEN e a fila de CLOSED deve estar vazia. Este detector d_0 tem seu centro localizado no centro do hipercubo e seu raio é igual a $\frac{\sqrt{n}}{2}$, o que significa que ele abrange todo o hipercubo.

O procedimento consiste em retirar um detector d da fila de OPEN e testar se este detector ataca o conjunto próprio. Caso ele não o ataque, deve ser armazenado na fila de CLOSED. Se atacar o conjunto próprio, ele deve ser particionado em 2^n detectores, onde n é a

dimensão do problema. Testado o detector, é verificado se todos os detectores de um mesmo raio já foram analisados, isto é, se já terminou a análise de um nível do algoritmo. Caso o tenha, verifica-se se a condição de parada foi atingida. Caso contrário, o algoritmo volta para analisar um outro detector.

Podem ser utilizadas diferentes condições de parada: raio mínimo para o detector, número máximo de detectores e volume ocupado pelos detectores. O cálculo do volume é feito através de uma integração de Monte Carlo. O principal papel do critério de parada é evitar o crescimento exponencial do particionamento Quadtree, especialmente em problemas de alta dimensão.

Em Amaral et al. (2007) foi proposto um método de geração de detectores que faz uso de Algoritmos Genéticos para gerar o conjunto de detectores. O sucesso deste método depende de como a solução é codificada no cromossomo e da função de aptidão escolhida para avaliar a solução.

O Algoritmo de Seleção Negativa com Detectores Variáveis foi o algoritmo utilizado neste trabalho. A geração destes detectores foi feita por meio de Algoritmos Genéticos e será apresentada mais detalhadamente no Capítulo a seguir.

3 MODELO UTILIZADO NA GERAÇÃO DE DETECTORES

A partir da análise do algoritmo do RNSA descrito no Capítulo anterior, o modelo para a geração de detectores com representação real será descrito mais detalhadamente e suas devidas otimizações abordadas.

3.1 Descrição do Método

No método proposto (Amaral, 2006a), o problema de geração de detectores é visto como um problema de otimização. A função a ser maximizada é o volume coberto pelos detectores, com a restrição de que não pode haver sobreposição com o conjunto próprio.

Esta geração de detectores foi feita através de um Algoritmo Evolucionário. Os Algoritmos Evolucionários são métodos estocásticos de busca e otimização globais inspirados na evolução natural. Eles operam sobre uma população de soluções potenciais através da aplicação do princípio de sobrevivência do mais apto, produzindo melhores soluções de forma sucessiva para o problema em questão.

Os quatro principais modelos de computação evolucionária existentes são: a Programação Evolutiva, as Estratégias Evolutivas, a Programação Genética e o Algoritmo Genético. Para maiores detalhes sobre cada modelo pode-se consultar respectivamente: Fogel e Fogel. (1995), Bäck (1996b), Kosa (1992) e Goldberg (1989), Michalewick (1996) e Mitchell (1996). Já para uma visão geral dos algoritmos evolucionários, consultar os seguintes textos: (Spears, 1993), (Bäck, 1996a).

O Algoritmo Evolucionário utilizado como algoritmo de otimização foi o Algoritmo Genético (GA).

O Algoritmo Genético, assim como qualquer Algoritmo Evolucionário, necessita dos seis componentes (Michalewick, 1996), mostrados na Figura 8 e citados a seguir:

- Uma representação genética para as possíveis soluções para o problema;
- Uma forma de criar uma população inicial de potenciais soluções;
- Uma função de avaliação que desempenha o papel do ambiente, classificando soluções de acordo com sua aptidão;

- Critério de parada do algoritmo;
- Operadores genéticos que modificam a composição dos descendentes (seleção, *crossover* e mutação), e;
- Valores para os vários parâmetros que o algoritmo genético utiliza (tamanho da população, probabilidades de aplicação dos operadores genéticos, etc.).

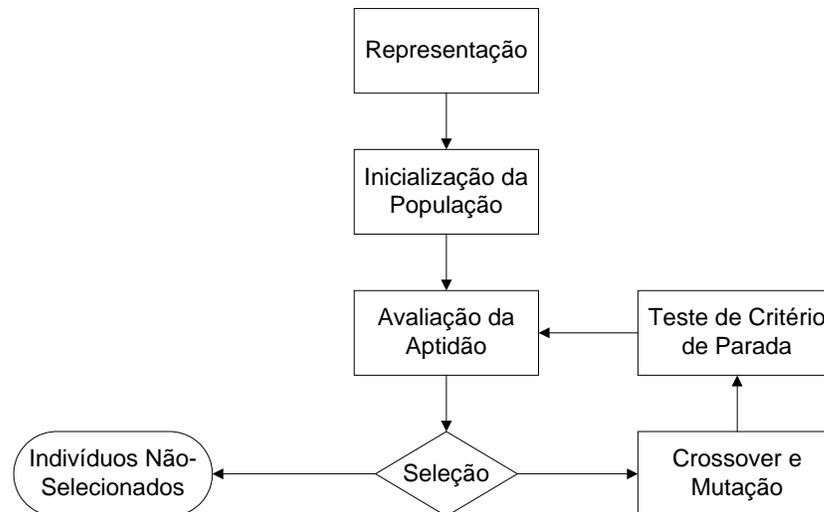


Figura 8 – Funcionamento de um Algoritmo Genético

No método utilizado, cada cromossomo representa um possível conjunto de detectores onde cada gene representa um ponteiro (índice) para um determinado ponto em uma seqüência de amostras que representa uma distribuição de probabilidade. Portanto, cada cromossomo determina que pontos da seqüência devam ser usados como centros dos detectores. A principal idéia por trás desta representação é reduzir o tamanho do cromossomo, especialmente em problemas de alta dimensão.

Independentemente da dimensão do problema, o tamanho do cromossomo é igual ao número máximo de detectores desejado. Um exemplo possível desta representação é mostrado na Figura 9, onde os círculos representam o conjunto próprio (*self*); os pontos identificados por '+' são candidatos a serem centro de detectores. No cromossomo mostrado à direita há cinco genes indicando que os pontos 3, 6, 10, 4 e 8 são os candidatos.

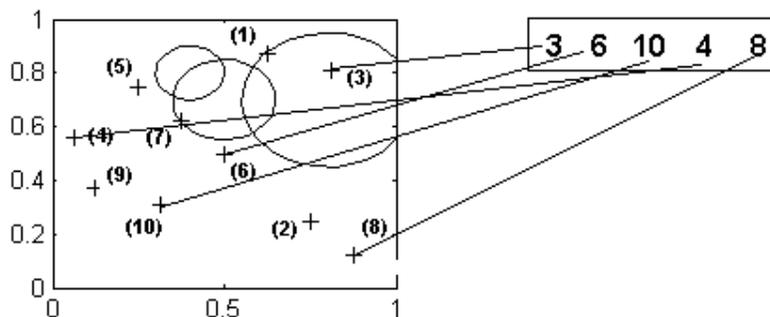


Figura 9 – Exemplo da codificação utilizada

Neste trabalho, ao invés de usar distribuições pseudo-aleatórias para determinar o centro dos detectores, como proposto por Ji e Dasgupta (2004), seqüências quase-aleatórias foram utilizadas, uma vez que estas apresentam um elevado grau de uniformidade nos espaços multidimensionais e podem proporcionar uma melhor cobertura do espaço do problema com menos amostras. Algumas das seqüências quase-aleatórias mais conhecidas são Faure, Sobol e Halton (Morokoff e Caflisch, 1994).

A seqüência de Halton usa uma base principal diferente para cada dimensão. Para a primeira dimensão usa-se base 2, para a segunda dimensão, base 3, para a terceira dimensão, base 5, e assim por diante. Bases altas significam ciclos mais longos e maior tempo de execução.

A seqüência de Faure é semelhante à seqüência de Halton, porém com duas diferenças: usa somente uma base para todas as dimensões e uma permutação dos elementos dos vetores para cada dimensão.

Por sua vez, a seqüência de Sobol tem, assim como a seqüência de Faure, uma mesma base para todas as dimensões e prossegue reordenando os elementos do vetor dentro de cada dimensão. A seqüência de Sobol é mais simples (e mais rápida) do que a seqüência de Faure pois usa a base 2 para todas as dimensões (Dias, 2004).

A função de avaliação é composta de duas partes: a decodificação do cromossomo e o cálculo da função objetivo propriamente dita.

A decodificação do cromossomo consiste em extrair a posição do centro dos detectores a partir dos índices que estão no cromossomo. Uma vez determinados os centros dos detectores, uma função de decodificação calcula o maior raio possível, levando em consideração que o conjunto de detectores não pode invadir o conjunto próprio e que deve haver certa quantidade de sobreposição entre os detectores. Se um dos centros escolhidos está

dentro do próprio conjunto, este é descartado. Por exemplo, na Figura 9, o centro do detector com a etiqueta (3) seria eliminado.

Após a função de decodificação ter calculado o maior raio possível, não invadindo o conjunto próprio, o raio pode ser ajustado para ter certo grau de sobreposição. O parâmetro α controla este grau de sobreposição. Se $\alpha = 0$, o raio é igual à distância entre os detectores; se $\alpha = 1$, não há sobreposição entre os detectores. O raio calculado de um detector baseado no valor de α é mostrado na Figura 10.

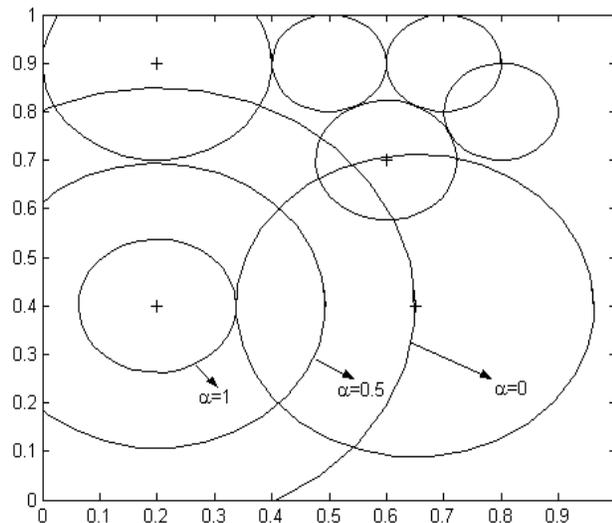


Figura 10 – Cálculo do raio em função do parâmetro α

O processo de posicionamento dos detectores, após o raio ter sido calculado pela função de decodificação, é mostrado na Figura 11. Inicialmente, a função de decodificação define o raio do primeiro detector para que não haja sobreposição com qualquer um dos círculos que representam o conjunto próprio, ou seja, o raio é máximo sem que toque o conjunto próprio. Um segundo detector é colocado com certo grau de sobreposição (*overlap*) com o primeiro. Esta regra - certa sobreposição com outros detectores e a não sobreposição a todos do conjunto próprio - é observada nos posicionamento dos demais detectores.

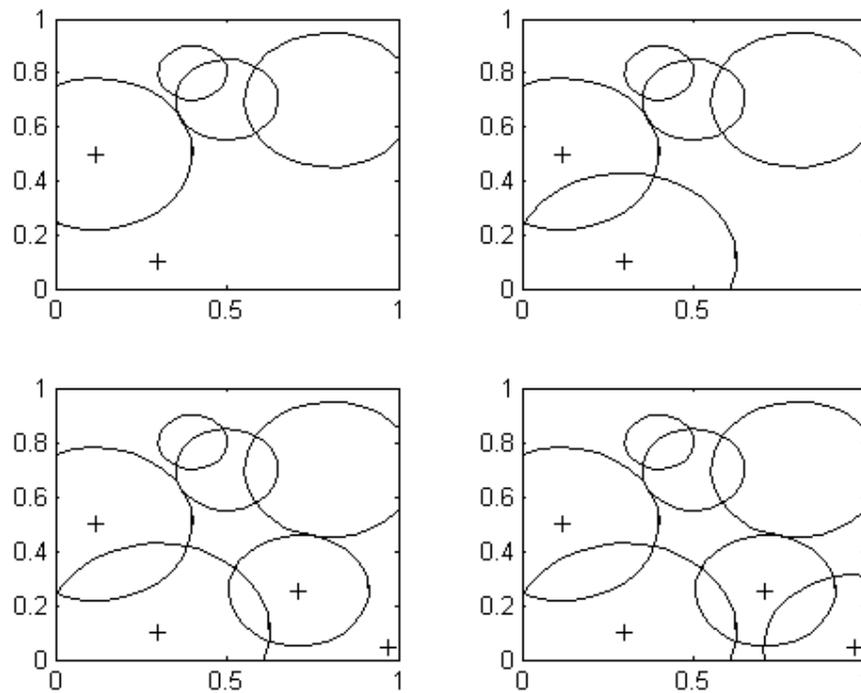


Figura 11 – Posicionamento dos detectores

O Algoritmo Genético necessita de uma função de avaliação que mensure quão adaptados estão os indivíduos da população. Esta função, também chamada de função de aptidão, avalia cada um dos membros da população com o intuito de determinar quais são os indivíduos mais aptos, pois estes são aqueles com maior probabilidade de se reproduzir.

A função de aptidão utilizada foi o volume coberto pelos detectores. A principal vantagem de se utilizar esta função é saber de antemão o seu valor com certa margem de erro.

Se o volume do conjunto próprio V_s puder ser calculado, o volume V_{ns} que deve ser coberto pelo conjunto de detector e pode ser facilmente calculado como $V_{ns} = 1 - V_s$, desde que o espaço próprio/não-próprio corresponda a um hipercubo unitário $[0,1]^n$. Portanto, quanto mais próximo do valor de V_{ns} estiver o volume do conjunto de detectores, melhor será a solução. Então, é possível configurar o algoritmo genético para executar um determinado número de gerações ou até que se encontre a cobertura desejada (desde que o V_{ns} possa ser estimado).

O volume do conjunto de detectores e o volume do conjunto próprio são calculados pela Integração de Monte Carlo.

Segundo a literatura, o nome "Monte Carlo" surgiu durante o projeto Manhattan na Segunda Guerra Mundial. No projeto de construção da bomba atômica, Ulam, Von Neumann e Fermi consideraram a possibilidade de utilizar o método, que envolvia a simulação direta de

problemas de natureza probabilística relacionados com o coeficiente de difusão do nêutron em certos materiais.

Já a denominação do método provém da cidade de Monte Carlo (“capital dos jogos de azar”), no principado de Mônaco, famosa pelos seus cassinos, e jogos de roleta, que são dispositivos que produzem números aleatórios.

A Integração de Monte Carlo é um método para integrar uma função de mais de um domínio, onde expressões analíticas são muito difíceis de serem aplicadas - por exemplo, o cálculo do volume de sobreposição hiperesferas.

Considere-se $X = [0,1]^n$ e $A \subseteq X$ é um subconjunto de X cujo volume tem de ser calculado. A integração pode ser expressa como:

$$I = \int_x f(x) dx \quad (2)$$

$$f(x) = \begin{cases} 0, & x \notin A \\ 1, & x \in A \end{cases}$$

Se $g(x)$ representa uma função densidade de probabilidade de uma variável aleatória uniformemente distribuída ao longo de $X = [0,1]^n$, a Equação (2) pode ser reescrita como:

$$I = \int_x f(x)g(x)dx = E(f(x)) \quad (3)$$

que é igual ao valor esperado para $f(x)$, $E(f(x))$. Este valor esperado pode ser encontrado obtendo a média da função $f(x)$ nos n pontos selecionados aleatoriamente em $[0,1]^n$. Portanto, a integral I pode ser estimada como:

$$I = \int_x f(x)g(x)dx = E(f(x)) = \frac{1}{n} \sum_{i=1}^n f(x_i) \quad (4)$$

O erro absoluto deste método é independente do número de dimensões do espaço X e diminui a uma taxa $1/\sqrt{N}$, onde N corresponde ao número de pontos utilizados. De modo a obter uma melhor convergência, distribuições quasi-aleatória são utilizadas novamente. Nestes, ao contrário das distribuições pseudo-aleatórias, amostras são determinísticas e calculadas de forma que "espaços" deixados entre as amostras anteriores sejam preenchidos por novas amostras. Isto reduz o desvio padrão e acelera uma simulação Monte Carlo (Morokoff, 1993). Quando distribuições quasi-aleatórias são usadas, o erro decresce a uma taxa de aproximadamente $1/N$.

O GA passa de geração a geração selecionando e reproduzindo pais até que um critério de parada seja satisfeito. Alguns critérios podem ser utilizados, como por exemplo, a convergência da população (Houck et al., 1995). No caso desta pesquisa foi utilizado como critério de parada um número máximo de gerações. Em outras palavras, ao fim de um número

especificado de gerações, há o término do algoritmo. Com isso, evita-se que o algoritmo permaneça em execução para sempre (*loop* infinito).

A seleção natural é um processo pelas quais características hereditárias que contribuem para a sobrevivência e reprodução se tornem mais comuns numa população, enquanto que características prejudiciais tornam-se mais raras. Isto ocorre porque indivíduos com características vantajosas têm mais sucesso na reprodução, de modo que mais indivíduos da próxima geração têm estas características herdadas.

Ao longo de muitas gerações, adaptações ocorrem através de uma combinação de mudanças sucessivas, pequenas e aleatórias nas características de cada indivíduo, e da seleção natural das variantes mais adequadas ao seu ambiente. Estas mudanças sucessivas são feitas através de mecanismos chamados operadores genéticos.

Estes operadores genéticos são necessários para que a população se diversifique e mantenha características benéficas adquiridas pelas gerações anteriores. Em outras palavras, os operadores genéticos fornecem a base de busca do Algoritmo Genético. Estes operadores são utilizados para criar novas soluções baseadas nas soluções existentes na população. Existem dois tipos básicos de operadores: cruzamento (*crossover*) e mutação.

O cruzamento leva dois indivíduos a se reproduzirem e gerarem dois novos indivíduos enquanto a mutação altera um indivíduo a fim de produzir uma nova solução. A aplicação destes dois tipos básicos de operadores e seus derivados depende da representação utilizada.

Os operadores genéticos utilizados foram:

- Mutação Uniforme: Seleciona aleatoriamente uma variável e a troca por um número aleatório uniforme.
- Mutação de Fronteira: Seleciona aleatoriamente uma variável e a troca pelo seu limite superior ou inferior.
- Mutação Não-Uniforme: Seleciona aleatoriamente uma variável e a troca por um número aleatório não-uniforme.
- Mutação Não Uniforme Múltipla: Aplica o operador de Mutação Não Uniforme para todas as variáveis do indivíduo-pai.
- Cruzamento Simples: Muda cada variável em todos os indivíduos da população com uma probabilidade pré-determinada.
- Cruzamento Aritmético: Produz duas combinações lineares complementares dos indivíduos-pai.

- **Cruzamento Heurístico:** Produz uma extrapolação linear de dois indivíduos. Este é o único operador que utiliza a informação de aptidão.

Para que a utilização destes operadores com a representação proposta fosse possível, foi necessário fazer uma adaptação, uma vez que o cromossomo é formado pelos índices que posicionam o centro de cada detector, ou seja, números inteiros. Em outras palavras, as operações executadas pelos operadores genéticos fracionam estes números, tendo que ocorrer um arredondamento destes índices cada vez que a função de aptidão é acionada.

Geração após geração, o Algoritmo Genético busca sempre encontrar indivíduos com melhor desempenho que seus predecessores. Porém, alguns problemas intrínsecos do algoritmo podem dificultar a procura ou mesmo conduzir a pesquisa a um rumo errado. Nestes casos, possivelmente um mau ajuste de determinados parâmetros do GA ocasiona falhas nos operadores.

Para esta pesquisa foram utilizados os seguintes parâmetros:

- Número de indivíduos da população: Uma população pequena pode conter amostras insuficientes na maioria dos hiperespaços, conduzindo o algoritmo a um mínimo local. Já uma população grande contém uma quantidade suficiente do hiperespaço, porém a execução do algoritmo será mais custosa computacionalmente, podendo se tornar inaceitável.
- Número de gerações: Um pequeno número de gerações pode conduzir o algoritmo a um mínimo local, enquanto que um grande número de gerações possivelmente conduzirá a uma boa solução. No entanto, há grandes possibilidades de o GA convergir muito antes de atingir o número de gerações especificado, isto é, haveria um gasto computacional desnecessário.
- Função de Seleção: A seleção dos indivíduos para produzir sucessivas gerações desempenha um papel extremamente importante no Algoritmo Genético. Foi utilizada a função de seleção baseada na normalização geométrica. Este método define a probabilidade de um indivíduo ser selecionado por:

$$P_j = q' (1 - q)^{s-1}, \quad (5)$$

onde,

$$q' = \frac{q}{1 - (1 - q)^{P'}}, \quad (6)$$

q é a probabilidade de seleção do melhor indivíduo, s é a classificação do indivíduo (1 sendo melhor) e P o tamanho da população.

3.2 Identificação de Limitações do Método

Ao longo do desenvolvimento desta pesquisa foram identificados alguns problemas, características ou limitações do algoritmo, em especial no cálculo da função de avaliação, que merecem destaque.

3.2.1 Ordenação dos Detectores

A função decodificadora utiliza uma estratégia na qual, segundo certa ordem, cada detector vai sendo colocado no espaço por vez (Amaral, 2006a). A idéia de ordenar os detectores, como feito originalmente, foi implementada com o intuito de que os primeiros detectores gerados estivessem posicionados o mais afastado possível do conjunto próprio.

A Tabela 1 apresenta os resultados da avaliação da ordenação dos detectores feitos para três conjuntos de dados sintéticos. Cada conjunto de dados apresenta a forma de um triângulo, uma cruz e uma rosca, respectivamente Figura 12a, b e c.

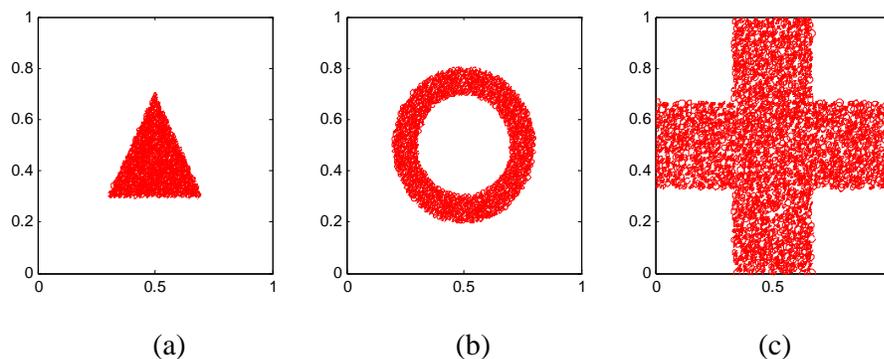


Figura 12 - Conjuntos de dados

(a) Forma de Triângulo; (b) Forma de Rosca; (c) Forma de Cruz

Essa avaliação foi feita com base no volume coberto pelo conjunto de detectores gerados no espaço não próprio. Todos os parâmetros do método foram mantidos os mesmos:

número máximo de detectores = 50, população = 100, gerações = 200, *overlap* $a = 0.5$, *crossover* = 65% e mutação = 0.8%. Os resultados presentes na figura são a média entre 10 experimentos.

Tabela 1 – Volume Coberto no Teste de Ordenação

Caso	Volume Coberto (em %)	
	Com Ordenação	Sem Ordenação
Triângulo	95,54	99,13
Cruz	84,08	90,81
Rosca	90,84	97,53

Pode ser observado que, para estes exemplos, a utilização da ordenação não se mostrou tão benéfica. Os resultados obtidos sem fazer esta ordenação são superiores, o que sugere que o próprio algoritmo genético, “naturalmente”, se encarrega de ordenar os detectores. A utilização da ordenação, portanto, se tornaria repetitiva, não ajudando o algoritmo, o que, conseqüentemente, a torna desnecessária.

3.2.2 Variação na Sobreposição

O objetivo de cada detector é envolver o maior volume possível do hipercubo unitário sem que haja sobreposição com os elementos do conjunto próprio e com certo grau de sobreposição com os outros detectores (Amaral, 2006a). Este grau de sobreposição é controlado pelo parâmetro α . Quando este tem valor igual a 1 não há sobreposição entre os detectores e quando tem valor igual a 0, o grau de sobreposição é o maior possível.

Foi realizado um experimento com o objetivo de observar a cobertura em função do parâmetro a . Na Figura 13a observa-se o primeiro conjunto de dados sintéticos. Este conjunto possui o formato de uma cruz sobre todo o espaço $[0,1]^2$, e foi utilizado para avaliar as propriedades do algoritmo *V-Detector* (Ji e Dasgupta, 2004), descrito na seção 2.4.4. No conjunto de treinamento desta pesquisa foram utilizados 300 pontos gerados aleatoriamente na região do conjunto próprio e 1000 pontos de teste gerados no espaço não-próprio. O raio dos indivíduos próprios é considerado uma propriedade do conjunto de dados e é estimado pela distância média ao vizinho mais próximo. Isto significa que, na média, outro ponto próprio está na dada distância do raio do conjunto próprio.

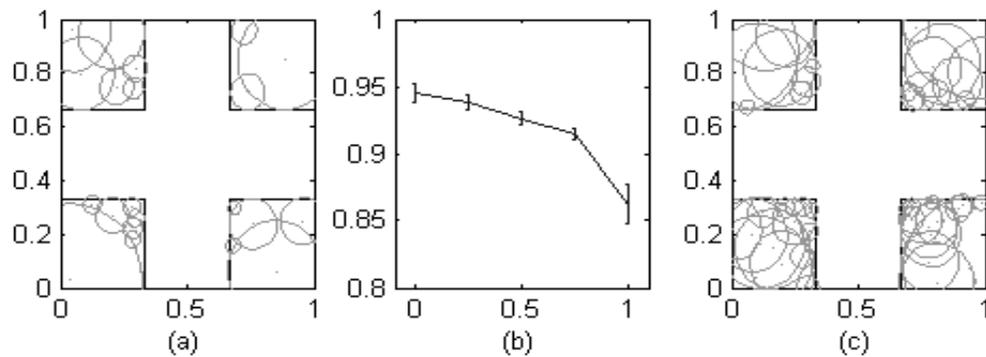


Figura 13 – Resultados para o conjunto da cruz

(a) Cobertura não-própria com $\alpha = 0$; (b) Cobertura não-própria versus α ; (c) Cobertura não-própria sem o controle de sobreposição

A Figura 13b apresenta a cobertura não-própria obtida para diferentes valores de α . Todos os parâmetros do método foram mantidos os mesmos: número máximo de detectores = 50, população = 100, gerações = 200, *crossover* = 65% e mutação = 0.8%. Os resultados presentes na figura são a média entre 20 experimentos e o desvio padrão é apresentado como uma barra de erro.

Pode ser observado que a cobertura aumenta e que o valor de α diminui. Isto significa que, se for dado um maior grau de sobreposição entre os detectores, maior será a liberdade para o algoritmo genético determinar o melhor conjunto de detectores. Também se pode perceber que o desvio padrão observado nestes experimentos é pequeno, o que sugere que o método é capaz de produzir resultados consistentes para os mesmos padrões.

O mesmo resultado pode ser visto na Figura 14, onde foi utilizada uma região do conjunto próprio no formato de um triângulo, similar ao utilizado para demonstrar o Algoritmo de Seleção Negativa, por Ji (2005).

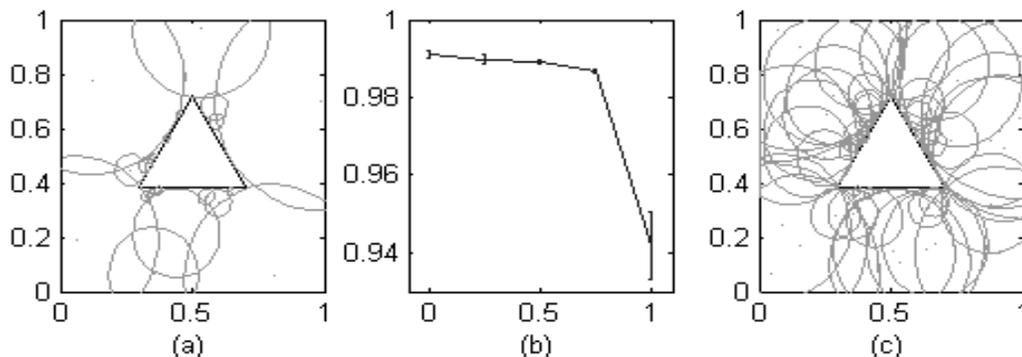


Figura 14 - Resultados para o conjunto do triângulo

(a) Cobertura não-própria para $\alpha = 0$; (b) Cobertura não-própria versus α ; (c) Cobertura não-própria sem o controle de sobreposição

Dos resultados obtidos por ambos os conjuntos de dados, pode ser observado que, se há a diminuição da sobreposição (aumento de α), o número de detectores também diminui, assim como a cobertura do espaço não próprio. Isto significa que o parâmetro α controla o número de detectores e a cobertura do espaço não próprio.

O algoritmo também pode ser executado sem nenhum controle de *overlap*. A Figura 13c e a Figura 14c mostram a cobertura do espaço não próprio sem o controle de *overlap*. Pode-se observar que há uma excessiva sobreposição entre os detectores e que é possível que um grande número de detectores pouco contribua para a cobertura do espaço não próprio. A partir desta observação, foi proposto o seguinte procedimento para se obter uma boa cobertura com um número menor de detectores. Inicialmente, executa-se o algoritmo com $\alpha = 0$, pois como pode ser visto na Figura 13b e na Figura 14b, este é o valor de sobreposição que fornece maior cobertura com um menor número de detectores. Ao final da execução, anota-se o número final de detectores obtidos. Executa-se novamente o algoritmo, agora utilizando como número máximo de detectores – isto é, como tamanho do cromossomo –, o número final de detectores encontrado anteriormente. Quando se comparam os resultados obtidos para o conjunto de dados em forma de cruz, utilizando a estratégia descrita anteriormente, com os resultados obtidos sem controle de sobreposição, observa-se que houve uma diminuição de apenas 3,75% na cobertura e de 46% no número de detectores. Já no caso do triângulo, a diminuição de cobertura foi de apenas 0,7% e a redução do número de detectores foi de 50%.

3.2.3 Comportamento em Altas Dimensões

Dados com altas dimensões são uma dificuldade para classificadores e outros problemas que envolvem aprendizado de máquinas. Muitos algoritmos podem não trabalhar muito bem com um alto número de dimensões, já que o espaço de busca cresce exponencialmente quando o número de dimensões aumenta (Ji e Dasgupta, 2007).

Para avaliação do comportamento do algoritmo em altas dimensões, foi seguido o mesmo processo que Ji e Dasgupta (2007) utilizaram.

Considerando o espaço de busca um hipercubo unitário $[0,1]^n$ de n-dimensões, este processo consiste em distribuir 1000 pontos aleatoriamente sobre a região própria e supor que uma hiperesfera está centrada no centro do hipercubo, onde, por sua vez, é gerado o conjunto de detectores usando estes pontos como amostras do conjunto próprio e, em seguida, tentando classificar os outros 1000 pontos de teste.

No algoritmo original, os parâmetros utilizados para o algoritmo genético foram: 100 gerações, 50 indivíduos e número máximo de detectores de 100. Os operadores de mutação e crossover utilizados, apontados no capítulo anterior foram: Mutação de Fronteira, Mutação Não Uniforme Múltipla, Mutação Não Uniforme e Mutação Uniforme. Os operadores de crossover utilizados foram: Crossover Aritmético, Crossover Heurístico e Crossover Simples (Houck, 1995). A Tabela 2 apresenta os resultados do algoritmo original para algumas dimensões.

Tabela 2 – Resultados do algoritmo original

Dimensão	Tempo de Execução (minutos)	Cobertura dos Detectores	Número de Detectores	Erro	Taxa de Detecção	Taxa de Alarme Falso
2	24.50±0.75	0.9913±0.0001	93.3±2.1	0.01±0	0.9879±0	0±0
4	29.55±0.51	0.9679±0.0011	99.5±0.6	0.03±0.001	0.9739±0.0084	0±0
6	31.05±0.70	0.9156±0.0045	99.5±0.5	0.084±0.0102	0.9158±0.0102	0±0
8	30.35±0.43	0.7170±0.0184	100±0	0.29225±0.01565	0.7078±0.0156	NaN
16	32.29±0.48	0.1241±0.0018	100±0	0.88425±0.01266	0.1158±0.0127	NaN

Observa-se que, apesar do tempo de execução ser grande (da ordem de dezenas de minutos), ele não aumenta significativamente com o aumento da dimensão. Também se pode notar que o número máximo de detectores utilizados não é suficiente para cobrir o espaço não-próprio em dimensões mais altas, dentro do número de gerações escolhido. Além disso, pode-se notar que o valor de cobertura obtido nos fornece uma boa indicação do desempenho

em relação à detecção de falhas. Quando menor a cobertura, menor é a taxa de detecção e maior é o erro.

Este algoritmo é eficaz quando avaliado sua cobertura pelos detectores, desde que seja ajustado de forma conveniente, mas o seu tempo de execução é alto. Isto ocorre porque os operadores utilizados não são apropriados para a representação empregada, fazendo-se necessária, portanto, a criação de operadores genéticos mais adequados.

3.3 Otimizações do Método

Ao longo do desenvolvimento desta pesquisa foram propostas algumas otimizações para o método. Estas otimizações visam, acima de tudo, diminuir o custo computacional.

3.3.1 Operadores Genéticos Mais Adequados

Como dito no item 3.1, para utilização dos operadores genéticos com a representação proposta tornou-se necessária uma adaptação, uma vez que o cromossomo é formado pelos índices que posicionam o centro de cada detector, ou seja, números inteiros. Em outras palavras, as operações executadas pelos operadores genéticos fracionam estes números, tendo que ocorrer um arredondamento destes valores a fim de que se tornem índices cada vez que a função de aptidão é acionada.

A fim de eliminar esta limitação, foram utilizados dois operadores mais adequados, sendo um de *Crossover* e outro de *Mutação*, ambos utilizados por Correa et al (2004).

Este novo operador genético de *Crossover* calcula dois vetores de troca para cada pai. Para cada gene (índices da coordenada do centro do detector) do pai 1, o operador verifica se o mesmo está presente (em qualquer posição) no genoma do pai 2. Se isso não acontecer, este gene é copiado para o vetor de troca do pai 1. Isto significa que os índices presentes no vetor de troca podem ser transferidos para o pai 2 como resultado de um *crossover*, uma vez que esta transferência não criaria qualquer gene duplicado no genótipo do pai 2. O mesmo procedimento é realizado para cada gene do pai 2. Por exemplo, suponha-se que dois pais é sejam compostos pelos seguintes cromossomos: P1 = [1, 2, 3, 4, 5] e P2 = [2, 5, 9, 10, 12].

Portanto, os respectivos vetores de troca são: $VP1 = [1, 3, 4]$ e $VP2 = [9, 10, 12]$. Uma vez que os vetores de troca foram identificados, o operador propriamente dito pode ser aplicado, como se segue.

Neste operador genético de cruzamento, a probabilidade de *crossover* utilizada não é fixa. O *crossover* é realizado sempre que os dois pais não são iguais um ao outro, ou seja, sempre que haja pelo menos um índice no vetor de troca de ambos os pais. Se os dois pais são iguais uns aos outros, ou seja, os seus vetores de troca estão vazios, um dos pais é reproduzido sem alterações para a próxima geração e o outro pai é apagado, para evitar que indivíduos duplicados sejam inseridos na população desnecessariamente.

A operação de *Crossover* é realizada da seguinte forma. É gerado um número natural aleatório c , variando de 1 até a quantidade de pontos quasi-aleatórios criados. Este número c determina quantos índices de cada vetor de câmbio serão efetivamente trocados entre os dois pais. Este procedimento garante que não haverá duplicação de índices em qualquer um dos dois indivíduos produzidos pelo cruzamento.

Já a mutação ocorre de maneira análoga à operação de *crossover*. O cromossomo a sofrer mutação tem seu gene substituído por outro gerado aleatoriamente, porém com a condição de que cada índice do genótipo do novo indivíduo não esteja presente no genótipo do indivíduo que sofrera a mutação.

Como parâmetro de comparação foi feito um experimento onde foi utilizado como conjunto próprio um círculo de raio 0,25 centrado no centro do hipercubo unitário (Ji e Dasgupta, 2006). Foram utilizadas 100 gerações, 50 indivíduos por população, número máximo de 200 detectores e 20.000 pontos quasi-aleatórios.

Os resultados apresentados na

Tabela 3 mostram que o algoritmo com operadores genéticos mais adequados se mostrou muito menos custoso computacionalmente, tendo reduzido em aproximadamente 75% do tempo de execução do algoritmo original. Com relação ao volume e ao número de detectores finais, ambos se mostram equivalentes. A taxa de detecção foi a mesma para ambos os algoritmos. Já a taxa de falso alarme foi 1,71% para o algoritmo original, enquanto o algoritmo com os novos operadores genéticos propostos teve a mesma taxa nula.

Tabela 3 – Comparação entre o algoritmo original e o algoritmo com os operadores genéticos mais adequados

Parâmetros	Algoritmo Original	Algoritmo com Operadores Genéticos mais Adequados
Tempo	70 minutos	18 minutos
Número de Detectores	184 detectores	176 detectores
Volume Coberto	0,8001	0,7989
Taxa de Falso Alarme	1,71%	0
Taxa de Detecção	98,79%	98,79%

3.3.2 Operadores de Seleção de Detectores

Com o algoritmo utilizado, sem o parâmetro de sobreposição α , não há controle sobre o número de detectores usados, fazendo com que o algoritmo tente utilizar tantos detectores quantos possíveis (normalmente o valor é muito próximo ao número máximo de detectores permitidos), desde a função de aptidão avalie somente a cobertura do espaço não próprio, e não o número de detectores que foram utilizados. Uma situação similar pode ser observada no algoritmo *V-Detectors* (Ji, 2004), já que este foi desenvolvido para alcançar uma certa percentagem de cobertura, e o número máximo de detectores foi proposto para evitar que o algoritmo ficasse em *loop* infinito.

Para controlar o número excessivo de detectores foram utilizados operadores de seleção de detectores, cuja finalidade é descartar detectores.

Estes operadores foram implementados como operadores de mutação e foram denominados: Mutação com Seleção Simples, Mutação com Seleção Múltipla e Mutação com Algoritmo de Poda.

O operador de Mutação com Seleção Simples é responsável por retirar apenas um detector, dado um limiar de probabilidade.

O operador de Mutação com Seleção Múltipla é responsável por retirar quantos detectores forem determinados, dado um limiar de probabilidade.

Já o operador de Mutação com Algoritmo de Poda é responsável por retirar detectores segundo o critério exposto a seguir.

Dado o número de detectores já colocados D , verifica-se qual o detector mais próximo, d_{PROX} , do detector d_0 que se quer colocar.

Então, calcula-se:

$$O_0 = e^{-\left(\frac{dist}{r_{PROX}}\right)^2}, \quad (7)$$

onde $dist$ é a distância entre d_{PROX} e d_0 ; e r_{PROX} é o raio do detector mais próximo d_{PROX} .

Por sua vez, é gerado um valor r aleatoriamente através de uma distribuição uniforme. Com isso, se $r > O_0$, o novo detector é inserido à população. Caso contrário, ele é considerado pouco útil e, conseqüentemente, podado, ou seja, não é incluído na população.

Como exemplo, pode-se citar três situações possíveis. Num primeiro momento, o detector d_0 é posicionado de tal forma que sua distância ao vizinho mais próximo é $2r$ (Figura 15a). Neste caso, após as devidas substituições na Equação 7, obtém-se o valor 0,0183 para O_0 , o que significa que este detector tem 98,17% de chance de ser aceito por esse operador. Em um segundo momento, o detector d_0 é posicionado a $0,9r$ de distância ao vizinho mais próximo (Figura 15b), o que implica em obter o valor 0,4448 para O_0 . Assim, este detector tem 55,52% de chance de ser aceito. E, por fim, em um terceiro momento, o detector d_0 é posicionado a $0,1r$ de distância ao vizinho mais próximo (Figura 15c), fazendo com que O_0 seja 0,9900 e, conseqüentemente, tenha 1% de chance de ser adicionado ao conjunto de detectores.

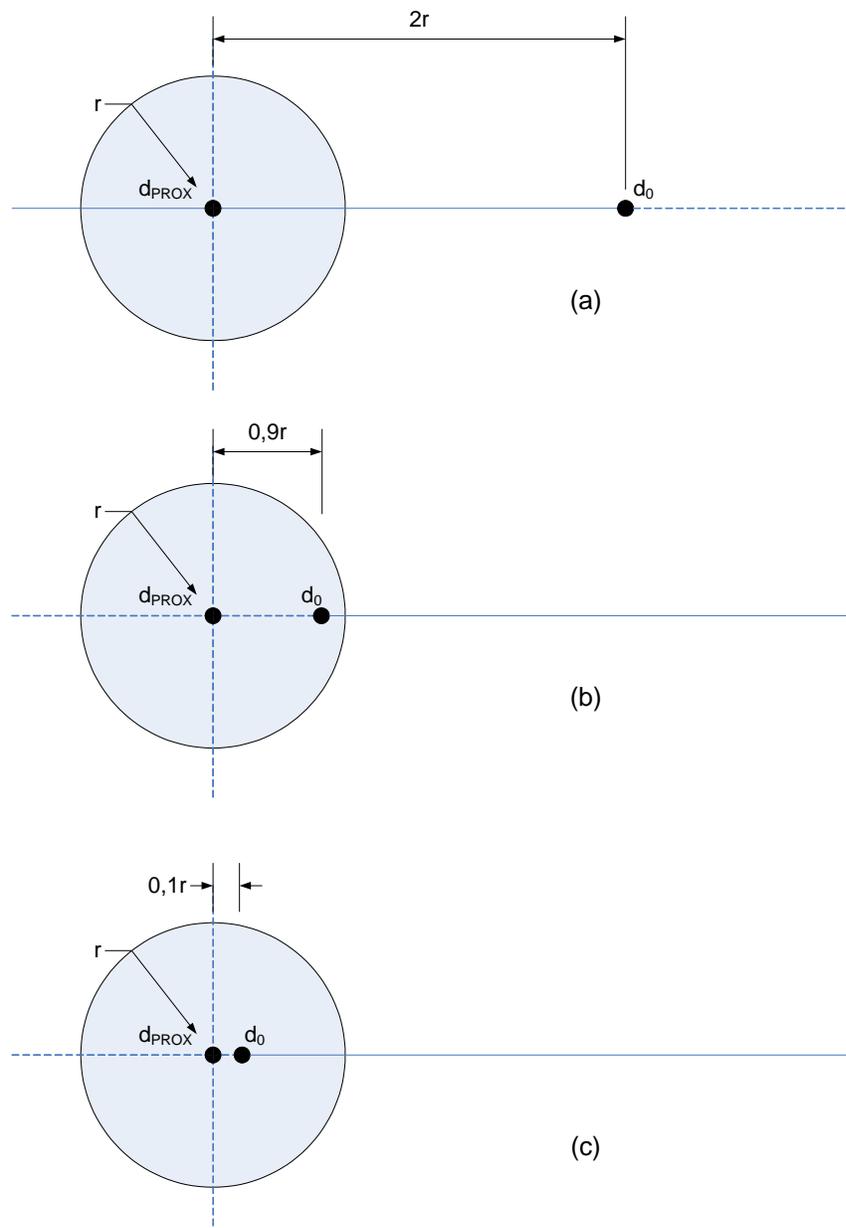


Figura 15 – Exemplos de Funcionamento do Operador de Seleção com Algoritmo de Poda

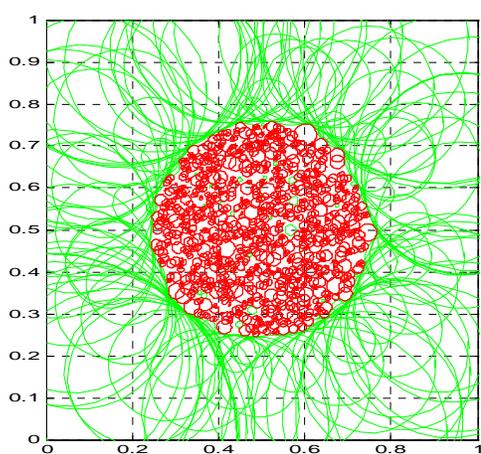
Assim como no item 3.3.1, foram feitos experimentos onde foi utilizado como conjunto próprio um círculo de raio 0,25 centrado no centro do hipercubo unitário (Ji e Dasgupta, 2006). Os parâmetros 100 gerações, 50 indivíduos por população, número máximo de 200 detectores e 20.000 pontos quasi-aleatórios foram utilizados, assim como os três operadores genéticos descritos neste item.

Os resultados apresentados na Tabela 4 mostram que o algoritmo com operadores genéticos mais adequados se mostrou menos custoso computacionalmente que o algoritmo

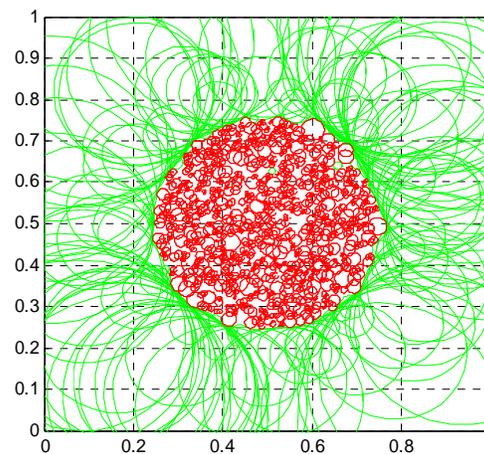
original, tendo reduzido o tempo de execução em aproximadamente 40%. Em termos do número de detectores finais, o algoritmo de poda de detectores utilizou 15% menos detectores do que o algoritmo original, ocupando uma área de 0,7972, contra a área de 0,8001 coberta pelo algoritmo original. A taxa de detecção para o algoritmo original foi de 98,79% e para o programa com os novos operadores foi de 98,67%. Já a taxa de falso alarme foi 1,71% para o algoritmo original, enquanto o algoritmo com algoritmo de poda teve a mesma taxa zero. Na Figura 16a está representado o algoritmo original com seus 184 detectores, enquanto que na Figura 16b está representado o algoritmo com o algoritmo de poda de detectores e seus 176 detectores, podendo se perceber a diferença no número de detectores finais.

Tabela 4 – Comparação entre o algoritmo original e o algoritmo com operadores de seleção

Parâmetros	Algoritmo Original	Algoritmo com Operadores de Seleção
Tempo	70 minutos	38 minutos
Número de Detectores	184 detectores	158 detectores
Volume	0,8001	0.7976
Taxa de Falso Alarme	1,71%	0
Taxa de Detecção	98,79%	98,67%



(a)



(b)

Figura 16 – Comparação entre algoritmo original e algoritmo de poda de detectores

3.3.3 Redução do Número de Pontos do Conjunto Próprio

Um bom conjunto de pontos próprios (*Self*), isto é, um conjunto de pontos com um número reduzido, mas que represente adequadamente a noção de *próprio*, melhora o desempenho do algoritmo, já que este diminui o número de iterações realizadas, o que decresce seu custo computacional.

Para que isto pudesse ser avaliado foi utilizado um conjunto de dados sintéticos que possui o formato de uma cruz, já que esta forma representa uma dificuldade maior para a geração de detectores com formato de hiperesferas.

Foram desenvolvidos três métodos de redução do conjunto próprio, todos baseados no método de clusterização *k-means*. Os centros dos clusters serão os pontos do conjunto próprio reduzido. Os raios para o conjunto próprio reduzido são calculados de três maneiras diferentes, descritas a seguir.

Na primeira maneira, o raio do conjunto próprio reduzido é dado pela maior distância do centro do *cluster* a um elemento deste mesmo *cluster*. Na segunda maneira, o raio é dado pela distância ao centro do *cluster* mais próximo.

Já na terceira maneira, calcula-se a distância ao centro do *cluster* mais próximo para todos os centros do *cluster*. A seguir, é construído um histograma destas distâncias para se obter sua densidade de probabilidade. Um número ótimo de divisões para o histograma é determinado com base no método de Shimazaki e Shinomoto (2007). Uma vez obtida a densidade de probabilidade, é possível escolher um valor de distância d tal que a probabilidade $P(x < d)$ seja igual a um limiar.

Para encontrar o número ótimo de pontos do conjunto próprio foi utilizado um algoritmo descrito a seguir.

- Dois conjuntos são criados. Um é o conjunto chamado Referência (REF), sendo composto por um número muito grande de pontos que representem o conjunto próprio, uma vez que este número é muito grande, ele é capaz de representar o conjunto próprio de forma adequada. O outro conjunto, denominado Compacto (COMP), é, como o seu próprio nome diz, uma representação compacta, isto é com um menor número de pontos do que o conjunto de Referência. Este menor número de pontos foi obtido através de um dos métodos de redução do conjunto próprio, descritos anteriormente.

- Observa-se então a porcentagem de pontos de REF contidos em COMP. Este valor, denominado N_{IN} , nos fornece a indicação de o quão boa é a cobertura das hiperesferas da representação compacta em relação aos pontos do conjunto de Referência.
- Geram-se pontos aleatórios dentro de COMP e observa-se quantos destes pontos não são capturados pelas hiperesferas REF. Este valor, chamado de N_{OUT} , nos fornece uma indicação de quanto hiperespaço, além daquele ocupado por REF, está sendo ocupado pela representação COMP.
- O ponto ideal é quando N_{IN} possuir valor 1 e N_{OUT} possuir valor 0, pois isso significa que a representação compacta captura todos os pontos de REF sem ocupar um hipervolume maior do que aquele ocupado por REF. Quanto mais próximo deste ponto estiver a representação, melhor será a representação compacta. Com isso, reduz-se o número de pontos, fazendo com que o algoritmo se torne menos custoso computacionalmente e, conseqüentemente, diminua seu tempo de execução. A Figura 17 mostra o ponto ideal.

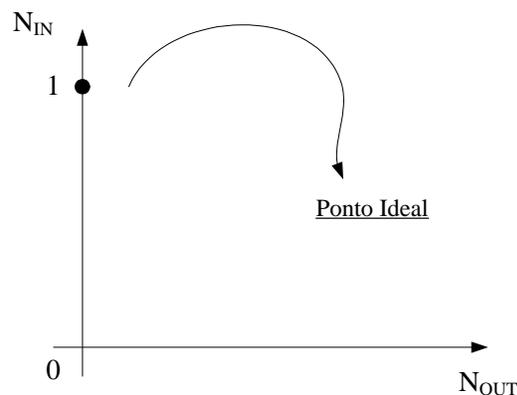


Figura 17 – Ponto ideal para a redução do conjunto próprio

Para exemplificar, foi utilizado o mesmo conjunto de dados sintéticos em formato de cruz como conjunto próprio e gerado 5000 pontos que foram utilizados como representação de referência. Foram geradas, ainda, distribuições compactas com 1000, 1500, 2000 e 2500 pontos. Cada uma destas distribuições compactas foi calculada 10 vezes e os valores de média e desvio padrão dos valores de N_{IN} e N_{OUT} são mostrados a seguir. A Tabela 5 mostra os valores de média e desvio padrão para *k-means* com raio ao elemento mais próximo. A Tabela 6 mostra os mesmo valores, mas para *k-means* com raio pelo centro mais próximo. Já a

Tabela 7 e a Tabela 8 apresentam, respectivamente, *k-means* pela probabilidade segundo limiares de 0.68 e 0.95.

Tabela 5 – *k-means* com raio ao elemento mais próximo

Número de Pontos da Distribuição Compacta	Média de N_{IN}	Desvio padrão de N_{IN}	Média de N_{OUT}	Desvio padrão de N_{OUT}
2500	0.81046	0.0047	0.0537	0.00248
2000	0.81914	0.00577	0.0727	0.0035
1500	0.85118	0.00280	0.1058	0.0032
1000	0.90254	0.0024	0.1673	0.00579

Tabela 6 – *k-means* com raio pelo centro mais próximo

Número de Pontos da Distribuição Compacta	Média de N_{IN}	Desvio padrão de N_{IN}	Média de N_{OUT}	Desvio padrão de N_{OUT}
2500	0.99544	0.00121	0.1620	0.00302
2000	0.99512	0.0012372	0.2000	0.004
1500	0.99552	0.00122275	0.236945	0.005
1000	0.99708	0.00066	0.27858	0.0035

Tabela 7 – *k-means* com limiar 0.68

Número de Pontos da Distribuição Compacta	Média de N_{IN}	Desvio padrão de N_{IN}	Média de N_{OUT}	Desvio padrão de N_{OUT}
2500	0.99314	0.0018	0.238707	0.0076
2000	0.9938	0.001771	0.2610	0.0050
1500	0.99494	0.000933	0.282252	0.0069
1000	0.99646	0.00128	0.2979	0.0045

Tabela 8 – *k-means* com limiar 0.95

Número de Pontos da Distribuição Compacta	Média de N_{IN}	Desvio padrão de N_{IN}	Média de N_{OUT}	Desvio padrão de N_{OUT}
2500	0.99998	0.0001	0.2887	0.006
2000	0.99992	0.00017	0.294	0.006
1500	0.99992	0.00014	0.3037	0.004
1000	0.99988	0.000139	0.311	0.0084

4 ESTUDOS DE CASOS

Para avaliar a proposta apresentada neste trabalho, alguns experimentos foram realizados utilizando-se o modelo de geração de detectores desenvolvido, baseado em Algoritmos Genéticos e apresentado no Capítulo anterior. Seus objetivos principais foram determinar a funcionalidade deste modelo e auxiliar na identificação das virtudes e deficiências do mesmo.

Os três primeiros experimentos consistem em avaliar o desempenho das melhorias sugeridas para o modelo de geração de detectores. O primeiro experimento consiste em avaliar o comportamento do algoritmo em altas dimensões. Já o segundo experimento avalia os métodos propostos de redução do conjunto do número de pontos do conjunto próprio utilizado na geração de detectores. No terceiro, comparam-se os métodos de seleção positiva e negativa.

A partir do quarto experimento até o último foram feitos estudos de casos com a utilização de exemplos reais.

O quarto experimento consiste na detecção de falhas paramétricas de um circuito eletrônico que implementa um filtro passa-banda do tipo Sallen Key (Arminian et al., 2000).

O quinto experimento trata de um filtro ativo que é utilizado como *benchmark* na área de detecção de anomalias em circuitos (Kaminska, et al., 1998).

O sexto experimento é relativo à detecção de anomalias em um filtro Leapfrog de quarta ordem e também utilizado como *benchmark* na área de detecção de falhas em circuitos (Kaminska, et al., 1998).

A detecção de anomalias em rolamentos (Ji e Dasgupta, 2004) foi feita no sétimo experimento.

No oitavo e último experimento foi realizada a detecção de anomalias em eletrocardiogramas (ECG) (Kim et al., 2004).

No quarto, quinto e sexto e oitavo experimentos foram pré-processados pelo algoritmo *Bitmap* (Wei et al., 2005), que por sua vez utiliza a representação simbólica para séries temporais SAX (Lin et al., 2003).

4.1 Avaliações

Para avaliar o desempenho das melhorias sugeridas para o algoritmo proposto, foram realizados alguns testes. Estes testes consistem na análise do comportamento do algoritmo para altas dimensões e na análise de alguns métodos de redução do número de pontos que compõem o conjunto próprio.

4.1.1 Desempenho do Algoritmo

Dados com altas dimensões são uma dificuldade para classificadores e outros problemas que envolvem aprendizado de máquinas. Muitos algoritmos podem não trabalhar muito bem com um alto número de dimensões, já que o espaço de busca cresce exponencialmente quando o número de dimensões aumenta (Ji e Dasgupta, 2007).

Para avaliação do comportamento do algoritmo em altas dimensões, foi seguido o mesmo processo empírico utilizados por Ji e Dasgupta (2007).

Considerando o espaço de busca um hipercubo unitário $[0,1]^n$ de n-dimensões, este processo consiste em distribuir 1000 pontos aleatoriamente sobre a região própria e supor que uma hiperesfera com raio 0.25 está centrada no centro do hipercubo, onde, por sua vez, é gerado o conjunto de detectores usando estes pontos como amostras do conjunto próprio. Em seguida tenta-se classificar os outros 1000 pontos de teste.

A Tabela 9 apresenta o volume a ser coberto pelos detectores para as dimensões 2, 4, 6, 8 e 16. O cálculo foi feito com base no processo utilizado por Stibor et al. (2006). Pode-se verificar que, com o aumento do número de dimensões, o volume do conjunto próprio diminui e, conseqüentemente, o volume a ser coberto aumenta.

Tabela 9 – Volume do espaço não-próprio para cada dimensão

Dimensão	Volume a ser coberto
2	0.80365
4	0.98073
6	0.99874
8	0.999938
16	1.00000

No algoritmo original, os parâmetros utilizados para o algoritmo genético foram: 100 gerações, 50 indivíduos e número máximo de detectores de 100. Os operadores de mutação e crossover utilizados, apontados no capítulo anterior, foram: Mutação de Fronteira, Mutação Não Uniforme Múltipla, Mutação Não Uniforme e Mutação Uniforme. Os operadores de crossover utilizados foram: Crossover Aritmético, Crossover Heurístico e Crossover Simples (Houck, 1995). A Tabela 10 apresenta os resultados do algoritmo original para algumas dimensões.

Tabela 10 – Resultados do algoritmo original

Dimensão	Tempo de Execução (minutos)	Cobertura dos Detectores	Número de Detectores	Erro	Taxa de Detecção	Taxa de Alarme Falso
2	24.50±0.75	0.9913±0.0001	93.3±2.1	0.01±0	0.9879±0	0±0
4	29.55±0.51	0.9679±0.0011	99.5±0.6	0.03±0.001	0.9739±0.0084	0±0
6	31.05±0.70	0.9156±0.0045	99.5±0.5	0.084±0.0102	0.9158±0.0102	0±0
8	30.35±0.43	0.7170±0.0184	100±0	0.29225±0.01565	0.7078±0.0156	NaN
16	32.29±0.48	0.1241±0.0018	100±0	0.88425±0.01266	0.1158±0.0127	NaN

Já para o algoritmo com melhorias, foi utilizado o mesmo número de gerações e população. Os operadores de mutação e crossover utilizados são os descritos na seção 3.3.1 e 3.3.2. A Tabela 11 apresenta os resultados do algoritmo com melhorias para algumas dimensões.

Tabela 11 – Resultados do algoritmo com melhorias

Dimensão	Tempo de Execução (minutos)	Cobertura dos Detectores	Número de Detectores	Erro	Taxa de Detecção	Taxa de Alarme Falso
2	18.50±0.60	0.9912±0.0003	89.5±2.1	0.0095±0	0.9885±0.0009	0±0
4	19.74±0.26	0.9690±0.0005	99.5±0.7	0.032±0	0.9684±0.0029	0±0
6	20.10±0.28	0.9284±0.0002	100±0	0.077±0.0057	0.9228±0.0057	0±0
8	21.50±0.02	0.7446±0.0182	100±0	0.2645±0.0148	0.7335±0.0148	NaN
16	22.19±0.21	0.132±0.0005	100±0	0.8775±0.0078	0.1225±0.0078	NaN

Como pode ser observado, o tempo de execução diminui de 25 a 32%. Já a cobertura dos detectores se manteve a mesma, superando um pouco a cobertura do algoritmo original à medida que o número de dimensões aumenta. O número de detectores se manteve bem próximo, assim como o erro do classificador. A taxa de detecção foi ligeiramente maior do que a do algoritmo original, enquanto a taxa de alarme falso se manteve a mesma.

4.1.2 Métodos de Redução do Conjunto Próprio

Para testar o método de redução, foram gerados detectores utilizando a distribuição de referência (5000 pontos) e a distribuição compacta (1000 pontos). Foi utilizada a técnica de redução usando o *k-means* com raio determinado pela distância ao centro do grupo mais próximo. Foi utilizado um conjunto próprio em formato de cruz, 200 gerações e um máximo de 100 detectores. Os resultados são apresentados na Tabela 12 e Tabela 13.

Tabela 12 – Resultados para algoritmo original

Numero de Pontos	Tempo de Execução	Cobertura dos Detectores	Número de Detectores	Erro	Taxa de Detecção	Taxa de Alarme Falso
1000	24 min	92,28%	87	0.038	91,34%	0%
5000	69,5 min	97,68%	84	0.023	96,7%	0.89%

Tabela 13 – Resultados para algoritmo com melhorias

Numero de Pontos	Tempo de Execução	Cobertura dos Detectores	Número de Detectores	Erro	Taxa de Detecção	Taxa de Alarme Falso
1000	17 min	92,35%	68	0.038	91,34%	0%
5000	40 min	97,61%	86	0.018	96,81%	0,71%

Como pode ser visto, no algoritmo original, para um a redução de 80% dos pontos, foi obtida uma grande diminuição do tempo de execução, um pequeno decréscimo na taxa de detecção e um pequeno aumento na cobertura, no erro e na taxa de alarme falso.

Já no algoritmo com as melhorias propostas implementadas, para os mesmo 80% de redução do número de pontos do conjunto próprio, foi obtida uma grande diminuição do tempo de execução, um pequeno decréscimo na cobertura e nas taxas de alarme falso e detecção e um aumento do erro.

4.1.3 Comparação entre Métodos de Seleção Positiva e Negativa

Muitas vezes questiona-se o porquê de se utilizar a seleção negativa e não simplesmente a positiva. Do ponto de vista da teoria da informação, os conjuntos normal e anormal contêm a mesma quantidade de informação, o que sugere que representações igualmente compactas são possíveis. Além disso, pode-se citar ainda outras ocasiões que apóiam o uso da seleção negativa (Martins et al., 2000).

- Quando o comportamento normal de um sistema é definido por um conjunto de padrões complexos, onde é muito difícil obter relações. Neste caso, pode ser mais fácil observar os padrões anormais do que os normais.
- Há sistemas nos quais a quantidade de padrões anormais possíveis é muito maior do que a quantidade de padrões normais. Já que treinar um sistema de detecção de falhas com um número grande de situações de falha se torna impraticável, é aconselhável primeiro detectar qualquer comportamento anormal e então tentar identificar sua causa.
- O algoritmo de seleção negativa pode ser usado para gerar amostras não próprias e fazer-se uso de um classificador comum (como uma rede neural) como um detector de falhas (Gonzalez, 2000)

A seguir serão mostrados dois exemplos onde a seleção negativa obtém bons resultados em relação à seleção positiva.

Voltando ao exemplo do conjunto de dados em forma de cruz, utilizando-se 1000 pontos para representar o conjunto próprio, usando como raio a distância ao vizinho mais próximo, vemos na Tabela 14 e na Tabela 15 os resultados alcançados pela representação negativa e positiva.

Tabela 14 – Resultados do algoritmo original utilizando representação positiva e negativa (1000 pontos)

Método	Erro	Taxa de Detecção	Taxa de Alarme Falso
Negativa	0.051	93,4%	3,9%
Positiva	0.191	95,6%	30,65%

Tabela 15 – Resultados do algoritmo com melhorias utilizando representação positiva e negativa (1000 pontos)

Método	Erro	Taxa de Detecção	Taxa de Alarme Falso
Negativa	0.042	92,7%	1,78%
Positiva	0.191	95,6%	30,65%

Pode-se observar que, no caso da seleção negativa, a taxa de detecção é um pouco menor, mas, em contrapartida, a taxa de alarme falso é bem menor. Isto acontece, porque a seleção negativa, quando cobre o espaço não próprio, tende a criar uma fronteira em torno dos pontos próprios. Esta fronteira é responsável por considerar que amostras anormais muito próximas ao conjunto próprio sejam consideradas normais, fazendo com que a taxa de detecção seja menor do que no caso da seleção positiva. Por outro lado, esta fronteira também faz com que pontos do conjunto próprio que não estejam dentro da hiperesferas dos pontos próprios, mas que pertencem ao conjunto próprio, sejam considerados próprios. Isto explica porque a taxa de alarme falso neste caso é maior para a seleção positiva. Este comportamento se aplica tanto ao algoritmo original como ao algoritmo com as melhorias implementadas. A Figura 18 e a Figura 19 mostram os resultados.

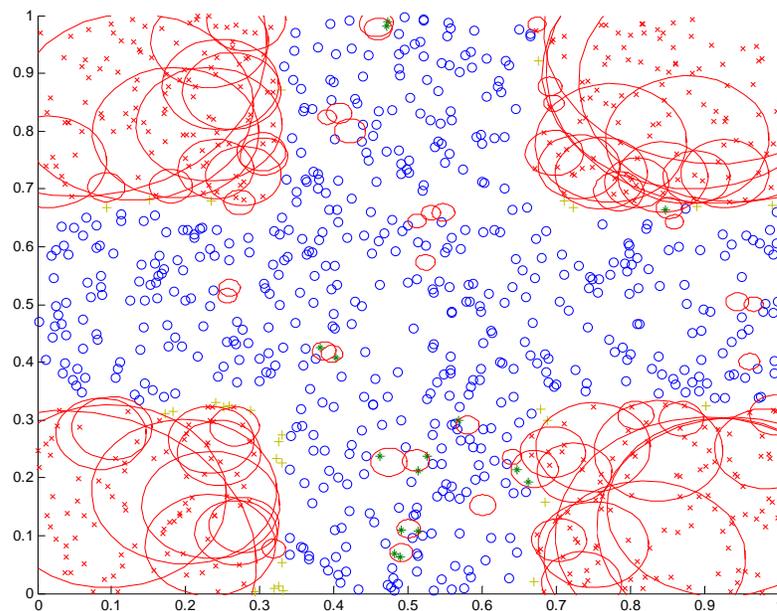


Figura 18 – Seleção negativa utilizando algoritmo original (1000 pontos)

o Pontos normais classificados corretamente. **x** Pontos anormais classificados corretamente.
± Sinais anormais classificados como normais. ***** Sinais normais classificados como anormais.

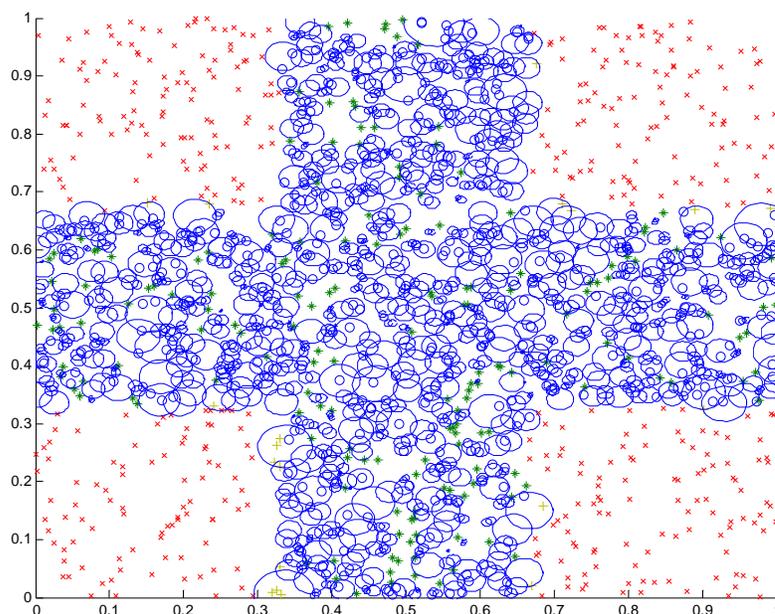


Figura 19 – Seleção positiva utilizando algoritmo original (1000 pontos)

No exemplo em formato de cruz, a área teórica que seria ocupada pelos pontos próprios seria de 0.5555, o que representa uma porção considerável do hipercubo unitário. Isto poderia justificar a dificuldade da seleção positiva, pois existem espaços dentro da região própria não cobertos pelas hiperesferas próprias. Observe que estes espaços também permitem o aparecimento de alguns detectores pequenos. Uma vez que o valor da função de avaliação é dado pelo volume dos detectores, detectores pequenos pouco contribuem e tendem a ser substituídos por detectores maiores. Para melhorar os resultados da seleção positiva, usa-se uma representação reduzida. Foi feita uma clusterização de 5000 pontos para 84 grupos (o mesmo número de detectores que foi obtido usando o algoritmo original) e o raio das hiperesferas próprias foi obtido considerando a distância ao centro do grupo mais próximo. A Tabela 16 mostra os resultados.

Tabela 16 – Resultados do algoritmo original utilizando representação positiva e negativa (84 grupos)

Método	Erro	Taxa de Detecção	Taxa de Alarme Falso
Negativa	0.051	93,40%	3,9%
Positiva	0.082	81,32%	0%

Observa-se que a taxa de alarme falso da seleção positiva caiu a 0%, mas houve uma queda significativa na taxa de detecção. Isto acontece porque as hiperesferas da representação reduzida do conjunto próprio classificam pontos anormais como normais. Na Figura 20 é mostrado o resultado da Seleção Positiva.

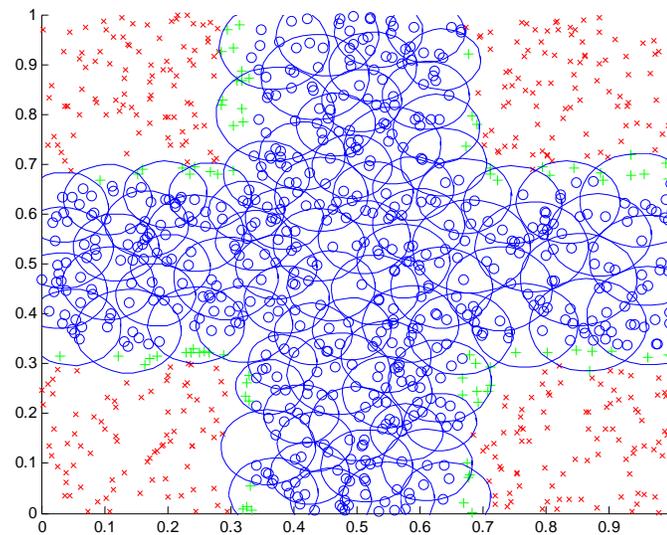


Figura 20 – Seleção positiva utilizando algoritmo original (84 grupos)

Observe-se agora, em um segundo exemplo, um conjunto de dados sintéticos com o formato de um círculo com raio igual 0.001 centrado em (0.5, 0.5). Para este exemplo, foram gerados 1000 pontos dentro deste círculo para representar o conjunto próprio. Neste exemplo o volume coberto pelo conjunto próprio é bem pequeno: 0.000003141.

Aplicando-se o algoritmo de geração de detectores. Verificou-se que um excelente valor de cobertura é atingido já nas primeiras gerações. Isto acontece porque os raios dos detectores crescem até o limite dado pelos raios do conjunto próprio. Como o volume do conjunto próprio é muito pequeno, o volume coberto pelos detectores é muito grande (teoricamente $1 - 0.000003141 = 0.999996859$). Uma vez que o valor da função de avaliação é dado pelo volume calculado por uma integração de Monte Carlo com 20 mil pontos, não é possível diferenciar valores de volume entre 0.99995 e 1. Isto faz com que todos os conjuntos de detectores que tenham valores maiores que 0.99995 tenham a mesma avaliação.

A Tabela 17 e a Tabela 18 mostram os resultados obtidos para este exemplo com o algoritmo original e com o algoritmo com melhorias, respectivamente. Foram utilizados como

pontos de teste 1000 pontos aleatórios dentro do Volume do conjunto próprio e 1000 pontos fora deste volume.

Tabela 17 – Resultado do algoritmo original

Método	Erro	Taxa de Detecção	Taxa de Alarme Falso
Negativa	0.0006	100%	0,06%
Positiva	0.1265	100%	25,29%

Tabela 18 – Resultado do algoritmo com melhorias

Método	Erro	Taxa de Detecção	Taxa de Alarme Falso
Negativa	0	100%	0%
Positiva	0.1265	100%	25,29%

Mais uma vez observa-se que a taxa de alarme falso na seleção positiva foi maior, pois este método considerou pontos próprios não cobertos por suas hiperesferas como pontos anormais. O que acontece é que a seleção positiva considera como anormais tanto pontos muito próximos dos pontos próprios, que não estão cobertos por suas hiperesferas, quanto pontos muito distantes que, normalmente, representam anomalias mais graves. Já a seleção negativa tem a tendência a considerar como pontos anormais aqueles mais distantes dos pontos próprios, mesmo que isso signifique que pontos anormais muito próximos dos pontos próprios sejam considerados normais. Pode-se observar ainda que, com as melhorias feitas no algoritmo, este tornou ainda mais discrepante a utilização do método da seleção negativa em detrimento da seleção positiva.

4.2 Estudo de Casos

Os seguintes experimentos foram propostos com o intuito de mostrar a eficiência do algoritmo em casos reais.

Para os três experimentos seguintes, filtro Sallen Key, filtro Universal e filtro Leapfrog, foi utilizada a técnica de pré-processamento *Bitmap*.

Os dois experimentos restantes são, respectivamente, detecção de anomalias em rolamentos e detecção de anomalias em eletrocardiogramas.

4.2.1 Filtro Sallen Key

Nos sistemas de processamento de sinais analógicos, os filtros Sallen Key de segunda ordem, que exigem apenas uma única fonte de tensão controlada (VCVS), são amplamente utilizados.

Neste trabalho é empregada a abordagem através da Resposta ao Impulso (*IR – Impulse Response*), pela qual as informações sobre circuitos lineares invariantes no tempo (LTI) podem ser obtidas e os parâmetros de desempenho podem ser inferidos implicitamente, em vez de serem diretamente medidos. Este é um fator significativo em vista do problema complexo de modelagem de falhas em circuitos analógicos (Amaral, 2006a).

Na metodologia de testes proposta, obtêm-se respostas impulsionais de circuitos. Posteriormente, utiliza-se a representação simbólica *Bitmap* como forma de pré-processamento dos dados. O uso da representação simbólica será apresentado a seguir, no item 4.2.1.1.

As falhas, por sua vez, são classificadas em falhas simples e falhas múltiplas. As falhas simples se caracterizam pelo circuito ter apenas um componente defeituoso, isto é, com seu valor nominal fora da tolerância. As falhas múltiplas são análogas às falhas simples, exceto pelo fato de se caracterizarem pelas falhas em mais de um componente. Em ambos os casos, quando ocorre uma falha em um determinado componente, este tem seu valor alterado aleatoriamente para as faixas de 25% a 50% ou de 150% e 175% do valor nominal.

Esta abordagem será usada para detectar falhas paramétricas no filtro Sallen Key, mostrado na Figura 21. Este circuito já foi utilizado em diversos trabalhos na área de detecção de anomalias, como por exemplo, em (Amaral et al., 2004 e 2007), (Augusto e Almeida, 2008), (Zhang e Xu, 2007), (Spina e Upadhyaya, 1997), (Arminian & Arminian, 2002), (Albustani, 2004) e (Savioli et al., 2004).

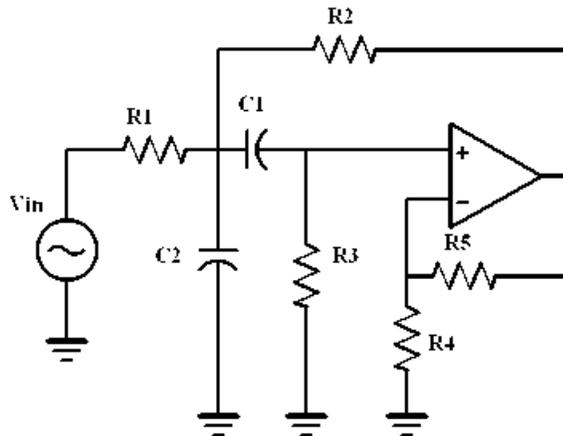


Figura 21 – Circuito do filtro Sallen Key

4.2.1.1 Uso de Representação Simbólica

Foi proposta a utilização da representação simbólica *Bitmap* (Wei et al., 2005). Esta representação permite aos usuários compreender uma série temporal de tamanho variado e identificar as partes de interesse. Ademais, esta representação ainda fornece a representação normalizada.

Para alcançar tal alvo, há a necessidade de uso de um método de discretização para transformar as informações de séries temporais contínuas em domínio discreto. Para tal fim, será utilizado o *Symbolic Aggregate approxImation* (SAX).

O SAX (Lin et al., 2003) foi a primeira representação simbólica para séries temporais que permitiu redução das dimensões do problema e indexação.

Em problemas clássicos de mineração de dados, como clusterização, classificação, indexação, etc., o SAX é uma representação tão boa quanto as Transformadas Discretas de Fourier e Wavelet, exigindo menos espaço de armazenamento. Além disso, a representação permite avaliar a riqueza de estruturas de dados e algoritmos em bioinformática ou busca textual, e também oferece soluções para os muitos desafios associados à mineração de dados.

Para transformar a série temporal de n dimensões para w dimensões, o dado é dividido em w seções de tamanhos iguais. O valor médio dos dados que caem dentro de uma parte é calculado e um vetor destes valores se torna a representação, com reduzida dimensão. Essa

representação é então discretizada, de tal modo que se obtenha uma palavra com símbolos. A Figura 22 ilustra a idéia de um sinal sendo convertido em um vetor SAX (Keogh et al., 2005).

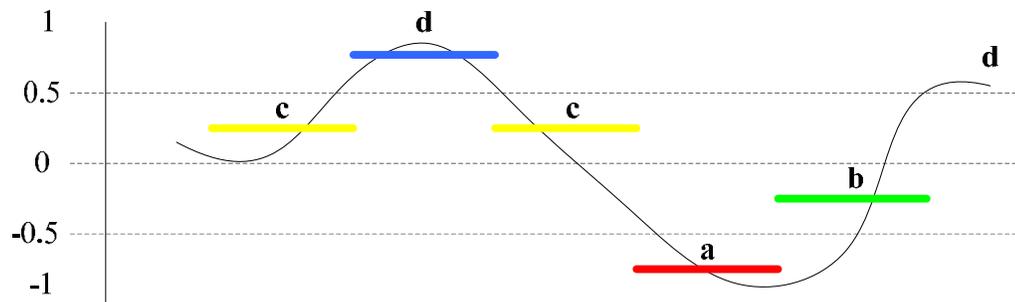


Figura 22 – Representação SAX

Após a conversão da série original para a representação SAX, podem-se contar as frequências de sub-palavras do SAX de comprimento L , onde L é o desejado nível de recursividade. Frequências de nível 1 são simplesmente a contagem de todos os tipos de símbolos. Já para as frequências de nível 2, contam-se os pares de sub-palavras de tamanho 2 (por exemplo, cc, cb, bc, etc.). Somente se contam sub-palavras de uma mesma série temporal, isto é, não se misturam as palavras. A Figura 23 ilustra este conceito.

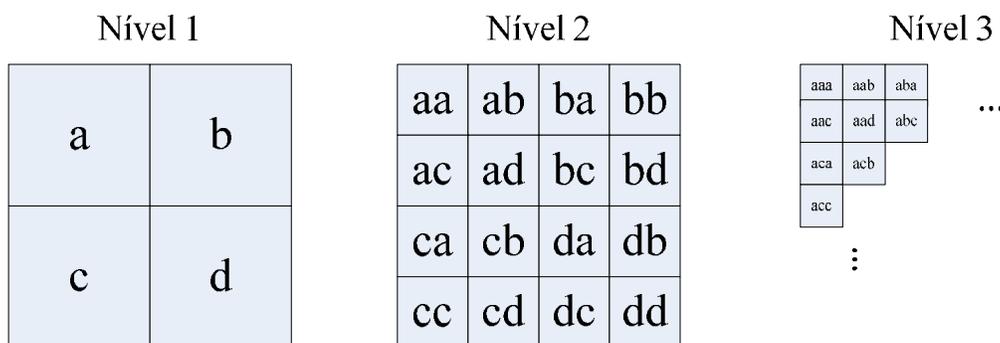


Figura 23 – Geração do *Bitmap*

Após este processo, há então uma normalização, tornando a saída deste pré-processamento entre 0 e 1, o que é adequado ao método utilizado, cujos dados devem ser representados em um hipercubo unitário.

Esta representação será utilizada como pré-processamento no estudo de casos, os quais fazem parte: os filtros Sallen Key, Universal e Leapfrog e a detecção de anomalias em eletrocardiogramas.

4.2.1.2 Resultados

Os valores nominais adotados para os componentes foram: $R1 = 5,18k\Omega$, $R2 = 1k\Omega$, $R3 = 2k\Omega$, $R4 = R5 = 4k\Omega$ e $C1 = C2 = 5nF$; e tolerância de 1%.

Os parâmetros utilizados para os algoritmos genéticos foram: 100 gerações e 50 indivíduos. Tanto os operadores de mutação e *crossover* mais adequados como os operadores de seleção de detectores foram utilizados.

A Tabela 19 mostra o número de detectores obtidos, o tempo de processamento e o volume coberto. A Tabela 20 apresenta as taxas de detecção e alarme falso e erro do classificador para falhas simples e múltiplas dos componentes.

Tabela 19 – Resultados do algoritmo genético para filtro Sallen Key

Parâmetros	Resultados
Detectores Obtidos	100
Tempo de Execução	2,40 minutos
Volume Coberto	1

Tabela 20 – Resultados do classificador para filtro Sallen Key

Parâmetros	Resultados	
	Falha Simples	Falhas Múltiplas
Taxa de Detecção	99,8%	98,2%
Taxa de Alarme Falso	0%	0%
Erro	0.001428	0.0129

Estes resultados foram obtidos já nas primeiras gerações. A ótima cobertura fornecida pelos detectores é evidenciada pelos resultados do classificador, onde houve uma diferença muito pequena entre as taxas de detecção para falhas simples e múltiplas. Pode-se destacar também o baixo tempo de execução do algoritmo.

4.2.2 Filtro Universal

O chamado filtro universal, mostrado na Figura 24, é um filtro ativo que incorpora filtros passa-baixa (LPO), passa-alta (HPO) e passa-faixa (BPO) em um mesmo circuito. Ele

faz parte de um conjunto de circuitos utilizados pelo IEEE Mixed-Signal Technical Activity Committee para a pesquisa e desenvolvimento de técnicas de detecção e modelagem de falhas, de geração de vetores de teste, de projeto visando a testabilidade e de BIST (*Built-in Self Test*) (Amaral, 2006a).

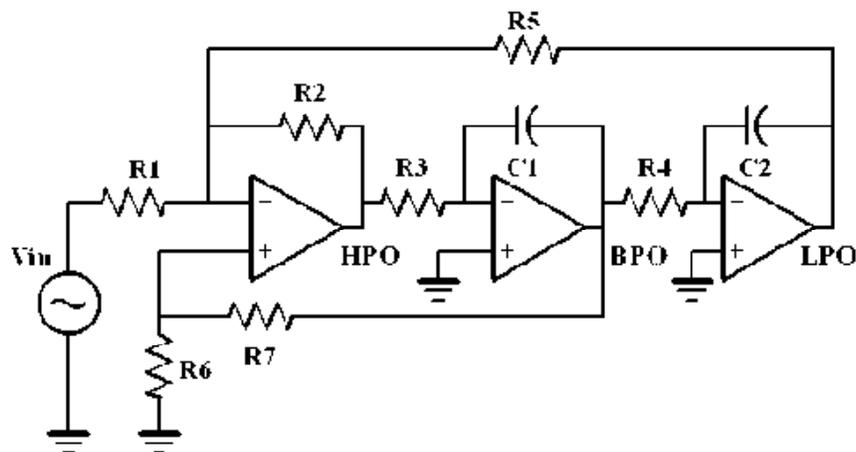


Figura 24 – Circuito do filtro universal

Para o tratamento dos dados deste circuito foi utilizado o mesmo procedimento descrito no item 4.2.1. Neste circuito de referência foram utilizados os seguintes valores nominais para os componentes: $R1 = R2 = R3 = R4 = R5 = 10k\Omega$, $C1 = C2 = 20nF$, $R6 = 3k\Omega$ e $R7 = 7k\Omega$ (Kaminska et al., 1998). A tolerância dos componentes foi de 1%.

Este circuito já foi utilizado em algumas publicações (Allipi et al., 2002), (Singh et al., 2004), (Albustani, 2004) com diferentes técnicas de pré-processamento. Neste experimento, o pré-processamento empregado é o *Bitmap*, descrito na seção 4.2.1.1.

A Tabela 21 mostra o número de detectores obtidos, o tempo de processamento, o volume coberto. Já a apresenta as taxas de detecção e alarme falso e erro do classificador para falhas simples e múltiplas dos componentes.

Tabela 21 – Resultados do algoritmo genético para filtro universal

Parâmetros	Resultados
Detectores Obtidos	100
Tempo de Execução	1,14 minutos
Volume Coberto	1

Tabela 22 – Resultados do classificador para filtro universal

Parâmetros	Resultados	
	Falha Simples	Falhas Múltiplas
Taxa de Detecção	100%	99%
Taxa de Alarme Falso	0	0
Erro	0	0.00714

Assim como no exemplo anterior, para o Filtro Universal pode-se observar a ótima cobertura fornecida pelos detectores e o baixo tempo de execução do algoritmo.

4.2.3 Filtro Leapfrog

O terceiro circuito é mostrado na Figura 25. Este circuito de *benchmark* é composto de treze resistores, quatro capacitores e seis amplificadores operacionais.

O filtro Leapfrog de quarta ordem é utilizado para eliminar as altas frequências. Portanto, o filtro permite a passagem desde a componente DC até as baixas frequências da operação desejada do sistema. Normalmente este filtro é utilizado em amplificadores de áudio (Kaminska et al., 1998).

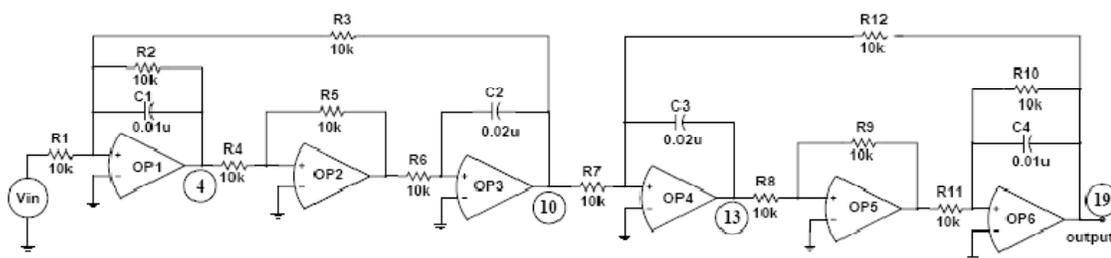


Figura 25 – Circuito do filtro Leapfrog

Para o tratamento dos dados deste circuito foi utilizado o mesmo procedimento descrito no item 4.2.1. Neste circuito de referência foram utilizados os seguintes valores nominais para os componentes: $R1 = R2 = R3 = R4 = R5 = R6 = R7 = R8 = R9 = R10 = R11 = R12 = 10k\Omega$, $C1 = C4 = 10nF$, e $C2 = C3 = 20nF$ (Kaminska et al., 1998). A tolerância utilizada para os componentes foi de 1%.

Este circuito já foi utilizado em diversos trabalhos na área de detecção de falhas, como por exemplo, em (Chakrabarti e Chatterjee, 2000), (Guo, 2003), (Guo e Savir, 2006) e (Mahdavi e Mohammadi, 2009). Neste experimento, o pré-processamento empregado é o *Bitmap*, descrito na seção 4.2.1.1.

A Tabela 23 mostra o número de detectores obtidos, o tempo de processamento e o volume coberto. A Tabela 24 apresenta as taxas de detecção e alarme falso e erro do classificador para falhas simples e múltiplas dos componentes.

Tabela 23 – Resultados do algoritmo genético para filtro Leapfrog

Parâmetros	Resultados
Detectores Obtidos	95
Tempo de Execução	1,40 minutos
Volume Coberto	1

Tabela 24 – Resultados do classificador para filtro Leapfrog

Parâmetros	Resultados	
	Falha Simples	Falhas Múltiplas
Taxa de Detecção	84,0%	95,8%
Taxa de Alarme Falso	0	0
Erro	0.1143	0.03

Pode-se observar a ótima cobertura fornecida pelos detectores e o baixo tempo de execução do algoritmo. Em contrapartida as taxas de detecção foram inferiores aos outros filtros. Isto acontece porque o filtro Leapfrog apresenta uma menor sensibilidade à variação de parâmetros do que os outros filtros. É necessário que haja uma variação muito grande nos valores dos componentes para que a forma da resposta ao impulso seja alterada.

4.2.4 Detecção de Anomalias em Rolamentos

Em um rolamento, a carga é transmitida do anel externo para o anel interno por meio de uma esfera, o que significa que as regiões de contato são bem pequenas, implicando em rotações bastante suaves. Entretanto, isto também significa que não existe muita área de contato para receber a carga, o que pode causar danos ao rolamento.

Os dados utilizados neste experimento foram utilizados em (Ji e Dasgupta, 2004). Os dados experimentais foram obtidos partir de medições da aceleração dos rolamentos. O pré-processamento usa momentos estatísticos para representar segmentos de 128 pontos dos sinais de aceleração. Os momentos de primeira (média), de segunda (variância), de terceira, de quarta e quinta ordens são utilizados, fazendo com que o problema tenha cinco dimensões.

Os parâmetros foram: 50 gerações e 200 indivíduos. Foram utilizados tanto os operadores genéticos mais adequados como os operadores de seleção de anomalias.

A Tabela 25 mostra o número de detectores obtidos, o tempo de processamento, o volume coberto, assim como as taxas de detecção e alarme falso e erro do classificador.

Tabela 25 – Resultados obtidos para experimento com rolamentos

Parâmetros	Resultados
Detectores Obtidos	200
Tempo de Execução	2,2 minutos
Volume Coberto	1
Taxa de Detecção	93,73%
Taxa de Alarme Falso	0
Erro	0,0282

Observa-se uma alta taxa de detecção aliada a uma taxa de alarme falso zero. O erro do classificador foi menor do que 3%, enquanto o do algoritmo original (Amaral, 2006a) foi de 9% e o alcançado por Ji e Dasgupta (2004) foi de 22,82%. O algoritmo foi executado em poucos minutos e utilizou o máximo de detectores estabelecido.

4.2.5 Detecção de Anomalias em Eletrocardiogramas

O eletrocardiograma (ECG) é um exame médico na área de cardiologia onde é feito o registro da variação dos potenciais elétricos gerados pela atividade elétrica do coração. Através destes registros é possível fazer um diagnóstico das condições cardíacas do paciente.

Os dados utilizados para este experimento foram retirados da pesquisa de Kim et al. (2004) e se reporta um acerto de 98% utilizando modelos ocultos de Markov. Foi escolhida pelos autores uma gravação de ECG e esta foi dividida em formas de onda individuais para obter 28 batimentos cardíacos normais e 28 batimentos cardíacos anormais de uma contração

ventricular prematura. Como pré-processamento dos dados, foi utilizado o Bitmap, descrito na seção 4.2.1.1.

A Tabela 26 apresenta os resultados tanto do algoritmo genético como do classificador. Para este experimento foram utilizados apenas 5 gerações, pois o algoritmo genético converge rapidamente para um bom resultado em poucas gerações. Com isso, decai o tempo de execução do algoritmo.

Tabela 26 – Resultados obtidos para experimento com eletrocardiograma

Parâmetros	Resultados
Detectores Obtidos	5
Tempo de Execução	0,15 minutos
Volume Coberto	99,99%
Taxa de Detecção	100%
Taxa de Alarme Falso	0
Erro	0

Para este experimento a taxa de detecção é de 100% e a de alarme falso é igual a zero. O erro é zero. O tempo de execução foi de poucos segundos, atingindo uma altíssima cobertura.

5 CONCLUSÃO

Esta dissertação apresentou um método de detecção de anomalias baseado em sistemas imunológicos artificiais, e, por sua vez, fundamentado em uma técnica de reconhecimento próprio/não-próprio chamada algoritmo de seleção negativa.

O objetivo deste trabalho foi aprofundar os estudos sobre este algoritmo, assim como aperfeiçoar suas funções a fim de se obter uma maior área de cobertura possível num menor espaço de tempo.

A metodologia adotada neste trabalho para análise e melhoria do método foi estruturada em três etapas principais: (i) identificação das limitações do algoritmo, (ii) melhorias do algoritmo e (iii) estudo de casos.

Na primeira etapa foram identificadas as limitações do algoritmo, como a verificação da utilização de ordenação e da variação da sobreposição dos detectores. Foi também verificado o comportamento do algoritmo tanto em baixas como em altas dimensões.

Na segunda etapa foram propostas e implementadas melhorias do mesmo algoritmos: a implementação de operadores genéticos mais adequados e de métodos de redução da quantidade de pontos do conjunto próprio.

Na terceira etapa, o algoritmo aprimorado foi avaliado e comparado à versão sem melhorias. Ainda nesta etapa foi efetuado um estudo para os seguintes casos de circuitos eletrônicos reais: filtro Sallen Key, filtro universal e filtro Leapfrog, com a técnica de pré-processamento *Bitmap*. Para os dois primeiros circuitos, o algoritmo alcançou bons resultados. Porém, para o experimento com o filtro Leapfrog, o algoritmo não obteve resultados tão bons quanto nos demais experimentos, porque a variação dos valores nominais não é suficiente para se caracterizar uma falha propriamente dita, ou seja, não modifica a forma de onda da resposta ao impulso o suficiente para ser considerada como falha.

A técnica de pré-processamento *Bitmap* se mostrou bastante apropriada, podendo ser aplicada a diferentes tipos de problemas, como detecção de anomalias em circuitos e em eletrocardiogramas. Outro aspecto interessante da técnica é o fato de ela normalizar de forma automática os pontos entre 0 e 1, o que é bastante conveniente para o método de seleção negativa empregado, que requer que os dados estejam representados em um hipercubo.

Foi também feito um estudo para os experimentos de detecção de falhas em rolamentos e para detecção de anomalias em eletrocardiogramas, nos quais o algoritmo se mostrou

eficiente (em problemas de naturezas distintas). No experimento do eletrocardiograma, o algoritmo com melhorias teve um resultado superior ao apresentado por Kim et al. (2004).

Em resumo, o estudo sobre o algoritmo permitiu uma melhor compreensão de suas características e melhorias implementadas, possibilitando ao algoritmo atingir bons resultados, inclusive quando comparado à versão anterior. Os resultados alcançados mostraram que o novo algoritmo atingiu o objetivo de reduzir o tempo de execução sem comprometer a eficiência.

A utilização de algoritmos evolucionários em Sistemas Imunológicos Artificiais é uma área com inúmeras possibilidades de pesquisa. Portanto, existem muitas direções que podem ser seguidas em trabalhos futuros.

Uma destas possibilidades é a utilização de um algoritmo evolucionário multi-objetivo com a finalidade de maximizar volume coberto e minimizar o número de detectores finais, já que mesmo com a utilização dos operadores de seleção de detectores, a minimização do número de detectores utilizados não se mostrou tão eficiente em alguns casos. Isto ocorre devido ao fato do algoritmo genético utilizado não ser multi-objetivo e, portanto, se focar apenas na maximização do volume a ser coberto pelos detectores.

Os resultados mostraram também que, em alguns casos onde o conjunto próprio é demasiadamente pequeno, há a necessidade de calcular o volume de forma diferenciada, fazendo um tratamento diferenciado para regiões próximas aos pontos próprios. Deste modo, o cálculo da função de aptidão seria composto de duas partes: a primeira seria o volume coberto pelos detectores utilizando um número moderado de pontos (da ordem de dezenas de milhares) em todo o hipercubo; numa segunda parte calcular-se-ia o volume próximo ao conjunto próprio. Existe, então, a necessidade de se estudar outros métodos de cálculo de volume.

Outra alternativa de trabalho futuro seria utilizar outras representações, como, por exemplo, hiperparalelogramos. Possivelmente, esta representação possibilitaria uma melhor cobertura do espaço do que as hiperesferas, mas o algoritmo teria que ser modificado para lidar com esta nova representação.

REFERÊNCIAS

ABE, N., ZADROZNY, B., LANGFORD, J. 2006. Outlier Detection by Active Learning. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006, p.504-509.

AKHAVAN, S.; CALVA, G. Automatic anomaly detection in ECG signal by fuzzy decision making. 6th International Conference on Fuzzy Theory and Technology, 1998, p.96-98.

ALBUSTANI, H. Modeling Methods for Testability Analysis of Analog Integrated Circuits Based on Pole-Zero Analysis. Ph.D Thesis, Universität Duisburg-Essen, 2004.

ALMEIDA, T. A.; YAMAKAMI, A.; TAKAHASHI, M. T. Sistema imunológico artificial para resolver o problema da árvore feradora mínima com parâmetros fuzzy. Pesquisa Operacional. 2007; 27(1): p.131-154. ISSN 0101-7438.

AMARAL, J. L. M.; AMARAL, J. F. M.; MORIM, D.; TANSCHKEIT, R. An Immune Fault Detection System with Automatic Detector Generation by Genetic Algorithms. 7th International Conference on Intelligent Systems Design and Applications, Rio de Janeiro, 2007, p. 233-238.

AMARAL, J.L.M. Sistemas Imunológicos Artificiais aplicados à Detecção de Falhas. Tese de Doutorado, PUC, Rio de Janeiro, 2006a.

AMARAL, J. L. M.; AMARAL, J. F. M.; TANSCHKEIT, R. An Immune Fault Detection System for Analog Circuits with Automatic Detector Generation. IEEE Congress on Evolutionary Computation, 2006b.

AMARAL, J. L. M.; AMARAL, J. F. M.; TANSCHKEIT, R.; PACHECO, M. A. C . An Immune Inspired Fault Diagnosis System for Analog Circuits Using Wavelet Signatures. NASA/DoD Conference in Evolvable Hardware, 2004, p. 138-141.

AMINI, M.; JALILI, R. Network-Based Intrusion Detection Using Unsupervised Adaptive Resonance Theory (ART). Proceedings of the 4th Conference on Engineering of Intelligent Systems, 2004.

ARMINIAN, F.; ARMINIAN, M.; COLLINS Jr, H. W. Analog Fault Diagnosis of Actual Circuits Using Neural Networks. IEEE Transactions on Instrumentation and Measurement, Vol. 47, No. 2, 2000, p. 554-550.

AUGUSTO, J. A. S.; ALMEIDA, C. B. A Tool for Single Fault Diagnosis in Linear Analog Circuits with Tolerance Using the T-vector Approach. VLSI Design, 2008.

BÄCK, T.; SCHWEFEL, H.. Evolutionary Computation: an Overview. In Proceedings of the Third IEEE Conference on Evolutionary Computation, IEEE Press, 1996a, p. 20-29.

BÄCK, T. Evolution Strategies: an Alternative Evolutionary Algorithm. In J. M. Alliot, E. Lutton, E. Ronald (Eds.), Artificial Evolution, pp. 3-20, Springer-Verlag, 1996b.

BARBOSA, C. et al. Nondestructive evaluation of steel structures using a superconducting quantum inference device magnetometer and a neural network system. *Review of Scientific Instruments*, Vol.71, No. 10, 2000, p. 3806-3815.

BARRETO, G. A.; MOTA, J. C. M.; SOUZA, L. G. M.; FROTA, R. A.; AGUAYO, L.; YAMAMOTO, S.; MACEDO, P. Competitive Neural Networks for Fault Detection and Diagnosis in 3G Cellular Systems. *LNCS Lecture Notes in Computer Science*, Vol. 3124, 2004, p. 207-213.

BÖRNER, M.; STRAKY, H.; WEISPFENNING, T.; ISERMANN, R. Model Based Fault Detection of Vehicle Suspension and Hydraulic Brake Systems. *Mechatronics* 12, 2002, p. 999–1010.

BISHOP, C. M. Novelty Detection and Neural Network Validation. In *Proceedings of IEEE Vision, Image and Signal Processing*. Vol. 141, 1994, p. 217-222.

CARPENTER, G. A.; GROSSBERG, S. Adaptive Resonance Theory. *The Handbook of Brain Theory and Neural Networks*, Second Edition, 2002.

CASTRO, L. N., TIMMIS J. Artificial Immune System: A New Computational Intelligence Approach. Springer-Verlag, 2002.

CASTRO, L.N. Engenharia Imunológica: Desenvolvimento e Aplicação de Ferramentas Computacionais Inspiradas em Sistemas Imunológicos Artificiais. Tese de Doutorado, 287 p., UNICAMP, Campinas, SP, 2001.

CASTRO, L. N.; VON ZUBEN, F. J. The Clonal Selection Algorithm with Engineering Applications. *Workshop Proceedings of GECCO*, 2000, p. 36-37.

CATELANI, M., FORT, A. Soft Fault Detection and Isolation in Analog Circuits: Some Results and a Comparison Between a Fuzzy Approach and Radial Basis Function Networks. *IEEE Transactions on Instrumentation and Measurement*, Vol. 51, No. 2, 2002, p. 196–202.

CHAKRABARTI, S.; CHATTERJEE, A. Partial simulation-driven ATPG for detection and diagnosis of faults in analog circuits. *Proceedings of the 2000 IEEE/ACM International Conference on Computer-Aided Design*, 2000.

CHAN, T. H. T., NI, Y. Q.; KO, J. M. Neural Network Novelty Filtering For Anomaly Detection Of Tsing Ma Bridge Cables. *Structural Health Monitoring 2000*, Technomic Publishing Co, Lancaster, 1999, p. 430–439.

CHANDOLA, V., BANERJEE, A., KUMAR, V. Anomaly Detection: A Survey. To Appear in *ACM Computing Surveys*, 2009.

CHOWDHURY, F. N.; CHRISTENSEN, J. P.; ARAVENA, J. L. Power System Fault Detection and State Estimation Using Kalman Filter with Hypothesis Testing. *IEEE Transactions on Power Delivery*, 1991.

CHUAH, M.; FU, F. ECG Anomaly Detection Via Time Series Analysis. Technical Report LU-CSE-07-001, 2007.

CORREA, E. S.; STEINER, M. T. A.; FREITAS, A. A.; CARNIERI, C. A Genetic Algorithm for Solving a Capacitated P-Median Problem. Numerical Algorithms. Kluwer Academic Publishers, Vol. 35, 2004, p.373-388.

CROOK, P. A., MARSLAND, S., HAYES, G., NEHMZOW, U. A Tale of Two Filters - On-Line Novelty Detection. Proceedings of International Conference on Robotics and Automation, 2002, p. 3894-3899.

D'HAESELEER, P.; FORREST, S.; HELMAN, P. An Immunological Approach To Change Detection: Algorithms. Proc Of The 1996 IEEE Symposium On Computer Security And Privacy, 1996, p. 110-119.

DASGUPTA, D. Advances in Artificial Immune Systems. IEEE Computational Intelligence Magazine, 2006, p. 40-49.

DASGUPTA, D.; JI, Z.; González, F. Artificial Immune System (AIS) Research in the Last Five Years. IEEE Congress Of Evolutionary Computation (CEC), 2003.

DASGUPTA, D.; KRISHNAKUMAR, K.; WONG, D.; BERRY, M. Negative Selection Algorithm For Aircraft Fault Detection. Artificial Immune Systems: Proceedings Of ICARIS 2004, SPRINGER, 2004, p. 1-14.

DASGUPTA, D.; FORREST, S. Novelty Detection in Time Series Data using Ideas from Immunology. 5th International Conference on Intelligent Systems, 1996.

DIAS, M. A. G. Real Options Approach in Petroleum Investment, <http://www.puc-rio.br/marco.ind/>, 2004, visitado em 23 de fevereiro de 2009.

DONAT ,W.; CHOI, K.; AN, W.; SINGH, S.; PATTIPATI, K. Data Visualization, Data Reduction and Classifier Fusion for Intelligent Fault Detection and Diagnosis in Gas Turbine Engines. ASME Turbo Expo, 2007.

EBNER, M., BREUNIG, H. G., ALBERT, J. On the use of negative selection in an artificial immune system. Proceedings of the Genetic and Evolutionary Computation Conference, 2002, p. 957-964.

EPSTEIN, B. R.; CZIGLER, M.; MILLER, S. R. Fault Detection and Classification in Linear Integrated Circuits: An Application of Discrimination Analysis and Hypothesis Testing. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1993.

FENTON, W. G., MCGINNITY, T. M., MAGUIRE, L. P. Fault diagnosis of electronic systems using intelligent techniques: a review. IEEE Trans. Systems, Man and Cybernetics – Part C, Vol. 31, 2001, p. 269-281.

FOGEL, G. B.; FOGEL, D. B. Continuous Evolutionary Programming: Analysis And Experiments. In Cybernetics And Systems, Vol. 26, 1995, p. 79-90.

- FORREST, S., et al. Self-Nonself Discriminator in a Computer. Proc. of the 1994 IEEE Symposium on Research in Security and Privacy, Los Alamitos, CA: IEEE Comput. Soc. Press, 1994, p. 202-212.
- FUKUNAGA, K.; NARENDRA, P. M. A Branch and Bound Algorithm for Computing k-Nearest Neighbors. IEEE Trans. Comput., 1975, p.750-753.
- GOLDBERG, D. E. Genetic Algorithms in Search, Optimization, and Machine Learning. Ed. Addison-Wesley, 1989.
- GONZÁLEZ, F., DASGUPTA, D., NIÑO, L. F. A Randomized Real-Valued Negative Selection Algorithm. Proceedings of the 2nd International Conference on Artificial Immune Systems (ICARIS), 2003, p.261–272.
- GONZÁLEZ, F., DASGUPTA, D., KOZMA, R. Combining Negative Selection and Classification Techniques for Anomaly Detection. Congress on Evolutionary Computation, IEEE, 2002, p.705–710.
- GORONZY, J. J., WEYAND, C. M. The Innate and Adaptive Immune Systems. In: Goldman, L.e Ausiello, D., Eds.Cecil Medicine, 23 ed, Saunders, 2008.
- GUO, Z. Coefficient-Based Parametric Faults Detection in Analog Circuits. GLSVLSI, 2003, p. 299-302.
- GUO, Z.; Savir, J. Coefficient-based test of parametric faults in analog circuits, IEEE Transactions on Instrumentation and Measurement 55, Vol. 1, 2006, p. 150–157.
- HOUCK, C. R.; JOINES, J. A.; KAY, M. G. A Genetic Algorithm for Function Optimization: a Matlab Implementation. Technical Report: NCSU-IE-TR-95-09, North Carolina State University, Raleigh, 1995.
- IVERSEN, A.; TAYLOR, N. K.; BROWN, K. E. Anomaly Detection by Auto-Association. Proceedings of the 7th Nordic Signal Processing Symposium, 2006.
- JANEWAY, C., How the immune system recognizes invaders. Scientific American, Vol. 269, no. 3, 1993, p. 72–79.
- JERNE, N. K. Idiotypic Networks and Other Preconceived Ideas. Immunological Reviews, Vol. 79, 1984, p. 5-24.
- JI, Z., DASGUPTA, D. Revisiting Negative Selection Algorithms. Evolutionary Computation, Vol. 15, No 2, 2007, p. 223-251.
- JI, Z., DASGUPTA, D. Applicability Issues of the Real Valued Negative Selection Algorithms. Genetic and Evolutionary Computation Conference – GECCO, 2006.
- JI, Z. A boundary-aware negative selection algorithm. Proceedings of IASTED International Conference on Artificial Intelligence and Soft Computing (ASC), 2005.

JI, Z., DASGUPTA, D. Real-Valued Negative Selection Algorithm with Variable-Sized Detectors. *Genetic and Evolutionary Computation, Part I*, Vol. 3102, 2004, p.287–298.

KAMINSKA, B.; ARABI, K.; BELL, I.; GOTELI, P.; HUERTAS, J. L.; KIM, B.; RUEDA, A.; SOMA, M. Analog and mixed-signal benchmark circuits – first release. *IEEE International Test Conference*, 1998, p.183-190.

KEOGH, E; LIN, J; FU, A. HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence. *5th IEEE International Conference on Data Mining*, 2005, p. 226-233.

KIM, S.; SMYTH, P.; LUTHER, S. Modeling Waveform Shapes with Random Effects Segmental Hidden Markov Models. *Technical Report UCI-ICS 04-05*, 2004.

KOZA, J. R. *Genetic Programming: Automatic Discovery of Reusable Programs*. MIT Press, 1992.

LI, Z.; WUB, Z.; HE, Y.; FULEI, C. Hidden Markov Model-Based Fault Diagnostics Method in Speed-Up and Speed-Down Process for Rotating Machinery. *Mechanical Systems and Signal Processing*, Vol. 19, Issue 2, 2005, p. 329-339.

LIN, J.; KEOGH, E.; LONARDI, S.; CHIU, B. A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. In *proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 2003.

MAHDAVI, S. J. S.; MOHAMMADI, K. Evolutionary derivation of optimal test sets for neural network based analog and mixed signal circuits fault diagnosis approach. *Microelectronics Reliability*, Vol. 49, 2009, p. 199–208.

MAIDON, Y.; JERVIS, B.W.; DUTTON, N.; LESAGE, S. Diagnosis of Multifaults in Analogue Circuits Using Multilayer Perceptrons. *IEEE Proceedings of Circuits Devices Systems*, Vol. 144, No. 3, 1997, p. 149–154.

MAK, M. W.; ALLEN, W. G.; SEXTON, G. C. Speaker Identification Using Radial Basis Functions. *Third International Conference on Artificial Neural Networks*, 1993, p.138 –142.

MARKOU, M.; SINGH, S. Novelty Detection: A Review Part1: Statistical Approaches. *Signal Processing*, 2003a.

MARKOU, M.; SINGH, S. Novelty Detection: A Review Part2: Neural Network-Based Approaches. *Signal Processing*, 2003b.

MARTINS, J.F; BRANCO, P.J. COSTA; DENTE, J.A. Fault Detection using Immune-Based Systems and Formal Language Algorithms, in *Proc of 39th Conference on Decision and Control*, 2000, p. 2633-2638.

MICHALEWICZ, Z. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, 1996.

MITCHELL, M. *An Introduction To Genetic Algorithms*. MIT Press, 1996.

- MOHAMMADI, K.; MONFARED, A.R.M; NEJAD, A.M. Fault Diagnosis of Analog Circuits with tolerances by using RBF and BP Neural Networks. Proc. of Students Conference on Research and Development, 2002, p. 317-321.
- MOROKOFF, W. J.; CAFLISCH R. E. Quasi-Monte Carlo Integration. Academic Press, 1993.
- MOROKOFF, W. J.; CAFLISCH R. E. Quasi-Random Sequences and Their Discrepancies. SIAM Journal on Scientific Computing, Vol. 15, Issue 6, 1994, p. 1251-1279.
- MUKKAMALA S.; JANOSKI G.; SUNG A. H. Intrusion Detection Using Neural Networks and Support Vector Machines. Proceedings of IEEE International Joint Conference on Neural Networks, 2002, p.1702-1707.
- ONDEL, O.; BOUTLEUX, E.; CLERC, G. A Method to Detect Broken Bars in Induction Machine Pattern Recognition Techniques. IEEE Transactions on Industry Applications, Vol. 42, No. 4, 2006, p. 916–923.
- PAALANEN, P.; KÄMÄRÄINEN, J.; ILONEN, J.; KÄLVIÄINEN, H. Feature Representation and Discrimination Based on Gaussian Mixture Model Probability Densities – Practices and Algorithms. Research Report, 1995.
- SMYTH, P. Probabilistic Anomaly Detection in Dynamic Systems. Neural Information Processing Conference, 1993.
- PARZEN, E. An Estimation of a Probability Density Function and Mode. Ann. Math. Statist., Vol. 33, 1962, p. 1065-1076.
- QIAO, Y.; XIN, X.W.; BIN, Y.; GE, S. Anomaly intrusion detection on HMM. Electronics Letters, Vol. 30, N° 13, p. 63, 2002.
- QINGYANG, X., XIANYAO, M.; NING, W. Gas Turbine Fault Diagnosis Based on ART2 Neural Network. Proceedings of the 7th World Congress on Intelligent Control and Automation, 2008, p. 5244-5248.
- RANGANATHAN, A. The Levenberg-Marquardt Algorithm. Disponível em <http://citeseer.ist.psu.edu/638988.html>. Acesso em 23 de dezembro de 2007.
- SAVIOLI, C. E.; SZENDRODI, C. E.; CALVANO, J. V.; MESQUITA, A. C. ATPG for Fault Dignosis on Analog Electrical Networks Using Evolutionary Techniques. Proc. 17th Symposium on Integrated Circuits and System Design. Brasil. 2004. p. 100-104.
- SHIMAZAKI, H.; SHINOMOTO, S. A Method for Selecting the Bin Size of a Time Histogram. Neural Computation, MIT Press, 2007.
- SINGH, A.; PATEL, C.; PLUSQUELLIC, J. On-Chip Impulse Response Generation for Analog and Mixedsignal Testing. IEEE International Test Conference, 2004, p. 262-270.
- SILVA, J. M. N. Redes Neurais Artificiais: Rede Hopfield e Redes Estocásticas. Estudo Orientado, UFF, 2003.

SPEARS, W.; DE JONG, K. A.; BÄCK, T.; FOGEL, D. B.; GARRIS, H. An Overview of Evolutionary Computation. In Proceedings of the European Conference on Machine Learning (ECML-93), LNAI, vol. 667, Springer-Verlag, 1993, p. 442-459.

SPINA, R.; UPADHYAYA, S. Linear Circuit Fault Diagnosis Using Neuromorphic Analyzers. IEEE Transactions on Circuits and Systems II, vol. 44, 1997, p.188-196.

STIBOR, T.; TIMMIS, J.; ECKERT, C. On the Use of Hyperspheres in Artificial Immune Systems as Antibody Recognition Regions. Lecture Notes in Computer Science, 2006.

STIBOR, T.; TIMMIS, J.; ECKERT, C. A Comparative Study of Real-Valued Negative Selection to Statistical Anomaly Detection Techniques. ICARIS, 2005a, p. 262–275.

STIBOR, T.; TIMMIS, J.; ECKERT, C. Is Negative Selection Appropriate for Anomaly Detection?. Genetic and Evolutionary Computation , 2005b.

SUN, Y.; CHEN, G.; LI, H. Analog Circuits Fault Diagnosis Using Support Vector Machine. International Conference on Communications, Circuits and Systems, 2007.

TIMMIS, J.; NEAL, M.; HUNT, J. An Artificial Immune System for Data Analysis. Biosystems, vol. 55, n. 1, 2000, p.143-150.

THOTTAN M.; JI, C. Anomaly Detection in IP Networks. IEEE Trans. Signal Process, 2003, p. 2191–2204.

VARELA, F., COUTINHO, A., DUPIRE, B., VAZ, N., Cognitive networks: immune and neural and otherwise. Theoretical Immunology: Part Two. SFI Studies in the science of Complexity, 1988, p. 359–371.

WEI, L.; KEOGH, E.; et al. Assumption-Free Anomaly Detection in Time Series. 17th International Scientific and Statistical Database Management Conference, 2005.

YE Z.; WU B.; SADEGHIAN, A. R. Induction Motor Mechanical Fault Online Diagnosis With the Application of Artificial Neural Network. Proceedings of IEEE Applied Power Electronics Conference (APEC), 2001, p.1015-1021.

YPMA, A.; DUIN, R.P.W. Novelty Detection Using Self-Organising Maps. Progress in Connectionist Based Information Systems, vol. 2, 1998, p. 1322-1325.

ZHANG, W.Q.; XU, C. Improved Algorithms for Circuit Fault Diagnosis Based on Wavelet Packet and Neural Network. International Symposium on Nonlinear Dynamics, 2007.

ZHENG, J.; HU, M. An Anomaly Intrusion Detection System Based on Vector Quantization Transactions on Information and Systems. IEICE, 2006.

This document was created with Win2PDF available at <http://www.win2pdf.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.
This page will not be added after purchasing Win2PDF.